Describing LRs with Metadata: Towards Flexibility and Interoperability in the Documentation of LR

Workshop Programme

22 May 2012

9:00 – 9:10 Welcome and Introduction

9:10 – 10:40 Overview Session

Daan Broeder, Menzo Windhouwer, Dieter van Uytvanck, Thorsten Trippel and Twan Goosen, *CMDI: a Component Metadata Infrastructure*

Maria Gavrilidou, Penny Labropoulou, Elina Desipri, Ioanna Giannopoulou, Olivier Hamon and Victoria Arranz, *The META-SHARE Metadata Schema: Principles, Features, Implementation and Conversion from other Schemas*

Gunn Inger Lyse, Carla Parra and Koenraad De Smedt, *Applying Current Metadata Initiatives: The META-NORD Experience*

10:40 – 11:00 Coffee break

11:00 – 13:00 Working and Experiencing the Component Model

Volker Boehlke, Torsten Compart and Thomas Eckart, *Building up a CLARIN resource center – Step 1: Providing metadata*

Thorsten Trippel, Christina Hoppermann and Griet Depoorter, *Infrastructure (CMDI) in a Project on Sustainable Linguistic Resources*

Hanna Hedeland and Kai Wörner, *Experiences and Problems creating a CMDI profile from an existing Metadata Schema*

Menzo Windhouwer, Daan Broeder and Dieter van Uytvanck, *A CMD Core Model for CLARIN Web Services*

13:00 – 14:00 Lunch break

14:00 – 16:00 General Impulses and Test Cases

Peter Menke and Philipp Cimiano, *Towards an ontology of categories for multimodal annotation*

Kristian Tangsgaard Hvelplund and Michael Carl, *User Activity Metadata for Reading, Writing and Translation Research*

Paula Estrella, *Metadata for a Mocoví - Quechua - Spanish parallel corpus*

Hennie Brugman and Mark Lindeman, *Publishing and Exploiting Vocabularies using the OpenSKOS Repository Service*

16:00 – 16:30 Coffee break

16:30 – 17:45 Metadata Applications: Overview Presentations and Demos

Peter Withers, *Metadata Management with Arbil*

Matej Durco, Daan Broeder and Menzo Windhouwer, *SMC4LRT - groundwork for query expansion and semantic search*

Martine de Bruin, Marc Kemps-Snijders, Jan Pieter Kunst, Maarten van der Peet, Rob Zeeman and Junte Zhang, *Applying CMDI in real life: the Meertens case*

Demos

17:45 – 18:30 Final Discussion and Wrap-up

Editors

Victoria Arranz	ELDA/ELRA, Paris, France
Daan Broeder	MPI, Nijmegen, The Netherlands
Bertrand Gaiffe	ATILF, Nancy, France
Maria Gavrilidou	ILSP/Athena R.C., Athens, Greece
Monica Monachini	CNR-ILC, Pisa, Italy
Thorsten Trippel	Universität Tübingen, Tübingen, Germany

Workshop Organizing Committee

Victoria Arranz	ELDA/ELRA, Paris, France
Daan Broeder	MPI, Nijmegen, The Netherlands
Bertrand Gaiffe	ATILF, Nancy, France
Maria Gavrilidou	ILSP/Athena R.C., Athens, Greece
Monica Monachini	CNR-ILC, Pisa, Italy
Thorsten Trippel	Universität Tübingen, Tübingen, Germany

Workshop Programme Committee

Helen Aristar-Dry	Michigan State University, USA
Núria Bel	UPF, Barcelona, Spain
António Branco	University of Lisbon, Portugal
Lars Borin	Språkbanken, Göteborg, Sweden
Khalid Choukri	ELDA/ELRA, Paris, France
Thierry Declerck	DFKI, Saarbrücken, Germany
Matej Durco	Austrian Academy of Sciences, Vienna, Austria
Gil Francopoulo	CNRS-LIMSI-IMMI + TAGMATICA, Paris, France
Francesca Frontini	CNR-ILC, Pisa, Italy
Olivier Hamon	ELDA/ELRA, Paris, France
Erhard Hinrichs	Universität Tübingen, Tübingen, Germany
Penny Labropoulou	ILSP/Athena R.C., Athens, Greece
Jan Odijk	Universiteit Utrecht, The Netherlands
Elena Pierazzo	Kings' College, London, UK
Laurent Romary	INRIA, Nancy, France
Andreas Witt	IDS, Mannheim, Germany
Peter Wittenburg	MPI, Nijmegen, The Netherlands
Tamás Varadi	Hungarian Academy of Sciences, Budapest, Hungary
Marta Villegas	UPF, Barcelona, Spain
Sue Ellen Wright	Kent State University, USA

Table of contents

Author Index

Preface/Introduction

The description of Language Resources (LRs) continues to be a crucial point in the lifecycle of LRs, and more particularly, in their sustainable exchange. This has been so for a number of repositories or LR distribution centres in place (ELRA, GSK, LDC, OLAC, TST-Centrale, BAS, among others), who house LR catalogues following some proprietary metadata schema. A number of projects and initiatives have also focused these past few years in the sharing of LRs (ENABLER, CLARIN, FLaReNet, PANACEA, META-SHARE), for example, for Language Technology (LT).
Based on these initiatives a consensus emerges that shows a number of requirements for standardized metadata:

- There should be a common publication channel for the LR descriptions in the world.
- This channel allows users to carry out easy and efficient LR data discovery and possible subsequent retrieval of LRs.
- Expert knowledge is required to create the data model for the metadata description.
- Subject matter experts (both researchers and LR/LT providers and developers) are required to provide the content for the data model.
- The data model needs to be clear, expressive, flexible, customizable and interoperable.
- Metadata have to provide for different user groups, ranging from providers to consumers (both individuals and organisations). This applies both to the information contained in the metadata and the supporting tool infrastructure for creating, maintaining, distributing, harvesting and searching the metadata.

Currently several initiatives focus on metadata. From the realm of work done within initiatives like ENABLER and CLARIN descended the Component MetaData Infrastructure (CMDI, ISO TC 37 SC 4 work item for ISO 24622), which allows the combination of standard data categories (for example from ISO 12620, isocat.org) to components, which are combined into metadata profiles. Early versions of this model have been operational in repositories such as ELRA's, which complied with the work done within INTERA. FLaReNet, as the result of a permanent and cyclical consultation, has issued a set of main recommendations where a global infrastructure of uniform and interoperable metadata sets appears among the Top Priorities for the field of LRs. For use within HLT, META-SHARE provides a fully-fledged schema for the description of LRs, in the framework of the component model, covering all the current resource types and media types of use, in all the stages of a resource's lifecycle. Our aim is to learn from one another's experiences and plans in this area.

The current state of the art for metadata provision allows for a very flexible approach, catering for the needs of different archives and communities, referring to common data category registries that describe the meaning of a data category at least to authors of metadata. Component models for metadata provisions are for example used by CLARIN and META-SHARE, but there is also an increased flexibility in other metadata schemas such as Dublin Core, which is usually not seen as appropriate for meaningful description of language resources.

Making resources available for others and putting this to a second use in other projects has never been more widely accepted as a sensible efficient way to avoid a waste of efforts and resources. However, when it comes to the details, there is still a vast number of problems. This workshop has aimed at being a forum to address issues and challenges in the concrete work with metadata for LRs, not restricted to a single initiative for archiving LRs. It has allowed for exchange and discussion and we hope that the reader finds the articles here compiled interesting and useful.

CMDI: a Component Metadata Infrastructure

Daan Broeder [1], Menzo Windhouwer[1], Dieter van Uytvanck[1], Twan Goosen[1], Thorsten Trippel[2]

[1]MPI for Psycholinguistics, Nijmegen, The Netherlands, [2]Eberhard-Karls-Universität Tübingen (Germany),
{daan.broeder|menzo.windhouwer|dieter.vanuytvanck|twan.goosen}@mpi.nl, thorsten.trippel@uni-tuebingen.de

Abstract

The paper's purpose is to give an overview of the work on the Component Metadata Infrastructure (CMDI) that was implemented in the CLARIN research infrastructure. It explains, the underlying schema, the accompanying tools and services. It also describes the status and impact of the CMDI developments done within the CLARIN project and past and future collaborations with other projects.

1 Introduction

Currently there is a fragmented world with respect to metadata for Language Resources (LR). However recently there have been initiatives that give some hope of creating interoperable schemas of high specificity that allow the creation comprehensive catalogues of LRs.

Before 2000 there were mainly the proprietary catalogues of the commercial companies and language resource centers as LDC and ELRA and the practice of inserting metadata in the transcriptions or annotation file headers as for example TEI and CHILDES formats support. Yet little attention was given to interoperability between archives and data centers using different metadata schema. Since 2000 we have seen the rise of new LR metadata schemas as IMDI, IMDI [2003], IMDI [2009] and OLAC but application and uptake of these schemas has been limited. Although OLAC is now used more or less as a standard for information exchange between LR archives, it is still delivering low specificity.

The experience in creating IMDI and trying to apply it to the variety of subdomains in linguistic research has helped to realize that a single metadata schema cannot succeed in conquering all sub fields of linguistics. The differences in needs, terminology and traditions will prevent uptake and acceptance of such a schema. Therefore, when there was a need to come to a comprehensive approach for metadata within the CLARIN infrastructure [Váradi et al., 2008], we chose to build an infrastructure permitting many different schemas to co-exist and supporting semantic interoperability by using a separate 'pragmatic reference system' for the semantics being implied. To support users with a low threshold for creating new schemas and reusing existing work at a conceptual level, an approach was chosen where small reusable snippets of metadata schema's can be created and recombined to form complete new schemas. This component based approach or Component Metadata Infrastructure [CMDI, Broeder et al.] is based on well-defined formal schemas and explicit semantics by using registries for the schema components, the final schema and the pragmatic ontology.

2 CMDI overview

CMDI is a flexible framework for metadata modelers and metadata creators to create and use appropriate metadata schemas for describing resources. It aims at making the metadata modeling process easy by allowing reuse of different snippets of metadata schemas or metadata components that bundle descriptions for certain resource characteristics. These components can be recombined to create a suitable metadata profile for describing a specific resource type. Components hence contain metadata elements or other components, forming profiles to be used either to describe singular resources or sets of related resources such as collections. Figure 1 illustrates the model.

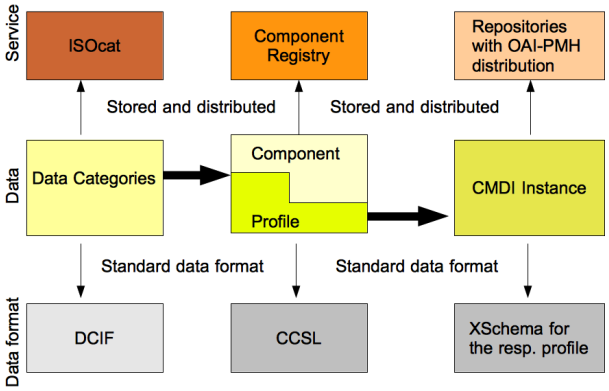


Figure 1: Model of the component metadata infrastructure

Each of the constituents of the model has a three layer structure, from the bottom: a data format, the data and a service storing and distributing the data.

Metadata modelers are able to use their own terminology deemed appropriate for the task in the components. This flexible use of terminology inevitably also creates semantic interoperability problems that we try to solve using a 'pragmatic' ontology, which is a combination of a concept registry — more specific the ISO data category registry [ISOcat] — and a relation registry [RELcat, Schuurman and Windhouwer, 2011]. The data in ISOcat is available in the Data Category Interchange Format (DCIF), which is a standard format as defined by ISO 12620 [2009].

Metadata registry and relation registy together provide the semantics of the metadata terms used and make possible relations between the metadata concepts explicit. Metadata modelers may also use their own terminology — or terms in their own language — for elements in the metadata components and remain interoperable by linking the component elements to the corresponding data category entry in ISO-

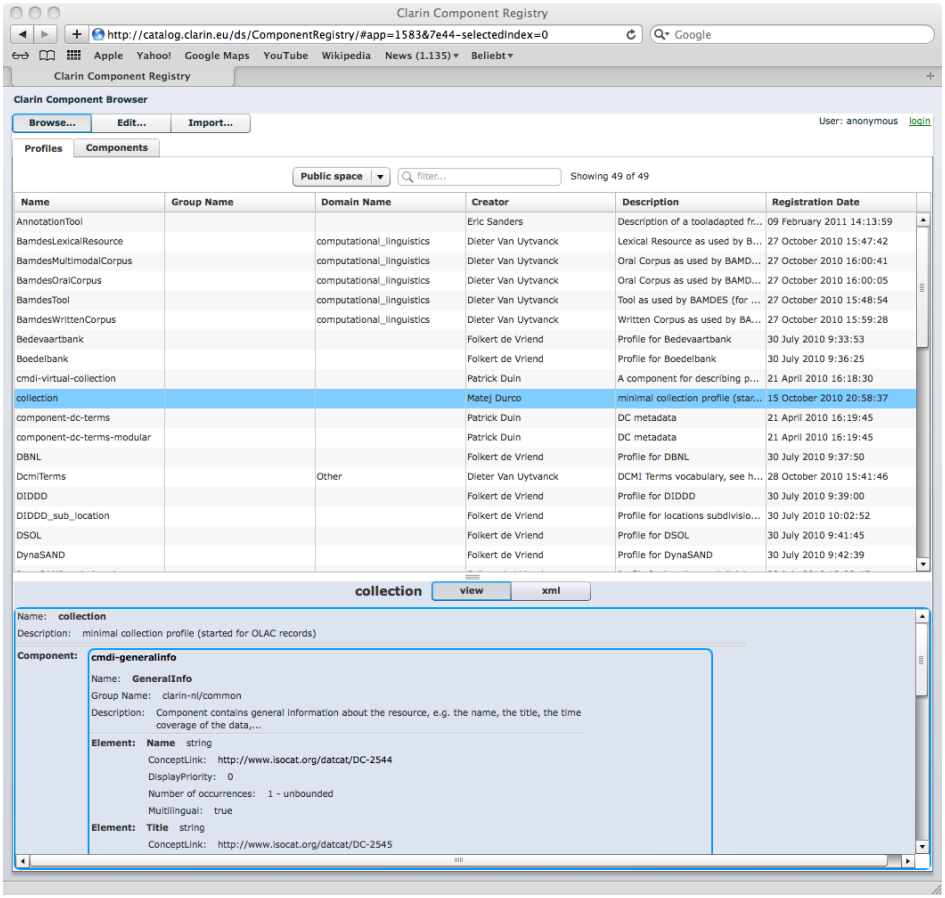


Figure 2: The CMDI Component Registry

cat. Different but similar elements can refer to the same entry if they are semantically equivalent or they can refer to different entries, for which their semantic similarity — their relation — is stored in the relation registry (RELcat).

The metadata components, the combined components and profiles are stored in the CMDI component registry, depicted in the center of Figure 1. They are defined in the CMDI-Component Specification Language (CCSL) and distributed using a REST-based API or a browser interface. Users can browse this registry and combine existing components in a new profile (see Figure 2). New components can be created using the component editor and stored in the component registry.

For creating actual instantiations of the metadata profiles, these are automatically transformed into XML schemas, also available from the component registry. They are used for validating the metadata instances, the metadata records that describe actual resources. These can be created in a variety of ways, for example by transforming legacy data. For direct creation we have developed ARBIL which is a versatile metadata editor. ARBIL allows users to manipulate and edit metadata of many metadata records by using table structures instead of the unformated XML-code.

Within the CLARIN infrastructure, CMDI is the metadata infrastructure of choice. The different CLARIN centers and others that act as LR providers share and distribute their metadata in CMDI format via OAI-PMH to be harvested by CMDI service providers. The right side of Figure 1 illustrates that. Such service providers may choose to harvest all or a sub-set of CMDI data-providers and aggregate the metadata in metadata catalogs. Within CLARIN we have currently the following catalogs: the CLARIN VLO [van Uytvanck et al., 2010] and the Meertens Institute CMDI Catalogue [CMDI MI Search Engine] and outside CLARIN there is the NaLiDa faceted browser [see NaLiDa FB], see Figure 3.

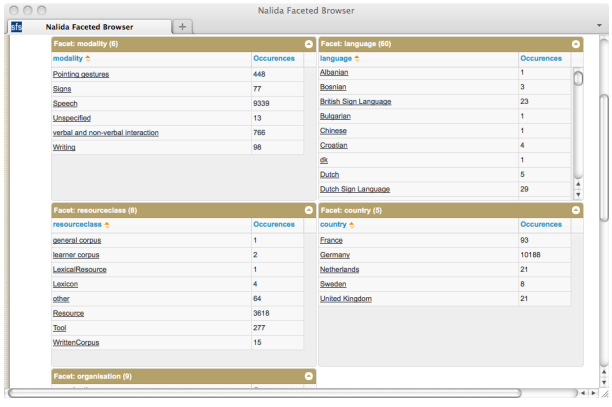


Figure 3: Faceted Browser for CMDI metadata

All the approaches mentioned above offer a faceted browser for structured access to the repositories, often combined

with a full text search of the metadata for example by using the Apache SOLR and Lucene combination [see SOLR]. This allows users to navigate in the harvested collection's metadata by defining criteria to be fulfilled by the searched resources. The possible criteria provided by a faceted browser are the facets, for this reason it is important to choose the facets appropriately to the intended use. As the facets may refer to different instantiations of similar data categories it seems appropriate to also use the terms and mappings from the ISOcat registry. The last requirement has been partly fulfilled in the VLO where facets are based on ISOcat data categories.

For more complex combinations of search terms, faceted browsing and searching has some limits, for example in the number of facets. To overcome these limits the Austrian CLARIN project is working on a prototype supporting complex CMDI metadata search queries. This work in progress aims at power-users that have a good grasp all the aspects of the CMDI infrastructure including the possibilities of varying precision and recall by varying the semantic mapping variables [Durco et al., 2012, submitted]. This prototype is the most complete implementation of the complete CMDI architecture that is shown in Figure 4.

The complete system for CMDI metadata creation and exploitation is depicted in Figure 4. At the (left) exploitation side CMDI metadata is harvested and put in a joint CLARIN metadata repository. There it is either consumed by simple but effective faceted browser tools as the VLO or by complex ones as the Austrian MD Search, making use of Semantic Mapping services provided by the pragmatic ontology using the combination of ISOcat and the Relation Registry.

3 Standardization efforts

An important step in making the component metadata approach successful and sustainable for long time archiving, is aiming a standardization of the framework. This also offers an opportunity to cooperate with like-minded projects such as META-SHARE [see Gavrilidou et al., 2011], which also wants to use a component metadata approach, to achieve interoperability. The standardization is running under the auspices of ISO TC37/SC4 offering an institutionalized platform for the involvement of relevant parties such as META-SHARE and CLARIN, the communities currently working with metadata components. This standardization bodies technical committee is also governing the means for solving semantic interoperability issues, the ISOcat data category registry, with ISO 12620:2009 being hosted by the sister subcommittee ISO TC37/SC3.

An important element of CMDI is the use of ISOcat to help solve issues of semantic interoperability where metadata modelers use different terminology. ISOcat is positioned as a general registry for linguistic data category definitions, and it was natural for the component metadata initiatives in the LR domain such as those from CLARIN and META-SHARE to use ISOcat to register metadata concept definitions. Currently, a group of experts informally termed 'Athens Core´ that is a broad representation from the LR community pushes the metadata concept ISO standardization process forward. More details on the CMDI related standardization processes are found in Broeder et al. [2012]).

4 Status of CMDI usage

Currently we know of the different national CLARIN projects, the German NaLiDa project and some smaller projects that have been using CMDI implementations or are planning to use it. We expect there to be some papers at the LREC 2012 'Describing Language Resources' workshop.

The VLO currently lists over 180000 resources, described by metadata, the component registry lists 49 different profiles and 218 components in the public section (as of February 2012), with about 15 commiters from various institutions. There are 62 registered users of the component registry. Registered users here means that they have created and modified components, read access does not require registration. Besides the public profiles and components there are currently 127 private profiles and 303 private components showing very active development going on.

5 Conclusion and future initiatives

It is too early to come to any final conclusions about the success of component metadata, also because its success cannot be measured only in acceptance by the metadata creators. It also depends if outside users can use CMDI to locate the resources they require, hence the success is depending on tools to work with CMDI.

At the moment on the metadata production side, things are coming along although some attention needs to be paid to the risk of insufficient reuse of existing CMDI components and profiles and proliferation of different profiles. At the metadata exploitation side there remain many challenges but we trust that there will be several solutions also because CLARIN centers are accepting CMDI tagged resources and will need to provide metadata exploitation solutions for their own users as well as for outside users.

We think that a communal standardization initiative of CLARIN and META-SHARE will lead to an acceptable implementation for all groups that are pledged to the use of metadata components and explicit semantics using ISOcat.

Another aspect of component metadata is that it is a very good candidate to be used by the projects working on research infrastructures catering for a variety of communities and disciplines. They have to deal with a large variety of data types and have to bridge differences in terminology used by different communities. One example is DASISH which is a community cluster project combining linguistics, wider humanities and the social sciences where CMDI could be successfully applied.

6 Acknowledgements

Work for this paper was conducted within the Clarin-NL project and in the NaLiDa project funded by the German Research Foundation (DFG) in the program for Scientific Library Services and Information Systems (LIS).

References

ARBIL. http://www.lat-mpi.eu/tools/ARBIL.

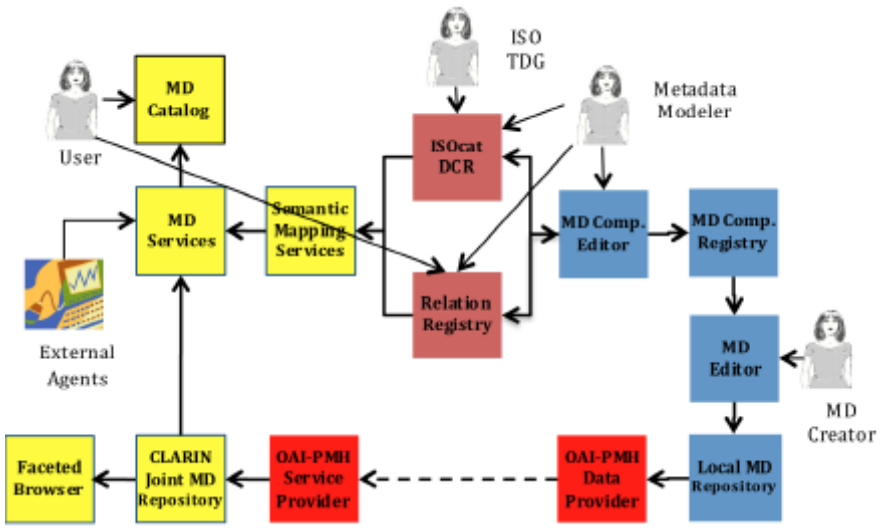


Figure 4: The CMDI Architecture

D. Broeder, O. Schonefeld, T. Trippel, D. van Uytvanck, and A. Witt. A pragmatic approach to xml interoperability - the component metadata infrastructure (CMDI). In *Balisage: The Markup Conference 2011*, volume 7.

D. Broeder, D. van Uytvanck, M. Gavrilidou, and T. Trippel. Standardizing a component metadata infrastructure. In *Proceedings of the 8th Conference on International Language Resources and Evaluation (LREC 2012)*, Istanbul, 2012.

CHILDES. http://childes.psy.cmu.edu.

CMDI. http://www.clarin.eu/cmdi.

CMDI MI Search Engine. CMDI Meertens Institute Search Engine. http://www.meertens.knaw.nl/cmdi.

DASISH. http://www.lat-mpi.eu/latnews/tag/dasish/, project website forthcoming.

M. Durco, D. Broeder, and M. Windhouwer. Semantic mapping - groundwork for query expansion and semantic search. 2012, submitted.

M. Gavrilidou, P. Labropoulou, S. Piperidis, M. Monachini, F. Frontini, G. Francopoulo, V. Arranz, and V. Mapelli. A metadata schema for the description of language resources (lrs). In *5th International Joint Conference on Natural Language Processing (IJCNLP 2011)*, 2011.

IMDI. http://www.mpi.nl/IMDI.

IMDI. Metadata elements for session descriptions, draft proposal version 3.0.4. `http://www.mpi.nl/IMDI/documents/Proposals/IMDI_MetaData_3.0.4.pdf`, October 2003.

IMDI. Metadata elements for catalogue descriptions, version 3.0.13. `http://www.mpi.nl/IMDI/documents/Proposals/IMDI_Catalogue_3.0.0.pdf`, August 2009.

ISO 12620. Terminology and other language and content resources - specification of data categories and management of a data category registry for language resource. Technical report, ISO, 2009.

ISOcat. http://www.isocat.org.

META-SHARE. http://www.meta-net.eu/meta-share, http://www.meta-share.eu/.

NaLiDa. http://www.sfs.uni-tuebingen.de/nalida/en/.

NaLiDa FB. NaLiDa faceted browser. http://www.sfs.uni-tuebingen.de/nalida/en/catalogue.html.

OLAC. http://www.language-archives.org/.

I. Schuurman and M. Windhouwer. Explicit semantics for enriched documents. what do isocat, relcat and schemacat have to offer? In *2nd Supporting Digital Humanities conference (SDH 2011)*, Copenhagen, November 2011.

SOLR. http://lucene.apache.org/solr/.

TEI. http://www.tei-c.org/.

D. van Uytvanck, C. Zinn, D. Broeder, P. Wittenburg, and M. Gardelini. Virtual language observatory: The portal to the language resources and technology universe. In *Proceedings of the 7th conference on International Language Resources and Evaluation*, Malta, 2010.

VLO. http://catalog.clarin.eu/ds/vlo/.

T. Váradi, P. Wittenburg, S. Krauwer, M. Wynne, and K. Koskenniemi. Clarin: Common language resources and technology infrastructure. In *Proeceedings of the 6th International Conference on Language Resources and Evaluation (LREC)*, 2008.

The META-SHARE Metadata Schema: Principles, Features, Implementation and Conversion from other Schemas

Maria Gavrilidou*, Penny Labropoulou*, Elina Desipri*, Ioanna Giannopoulou[&], Olivier Hamon[&], Victoria Arranz[&]

*Athena R.C./ILSP, [&]ELDA

*Athens, Greece, [&]Paris, France,

E-mail: {maria, penny, elina}@ilsp.athena-innovation.gr, {ioanna, hamon, arranz}@elda.org

Abstract

The current paper focuses on the presentation of a metadata model for the description of language resources proposed in the framework of the META-SHARE infrastructure, aiming to cover both datasets and tools/technologies used for their processing. It presents the rationale/background for its creation, the basic principles and features of the model, describes the process and results of its current application to the META-SHARE nodes, including the conversion from previous schemas, and concludes with work to be done in the future for the improvement of the model.

Keywords: metadata, META-SHARE, LRs description

1. Introduction

The META-SHARE metadata model is used for the description of Language Resources (LRs) focusing on the area of Human Language Technology (HLT). These resources encompass both data (textual, multimodal/multimedia and lexical data, grammars, language models, etc.) and technologies (tools/services) used for their processing. Its primary objective is to support the functions of META-SHARE (www.meta-share.eu), which is an open, integrated, secure and interoperable exchange infrastructure dedicated to LRs.

More specifically, META-SHARE serves as a space where LRs are documented, uploaded and stored in repositories, catalogued and announced, downloaded, exchanged and discussed, aiming to support a data economy. META-SHARE brings together knowledge about LRs and related objects and processes and fosters their use by providing easy, uniform, one-step access to LRs through the aggregation of LR sources into one catalogue; it facilitates LRs' search and retrieval processes, and encourages (re-)use and new use of LRs (Piperidis, 2012). The metadata descriptions constitute the means by which LR producers describe their resources and LR users identify the resources they seek. Thus, the META-SHARE metadata model forms the core engine driving the META-SHARE access interfaces to the LRs catalogue.

In this framework, interoperability at all levels is a critical issue. The adoption of a common metadata schema for all HLT resources, with mappings to other widespread schemas in the broader area of LRs, is crucial to the success of the endeavour.

The current paper gives an overview of such schema, going from its background and design principles to its description features. Then we guide the reader through the process and results of its current implementation on the META-SHARE nodes, with a focus on the import of LRs and the work done on the conversion of other proprietary metadata schemas into the one proposed in this work. An analysis of the findings so far is also provided together with a to-do list for the coming work.

2. Background

A variety of metadata schemas and sets of descriptive elements from LR catalogues are already available for the description of LRs in the wider area of language-related activities, as seen in Gavrilidou et al. (2011). However, interoperability problems between them are evident, given that they come from various backgrounds and focus on the needs of the specific communities that have devised them.

To overcome these issues, the ISO Data Category Registry (ISO 12620, 2009)[1] has been set up. The ISOcat DCR caters for semantic interoperability through the registration of *elements* ("data categories"), which refer to widely used concepts in the linguistics domain; users can then link their own elements to them (or add new ones according to the ISO 12620 framework requirements), thus achieving common terminology. A thematic area dedicated to metadata is included therein.

The component-based mechanism, as described in Broeder et al. (2008) and in Broeder et al. (2010), complements the ISOcat DCR by introducing the notion of *components*, which are groups of semantically coherent metadata elements and act as placeholders for well defined categories for the documentation of LRs (e.g. identification properties, usage, validation, licensing).

The META-SHARE metadata model builds upon this framework in order to provide the necessary equipment for describing LRs in the wider context of the Language Technology community. For the general principles of the model, we have taken into

[1] http://www.isocat.org

consideration the user needs (as collected through interviews with a variety of stakeholders and documented in Federmann et al. (2011)) as well as an overview of the most widespread metadata models in HLT and LR catalogue descriptions (Gavrilidou et al., 2011). As a result, we have adopted the following design principles:

- expressiveness: with the proposed LR typology we aim at covering any type of resource;
- extensibility: the modularity of the schema allows for future extensions, to cover more resource types as they become available; the schema will also cater for combinations of LR types for the creation of complex resources;
- semantic clarity: to achieve clear articulation of a term's meaning and its relations to other terms, each element of the schema is accompanied by a bundle of information constituting its identity, comprising its definition, its type, its domain and range of values, an example, the relations to other components/elements and links to the appropriate DC and ISOcat DCR terms (where applicable);
- flexibility: by the definition of a two-tier schema (minimal and maximal), we cater for the possibility for exhaustive but also for minimal descriptions (cf. Section 5);
- interoperability: this is guaranteed through the mappings to widely used schemas (mainly DC, and ISOcat DCR).

3. The META-SHARE ontology

The META-SHARE focus lies on the description of LRs, covering, as aforesaid, data resources and tools/services used for their processing.

META-SHARE remains at the level of *resource* rather than *individual item*, in the sense that it targets to describe whole sets of text/audio/video etc. files (corpora), sets of lexical entries (lexical/conceptual resources), integrated tools/services and so on, rather than individual items. For individual items, the META-SHARE model refers users to the recommended standards and/or best practices reported in (Monachini et al., 2011). However, the schema can handle *resource parts*, which are crucial for all multimedia-type resources, for instance, and it has in mind *resource collections*, which will be handled in the near future.

The central entity of the META-SHARE ontology is the LR per se. However, in the ontology, LRs are linked to other satellite entities through relations that in the model are represented as basic elements. The interconnection between the LR and these satellite entities pictures the LR's lifecycle from production to use: reference documents related to the LR (papers, reports, manuals, etc.), persons/ organizations involved in its creation and use (creators, distributors, etc.), related projects and activities (funding projects, activities of use, etc.), accompanying licenses, etc. Thus, the META-SHARE model recognizes the following distinct entities:

- the *resource* itself, i.e. the LR being described,
- the *actor*, further distinguished into *person* and *organization*,
- the *project*,
- the *document*, and
- the *licence*.

It should be noted, however, that the satellite entities are described only when the case arises, i.e. when linked to a specific resource. For their description, the metadata schema takes into account schemas and guidelines that have been devised specifically for them (e.g. BibTex for bibliographical references).

4. Main features of the model

As aforesaid, the META-SHARE metadata model is inspired by the component-based mechanism. *Components* consist of semantically coherent *elements* (data categories) that encode specific descriptive features; elements are also used to represent *relations* in the current version of the schema. The relation mechanism represents the encoding of linking features between resources. Relations hold between various forms of a LR (e.g. raw and annotated resource), different LRs included in the META-SHARE repository (e.g. a language resource and a tool that has been used to create it, etc.) but also between LRs and satellite resources such as standards used, related documentation, etc.

The core of the model is the *resourceInfo* component (Figure 1), which contains all information relevant for the description of a resource. It subsumes components that combine together to provide the full description of a resource.

Administrative components are common to all LRs and provide information on the various phases of the resource's life cycle, e.g. creation, validation, usage, distribution, etc. (LRSLM, 2010).

Further sets of components are provided depending on the LR type. The META-SHARE model recognises two main classification axes: *resourceType* and *mediaType* (i.e. the medium on which the LR is implemented). This choice has been dictated by the fact that they both bring to the description of the LRs distinct sets of features: for instance, *resourceType*-specific information includes annotation features (for corpora), types of encoding contents (for lexica and grammars), performance (for grammars), while *mediaType*-specific information refers to the actual medium of the LR, and includes features like format (wav/avi etc. for videos, txt/doc/pdf/xml for texts etc.) and size (sentences/words/bytes for text corpora, duration for audio/video corpora, entries/items for lexica, etc.).

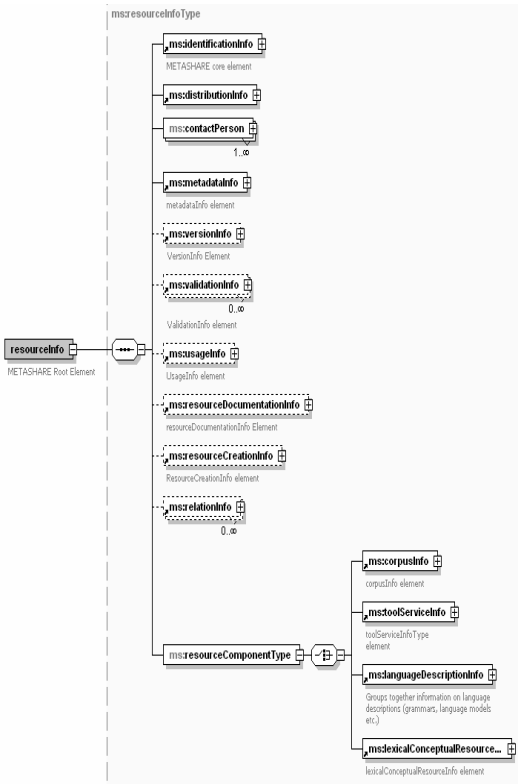


Figure 1: Common components for all LRs and resourceType components

More specifically, the following four values are suggested for the element *resourceType*:

- *corpus* (including written/text, oral/spoken, multimodal/multimedia corpora),
- *lexical/conceptual resource* (including terminological resources, word lists, semantic lexica, ontologies, etc.),
- *language description* (including grammars, typological databases, courseware, etc.),
- *tool/service* (including processing tools, applications, web services, etc. required for processing data resources).

Each LR receives only one *resourceType* value, but naturally it may take more than one *mediaType* value since LRs can consist of parts belonging to different types of media: for instance, a multimodal corpus includes a video part (moving image), an audio part (dialogues) and a text part (subtitles and/or transcription of the dialogues); a multimedia lexicon, besides the textual part, also includes a video and/or an audio part. Thus, for each part of the resource, the respective feature set (components and elements) should be used: e.g. for a spoken corpus and its transcriptions, the audio feature set will be used for the audio part and the text feature set for the transcribed part.

The following media type values are foreseen: *text*, *audio*, *image*, *video*. Two additional values are introduced, although they are not really distinct media type values: these correspond to numerical text resources (value *textNumerical*) and n-grams (value *textNgram*). These are actually subtypes of text resources but they present further descriptive particularities due to their contents: numerical data (e.g. biometrical, geospatial data, etc.) for the former, and items with frequency counts for the latter.

LR type-specific components are all located under the *resourceComponentType* component. Similarly, for each LR type, particular medium-dependent components are created to group together sets of features relevant to each LR/media type, given that media types and the relevant information differs across LR types; these are again grouped under an *xMediaType* component, where x stands for each of the LR type values. For instance, *corpusTextInfo*, *corpusAudioInfo*, *corpusVideoInfo*, *lexicalConceptualResourceTextInfo*, *lexicalConceptual- ResourceVideoInfo*, etc. provide information depending on the media type of each LR type and include the *mediaType* element with the values *text*, *audio*, *video* etc. accordingly.

Broadly speaking, the resource/media type-specific components[2] cover the following information types:

- contents: components mainly referring to languages covered in the resource, types of content (e.g. for images: drawings, photos, histograms, animations etc.), modalities included (e.g. written / spoken language, gestures, eye movements, etc.), etc.
- classificatory information: components including resource-type subclassification (e.g. subtypes of lexical/conceptual resources, tools/services etc.) as well as classification of the contents of the resource; this can be cross-media (e.g. domains, geographic and time coverage, etc.) as well as media-dependent (e.g. text type, audio genre, setting, etc.).
- formatting: file format, character encoding etc.; obviously, this information is more media-type-driven (e.g. different file formats for text, audio and video files).
- details on creation: it refers to the creation of the specific resource parts, e.g. the original source, the capture and recording methods (scanning and web crawling for texts vs. recording methods for audio files). These components are different from the *resourceCreationInfo* component attached at the resource level, which is used to give information on anything that concerns the creation of all resource and media types (e.g. creation dates).

[2] In fact, it should be noted that components are divided into three classes: (a) components common to all types of resources ("administrative" ones), (b) components re-usable for more than one resource / media type but not globally applicable (e.g. capture information for audio, video and image resources) and (c) the ones strictly applied to specific resource and media types (e.g. evaluation for tools, audio content for audio resources).

- performance: information regarding the performance of the resource; it is resource-type driven, given that the measures and criteria differ across resource types.
- operation: information relevant to the operation requirements of the resource (e.g. the hardware and software prerequisites for running a tool/service).
- input and output: these components are specific to tools/services; they can be used to provide information on the media type, format, language, etc. that the tool/service can take as input and the resulting output.
- finally, the special *linkToOtherMediaInfo* component,is provided for linking between the various media type parts of the resource.

5. Maximal vs. minimal schema

The set of all the components describing specific LR types and subtypes constitute the *profile* of each type. When describing a resource, users are presented with proposed profiles for each type, which can be used as templates and guidelines for the completion of the metadata description. Moreover, exemplary instantiations (e.g. for wordnet-type resources, for parallel corpora, for multimodal resources, for treebanks, etc.) will be made available as guiding assistance to LRs metadata providers.

In order to accommodate flexibility, both components and elements belong to two basic levels of description (stepwise approach):

- an initial level providing the basic components/elements for the description of a resource (*minimal schema*), and
- a second level with a higher degree of granularity (*maximal schema*), providing detailed information on a resource and covering all stages of LR production and use.

The minimal schema contains those elements considered indispensable for LR description (from the provider's perspective) and identification (from the consumer's perspective).

As regards the administrative set, there are four obligatory components: 1) addressing the needs for *identification* of the resource (its name, identifier, a free text description), 2) the *terms under which it may be distributed* and, if available, licensing details, 3) details of the *contact person* (surname and email) and 4) information on the *creation of the metadata record* (at least the date of creation).

Further obligatory components and elements are specified for each LR/media type. In general, the mandatory information is restricted to basic information so as not to intimidate metadata creators: size and languages for datasets, subtype for all (with value sets depending on the resource type), encoding level for language descriptions, etc. The further characterisation of specific components and elements as "recommended" prompts the providers to input richer descriptions for their resources.

6. Implementation of the model and workflow

The model has been automatically implemented as an XML schema (Federmann et al., 2012), documented also as a user manual[3], which contains detailed information, including definitions, examples and guidelines for the usage of the whole schema and each element (Desipri et al., 2012).

The model is conceived as a living entity evolving according to needs and developments in the field. The workflow that has been adopted consists of an iterative process of proposal, evaluation and updating. More specifically, a proposal is drawn by the core META-SHARE metadata group and put up for evaluation by metadata experts and the related projects (CESAR, METANET4U and META-NORD) who actually use the schema for the description of their LRs. Their comments and feedback are taken up by the core group, who coordinates the tasks and decides upon the updating of the schema. This workflow allows us to better coordinate the relevant activities, especially as META-SHARE has already integrated into a uniform catalogue the resources of 38 organisations. Browsing thereof is performed on the basis of the common metadata schema.

7. META-SHARE environment

An integrated environment for the support of META-SHARE functions has been developed (Federmann et al., 2012), serving both LR consumers and providers:

- Consumers of LRs (end users) will be able to: register and create a user profile, log-in to the repository network (single sign-on), browse and search the central inventory using search facilities, access the actual resources by visiting the local (or non-local) repositories for browsing and downloading them, get information about the usage of specific resources, their relation (e.g. compatibility, suitability, etc.) to other resources, as well as recommendations, download resources accompanied by easy-to-use licensing templates, including both free and for-a-fee resources, provide feedback about resources and exploit additional functionalities.
- Providers of resources will additionally be able to: create from scratch, store and edit resource descriptions by using the metadata editor, convert from an existing metadata schema into the META-SHARE metadata model (cf.

[3] cf. http://www.meta-net.eu/meta-share/META-SHARE%20%20documentationUserManual.pdf

Section 9), upload actual resources directly or by contacting support staff for large volume resources, get reports and statistics on number of views, downloads, types of consumers, etc. of LRs, as well as feedback from consumers.

META-SHARE is open-source, available on GitHub at https://github.com/metashare/META-S HARE.

8. Importing Language Resources in META-SHARE

In order to populate META-SHARE, it is possible to provide XML metadata files for each LR that follows the META-SHARE metadata. This allows skipping the use of the META-SHARE metadata editor when a large amount of metadata descriptions is available. Therefore, a LR provider may create his/her own XML files compliant with the META-SHARE *XML schema (XSD)*.

The importing of LRs may be done by using either the META-SHARE editor (see previous section) or an import script. Both can handle either single XML files or ZIP archives. Although the importing process can take some time, it really facilitates the population of existing LRs into META-SHARE. After successful import of these files, the LR provider should have the new LRs available on his/her META-SHARE node.

9. Conversion of existing LRs

Given the existence of various metadata schemas already used for the description of LRs by different organizations, the (semi-)automatic conversion thereof to the META-SHARE schema is particularly important. To illustrate the effort required for this endeavour, we present the conversion process adopted for LRs derived from the ELRA catalogue (1,008 as a whole), which has been made using XSL transformations. This was particularly needed given the large number of LRs to be imported.

It should be mentioned, though, that LRs have also been converted from other META-SHARE partners (CNR, DFKI, FBK and ILSP) as well as from collaborating projects (CESAR, METANET4U and META-NORD). Moreover, this conversion has been adapted recently, according to the updates in the latest version of the metadata schema.

Regarding the conversion process, for a given list of LRs, the basic steps are the following:

1. *Preparing* XML files containing the metadata descriptions of the LRs, following the resources' original XML schema.
2. *Mapping* the elements of the original XML schema to the elements of META-SHARE XML schema, one by one.
3. *Creating* an XSL file: this is an XML file containing the structure of META-SHARE XML schema, along with XSL transformations, which allow harvesting the information from the files of the original XML schema format.
4. *Running* the conversion for the XML files.

9.1 Preparing the XML files

The metadata of the existing LRs may be stored in different formats (e.g. database entries, ontology entities, etc.). The first step for the conversion process is to extract this metadata in XML format, a trivial task in most cases.

In the case of ELRA's LRs conversion, the XML files extracted from the database presented the metadata in a highly complex way, mostly due to the complexity of the ELRA database schema. To facilitate the conversion process, these XML files were converted into very simple XML "flat list" files, with the following format:

```
<resource id="1">
    <resource_reference>ELRA-S0148</resource_reference>
    <resource_fullname
language="English">WEBCOMMAND</resource_fullname>
<date_added>2004-09-14</date_added>
    …
</resource>
<resource id="2">
    <resource_reference>ELRA-S0034</resource_reference>
    …
</resource>
```

Files with this format have served as the input to the conversion process, which is described below.

9.2 Mapping between the two Schemata

An essential part of the pre-conversion process is to map the elements of the two schemata in order to have a clear list of corresponding fields. This has been done by metadata experts with a good understanding of the description contents.

In some conversion cases, this mapping was done rather fast since only a few elements had to be mapped. However, in the case of more complex input schemas, like ELRA's, the mapping could have taken much longer. In that particular case, the mapping of around 100 elements from the ELRA catalogue required various meetings of the metadata team and needed several days of work.

On the technical side, the need for visualising these mappings was covered by the Altova-MapForce software[4] (Figure 2), which has proved to be very helpful given the large amount of elements in the META-SHARE schema. Within this software, the original "input" and the META-SHARE "output" XML schemata were imported and the corresponding elements were linked with arrows.

[4] http://www.altova.com/mapforce.html

To give an estimate of the time needed to work on this software, around 100 elements in the ELRA schema have been mapped to the corresponding elements in the META-SHARE schema by one person in just a few minutes.

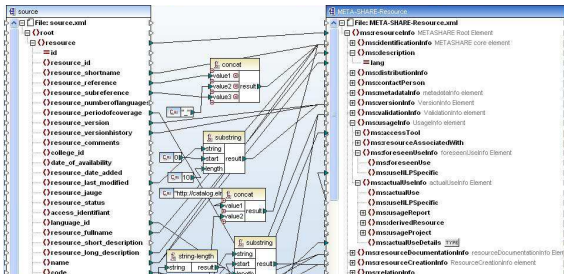


Figure 2: Screenshot from Altova-MapForce: mapping from ELRA into META-SHARE schema

9.3 Creating the XSL file

The XSL file should be looked at as the intermediate level between the input and the output schema. This file contains the XML structure of the output, i.e. the META-SHARE schema, and allows harvesting information from the input, e.g. the ELRA schema, by means of XSL transformations. Below follow two examples to illustrate the mappings taking place:

Example 1:

The element *resource_periodofcoverage* from the ELRA schema has been mapped to the *timeCoverage* element in META-SHARE schema.

Line in ELRA XML file:

```
…
<resource_periodofcoverage>Between 1992 and
1999</resource_periodofcoverage>
…
```

Lines in XSL conversion file:

```
…
<xsl:variable name='time' \
   select='preceding-sibling::resource_periodofcoverage'/>
     <xsl:if test='$time!=""'>
       <timeCoverageInfo>
         <timeCoverage>
           <xsl:value-of select='$time'/>
         </timeCoverage>
       </timeCoverageInfo>
     </xsl:if>
…
```

Lines in final META-SHARE XML file:

```
…
  <timeCoverageInfo>
    <timeCoverage>Between 1992 and 1999</timeCoverage>
  </timeCoverageInfo>
…
```

In this example, the element *timeCoverage* is part of the META-SHARE XML structure. The content inside the element is added by the line *<xsl:value-of select='$time'/>*, where *time* is a XSL variable, defined before, as the *XPath* "*preceding-sibling*" with the name "*resource_periodofcoverage*". If the variable is empty, the element *timeCoverageInfo* will not take place at all in the final XML file.

The size and complexity of the META-SHARE schema as well as the significant differentiation between this and the original formats, called for far more complicated transformations than the one previously displayed. Some examples are META-SHARE mandatory fields, where information was required but could be missing in some files, repeated fields for which complex loops were needed, elements with enumeration constraint, for which exhausting controls were developed, etc.

Example 2:

Here is the case where the ELRA LR metadata contains the description of some documentation files that come with it. Here the mapping of the elements is more complex and so are the transformations.

Line in ELRA XML file:

```
…
  <file_id>496</file_id>
  <file_fullname>The full name of the file 496</file_fullname>
  <type>S</type>
  <file_name>S_S0153_DB.pdf</file_name>
   <file_id>497</file_id>
  <file_fullname>The full name of the file 497</file_fullname>
  <type>A</type>
  <file_name>A_S0154_DB.pdf</file_name>
…
```

Lines in XSL conversion file:

```
…
<xsl:for-each select="*[local-name()='file_id' and
namespace-uri()='']">
  <resourceDocumentationInfo>
    <documentation>
      <documentInfo>
        <xsl:choose>
          <xsl:when test="string(.)='S'">
            <documentType>unpublished </documentType>
          </xsl:when>
          <xsl:when test="string(.)='A'">
            <documentType>article</documentType>
          </xsl:when>
        </xsl:choose>
        <title>
          <xsl:value-of \
              select="following-sibling::file_fullname[1]"/>
        </title>
      </documentInfo>
    </documentation>
    <xsl:if test="string(.)='S'">
      <samplesLocation>
        <xsl:value-of select="concat('http://catalog.elra.info/ \
            product_info.phpaction=download&amp; \
            filename=', following-sibling::file_name[1])"/>
      </samplesLocation>
    </xsl:if>
```

```
    </resourceDocumentationInfo>
</xsl:for-each>
…
```

Lines in final META-SHARE XML file:

```
…
<resourceDocumentationInfo>
   <documentation>
      <documentInfo>
         <documentType>unpublished</documentType>
         <title>The full name of the file 496</title>
      </documentInfo>
   </documentation>
   <samplesLocation> \
      http://catalog.elra.info/product_info.php \
      action=download&amp;filename=S_S0153_DB.pdf \
   </samplesLocation>
</resourceDocumentationInfo>
<resourceDocumentationInfo>
   <documentation>
      <documentInfo>
         <documentType>article</documentType>
         <title>The full name of the file 497</title>
      </documentInfo>
   </documentation>
</resourceDocumentationInfo>
…
```

In the example above, the XSL contains a loop for all the *file* metadata, as there may be more than one. The *documentType* is an element with enumeration constraint (i.e. it can only contain a value from a set of acceptable values). A control is checking the *type* of the file (A for article, S for sample, etc.) and assigning a valid value to this element. The *title* element is filled in with the content of *file_fullname.* Finally, the *samplesLocation* element only occurs if the *type* is a sample, and is filled in by a concatenation of strings: the fixed prefix of the ELRA URL and the name of the file.

In the case of ELRA's LRs conversion, simple XSL transformations, like the one described in the first example, were automatically produced by the Altova-MapForce software after the mapping. The more complex transformations, like the one described in the second example, were manually developed, which required several days of work.

9.4 Running the conversion for all LR files

The production of the META-SHARE compliant files was performed by applying the transformations of the XSL file on all the files of the original schema. A JAVA program was developed and used to run the conversion for ELRA's LRs. The duration of the conversion depends on the amount of LRs, but generally it is quite fast (ELRA's 1,008 LRs were converted within 3 minutes).

The conversion, however, was not the end of the process, as the produced files had to be successfully imported into META-SHARE. The import served as a kind of debugging process as it highlighted the errors on the validation of the files and helped improving and finalizing the XSL file.

10. Current situation and observations on LR description trends

The schema has been adopted by the different node repositories within META-SHARE, namely CNR, DFKI, ELDA, FBK and ILSP, as well by its related projects CESAR, METANET4U and META-NORD. All of them have converted their LR descriptions into the latest version of the schema, which allows unified LR description and common resource search among all the catalogues. These repositories cover a broad variety of languages, resource (datasets and tools) and media types, described according to the META-SHARE schema and available through www.meta-share.eu.

Statistical observations on the LRs metadata descriptions reveal the current situation and interesting trends for the development of the field.

Corpora are still the highest represented resource type in the LRs catalogue with a representation of 51.5%, followed by lexical/conceptual resources with 46.4%. On the other hand, tools/services are represented with only 2.6%. This low representation could be attributed to the following:

- The fact that some of the larger node repositories contain no tools or services as individual items but mostly data resources.
- More interestingly, the fact that the concept of having the same model describing both datasets and tools/services has only recently been introduced to the LR community. It is possible that LRs providers using the META-SHARE schema are not familiar with this concept, as they only fill in the basic metadata fields. However, the description of tools and services should be particularly encouraged in order to ensure interoperability between different types of LRs.

Further on interoperability, it should be noted that there are some important metadata fields that should be represented, such as: annotation type and format, mime type etc. Observations on the metadata records show an increasing tendency in the encoding of annotation types (20%), followed by the encoding of annotation format (6.2%). Mime type, although very important, is not highly represented (2.9%) and is mostly encoded for text corpora.

The most interesting findings concern the use of the *recommended* components. Providers have started to fill in not only the required information that corresponds to the minimal schema, but also information related to creation, usage and documentation of LRs. This may have been achieved through manual contributions done by the

above-mentioned related projects or through the conversion of LRs done within META-SHARE. In any case, this shows the providers' interest to describe their resources in detail. Specifically, 39.3% of metadata records include creation details and 35.4% also report on funding projects.

Usage information is represented in 6.2% of metadata records and it is mainly encoded for text corpora and lexical resources.

Resource documentation is also one of the blocks of recommended information that providers show a tendency to fill in. 9.7% metadata records report on publications elaborating on various levels of the LR lifecycle. Direct linking to the relevant documents – wherever provided – could prove very useful for a complete quick overview of the resources.

The general remark drawn from the above statistics is that providers are willing to describe their resources (with a potentially-interesting starting point in their own repositories) and also to extend the metadata descriptions beyond the minimal required information. The most prominent "additional" information is that which either increases awareness of their resources or ensures interoperability between different types of LRs.

11. Future work

Work in the future includes the evolution of the schema as regards breadth (i.e. coverage of more types as they emerge) and depth (i.e. enrichment and updating of the controlled vocabularies, representation of additional relations, improvements based on future feedback, etc.).

Mapping to other schemas is also of priority to support interoperability between LR descriptions. In addition to the currently existing linking of elements to the corresponding DC[5] and ISOcat ones, links to OLAC[6] elements are foreseen in the future.

12. Acknowledgements

This paper presents work done in the framework of the project T4ME, funded by DG INFSO of the European Commission through the 7th Framework Program, Grant agreement no.: 249119.

Many thanks are due to all the colleagues from the META-SHARE metadata working group and implementation team and from the colleagues of the collaborating projects, for their valuable feedback.

13. References

Broeder, Dan, Thierry Declerck, Erhard Hinrichs, Stelios Piperidis, Roland Romary, Nicoletta Calzolari, and Peter Wittenburg. 2008. "Foundation of a Component-based Flexible Registry for Language Resources and Technology." In Proceedings of the 6th International LREC Conference, Marrakech.

Broeder, Dan, Marc Kemps-Snijders, Dieter Van Uytvanck, Menzo Windhouwer, Peter Withers, Peter Wittenburg, and Claus Zinn. 2010. "A Data Category Registry- and Component-based Metadata Framework." In Proceedings of the Seventh International LREC Conference, Malta.

Desipri, Elina, Maria Gavrilidou, Penny Labropoulou, Stelios Piperidis, Francesca Frontini, Monica Monachini, Victoria Arranz, Valerie Mapelli, Gil Francopoulo, and Thierry Declerck. 2012. "META-NET Deliverable D7.2.4 – Documentation and User Manual of the META-SHARE Metadata Model (final)." Ed. Penny Labropoulou and Elina Desipri.

Federmann, Christian, Byron Georgantopoulos, Ricardo del Gratta, Bernardo Magnini, Dimitris Mavroeidis, Stelios Piperidis, and Manuela Speranza. 2011. "META-NET Deliverable D7.1.1 – METASHARE Functional and Technical Specifications."

Federmann, C.; Giannopoulou, I.; Girardi, C.; Hamon, O.; Mavroeidis, D.; Minutoli, S. and Schröder, M. (2012). META-SHARE v2: An Open Network of Repositories for Language Resources including Data and Tools. In Proceedings of the Eighth International Conference on Language Resources and Evaluation (LREC2012), Turkey.

Gavrilidou, Maria, Penny Labropoulou, Stelios Piperidis, Manuela Speranza, Monica Monachini, Victoria Arranz, and Gil Francopoulo. 2011. "META-NET Deliverable D7.2.1 - Specification of Metadata-Based Descriptions for Language Resources and Technologies."

ISO 12620. 2009. "Terminology and Other Language and Content Resources -- Specification of Data Categories and Management of a Data Category Registry for Language Resources." http://www.isocat.org.

LRSLM. 2010. Workshop on "Language Resources: From Storyboard to Sustainability and LR Lifecycle Management" In Seventh International LREC Conference, Malta: http://workshops.elda.org/lrslm 2010/

Monachini, Monica, Valeria Quochi, Nicoletta Calzolari, Nuria Bel, Gerhard Budin, Tommaso Caselli, Khalid Choukri, et al. 2011. "The Standards' Landscape Towards an Interoperability Framework". FLaReNet, CLARIN,META-NET. http://www.flarenet.eu/sites/default/files/FLaReNet_Standards_Landscape.pdf.

Piperidis, S. (2012). The META-SHARE Language Resources Sharing Infrastructure: Principles, Challenges, Solutions. In *Proceedings of the Eighth International LREC*,,Turkey.

[5] http://dublincore.org

[6] http://www.language-archives.org

Applying Current Metadata Initiatives: The META-NORD Experience

Gunn Inger Lyse, Carla Parra Escartín, Koenraad De Smedt

University of Bergen
Bergen, Norway
gunn.lyse@uib.no, carla.parra@uib.no, desmedt@uib.no

Abstract

In this paper we present the experiences with metadata in the Norwegian part of the META-NORD project, exemplifying issues related to the top-level description of language resources and tools (LRT). In recent years new initiatives have appeared as regards long-term accessibility plans to LRT. The META-NORD project, and the broader META-SHARE initiative in the META-NET network, are among the initiatives working on the standardization of the description of linguistic resources as well as on the creation of infrastructures that ensure a long-term curation and distribution of LRT. We present the use cases we have been dealing with in Norway as part of this effort. We also report on the importance of dealing with real user case scenarios to detect and solve potential problems concerning the construction of a larger open infrastructure for LRT.

1. Introduction

In recent years the need for using metadata and standards in the description of language resources and tools (LRT) has gained importance. This is not a novel problem as it has been addressed in several projects and initiatives before (Parra et al., 2010). We will not refer to all earlier initiatives now as they are well described and documented in Deliverable 2.2 of the ENABLER project (Gavrilidou and Desypri, 2003) and more recently in Deliverable 7.2.1 of the META-NET project, in which all initiatives are not only described but also compared and analyzed with the aim of specifying metadata-based descriptions for LRT (Gavrilidou et al., 2011, p. 7). In particular, they state as follows:

> In this effort, we intend to build upon previous initiatives so that the model is easily adopted by the target community. The aim is not to create yet another competing metadata model but rather to adapt existing resource description models to a unified proposal catering for the specific requirements of the community.

We find particularly relevant that they state as an aim that the model proposed will be easily adopted by the target community because in the past this has been proven both a challenge and a need. In fact, a survey carried out within the FLaReNet project has pointed out how difficult it was to gather information on LRT and to create and to compile and maintain LRT catalogs and observatories. This makes, in turn, the curating costs of such infrastructures high. FLaReNet Deliverable D6.1a (Parra et al., 2009, p. 4) reports the situation as follows (with our emphasis):

> The compilation of information for this first survey was harder than expected because of the **lack of documentation** for most of the resources surveyed. Besides, **the availability of the resource itself is problematic**: Sometimes a resource found in one of the catalogues/repositories **is no longer available** or simply **impossible to be found**; sometimes it is **only possible to find a paper reporting on some aspects of it**; and, finally, sometimes **the information is distributed among different websites, documents or papers at conferences**. This made it really difficult to carry out an efficient and consistent study, as the information found is not always coherent (e.g. not every corpus specifies the number of words it has) and sometimes it even differs from the one found in different catalogues/repositories.

CLARIN's Virtual Language World and Virtual Language Observatory have contributed a major effort to increase the visibility of LRT. With a slightly different perspective, the META-SHARE initiative is also actively working on this matter. As stated on their website, *"META-SHARE aims at providing an open, distributed, secure, and interoperable infrastructure for the Language Technology domain"*.
In what follows we illustrate the use of the metadata schema in the META-NORD initiative in the upload of the first batch of LRT in November 2011. In the next batch upload, foreseen for June 2012, new LRT will be described with an updated version of the metadata scheme and previously uploaded LRT will be updated accordingly to comply with the last modifications to the schema. The data must be understood as work in progress within the more general META-SHARE framework. The following sections report on the META-NORD initiative itself (section 2.) and the user cases studied and included into the META-NORD network in Norway (section 3.) before discussion and conclusions as well as further work to be done (sections 4. and 5.).

2. The META-NORD initiative

The META-NORD project[1] aims to establish an open linguistic infrastructure in the Baltic and Nordic countries to serve the needs of the industry and research communities. The project focuses on the national languages of the eight Nordic and Baltic countries: Danish, Estonian, Finnish, Icelandic, Latvian, Lithuanian, Norwegian and Swedish, which all have less than ten million speakers. The project cooperates closely with projects such as T4ME, CESAR and

[1] http://www.meta-nord.eu/

METANET4U, is a part of the common META-NET network and provides input to META-SHARE.
A major aim of META-NORD is to upgrade, harmonize, document and catalogue language resources and tools in order to make them interoperable, within languages and across languages, with respect to their data formats and — to the extent possible — their content. The idea is to make LRTs searchable through metadata, facilitating their access.
With many LRTs that are easily accessible, an application programming interface (API) may allow the assembly of on-the-fly tool chains (workflows) made up of standardized component language technology tools. In turn, these workflows will be able to process distributed — and in many cases interlinked — language resources in standardized formats (Borin and Lindh, 2011). To this end, META-NORD depends on standardized resource and tool metadata, standardized tool APIs and standardized mechanisms for publishing and making the metadata harvestable. LRTs may thus be utilized in language technology applications efficiently, both for academia and industry.
The ownership of the resources and tools is usually not taken over by META-NORD. Instead, META-NORD aims to store links to the resources and tools, along with metadata, in a mirrored database. META-NORD develops standardized top-level resource descriptions (metadata) for all relevant types of LRT, based on a recommended set of metadata descriptors for documenting resources provided by META-SHARE.[2] The latter is a network of repositories of language resources, including both language data and language tools, described through a set of metadata (Magnini and Speranza, 2011).

2.1. The metadata scheme and database from META-SHARE

META-SHARE has developed a set of metadata descriptors for documenting resources. All observations in this paper are based on version 1 of the META-SHARE metadata schema. This was the version used for the LRTs referred to in this paper which were uploaded in November 2011. A newer version has been released in March 2012. This version, which is still under review, has not been sufficiently tested yet in our context and will be discussed in future work.
As stated in Deliverable 7.2.1 of the META-NET project (Gavrilidou et al., 2011), the META-SHARE metadata model is based, on the one hand, on the results of a user requirements survey carried out by means of interviews during last LREC 2010 and, on the other hand, on the overview of metadata schemas and catalogs.
The principles of the resulting proposal stem from the following observations on the needs of the Human Language Technologies domain (Gavrilidou et al., 2011, p. 26):

- the need for a taxonomy of LRT which would define the various types of LRT (corpora, collections, annotations, speech corpora, multimodal corpora...) and the relations between them;
- the need for a common shared terminology;
- the need for a minimal sets of metadata that would facilitate and not hamper LRT description and harvesting;
- the need for a clear and non-complex structure of elements;
- the need for clear semantics of the elements (definitions, relations);
- the need for the interoperability of metadata between repositories and between resources and tools/services.

META-SHARE suggests an initial level providing the basic elements for the description of a resource (minimal schema), and a second level with a higher degree of granularity (maximal schema), providing more detailed information on each resource. The minimal schema contains the elements that are assumed to be indispensable for an adequate LRT description from the provider's perspective as well as search and identification from the consumer's perspective. In the final version of the deliverable (Gavrilidou et al., 2011, p. 38 ff.), the obligatory components that constitute the minimal schema are listed as follows:

(i) *IdentificationInfo*: groups together information needed to identify the resource (resourceTitle, persistent identifier, unique identifier).

(ii) *ContentInfo*: groups together information on the contents of the resource, and comprises a prose description, resourceType (corpus; lexicalConceptualResource; languageDescription) and mediaType (values: text; audio; video; image; tactile).

(iii) *DistributionInfo*: groups information on the distribution of the resource and comprises the elements Availability and distributionMedium (e.g. internetBrowsing, download, CDROM, etc.) and the component licenseInfo.

(iv) *ValidationInfo*: Indication of the validation status of the resource, contains only one element (validated, with values *yes* or *no*).

(v) *MetadataInfo*: groups information on the metadata record itself (metadataCreationDate, harvestingDate, originalMetadataLink).

(vi) *FundingInfo*: information on all projects that have funded the resource; repeated for each project, includes the component ProjectInfo (projectTitle, fundingType).

(vii) *PersonInfo*: groups information on the contact person (surname, givenName, CommunicationInfo).

(viii) *OrganizationInfo*: groups information the organization (organizationName, CommunicationInfo).

(ix) *CommunicationInfo*: groups information on communication details (address, email etc.) and can be attached to either PersonInfo or OrganizationInfo.

[2] http://www.meta-net.eu/meta-share

In the case where the *mediaType=text*, there are some type-dependent components and elements, listed here:

(i) *LanguageInfo*: information on the language(s) of a resource; repeated for each language in the case of bi- or multilingual resources (languageCoding, languageId, languageName).

(ii) *SizeInfo*: this component can be attached to every component that needs a specification of size; it includes two elements (size and sizeUnit).

(iii) *FormatInfo*: the mime-type of the resource which is a formalized specifier for the format included. Takes values from the Internet Assigned Numbers Authority (IANA).[3]

(iv) *CharacterEncodingInfo*: Groups together information on character encoding of the resource; repeated if parts of the resource have different character encodings (characterEncoding, sizeInfo).

(v) *DomainInfo*: Groups together information on domains of a resource; can be repeated for parts of the resource with distinct domain (domain, sizeInfo).

(vi) *AnnotationInfo*: (annotationType, which specifies the types of annotation levels provided by the resource).

In the future, META-SHARE will provide similar extensions to other media types (audio, video, image, tactile) and other LRT types (lexicalConceptualResource, languageDescription; technologyToolService). It also seems that the number of obligatory components will change in future versions. The challenge for META-NORD and for META-SHARE in general will be to convince resource owners to fill in the complete metadata forms and not only the minimal schema. Despite agreeing that at least a minimal set of metadata should be always provided, we cannot see that any of the other metadata in the extended version contain superfluous information. Informative metadata will certainly enhance the visibility and durability of LRT, as well as their usage for other purposes. Efforts should therefore be made to highlight the long-term benefits of taking the time to fill in the full metadata forms.

Nevertheless, one should also consider the fact that currently it is often necessary to fill in metadata in retrospect for a range of already existing resources. This is, for instance, mostly the case for the metadata delivered thus far in META-NORD. It is challenging to find information about a resource which has been incompletely documented. The resource creator is normally the best person to fill in metadata. As for the resources discussed in the current paper, the only metadata which were filled in by the developer were the ones associated to the TRIS corpus and the SCARRIE lexicon.

The first scheduled META-NORD upload of metadata and resources was performed in November 2011. At that time, the META-SHARE online metadata tool was still under development. Therefore the data was manually edited into an XML schema which was uploaded to an SVN versioning repository at the META-NORD partner University of Gothenburg, where the metadata were validated and stored. Later, the metadata were imported into an online database released by META-SHARE. The metadata from Norway and the other countries in META-NORD have been made available at the Tilde META-NODE where they are browsable and searchable by language, resource type and media type.[4]

3. META-NORD user cases in Norway

Norway delivered thirteen resources and tools for the first batch, listed in Table 1. For none of these resources, adequate metadata descriptions existed, so it was necessary to collect and structure the necessary information in terms of the adopted metadata scheme.

Several of these LRTs are currently being distributed by *The Norwegian Language Technology Resource Collection – Språkbanken*[5], established in 2010, with which we cooperate with respect to the creation of metadata. Two of the resources are downloadable from University of Bergen (UiB) and one LRT is downloadable from University of Oslo (UiO). The metadata are available in XML from the University of Gothenburg[6] and they are also browsable at the Tilde META-NODE. The LRT are classified as either Corpus, LexicalConceptual or ToolsServices. Within these categories, each LRT is assigned a number; this number is given in the leftmost column in Table 1. Thus, for instance, the TRIS corpus (which has been assigned the number 12) is found in the entry named *UIB-M10-12*.[7]

As regards availability and licensing, the majority of the presented resources are freely available, corresponding to a *Creative Commons Zero* license.[8] However, as Språkbanken is still in the process of defining a licensing scheme, we recommend that these resources are categorized for now in META-SHARE with the value *own* (denoting that currently, no standard licensing scheme is being applied). One of Språkbanken's resources, Norsk ordbank, is restricted in use and the user needs to apply for a user name and password. Three of the resources in Table 1 are currently categorized with respect to the CLARIN classification scheme (Váradi et al., 2008; Oksanen and Lindén, 2011). CLARIN RES covers LRT with special restrictions.

In what follows, we will discuss metadata issues more in detail with reference to the first metadata delivery in META-NORD.

3.1. Written corpora

3.1.1. The TRIS corpus

In cooperation with the CLARA project[9], the TRIS Spanish-German corpus was made available in META-NORD. This is a parallel corpus from the European database of Technical Regulations Information System, with documents written

[3] http://www.iana.org/assignments/media-types/

[4] http://metanode.tilde.com

[5] http://www.nb.no/spraakbanken/english

[6] https://svn.spraakdata.gu.se/repos/metanord/pub/uib/

[7] https://svn.spraakdata.gu.se/repos/metanord/pub/uib/Corpus/UIB-M10-12.xml

[8] http://creativecommons.org/publicdomain/zero/1.0/

[9] http://clara.uib.no

No.	Resource/tool	Resource Type	Download location	Availability	Licence
1	Oslo-Bergen tagger	Tools	UiO	available-unrestrictedUse	GPL
2	Sofie Treebank	Corpus	UiB	available-restrictedUse	CLARIN RES
3	Lexical database for Norwegian	Lexical resources	Språkbanken	available-unrestrictedUse	own
4	Lexical database for Swedish	Lexical resources	Språkbanken	available-unrestrictedUse	own
5	Lexical database for Danish	Lexical resources	Språkbanken	available-unrestrictedUse	own
6	Acoustic database for Norwegian	Speech resources	Språkbanken	available-unrestrictedUse	own
7	Acoustic database for Swedish	Speech resources	Språkbanken	available-unrestrictedUse	own
8	Acoustic database for Danish	Speech resources	Språkbanken	available-unrestrictedUse	own
9	Norsk ordbank, Bokmål	Lexical resources	Språkbanken	available-restrictedUse	own
10	Norsk ordbank, Nynorsk	Lexical resources	Språkbanken	available-restrictedUse	own
11	SCARRIE Lexical Resource	Lexical resources	Språkbanken	available-unrestrictedUse	CC-BY
12	TRIS Spanish-German	Corpus	-	available-restrictedUse	CLARIN RES
13	Parallel Treebank	Corpus	UiB	available-restrictedUse	CLARIN RES

Table 1: Tools and resources delivered from Norway for the first META-NORD upload

in Austria, Germany and Spain and their translations into Spanish and German respectively. The database is aligned at sentence level and consists of 995 file pairs corresponding to 10 different domains. This is a corpus under development, and the first version was released in the first META-NORD upload of metadata and resources in November 2011. This version consists of 97 files aligned at sentence level, containing approximately 686,649 words. A second version is ready to be uploaded for the next META-NORD batch upload in June 2012, containing 205 files which have been completely aligned at sentence level and which account for approximately 1,563,000 words (Parra, 2012).
The TRIS Spanish-German corpus is the only one in Table 1 which is not directly downloadable. It is to be made available with restrictions, in particular for research only, whereas a special license will be issued for commercial exploitation. Currently, the META-SHARE repository does not offer practical technical solutions for handling commercial licenses involving binding agreements, payment, etc. and therefore the TRIS corpus is not yet directly downloadable. A screenshot of the current metadata available for the TRIS corpus at the Tilde META-NODE is shown in Figure 1. As the screenshot shows, the metadata are organized in a human-readable and systematic way.

3.1.2. The Sophie Treebank and Parallel Treebank

Two treebanking resources resulting from the INESS project[10] have been made available in META-NORD. The Sofie Treebank is a syntactically annotated corpus of 255 sentences from the novel *Sofies verden* (Sophie's world) in the original language Norwegian (Gaarder, 1991). Each sentence is automatically analyzed with the Lexical-Functional Grammar (LFG) and disambiguated and verified by human annotators supported by a discriminant-based tool (Losnegaard et al., to appear; Rosén et al., 2009).
The Parallel Treebank is a multilingual treebank containing alignments of the above-mentioned Norwegian Sofie Treebank and treebanks of this text's translation into other languages: Estonian, Danish, German, Icelandic and Swedish. This work was initiated by the Nordic Treebank Network (Nivre et al., 2005) and some of the resources have been kindly passed on to META-NORD from Tekstlaboratoriet at the University of Oslo. The alignment has been done in cooperation with INESS. It has been necessary to clear the rights again for META-NORD, since the original license was made specifically between University of Oslo and each publishing house for each relevant language.

3.2. Speech corpora and speech databases

Three acoustic databases were released in the first META-NORD batch; these are categorized as Speech resources in Table 1. The acoustic databases are for Danish, Swedish and Norwegian, respectively, and are freely available through the National Library of Norway (Språkbanken). They were developed for R&D in speech recognition and for speech synthesis by the company Nordisk språkteknologi holding AS (NST), which went bankrupt in 2003. The resources in the estate were bought by a consortium consisting of the universities in Oslo, Bergen and Trondheim, the Norwegian Language Council and the company IBM in order to ensure that the resources developed at NST should not be lost. After *The Norwegian Language Technology Resource Collection – Språkbanken* was established in 2010, the NST resources were transferred to this organization in 2011.
This material consists of sound files and their corresponding written annotation, where each recording has been validated by trained linguists, marking for instance erroneous pronunciation or non-verbal events such as coughing. Thus, the resources are marked as having *MediaType=text audio*, i.e. both text and audio.
Part of the documentation for this resource[11] is only available in Norwegian, which needs to be changed if it is to conform to ELRA's specifications.

3.3. Lexical resources

Several lexical resources were released in the first META-NORD batch, as shown in Table 1. The SCARRIE lexical resource was originally developed for use in automatic proofreading of Norwegian Bokmål (*nob*). It was coded as a set of files in IDF (Intermediate Dictionary Format), a format used only in the SCARRIE project. This situation is illustrative for the status of the field, where there is an abundance of

[10] http://iness.uib.no

[11] http://www.nb.no/content/download/12321/78144/version/1/file/nst_taledat_no.pdf

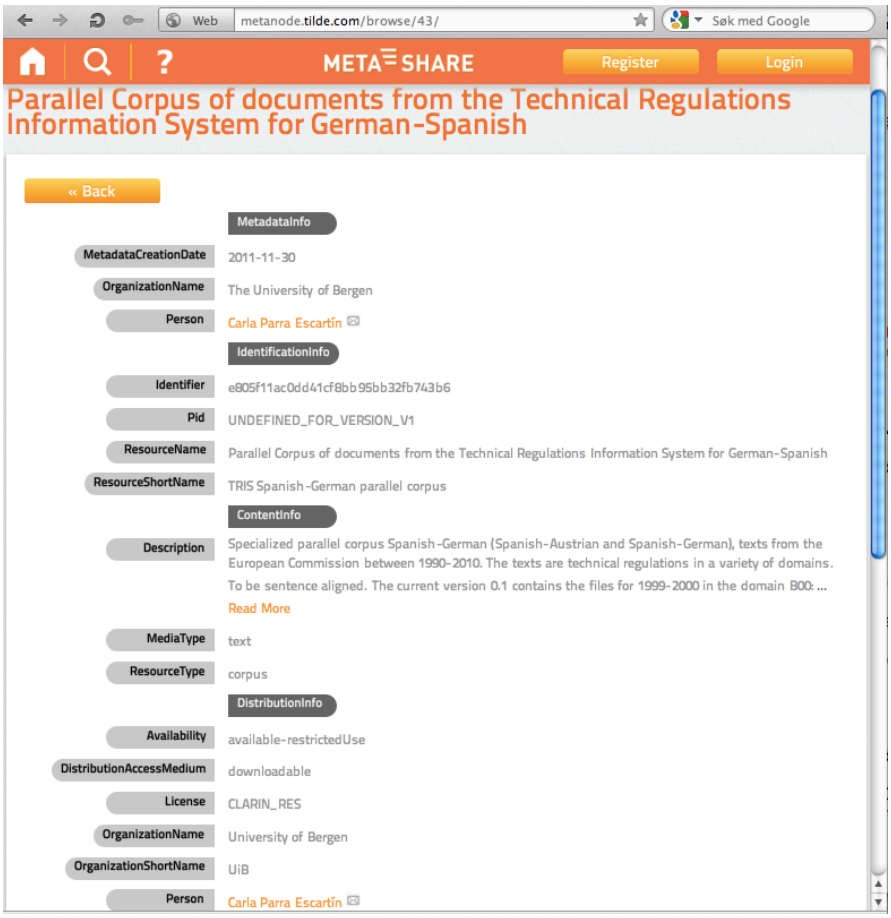


Figure 1: Screenshot of some of the current metadata available for the TRIS corpus at metanode.tilde.com

formats which cannot be readily handled by generic tools. In the context of the META-NORD project, it was decided to make this resources available in a format adhering to the Lexical Markup Format (LMF) standard.[12] The current LMF compliant version was derived from the original IDF lexicon files by means of a script written by Koenraad De Smedt at the University of Bergen in the context of the META-NORD project, with the aim of making the resource easier to share and reuse. This resource has been made available through Språkbanken.

Moreover, Språkbanken has made five other lexical resources available (cf. Table 1). Norsk ordbank is owned by the University of Oslo and The Norwegian Language Council. It is a fullform lexical database for the two written norms in Norwegian: Bokmål (*nob*) and Nynorsk (*nno*). Since the two written norms exist in parallel, the resource is split into two separate resources, one for each.

The metadata for Norsk ordbank delivered in META-NORD batch 1 illustrates the need for a simple and intuitive documentation of metadata elements and their possible values when confronted with metadata providers with limited experience. The META-SHARE metadata concerning distribution information has several components that may be filled in by the resource provider in order to specify the availability and restrictions of use. In the case of Norsk ordbank, a user agreement needs to be signed where the user accepts not to redistribute the resource. Therefore the resource is listed with a restrictive license, namely CLARIN RES. At the same time, the resource was defined as having *unrestrictedUse* with respect to the field *availability* in batch 1, whereas the correct value should probably be *availability=restrictedUse*. The metadata for this resource were filled in by a third party and the *unrestrictedUse* value was probably chosen because one might argue that the resource is freely available in virtue of being available *for free* for *anyone accepting the user terms*. A clear documentation might shed light on such uncertainties. Furthermore it might be helpful if the possible values of some fields are restricted in accordance with the choices made in other, related fields. For instance, the metadata value for *RestrictionsOfUse* for Norsk ordbank was given the value *RestrictionsOfUse=attribution* (and strictly speaking, a NoRedistribution restriction also applies with the current user agreement). If the RestrictionsOfUse field is used, implying that there are restrictions, this could automatically exclude the possibility of choosing *availability=unrestrictedUse*.

There are also three lexical databases for Danish, Swedish and Norwegian, respectively (these lexical databases were also part of the material from NST, as described in the section on speech corpora and speech databases). These are freely available and are well provided with metadata.

[12] ISO-24613:2008, http://www.lexicalmarkupframework.org/

4. Discussion

As we have seen in the previous section, in spite of spent efforts to create appropriate schemas to properly describe LRTs, nontrivial problems arise once researchers are asked to describe their resources. We will discuss these problems more in depth in the current section.

4.1. Language codes

In our experience, some decisions on metadata may severely affect the searchability of the metadata repository. Consider the issue of language coding, for which the ISO-639-3 standard was adopted. The Norwegian language is coded as *nor* in this standard, but two written norms exist, notably Bokmål, coded as *nob* and Nynorsk, coded as *nno*. If a language resource adheres to one of the written norms, then *nob* or *nno* is used in the metadata, whereas *nor* is used if the written norm is mixed, unspecified or irrelevant, as in the case of spoken dialect. Even though this coding scheme seems natural, it creates problems for retrieval. When Norwegian resources are searched (using *nor*), this will not match records coded with *nob* or *nno*. Thus, we have learned that language codes cannot always be treated as separate designators, but they can be supersets of others. A solution might be to code Bokmål resources with both *nob* and *nor*. It was, however, not possible in the META-SHARE-schema version 1 to provide multiple language codes to monolingual resources. In version 2 it seems that it might be possible, which may solve the problem. Finally, we would also like to highlight that language codes may not cover the needs of spoken data which contain specific dialects and that this issue should be discussed further within the META-SHARE network.

4.2. Complex resources

When preparing the first batch of resources to be uploaded to META-NORD, we had three valuable experiences as regards resources that are complex in virtue of containing multiple databases: the NST acoustic databases, the SCARRIE lexical resource and Norsk ordbank. Based on the experience reported in what follows, we suggest that guidelines for the minimal documentation of complex resources are developed within the META-SHARE network.

4.2.1. The NST acoustic databases

The NST acoustic databases are complex resources since they include several parts: the audio files, the text files, the tagset used as well as the instructions given to the informant. However, it is unclear from the metadata what the contents actually are, so end users must read the documentation in detail and inspect the various files in order to get a detailed overview. Problems may arise when the resource contains audio files as well as text files, since the different media types need different metadata properties. For instance, the size of resources may be in terms of duration of video/audio, or in terms of sentences or words in the case of a text. There is a lack of documentation regarding the application of size values.

META-SHARE does not cater for complex resources and therefore its metadata schema is not adequate to fully describe this type of resources. Therefore, we suggest that the metadata for this kind of resources are revised and better ways are established to display and integrate all the information available. We believe that an effort should be made to provide metadata on every part of a complex resource and those metadata sets should be grouped in the entry for the resource as a whole. Thus, a particular end user will gain a better insight into the whole as well as all parts of a complex resource and will have better information to select parts of the resource as needed.

4.2.2. The SCARRIE lexical resource

The SCARRIE lexical resource recoded in LMF consists of several different lexicons grouped together in a single XML file. This is possible due to the fact that the LMF top-level element *LexicalResource* may contain more than one element *Lexicon*. Within its single file, SCARRIE has separate lexicons for prefixes, suffices, grammatical words, elements of abbreviations, words only occurring in multiword expressions, and a main lexicon of open class words. In the metadata, there is no structured way of describing the contents and size of each lexicon. Furthermore, it would be desirable in the case of LMF resources that part of the metadata description would somehow reflect the LMF buildup in a systematic way.

4.2.3. Norsk ordbank

The lexical resource Norsk ordbank illustrates an approach opposite to that in SCARRIE: The resource was split by the creator into two separate resources, one for each Norwegian written norm (Norwegian Bokmål and Norwegian Nynorsk). Although the availability of this resource as separate files may be practical, it does not take into account that they share important metadata properties. They could be put together in one LMF top-level element *LexicalResource* like SCARRIE, but this may not be practical if the lexicons should be individually downloadable.

4.2.4. Conclusions on complex resources

Currently META-SHARE offers one option, namely to create one resource with one metadata scheme. In other words, complex resources must have two different metadata schemas and will *prima facie* be separate resources. A better option would be to have a metadata scheme for complex resources allowing for separate metadata descriptions for every subpart that can be considered an individual resource. As we have seen, many of the resources described in this paper encounter this problem. A schema for complex resources should make it possible to search and retrieve all parts of the complex resource, or to retrieve only the subpart that a user is looking for.

4.3. Rightholders

One resource could not be uploaded and catalogued for batch 1 because the IPR issues are complicated and could not be fully resolved. The Norwegian-Spanish Parallel Corpus has been developed with funding coming from more than one funder. The IPR has been cleared by the developer with respect to the authors of the original texts and the translators. The OCR and the alignment have been financed partly by the faculty, partly by META-NORD and partly a private person (the resource developer). Consequently, it has not

been clarified who owns the IPR of the parallel corpus. Similarly, the Sofie treebank has one IPR for the original texts and the translators and one for the linguistic annotations provided in the INESS project. META-SHARE currently does not seem to cater for situations where different IPRs are associated with different subparts of the resource. We deem it an important issue to be tackled in future versions of the metadata schema.

4.4. Licensing

A related challenge concerns licensing issues. In principle, we support attempts to use standardized licenses, but we have also experienced that Språkbanken, which is currently the major Norwegian LRT repository, sees a need to develop their own licensing schemes. It should be feasible to convince resource owners to use standardized licenses, but this will only be possible if the inventory of standardized options in a license covers the needs of the LRT owner.

The META-NORD experience has shown that the standardized licenses available at the time of the first upload only accommodated the needs reported by the LRT owners to a moderate extent. Succeeding the first batch of resources in META-NORD, however, META-SHARE has already developed a richer set of standardized META-SHARE licenses that (among other things) accommodate *NoRedistribution* restrictions. In general, handling licenses for individual users is challenging, since many resources are to be provided with a restricted license even if the restrictions are minor (e.g. CC-BY-SA). Furthermore, it would also clearly be helpful if there is a clear and intuitive correspondence between license types in the license schema and in the metadata schema. As of today, the META-NORD partners as well as the LRT owners cannot be expected to handle licensing issues, and good documentation is therefore pertinent.

4.5. Metadata providers

The relationship between the metadata provider and the resource creator was another interesting issue we faced. None of the resources had adequate metadata when the resource was first provided. In the case of the NST resources, the resource creator was no longer available, which made it particularly challenging to fill in the metadata and to provide documentation. The lack of documentation and the unavailability of resource providers is not a novel issue (Parra et al., 2009) and therefore ways to ensure that this will not happen in the future should be established. However, for those cases in which a third party tries to provide a resource with metadata and does not know a particular detail, we suggest that the same value is always assigned to that particular attribute: i.e. a clear distinction between the current *unknown* and *unspecified* should be established to avoid misunderstandings and make clear which information was not found.

In the case of the SCARRIE lexical resource and the TRIS corpus, we were able to approach the resource developers and ask them to fill in metadata. As regards the TRIS corpus, the resource provider had already considered metadata when designing the resource and therefore already had decided which kind of information was to be included in the metadata. Several questions arose with respect to how to add attributes and values to the maximal schema.

First, all corpus files contain information about the country of origin of the resource, source and target language, domain and year. This additional information should be added to the metadata to allow the final users select the whole corpus or just a subcorpus that they could create according to their needs.

In fact, the TRIS corpus is currently available in just one format, TMX (Translation Memory eXchange), but it will also be released in other formats such as raw monolingual texts and maybe also PoS-tagged texts. Again, this information should be available in the metadata and it should be left to the final user to choose the format suitable for a particular research purpose, i.e. it should be the final user who actually filters the corpus and adapts it to actual needs. Finally, the information as regards the size can be provided in different units (number of documents, number of sentences, or number of words). Both the number of sentences and the number of words could be retrieved from metadata at document level instead of at resource level and this information could be useful for the users who want to adapt the corpus to their own needs and not just use it "as is". We therefore also suggest that META-SHARE takes this into consideration and studies how to integrate and combine document and resource level information into their platform so that all this information can be use in a dynamic way by end-users.

4.6. Acquisition and versioning

The current metadata scheme does not seem to have an element for documenting the acquisition method (automatic, semi-automatic, manual, crowdsourcing, etc) which may be very relevant to the user in terms of assessing the quality of a resource. There is an element for validation, but it could be more closely related to the acquisition method.

Finally, META-SHARE may need better handling of resources which have more than one version, as will be the case of the TRIS corpus reported here. Even if the metadata schema has an element for indicating the version, it seems that different versions are treated as unrelated resources, whereas there is clearly a need for establishing references between different versions of a resource. Furthermore, there could be a need for procedures to retract earlier versions of a resource in case errors are detected.

5. Conclusions and future work

We have reported on the collection and structuring of metadata for a number of resources to be incorporated in META-SHARE. All observations have been made based on version 1 of the META-SHARE metadata scheme. Our main conclusions are that the definition of metadata, which at first sight could be a simple administrative task, is in fact far from trivial. Experiences with subsequent versions will be addressed in future work, in particular by evaluating whether issues discussed in the present paper have been resolved.

Regarding metadata that were filled in for the first batch in the META-NORD project, we have discussed issues with language codes, complex resources, rightholders, licensing, metadata providers and acquisition and versioning (cf. section 4.). A general conclusion is that the META-SHARE metadata schemas and procedures for their use need further

work so that LRT catalogs can be feasible and resources will be able to be exploited as much as possible.

This paper shows the importance of cooperating with resource providers, both those that are familiar with metadata schema and not, as this interaction makes different questions arise, highlighting problematic areas that need to be catered for in the succeeding work with metadata. A simple and intuitive documentation of metadata elements and values is also needed. We believe that META-SHARE should also develop guidelines as regards to how to describe LRT appropriately and in an efficient manner.

It is currently possible to provide metadata to META-SHARE that is inconsistent with the metadata in the header of the actual resource. In order to promote consistency, there should ideally be a closer cooperation between developers of novel LRT formats (such as LMF) and developers of catalogs and repositories (such as META-SHARE and CLARIN) allowing the extraction of header metadata and its automatic conversion into metadata for catalogs and repositories.

We also believe that metadata descriptions should become mandatory for resource providers and therefore the advantages of providing that information has to become clear for LRT creators, so that they actually are eager to describe their resources. Even though in the past few years this need has become clearer, a lot of efforts are still to be done. Initiatives such as the LREC Map are contributing a lot to this purpose, but other possibilities such as lobbying for making metadata descriptions of LRT mandatory in future EU funded projects.

Finally, we think that it will also be important to have a strategic dissemination plan to ensure that all parties (both resource providers and resource users) know the META-SHARE metadata schema as this will in turn ensure that the final platform will be easily adopted by the target community and therefore that it will be a success.

6. Acknowledgements

The research reported in this paper has received funding from the EU under CIP ICT PSP, grant agreement no 270899 (project META-NORD) and FP7, Marie Curie Actions, SP3 People ITN, grant agreement 238405 (project CLARA).

The authors would also like to acknowledge all the people involved in the description of resources delivered from the University of Bergen for META-NORD, in particular Anje Müller Gjesdal (META-NORD), Gyri Smørdal Losnegaard (META-NORD) and Arne Lindstad (National Library of Norway).

7. References

Lars Borin and Jonas Lindh. 2011. Deliverable D4.1: Metadata descriptions and other interoperability standards. version 1.0, 2011-05-02. Deliverable in the META-NORD project (CIP 270899).

Jostein Gaarder. 1991. *Sofies verden: roman om filosofiens historie*. Aschehoug, Oslo, Norway.

Maria Gavrilidou and Elina Desypri. 2003. Deliverable D.2.2: Report for the definition of common metadata description for the various types of national LRs, ENABLER project. Deliverable in the ENABLER project.

Maria Gavrilidou, Penny Labropoulou, Stelios Piperidis, Manuela Speranza, Monica Monachini, Victoria Arranz, and Gil Francopoulo. 2011. Deliverable D.7.2.1 Specification of Metadata-Based Descriptions for LRs and LTs. Deliverable in the T4ME project (META-NET).

Gyri S. Losnegaard, Gunn Inger Lyse, Martha Thunes, Victoria Rosén, Koenraad De Smedt, Helge Dyvik, and Paul Meurer. (to appear). What We Have Learned from Sofie: Extending Lexical and Grammatical Coverage in an LFG Parsebank. Accepted paper for META-RESEARCH Workshop on Advanced Treebanking. The Eighth conference on International Language Resources and Evaluation, Istanbul (LREC'12) 21.-27. May 2012.

Bernardo Magnini and Manuela Speranza. 2011. Metashare v1 user manual. Technical report, META-NET: A Network of Excellence forging the Multilingual Europe Technology Alliance, September 26th, 2011.

Joakim Nivre, Koenraad De Smedt, and Martin Volk. 2005. Treebanking in Northern Europe: A white paper. In Henrik Holmboe, editor, *Nordisk Sprogteknologi 2004. Årbog for Nordisk Sprogteknologisk Forskningsprogram 2000-2004*, pages 97–112. Museum Tusculanums Forlag, Copenhagen.

Ville Oksanen and Krister Lindén. 2011. Open content licenses: How to choose the right one. Workshop on visibility and availability of LT resources. NoDaLiDa, 2011, Riga, Latvia.

Carla Parra, Nuria Bel, and Valeria Quochi. 2009. Deliverable D6.1a: Survey and assessment of methods for the automatic construction of LRs. report on automatic acquisition, repurposing and innovative proposals for collaborative building of LRs. FLaReNet project.

Carla Parra, Marta Villegas, and Núria Bel. 2010. The BAsic Metadata Description (BAMDES) and TheHarvestingDay.eu: Towards sustainability and visibility of LRT. In Nicoletta Calzolari, editor, *Proceedings of the Seventh International Conference on Language Resources and Evaluation (LREC)*, pages 49–53. European Language Resources Association (ELRA).

Carla Parra. 2012. Design and compilation of a specialized parallel corpus Spanish–German. In Nicoletta Calzolari, editor, *Proceedings of the Eighth conference on International Language Resources and Evaluation, Istanbul (LREC'12)*, Paris. ELRA.

Victoria Rosén, Paul Meurer, and Koenraad De Smedt. 2009. LFG Parsebanker: A toolkit for building and searching a treebank as a parsed corpus. In Frank Van Eynde, Anette Frank, Gertjan van Noord, and Koenraad De Smedt, editors, *Proceedings of the Seventh International Workshop on Treebanks and Linguistic Theories (TLT7)*, pages 127–133, Utrecht. LOT.

Tamás Váradi, Steven Krauwer, Peter Wittenburg, Martin Wynne, and Kimmo Koskenniemi. 2008. Clarin: Common language resources and technology infrastructure. In *Proceedings of the Sixth International Conference on Language Resources and Evaluation (LREC)*.

Building up a CLARIN resource center – Step 1: Providing metadata

Volker Boehlke, Torsten Compart, Thomas Eckart
NLP Group – Department of Computer Science – University of Leipzig
Universität Leipzig, Institut für Informatik, PF 100920, 04009 Leipzig, Germany
E-mail: {boehlkev, compart, teckart}@informatik.uni-leipzig.de

Abstract

In this paper we will describe the different problems that need to be solved in case one decides to provide metadata according to the CLARIN specifications for resource centers. We will report on why we decided to use the repository system fedora and how we configured it in order to serve our purposes. We will also describe how we designed CMDI components and profiles for our resources and how we dealt with the issues of granularity, updates/versioning of metadata. Additionally the usage of PIDs and PartIdentifiers will be discussed.

1. Introduction

The NLP group of the Department of Computer Science of the University of Leipzig [1] takes part in the CLARIN-D[2] project and is currently in the process of setting up an infrastructure that fulfills all the requirements of a CLARIN (resource/service) center[3]. One of those requirements is the setup of a repository system containing metadata, preferably in the CMDI format, that is harvestable via OAI-PMH [4]. Besides deploying an existing repository system, designing and adding metadata to it proved to be challenging.

The following problems had to be addressed:

- decide which repository system to use
- installation and configuration of the repository system
- decide on the granularity of the metadata to be provided
- designing CMDI components/profiles
- how to deal with updates and versioning of metadata
- handling PIDs and PartIdentifiers

In the following chapters we want to describe our take on the solution to these problems.

2. Repository Systems

There are different approaches on how to rank existing repository systems. One can be found on the webometrics website[5] and is documented in *[3] Aguillo, Ortega* and *[4] Aguillo*. During our research on repository systems DSpace[6] by the DSpace Foundation, EPrints[7] developed by the University of Southampton, Fedora[8] by Fedora Commons and OPUS [9] maintained by Universitätsbibliothek Stuttgart (OPUS3 [10]) and by Kooperativer Bibliotheksverbund Berlin-Brandenburg (OPUS 4 [11]) stood out as the most prominent ones. DSpace, EPrints and Fedora are regularly present with dedicated user group events at the OpenRepositories[12] conference. All four of them are listed among the most popular systems used by the community on OpenDOAR[13] and ROAR[14].

OPUS and also MyCoRe[15] are mainly used in Germany. In 2009 Fedora Commons and the DSpace Foundation joined their forces and created the DuraSpace [16] organization. The two organizations, that operated separately before DuraSpace was created, had, in our view, a long, active and stable history, which, since DuraSpace is willing to support both solutions, qualified those systems for long term usage.

From a technical point of view we needed a very flexible system that not only managed resources and metadata, but was compatible to our specific needs. Additional requirements are:

- handle huge amounts of data
- handle huge amounts of entities/resources
- allow to define/use own metadata formats
- allow to store/handle data and metadata externally
- free of charge and open source
- active user community that has been stable for a longer period of time (no newly developed but well documented, mature systems)

Other features like nice GUIs or rapid out of the box useability were less important. Fedora fitted that profile

[1] http://asv.informatik.uni-leipzig.de/
[2] http://de.clarin.eu
[3] http://www.clarin.eu/files/centres-CLARIN-ShortGuide.pdf
[4] http://www.openarchives.org/OAI/openarchivesprotocol.html
[5] http://repositories.webometrics.info/
[6] http://www.dspace.org/
[7] http://www.eprints.org/
[8] http://fedora-commons.org/
[9] http://www.opus-repository.org
[10] http://elib.uni-stuttgart.de/opus/
[11] http://www.kobv.de/opus4/
[12] http://sites.tdl.org/openrepositories/
[13] http://www.opendoar.org/
[14] http://roar.eprints.org/
[15] http://www.mycore.de/
[16] http://www.duraspace.org/

best because, instead of defining a process on how to add and manage resources and encapsulating this functionality in a GUI, Fedora focuses on the specification of a flexible data model. A webservice API allows to add and manipulate resources programmatically and hides the internal implementation behind that interface. Through configuration of the system and by making use of this data model Fedora is adaptable to various usage scenarios. Of course this flexibility is traded for complexity (mainly on the configuration part) of the system.

3. Fedora

As stated above, Fedora allows to manage resources in a very flexible way. Entities are represented as fedora digital objects, in short FDOs[17], which consist of basic metadata (id, label) and datastreams. Datastreams may contain any kind of data. They can be either stored directly inside the repository (inline datastreams) or externally (external datastreams; an url which points to an external resource).

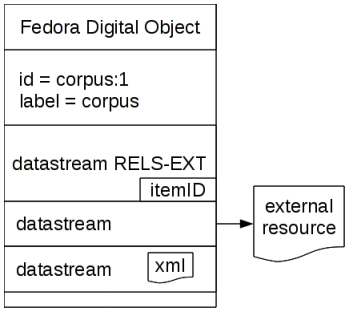


Figure 1: FDO

A RESTfull interface[18] provides the possibility to access (API-A) and manipulate (API-M) FDOs.

In case only Dublin Core[19] metadata has to be available via OAI-PMH no further configurations or components are needed. In order to be CLARIN compliant, a resource center has to provide metadata according to the CMDI standard. In order to do this an additional oaiprovider-module[20] based on ProAI[21] can be used. The oaiprovider-module can be configured to fetch data from a specified datastream of any given FDO in case metadata in a certain format is requested through the OAI-PMH interface. The content of a datastream represents a dissemination of the FDO it is attached to. For example a datastream `cmdi` may contain metadata for the given FDO in CMDI and is used by the oaiprovider-module when metadata records in CMDI are requested via OAI-PMH.

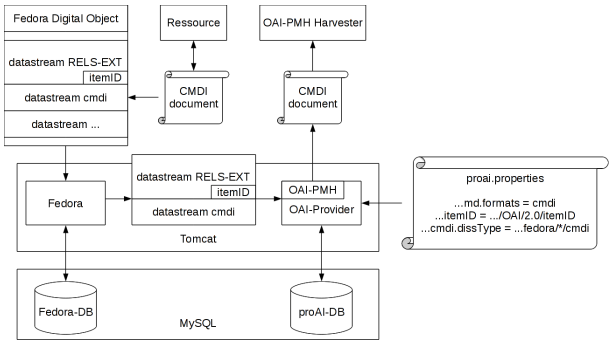


Figure 2: Fedora & OAI-PMH

The complete workflow of adding metadata consists out of the following subtasks:

- create a FDO representing a resource
- create metadata for this resource in the desired format (in our case CMDI)
- add a metadata-datastream to this FDO and push metadata into it

We chose to implement code that made use of the RESTfull webservice interface that comes with fedora in order to add FDOs and datastreams. Therefore we first implemented a very basic Java library called "FedoraAPI", which encapsulates the functionality that was needed. For example the task of creating an FDO and attaching CMDI metadata to it using the webservice interface consists of the following steps:

- call the API-M method `ingest` to create a new FDO
- create a relationship by adding a "literal" relationship to the newly created FDO with type "http://www.openarchives.org/OAI/2.0/itemID"
- create links to the id of the FDO (the same one defined by driver.fedora.itemID in the `proai.properties` file)
- call the API-M method `addDatastream` to add the CMDI datastream

The second step is needed because an `itemID` needs to be present in the `RELS-EXT` datastream (in which relationships are stored) before any dissemination of the object (including the one represented by our cmdi-datastream) will be included into answers of OAI-PMH requests computed by the oaiprovider-module/proai.

[17] https://wiki.duraspace.org/display/FEDORA35/Fedora+Digital+Object+Model
[18] https://wiki.duraspace.org/display/FEDORA35/REST+API
[19] http://dublincore.org/
[20] http://sourceforge.net/project/downloading.php?group_id=177054&filename=oaiprovider-1.2.zip
[21] http://proai.sourceforge.net/

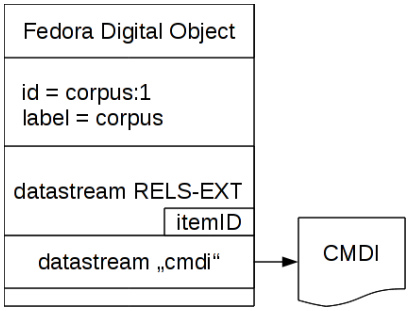


Figure 3 : FDO and CMDI

Using the FedoraAPI, this complex and sometimes confusing functionality is reduced to the simple task of declaring and manipulating an object that represents the FDO and calling two methods in order to ingest it and add a datastream to it:

```
FedoraObject corpus = new FedoraObject();
corpus.pid("corpus:1");
corpus.state(State.ACTIVE);
corpus.label("corpus");
corpus.ownerId("some owner id");
fedoraApiSupport.ingest(corpus);
fedoraCMDISupport.addCMDIDatastream(corpus.pid(),
"someCMDICode");
```

Snippet 1: adding a fedora object and CMDI metadata

Please note: The FedoraObject-Implementation was taken from an external provider[22].

Alternatively the "MediaShelf"[23] Java client, which also implements the Fedora Rest API, may be used. A detailed description of the installation and configuration of Fedora for our usage scenario is available on the CLARIN website[24].

4. Granularity

The first resources we chose to add to our repository are corpora of the Wortschatz project [25] (*[1] Quasthoff, Richter, Biemann*). The data of these corpora is usually collected from online newspapers, Wikipedia and several other sources and may be described as a collection of sentences. The original texts are not reconstructable due to copyright reasons. Based on this data statistical information like word frequencies, co-occurrences etc. are calculated. This data is available for more than 130 different languages. Users may access these corpora through a web portal or download differently sized variants as text dumps or MySQL databases. Additionally some corpora are accessible via webservices (*[2] Büchler, Heyer.*).

Metadata is available for each of the sentences:

- date (of crawling)
- source (usually an url)
- statistical data like length, number of tokens, …

This metadata plays an important role when discussing the granularity of accessible data and according metadata. Of course a corpus should be accessible as a whole and should therefore also be described by metadata. But usage scenarios in which smaller portions of the available data are of interest do exist too. Therefore these smaller portions of the available data should be described by metadata too. Two obvious ones are:

- all sentences of a certain document
- all sentences collected from a certain source
- all sentences collected over a certain period of time (e.g. each day)

In case of the wortschatz data it would be possible to choose an even finer level of granularity. Data and metadata could be made available on sentence or word level. This would result in very large metadata documents (CMDI of several gigabytes in size), which makes them unusable, or in a huge number (several billions) of entities that need to be handled by the repository system. While Fedora was tested in the past[26] and qualified itself for the usage with 14 million elements and 750 million relations, the underlying database solutions will perform badly for several billions of entries.

The following table provides a quick overview of the different possibilities and the resulting number of elements the repository system would need to handle in case metadata for a Wortschatz corpus of roughly 259 million sentences should be harvestable via OAI-PMH on certain levels:

Level Of Granularity	Number Of Elements
document	13,216,594
source	536,288
sentences for each day	5,232
sentence	259,081,726
word form	37,699,483

Table 1: number of elements per level of granularity

In order to balance between maximum granularity and the number of elements that need to be handled, the following criteria were used:

- Which are (simply due to the structure of the resource) "natural ways" of splitting up the resource?
- Which are common research questions people using the resource are working on? Which granularity is needed?

[22]http://cwilper.github.com/fcrepo-misc/
[23]http://mediashelf.github.com/fedora-client/
[24]http://www.clarin.eu/faq/3485
[25]http://wortschatz.uni-leipzig.de/
[26]http://fedora.fiz-karlsruhe.de/docs/Wiki.jsp?page=Main

- For which levels of granularity can metadata be provided?
- Which level of granularity is technically feasible
    - to be handled in a repository system?
    - to provide access to (on data level)?

In case of the wortschatz data all the questions stated above lead to the decision to provide metadata on the level of sources (and days; will be added in the future) on the top level of OAI-PMH. While it was possible to formulate use cases in which providing metadata (and access to the data) on these levels of granularity made sense, the *sentence*, *wordform,* and *document* levels simply resulted in too many elements for the workflow and fedora to handle (see section 5 for details). It remains an open task for infrastructure projects like CLARIN to define best practices concerning the granularity of metadata and data access to resources one should provide.

5. CMDI components and profiles

CMDI is able to represent metadata of linked resources. It also provides rich functionality like the referencing of sub- or super-components. These features were essential for the solution implemented in Leipzig. This solution is based on a CMDI-profile, whose components are linked with each other, and a webservice named CMDI-WS. The CMDI-WS is a RESTful webservice that creates CMDI profile instances on-demand. It is used in order to provide the metadata files utilised by other components (Fedora Repository / OAI provider module) of the infrastructure.

The wortschatz CMDI profile is based on the central component `Corpus`. A `Corpus` consists of a set of documents (`DocumentList`) with additional attributes. Every `DocumentList` contains at least one document (`Document`). A document is a collection of sentences (`SentenceList`). The component at the finest level of granularity is `Sentence` (figure 4). All of these components can be found by searching the component registry[27].

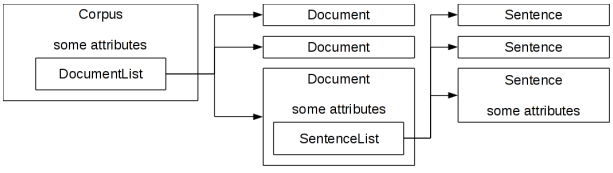


Figure 4 : CMDI profile

This highly flexible component oriented design is reflected in the structure of the CMDI profile and by the interface of the CMDI-WS. This design of the profile and CMDI-WS can serve all of the previously mentioned (see section 4 "Granularity") levels of granularity by generating only the relevant parts of metadata of the requested entity (which may also be additional ones consisting out of these components). Related parts are referenced using the elements `ResourceProxyList` or `IsPartOfList` of the CMDI basic structure.

A resource proxy element allows to differentiate between two resource types:

- `Metadata`: instances of other CMDI profiles
- `Resource`: a representation of the described resource (e.g. a text dump of a corpus)

CMDI currently allows two approaches of referencing metadata:

The first approach is to use a "quiet reference": The element is not displayed in the component tree, but the resource proxy allows to find related metadata.

```
<?xml version="1.0" encoding="UTF-8"?>
…
 <Resources>
  <ResourceProxyList>
   <ResourceProxy id="lrt">
    <ResourceType>Metadata</ResourceType>
     <ResourceRef>myressource.cmdi</ResourceRef>
   </ResourceProxy>
…
</Resources>
<Components>
 <!-->No reference to the resource proxy.<-->
…
</Components>
</CMD>
```

Snippet 2: referencing via "quiet reference"

The second more explicit approach is to create an empty element that points to the created resource proxy by directly using the XML schema instance `ID` and `IDREF` attributes. This way crawlers are able to crawl through metadata files while at the same time these CMDI files are successfully validated by XML schema validators. Due to these benefits, this option is used in the metadata documents created by the CMDI-WS.

```
<?xml version="1.0" encoding="UTF-8"?>
…
 <Resources>
  <ResourceProxyList>
   <ResourceProxy id="component1">
   </ResourceProxy>
  </ResourceProxyList>
…
 </Resources>
 <Components>
  <SourceProfile>
   <Source
ComponentId="clarin.eu:cr1:c_1311927752347"
ref="component-1"/>
   </SourceProfile>
  </Components>
</CMD>
```

Snippet 3: referencing via `ID` and `IDREF`

The main difference between both approaches is whether the referenced additional metadata is obligatory or not. A "quite reference" means additional metadata, which is not required by e.g. XML schema. The second approach references metadata explicitly required by an XML schema.

Please note: Circular dependencies between components are resolveable by using the reference functionality of

[27] http://catalog.clarin.eu/ds/ComponentRegistry

CMDI. The CLARIN component registry (*[6] Broeder. Kemps-Snijders, Van Uytvanck, Windhouwer, Withers, Wittenburg, Zinn*) accepts the definition of circular dependencies but is currently unable to generate XML schemas from component profiles containing such circles[28].

While all of the different levels of granularity are accessible via CMDI-WS, only some of those (corpus, every source) are used and translated directly into unique entities registered to the Fedora repository and therefore represented by OAI-PMH records. Finer levels of granularity are referenced in those CMDI components, but not registered to fedora as unique entities.

This is due to the high number of components and therefore the high number of metadata documents on these levels of granularity. Up to now corpora in more than 130 different languages were published on the wortschatz corpora portal and it is planned to provide more in the near future. The size of each corpus is usually measured by the number of sentences contained in it. Common sizes for our norm size corpora are 10,000, 100,000, 1,000,000, 3,000,000 and 10,000,000 sentences. As there already exist corpora with up to several 100 millions of sentences, future norm size corpora may even be significantly larger.

As stated in section 4 "Granularity" fedora is able to handle "only" several millions of entities. In case several hundreds of the wortschatz corpora need to be added to fedora, the levels of granularity to provide metadata for are limited to the levels of sources and sentences per day. Otherwise fedora would need to handle billions of entries, which is not manageable.

Therefore the decision to use the level of sources as the finest level of granularity managed by fedora and accessible via OAI-PMH was made. In order to reflect this decision in the profile, the components `Source` and `SourceList` were introduced in between `Corpus` and `DocumentList`. Every `SourceList` has at least one source (`Source`) and each of those represents a set of documents and is harvestable as an individual OAI-PMH record (figure 5).

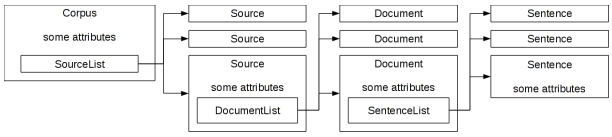


Figure 5 : CMDI profile including SourceList and Source

In order to give an example on the difference between source and document: In the newspaper corpora a source represents the web portal of a specific newspaper, whereas a document corresponds to an article that was published on this portal.

Following this approach only a manageable amount of entities has to be added to the Fedora Repository resulting in the same manageable amount of unique metadata records harvestable via OAI-PMH. Even one of the larger wortschatz corpora of about 260 million sentences contains "only" 536.288 sources (see table 1 in section 4 "Granularity"). This means even a few hundreds of these corpora are manageable using a single fedora instance.

Since the contents of this corpus were collected by "free" webcrawling, we do not expect the #sources / #sentences ratio to be much bigger in other cases (except for corpora created from large, single (re)sources like wikipedia).

Metadata on documents and sentences is still published by adding this metadata to the metadata record of each `Source`. Webservices providing data access to this level (and the corpus level) will be available in the future too.

6. Updates and Versioning of Metadata

In order to clarify things: We will not talk about versioning of `data (FDOs, entities)` in this section. Fedora offers support for versioning of the handled entities/resources. Since we are using fedora just to provide metadata on resources, but not to handle/archive the resource itself, we will probably not make use of these capabilities and cope with the versioning of data independently in our archiving system. Therefor we will just talk about updates and versioning of `metadata` from here on.

As stated above, instead of adding all the metadata directly to the `cmdi` datastream of the FDO, a link that points to a RESTful webservice was used. Since the process of adding huge amounts of FDOs is time consuming, this allows for easier updates of the associated metadata. It also significantly reduces the amount of data that needs to be stored by the repository (or any other system managed externally) since in case of the wortschatz corpora all metadata is either already present in the databases itself or can be created on the fly.

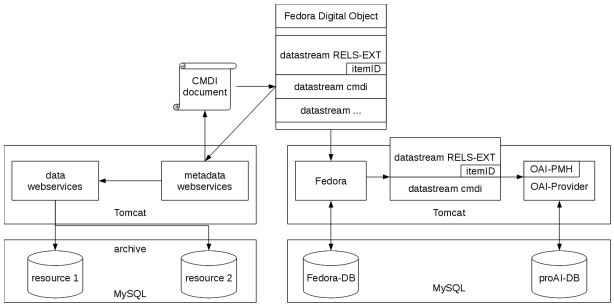


Figure 6: metadata webservice

There is no need to make changes to the data stored in the repository system in case updates of the metadata are available. Usually only a re-deployment of the updated metadata webservices and in some case additional data stored in the archive system is necessary. Adding a common German corpus of newspaper texts containing one million sentences to fedora resulted in adding 801 elements (one element that represents the whole corpus and 800 elements representing each source). In our setup

[28] http://catalog.clarin.eu/ds/ComponentRegistry/rest/registry/profiles/clarin.eu:cr1:p_1320657629642/xsd

this took only a few minutes. But while updating the metadata of one resource by updating all associated elements stored in the repository is no problem, it gets very time consuming in case this has to be done for hundreds of them that in some case might be several orders of magnitudes larger (resulting in much more elements to add) than one millions sentences.

When talking about handling updates, the problem of versioning has to be addressed as well. In the described setup versioning of metadata means, that older versions of the metadata webservice are not removed on updates. The datastreams stored in the FDOs should point to a current version of the metadata webservice, for example:

```
http://www.myhost.de/metadata/current/
resource123
```

while older versions remain available through:

```
http://www.myhost.de/metadata/{someVer
sion}/resource123
```

OAI-PMH requests to the repository will always make use of the most recent version of the metadata webservice and therefore will be answered with up to date metadata contained in the OAI-PMH records. But in case one points to the metadata using a PID, the "resource" (metadata document) stays available and unchanged since one could point and resolve to a concrete version:

```
http://www.myhost.de/metadata/{someVer
sion}/resource123
```

By using a metadata webservice and the versioning approach described above not only storing a huge amount of metadata documents describing a certain resource, but als storing various versions of this data in the repository system is avoided. Since metadata in some cases can be even larger than the data that is actually described, this results in saving a fair amount of storage space.

In order to be independent from the repository system and probably also because of performance issues the oaiprovider module caches all records. Therefore in case of changes to the `cmdi` datastream (for example because of updates of the metadata) these changes are not reflected by the oaiprovider-module. Even in case a FDO is completely removed from the fedora repository, the oaiprovider module still provides the metadata previously available for this entity. The only solution we found to this is to manually remove all of the data cached by the oaiprovider-module on update. This triggers a re-fetch of this data and only from this point on the updated metadata records are provided via OAI-PMH by the oaiprovider module.

Since doing mass ingests on fedora produces a fair load on the system, we use two separate fedora instances running in two otherwise identical virtual machines. One system is used for testing purposes while the other acts as the productive system. In case new data has to be added to the repository, the current state of the productive system is mirrored to the test installation and all ingests are done on this instance. Once completed some consistency checks are performed and, if successful, the state of the test machine is mirrored back to the productive system.

7. PIDs & PartIdentifiers

The usability of an infrastructure strongly depends on the way data and meta data can be accessed and referenced. It is well known that using Uniform Resource Locators (URLs) as the standard mean for referencing resources can not guarantee persistence which is essential for a long-term project like CLARIN. To overcome this problem CLARIN uses the Persistent Identifier Service based on the Handle system that allows stable references for every possible resources (*[5] Sun 2001*). The Handle system uses an infrastructure of distributed servers where data provider can register new resources and registered handles are maintained. Every resource is assigned a persistent identifier (PID) that references the respective resource. Users that want to resolve a PID (i.e. finding the "physical" location of the designated resource) use the resolving service of the system.

The identifiers that are used have a two-staged structure to address both resources as a whole and specific parts. This distinction is crucial for complex and comprehensive resources like the corpora provided by the Leipzig Corpora Collection: assigning single PIDs to every possible level (like every source, document or sentence) would require millions of handles even for medium sized corpora. As a consequence every corpus is assigned a persistent identifier whereas the parts of the particular corpus are addressed by so called part identifiers that are only valid in respect to the PID.

Handles correspond to a specified structure, including an identifier for the PID service, the institution (in this case `229C` for the NLP group in Leipzig) and the specific resource. In contrast part identifiers follow no schema and fall to the authority of the institution responsible for the registration of the handle. These part identifiers reflect the granularity that was sketched above (cf. section 4 on granularity), supporting an easy "zoom in"-functionality into a resource. An example of a valid PID is sketched in figure 7. The handle

`11858/00-229C-0000-0002-7EC9-9`[29]

is the PID for a corpus.

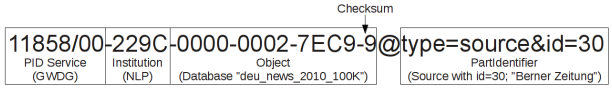


Figure 7: PID

By resolving the PID one can find the link to the CMDI file that contains all meta data about the corpus at the Leipzig resource center[30]. To address parts of the corpus a

[29] http://handle.gwdg.de:8080/pidservice/read/view?pid=11858/00-229C-0000-0002-7EC9-9

[30] http://clarinws.informatik.uni-leipzig.de:8080/cmdi/11858/00-229C-0000-0002-7EC9-9

part identifier is attached to the PID separated by an '@' character. For instance the PID

`11858/00-229C-0000-0002-7EC9-9@type=source&id=30`

addresses meta data about a specific source (in this case articles of a Swiss newspaper).

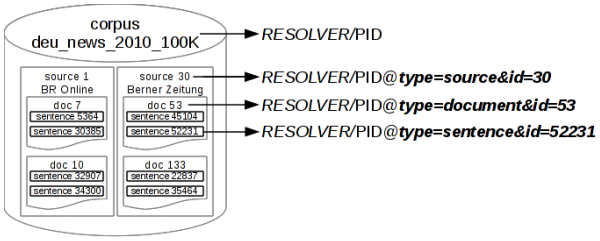


Figure 8: PIDs, PartIdentifiers and granularity

Figure 8 shows some examples for addressing parts of a corpus. The illustrated part identifiers reflect the natural structure of the LCC corpora (source, document, sentence). Additionally references to further parts are supported, like "all documents of a specific source" or "all sentences of a specific document".

The task of maintaining and resolving the persistent identifiers of the Leipzig CLARIN resource center was delegated to the GWDG[31] PID Service[32]. Unfortunately at the moment the resolving of PIDs with part identifier is not supported. Therefore it is only possible to address parts of resources by manually using the Leipzig meta data web service, instead of being redirected by the GWDG resolver automatically. The support of this enhanced functionality is expected soon.

8. Conclusion

Providing metadata is a complex problem. Although supportive standards (OAI-PMH, CMDI) and software solutions (repository systems) do exist, configuring and using them is only part of the problem. Additionally the topics granularity, updates/versioning and persistency have to be addressed.

From our experience, the steps to take in order to provide metadata are the following:

1) define the level(s) of granularity to provide metadata at, by taking into account in which levels of granularity:
   - o users of the ressource (want to) make use of the data/metadata
   - o data can be accessed
   - o metadata is available
   - o access is technically feasible
2) decide for standards to be supported
   - o format of the metadata
   - o interfaces to the repository system (e.g. OAI-PMH)
3) decide for a repository system that
   - o was already successfully tested in similar usage scenarios
   - o offers the functionality needed in order to provide metadata in the formats and levels of granularity previously defined
   - o supports the obligatory standards/interfaces (e.g. OAI-PMH)
   - o fits the legal and financial constraints (e.g. open/closed source; free/commercial solution)
   - o is actively developed and supported and will be so in the foreseeable future
4) design and implement a workflow that
   - o creates/converts the metadata
   - o adds metadata to (or provides metadata for) the repository system
   - o is able to handle updates/versioning

Infrastructure projects like CLARIN simplify some of these tasks, since several of the questions stated above (e.g. interfaces, formats) are already answered by standardization. Additional guideslines (e.g. best practices on granularity) are usually available too.

CMDI proved to be a complex but fitting solution. The component based approach allows for flexibility and therefore easy adaption of the defined metadata profile/schema. In our specific case this flexibility enabled the late decision on the question of the level of granularity.

The granularity of metadata remains an interesting question to tackle in the infrastructure projects. This is not limited to repositories and OAI-PMH but also incorporates other infrastructure components that handle "query for data" scenarios that not only return data but also attach metadata to the resultset.

9. References

[1] Quasthoff, U.; M. Richter; C. Biemann (2006): *Corpus Portal for Search in Monolingual Corpora.* Proceedings of the fifth international conference on Language Resources and Evaluation, LREC 2006, Genoa, pp. 1799-1802.

[2] Büchler, M. and Heyer, G. (2009): Leipzig Linguistic Services - *A 4 Years Summary of Providing Linguistic Web Services.* Gerhard Heyer (Editor): Text Mining Services – Building and applying text mining based service infrastructures in research and industry., Leipzig, Germany.

[31] Gesellschaft für wissenschaftliche Datenverarbeitung mbH Göttingen, http://www.gwdg.de/index.php

[32] http://handle.gwdg.de:8080/pidservice/

[3] Aguillo, I.F.; Ortega J.L.; Fernández M.; Utrilla A.M. (2010): *Indicators for a webometric Ranking of Open Access Repositories.* Scientometrics, 82 (3): 477-486.

[4] Aguillo, I.F. (2011): *The July 2011 Webometrics repository ranking.* DSpace User Group Meeting, OAI7 CERN Workshop on Innovations on Scholarly Communication, Geneve Switzerland, 21-24 June 2011.

[5] Sam X. Sun (2001): *Establishing Persistent Identity Using the Handle System.* Tenth International World Wide Web Conference, Hong Kong, May 2001.

[6] Broeder D.; Kemps-Snijders M.; Van Uytvanck D.; Windhouwer M.; Withers P.; Wittenburg P., Zinn C. (2010): *A Data Category Registry- and Component-based Metadata Framework.* LREC 2010; Malta.

The Component Metadata Infrastructure (CMDI) in a Project on Sustainable Linguistic Resources

Thorsten Trippel[1], **Christina Hoppermann**[1], **Griet Depoorter**[2]

[1]Eberhard Karls University Tübingen (Germany), [2]Institute for Dutch Lexicology (The Netherlands)
[1]first.last@uni-tuebingen.de, [2]first.last@inl.nl

Abstract

The sustainable archiving of research data for predefined time spans has become increasingly important to researchers and is stipulated by funding organizations with the obligatory task of being observed by researchers. An important aspect in view of such a sustainable archiving of language resources is the creation of metadata, which can be used for describing, finding and citing resources. In the present paper, these aspects are dealt with from the perspectives of two projects: the German project for Sustainability of Linguistic Data at the University of Tübingen (NaLiDa, cf. `http://www.sfs.uni-tuebingen.de/nalida`) and the Dutch-Flemish HLT Agency hosted at the Institute for Dutch Lexicology (TST-Centrale, cf.`http://www.inl.nl/tst-centrale`). Both projects unfold their approaches to the creation of components and profiles using the Component Metadata Infrastructure (CMDI) as underlying metadata schema for resource descriptions, highlighting their experiences as well as advantages and disadvantages in using CMDI.

Keywords: CMDI profile creation, CMDI experiences, CMDI infrastructure use

1 Introduction and Motivation

In the field of archiving language resources (LRs), there is a general need for describing primary research data by metadata. Metadata contribute to the sustainability of resources by being searchable, accessible and citable. Finding resources itself is essential for their reuse, quality assurance, establishment of cooperations and citations in publications. Here, reuse means that resources, such as corpora or lexical databases, could be used by others who are not necessarily their creators. Likewise, such resources could also be applied to other purposes than originally foreseen when designing the resource. Quality assurance (QA), on the other hand, enables a reviewing process to check the achieved results, recalculate figures and hence to prevent fraud and plagiarism. It is part of the academic tradition to foster reproducible results that also enable competing ideas to achieve comparable outcomes. Reuse and QA are increasingly important for funding organizations, such as the German Research Foundation (DFG), which, for instance, requires researchers to store primary data for a period of ten years (cf. Deutsche Forschungsgemeinschaft, 1998).

In practice, metadata descriptions are made available to services such as search engines. These services enable users to find information on LRs and provide them with a first impression of the research data as such (cf. Barkey et al., 2011).

Utilising this scope of application, metadata descriptions take an essential function in the context of archiving language resources and primary research data. Metadata is used in a similar way as in library catalogues: the catalogue files contain relevant information for users to find literature, cite it and to be provided with pointers to the storage location of books. This procedure is also applied to the use of metadata descriptions within an archive. In this context, metadata contributes to the ability of finding, citing and persistently accessing resources.

The requirements on metadata schemas for describing different types of language resources differ significantly from those on entrenched standards in the librarianship, such as Dublin Core (see Section 2). Addressing these requirements, the Component Metadata Infrastructure (CMDI, cf. `http://www.clarin.eu/cmdi`) was developed within the context of the European project CLARIN (cf. `www.clarin.eu`).

The approaches to the creation of CMDI metadata schemas presented in this paper are based on work within two projects: the German project for Sustainability of Linguistic Data at the University of Tübingen (NaLiDa) and the Dutch-Flemish HLT Agency hosted at the Institute for Dutch Lexicology (TST-Centrale). Both projects were involved in the creation of some of the first CMDI components and profiles prior to the availability of the complete functionality of the CMDI infrastructure such as the Component Registry (cf. Section 2). First components and profiles were created using standard XML techniques (cf. `http://www.clarin.eu/toolkit`). With the increased functionality of the Component Registry, the authors have been among its first productive users, providing the Registry's developers with feedback based on their experiences.

With a background in archiving linguistic resources, the present papers reports on the principles applied for creating components and profiles in CMDI. It also highlights the authors' experiences during this process. Concentrating on the design principles and experiences, the paper is structured as follows: Section 2 introduces the background of our work. Section 3 deals with the entire process of creating CMDI components and profiles whereas Section 4 gives account of the challenges within this procedure. The interoperability of CMDI with existing metadata schemas is taken into consideration within Section 5, and Section 6 concludes and gives an outlook on future work.

2 Background

For the purpose of describing linguistic research data, various metadata schemas have been used in the past. Among them are, inter alia, Dublin Core (DC, cf. http://dublincore.org/), OLAC (Open Language Archives Community, cf. http://www.language-archives.org/) and IMDI (ISLE Metadata Initiative, cf. http://www.mpi.nl/IMDI/).

Dublin Core is a metadata schema which is mainly used for (printed) publications within the librarianship. Metadata categories that are traditionally associated with DC are, for instance, title, author, date, etc. Although Dublin Core has become much more varied in the meantime — allowing subspecifications of metadata categories (by "qualifying" them) — often only the 15 core metadata categories are meant by referring to Dublin Core (cf. Hillmann, 2005). For language resources, DC lacks some levels of expressiveness, since, for example, the different roles of persons involved in the creation of a resource cannot appropriately be represented or the project context cannot be embedded.

Increasing requirements for archiving language-related material led to the development of the Open Language Archives Community's metadata set (OLAC, cf. Simons and Bird, 2008). This extension of the original Dublin Core was better adjusted to the needs of language archives. For example, it introduced qualifiers that had not been available in Dublin Core, such as participant roles, linguistic fields, etc. Nonetheless, the schema's expressive power remains similar to that of Dublin Core.

In contrast to OLAC, supposed to maintain compatibility with Dublin Core, the ISLE Metadata Initiative developed a metadata set to describe and differentiate particular primary data (IMDI, 2003; IMDI, 2009). This format was especially created for resources involving the recording of one or more persons and annotating these signal files. Other classes were not described. Still, IMDI is very detailed and users can be overwhelmed by the large amount of data they can or even have to provide.

A recent approach, the Component Metadata Infrastructure (CMDI, cf. Broeder et al., 2010; Broeder et al., 2012; de Vriend et al., 2010), addresses these issues, allowing a flexible possibility of including metadata categories required for specific classes of resources, while reusing existing structures and parts. To establish CMDI as an international standard, a new work item for ISO 24622 has been initiated within ISO TC 37 SC 4. Due to its flexibility, other metadata schemas can be easily represented in CMDI. For individual types of resources it is possible to create adjusted metadata schemas. In this paper, we use the CMDI terminology referring to the three base concepts underlying the schema: profiles, components and elements with their values. All of them are implemented as XML elements in the metadata instances. For not confusing XML elements with metadata elements, we use the ISOcat terminology of *(meta)data categories* to refer to the conceptual level of those elements (i.e. in XML: terminal elements) that have a value and whose concepts are defined in a data category registry. Further, *components* are collections of semantically grouped metadata categories, which serve as building blocks for profiles and which may also contain further components. A *profile* is a metadata component which is used as a template for describing a specific resource class and which is not embedded into other components. Both profiles and components are registered within the Component Registry (cf. http://catalog.clarin.eu/ds/ComponentRegistry/) whereas the metadata categories are linked to concepts in ISOcat, the Data Category Registry for ISO TC 37 (cf. http://www.isocat.org; ISO 12620, 2009). In practice, CMDI is not only used, for instance, within CLARIN(-related) projects. Other groups using the component model for metadata include, for example, META-SHARE (cf. www.meta-net.eu/meta-share/) with their metadata model oriented towards NLP resources (cf. http://www.meta-net.eu/meta-share/metadata-schema/).

3 Creating CMDI Components for Different Resource Types

This section discusses the creation of CMDI metadata components. For this purpose, emphasis is placed both on the reuse and the modification of substructures as well as on the creation of new substructures. In terms of substructures, it is distinguished between complex (high-level components having sub-components) and simple substructures (low-level components not containing any sub-components).

3.1 Types of Resources and User Groups

In various project contexts, we worked with different types of resources forming the basis for the CMDI profiles and components discussed here. Among them are: lexical resources, text and speech corpora, grammars, experimental data, tools and web services. These kinds of resources all have in common that they are electronically available and were created in the contexts of research projects. Some of them have restricted uses, others are freely available.

Due to this variety of different kinds of primary research data, various components were needed for the different levels of description. Though this increases the variety of the full metadata schema, many substructures could be reused. Moreover, the functionality of the components had to meet the requirements of the project's user groups. These groups were mainly composed of researchers in the field of linguistics who were neither experienced in the provision of metadata nor necessarily in XML technologies.

3.2 Reuse of Components

For most of these resources, we were able to reuse a wide range of components that had already been available within the public space of the Component Registry (cf. http://catalog.clarin.eu/ds/ComponentRegistry). Figure 1 illustrates a very general organization into components that can easily be reused, since they are not specific to a particular type of resource.

For the purpose of reuse, high-level components have the advantage that they can easily be integrated and that their implicit structure is already rich for being used in applications. Some of the high-level components we found suitable for reuse are illustrated in Table 1 (cf. third column). These components are comparatively general and

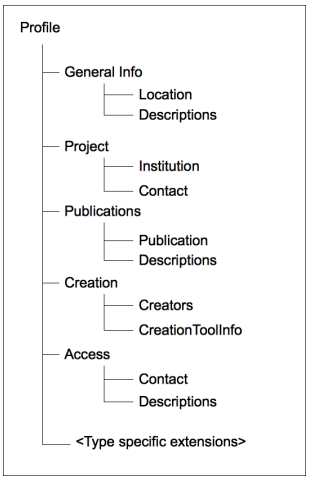


Figure 1: Component structures often used in the creation of profiles.

come with elaborate substructures, though none of these is mandatory.

In terms of low-level components, they distinguish themselves by their ability of being integrable into a number of other higher-level components. This integration is possible, as they do not contain further components themselves and are thus not as restricted to specific uses as high-level components. This characteristic results in a frequent reuse of lower-level components within our projects. Some examples are provided in Table 2 (cf. third column).

3.3 Recycling of Components

Especially within the German NaLiDa project, the reuse of existing components was often not sufficient. This was, for instance, the case when metadata categories were missing or further low-level components were needed. Therefore, existing components were adjusted by recycling them. This recyling process was mainly conducted for those components that were originally developed by Clarin-NL within the project *Creating and Testing Metadata Components*.

When recycling, the aforementioned higher-level components were used as a basis for creating derived components that are illustrated in Table 1 (cf. 4th column). Likewise was the procedure for recycling lower-level components, as shown in Table 2 (cf. 4th column).

Apart from creating new components (cf. following section), the reasons for not only reusing existing components but also changing them within the development process of a profile are manifold. For instance, the low-level *description*-component (cf. Table 2) needed to be modified due to changes within the general schema of CMDI. With these changes, the option of indicating languages used within the metadata categories' values (i.e. strings) became available by indicating the *xml:lang*-attribute. The recycled *description*-component allowed this attribute so that a new version was generated. Moreover, to allow for multilingual metadata, we also changed the metadata category's cardinality from one to unbounded. Thereby, the use of various *description*-elements was enabled within the component and, thus, also within the metadata instance whose contents are written in different languages.

Cardinality is a frequent cause for minor modifications of existing components. For instance, restricting the use of a metadata category such as *person* in a way that it can only occur once could lead to inserting enumerations of proper names into a single data category's value. Because this may require additional processing, there should be one metadata field per name of a person.

Another reason for modifying existing components is the need to add further metadata categories. This process is quite frequent during the development of a CMDI profile. Some components cannot be reused because they are to restricted by definition. Thus, they are designed for specific situations which makes it hard to reuse them for other purposes.

In all of these cases, recycling of components offers the possibility of both improving existing components by providing extensions and creating almost new components by using already defined substructures. Recycling also results in the disadvantage of enlarging the Component Registry and contributing to its complexity. This is less user-friendly, since often almost identical components are registered twice without being highlighted as such. This aspect is currently dealt with by the Component Registry's developers at the Max Planck Institute for Psycholinguistics in Nijmegen and will eliminate the aforementioned disadvantage after its realisation.

3.4 Creating New Components

When there is either no component available for describing a particular aspect of a resource or no existing component can be recycled, a new one has to be created. First of all, a component should be a collection of related metadata categories (and components). For instance, the component *speech-technical* describes all technical metadata about speech in a corpus (such as bit resolution, compression, number of channels, etc.). Second, when a (group of) metadata category(ies) can be used more than once within the same profile or for various profiles, it should be incorporated in a separate component to facilitate the reuse of the component. Once the content of the new component is established, the decision has to be made whether the metadata categories (or incorporated components) could occur more than once (i.e. defining the cardinality) and whether they should be mandatory. A new component does not always need to be entirely built from scratch: often it is possible to incorporate existing components, especially low-level components, such as the *language*- or *location*-component.

3.5 Selecting, Adding and Modifying Data Categories

To guarantee semantic interoperability, each metadata category (and its values) should be linked to one (widely agreed upon) concept that is either registered in ISOcat or in other trusted registries. Those concepts are called data categories and they are uniquely identifiable by a persistent identifier (PID), which is a uniform resource identifier (URI). If a concept has not yet been registered, it is possible to add it in the so-called ISOcat user space. Those new concepts can be submitted to a standardization process, after which they can gain the status of being standardized.

Table 1: Examples of reused and recycled high-level components.

Name	Description	Original Component for Reuse	Derived Component for Recycling (unpublished)
General Info	A component containing general information on a resource by grouping various metadata categories as used by Dublin Core, such as the title or name of a resource, its version, legal owner, location or description.	http://catalog.clarin.eu/ds/ComponentRegistry/rest/registry/components/clarin.eu:cr1:c_1271859438123/xml	http://catalog.clarin.eu/ds/ComponentRegistry/rest/registry/components/clarin.eu:cr1:c_1290431694495/xml
Project	A component consisting of details on the project in which a resource was created in.	http://catalog.clarin.eu/ds/ComponentRegistry/rest/registry/components/clarin.eu:cr1:c_1271859438125/xml	http://catalog.clarin.eu/ds/ComponentRegistry/rest/registry/components/clarin.eu:cr1:c_1290431694522/xml
Creators	A component documenting the creators of a resource, such as their names, roles within the creation process, contact information, etc.	http://catalog.clarin.eu/ds/ComponentRegistry/rest/registry/components/clarin.eu:cr1:c_1271859438134/xml	http://catalog.clarin.eu/ds/ComponentRegistry/rest/registry/components/clarin.eu:cr1:c_1290431694499/xml
Access	A component specifying the possibilities of accessing a resource, such as its availability, legal issues, contact details, etc.	http://catalog.clarin.eu/ds/ComponentRegistry/rest/registry/components/clarin.eu:cr1:c_1271859438124/xml	http://catalog.clarin.eu/ds/ComponentRegistry/rest/registry/components/clarin.eu:cr1:c_1290431694501/xml
Subject Language	A component identifying the language(s) included in a resource and indicating whether each language is the dominant language, the source language and/or the target language.	http://catalog.clarin.eu/ds/ComponentRegistry/rest/registry/components/clarin.eu:cr1:c_1271859438126/xml	http://catalog.clarin.eu/ds/ComponentRegistry/rest/registry/components/clarin.eu:cr1:c_1290431694564/xml

Table 2: Examples of reused and recycled low-level components.

Name	Description	Original Component for Reuse	Derived Component for Recycling (unpublished)
Descriptions	A component allowing prose text written in various languages, indicated by using the *xml:lang* attribute. This optional component is reused in almost every component, as users uttered their demand of free-text fields in addition to the semantically specified values of other data categories.	http://catalog.clarin.eu/ds/ComponentRegistry/rest/registry/components/clarin.eu:cr1:c_1271859438118/xml	http://catalog.clarin.eu/ds/ComponentRegistry/rest/registry/components/clarin.eu:cr1:c_1290431694486/xml
Country	A component indicating the location of something by giving country names with their corresponding ISO country codes.	http://catalog.clarin.eu/ds/ComponentRegistry/rest/registry/components/clarin.eu:cr1:c_1271859438104/xml	http://catalog.clarin.eu/ds/ComponentRegistry/rest/registry/components/clarin.eu:cr1:c_1290431694493/xml

4 Challenges of the Current Profile Creation Workflow

For creating a new profile, the ideal workflow is to have all data categories available in a data category repository, to select all required components and finally combine those components to a profile. In practice, this workflow cannot always be followed. Deviations and procedures to avoid potential problems are discussed within this Section.

4.1 The Cascade of Updating Components

For updating components an additional level of complexity is added. If a component is used by another component, updating this component may lead to a cascade of further changes in other components. In the following, the reason for this cascade will be explained, providing an estimate of the work involved and presenting a strategy for avoiding such a cascade of changes.

In CMDI, a component may make use of other components

by referring to them. For creating these references, each component has a unique and persistent identifier. However, if a component is published in the Component Registry, the mapping of the identifier to the component is defined as persistent. This persistency includes that changes to a published component are not allowed. Consequently, a change of a component results in a new derived component with a new identifier (i.e. a copy of the original component including the modifications) while the original one remains unaltered.

The most frequent case for such changes experienced by the authors is that a component *A* is extended by new optional metadata categories, resulting in a derived component *A'*. If such a component *A* has already been referred to within another component *B*, the component *A* should be replaced there as well by its derived version *A'* to allow for interoperability and to foster the reuse of components. However, such a replacement also results in a new component *B'*. As component *B* may have been referenced as well, the references need to be updated sequentially. Finally, the profiles making use of the old components are updated. As the profiles' persistent identifiers are used for identification in the validation process, the instances also need to be updated, even if their structure can remain the same according to the new schema.

An example for such a cascade provides the component *contact* containing metadata categories for street address, email, url, phone number, etc. As the use of fax numbers is not frequent any longer, they were left out in the initial definition of the component. The analysis of legacy data, however, required to include fax numbers to allow a lossless representation of the original data. Due to the fact that the component was already published, it was impossible to add the relevant metadata category. Instead, the existing component was reused, resulting in a copy of the original component plus the additional metadata category for indicating a fax number. To make the (optional) fax number available in all components, all references to the old component in the authors' components and profiles were replaced successively by the new component. As a result, there was a number of new derived components that only differed in view of the persistent identifier for referring to the new *contact*-component.

4.2 Estimating the Consequences of an Update

The number of changes resulting from updating a component highly depends on the number of components referring to the component. As a rule of thumb, it can be summarized that the lower the level of a component which is to be modified (i.e., the more other components make use of it), the more changes are required also for other components referencing to this component.

4.3 Avoiding the Cascade

Avoiding the cascade of changes is possible and easy: components that are not stable but under development are not to be published, since non-public components can be edited while maintaining their identifiers. Hence, new functionalities become effective after saving the changes. The reuse of existing public components then preliminarily only results in copies that are very similar in the private workspace before they are published. This procedure is also conducted by the authors. Thereby, it is ensured that components/profiles stay editable as long as there are still new resource types to be added within the development process that require the creation of new or the modification of existing components/profiles. The authors use this method for extending components by optional structures to maintain compatibility to older versions.

Unpublished, non-persistent components have the disadvantage of being of a temporary nature. Hence, when a component is stable, it should be published. In situations where a close collaboration between different working groups results in intensive discussions, private workspace components can be shared using the REST-based interface to the Component Registry[1]. At present, a sharing in the Component Registry's interface is only possible by creating a group account used by multiple users with all implied problems.

4.4 Consequences of the Cascade in Practice

The cascade of changes when applied to published components results in a growing list of available components. A single change in an optional metadata category may result in many components that have almost the same functionality and possibly the same name. Figure 2 shows the Component Registry with a sample of components encountering this problem, as, for example, *Annotationstypes-SoNaR*, *Author*, or *BroadcastPublication*, only distinguishable here by the date; the creator name not shown here could also be used to distinguish the components. For displaying the persistent identifier, the context menu of the particular component needs to be opened in the XML view mode.

The list of components, as is the corresponding list of profiles, shows a flat organization. Per default the components are sorted by their names. As of winter 2012 (2012-02-14) there are 214 components and 49 profiles registered in the public space, many more are probably present or almost stable in the private workspace[2].

Searching for adequate components requires considerable background and understanding, both of the model and of the individual components. Some of the descriptions are not very helpful either, because the different versions of similar components are not distinguished. Additionally, components in the private workspace are currently not searchable in the public space. This situation will result in incompatible and inconsistent but very similarly named components when the private ones are made accessible in the Component Registry's public space.

One way of solving this challenge is by cooperation with research partners and offering complete examples of CMDI files, for example, via OAI-PMH (i.e. the Open Archives Initiative Protocol for Metadata Harvesting). The distributed examples contain references to the schemas whereas, in turn, the schemas refer to the components. This

[1]The REST-based interface is also used for the references to the private components within this paper, cf. Tables 1 and 2.

[2]The authors, for example, defined 139 private components and 10 private profiles within the NaLiDa project so far, some resulting from a cascade of changes.

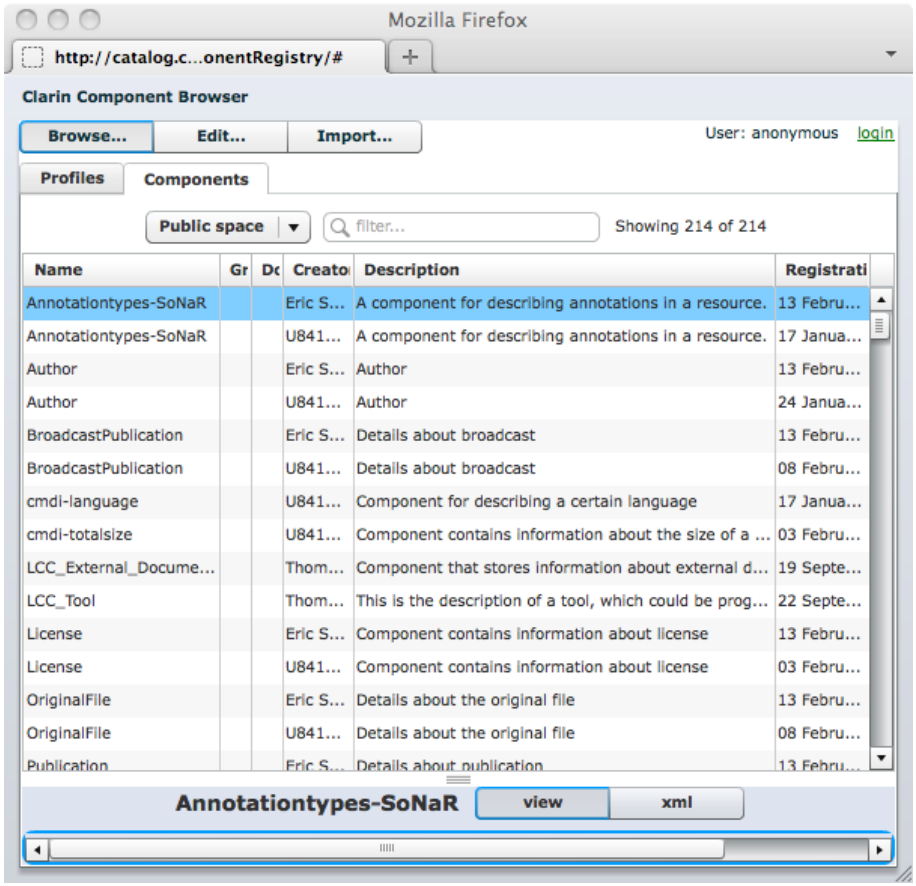


Figure 2: Published components in the Component Registry (2012-02-14) with 214 components .

way, sharing metadata files helps to communicate even on unpublished components and profiles.

In the long term we expect that there will be a mechanism in the Component Registry which supports a more transparent development of components by allowing those under development to be found in the Component Registry, flag-superseded and retired. Using such a mechanism will provide a way of maintaining all versions while indicating which ones should be used for new developments. Furthermore, it is expected that a list of recommended components will be established with a non-normative character, which forms best practice within cooperative projects, archives and data centers.

4.5 ISOcat Structures

Another challenge for the creation of CMDI profiles is the structure of data category repositories. The thematic domain group of metadata in ISOcat, for example, currently lists about 700 data categories (retrieved: 2012-02-14). Finding the relevant data categories, however, is not alway obviously. Reasons for this include, among others, the difficulty of using a search function without knowing how people name concepts within the registry as well as the lack of relations or an overview of existing concepts. Additionally, though the data categories are supposed to be standardized, the standardization process is not far advanced yet.

ISOcat consciously refrains from creating hierarchies of data categories, since those might pose a focus on specific theories and create a bias which is unwanted for a standard. Besides the curation of definitions and the standardization of ISOcat categories, no major improvement is being expected here. Figure 3 shows the ISOcat registry with a couple of entries visible in its flat structure.

Usability improvements of ISOcat can be created externally as a layer on top of it. Using ISOcat definitions and data categories it is possible, for example, to create a hierarchy (or many of those) of data categories to assist users in locating appropriate data categories, seeing what categories are already available and filling in required additions. Figure 4 gives an example of such a hierarchy in HTML used within the NaLiDa project's website. The hierarchy is also accessible as OWL representation (cf. Zinn et al., 2011).

5 Interoperability of CMDI with Other Metadata Schemas

One reason for referencing metadata categories and their values to ISOcat and Dublin Core is the aim of ensuring the interoperability of different metadata schemas. On the one hand, this is desirable especially in the case of transforming already existing metadata into another format. On the other hand, it also enables the application of existing systems or tools for the purpose of, for instance, searching for resources by means of metadata or distributing both metadata and language resources.

In principle, there are two different approaches: first the metadata is stored in parallel in different formats that are

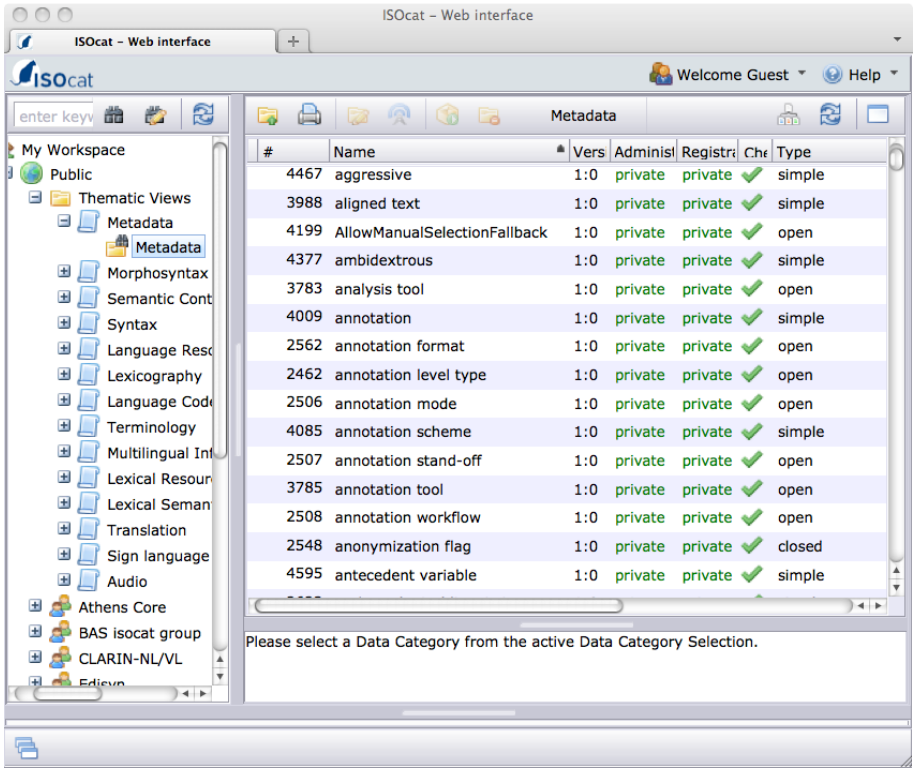


Figure 3: Metadata thematic domain in ISOcat with 700 data categories (only partially visible).

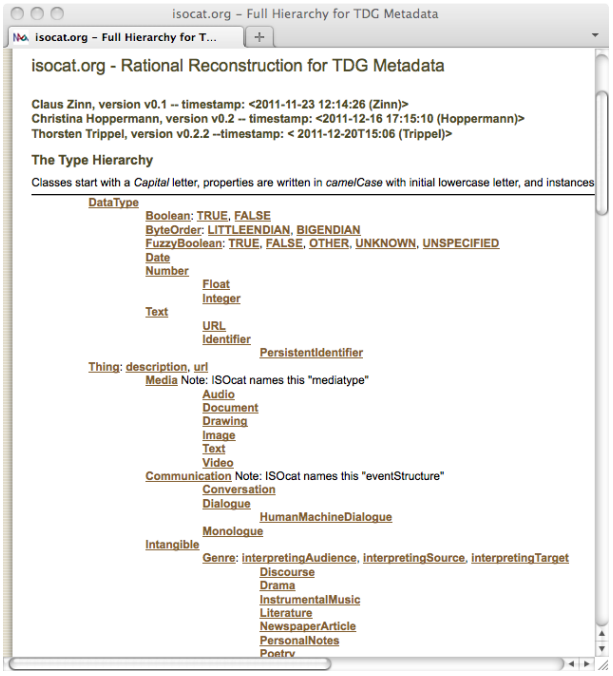


Figure 4: Part of a hierarchy layer on top of ISOcat.

independent of each other. Such a double maintenance would be labor-intensive, costly and error-prone. The second option is more appealing, which is the transformation of CMDI files to these other formats, as it is automatically done and requires no additional maintenance.

Additionally, there are further reasons for such a mapping between metadata schemas:

Dublin Core for OAI-PMH: Sharing metadata using OAI-PMH allows the use of any number of metadata formats, but the 15 core DC metadata categories are part of the *oai_dc data format* specified by the protocol. It would be possible to maintain separate Dublin Core files, but as they usually contain less information, this information can also be extracted from the CMDI files. In contrast, due to its complexity, an entire CMDI file could not be mapped to DC, so the transformation to DC is lossy.

OLAC: Worldwide, the community of OLAC users has services for harvesting and searching for resources. As it is the case with DC, the CMDI community can easily support the OLAC formats by a lossy transformation.

IMDI: In the area of spoken language documentation, tools have been developed for working with IMDI metadata. The mapping to IMDI could be desired, but as it is not as flexible as CMDI an IMDI conversion is only possible for resources that could also have been described with IMDI from the start. It is expected that in due time IMDI tools and descriptions will be replaced by CMDI.

Technically, the transformation of CMDI metadata to other schemas can be achieved by mapping ISOcat data categories onto the corresponding metadata categories of the target format. For instance, taking the 15 core metadata categories of Dublin Core, a transformation process first needs to search for equivalent metadata categories in the CMDI

profiles (by means of the concept links for each metadata category). Then the corresponding values of these metadata categories need to be extracted from the CMDI files and inserted into a Dublin Core template serving as output. First transformations have already been conducted by the authors, although a productive implementation is still pending.

6 Summary and Outlook

In this paper, we presented the use of the Component Metadata Infrastructure as underlying metadata schema for creating resource descriptions in two research projects. CMDI was compared to other metadata standards while highlighting its suitability for describing different types of resources. The paper's main focus, however, represented the introduction of principles for creating, re-using and recycling components and profiles in CMDI. Additionally, the challenges within the CMDI profile creation workflow were addressed and recommendations for solving these challenges were given. Finally, the interoperability of CMDI with other metadata schemas was considered to provide a complete picture of its functionalities.

In the future, we expect further tools to be developed working with CMDI formats and archives making their CMDI descriptions available. Additionally, we advocate that CMDI will be standardized as a long-term metadata formalism. This standardization process has been initiated within ISO TC 37 SC 4 as standardization work item (ISO 24622). The standardization will allow a long-time use of this flexible metadata schema for a large variety of resources not catered for by other standards.

7 Acknowledgements

Work for this paper was conducted within the Centre for Sustainability of Linguistic Data (NaLiDa), funded by the German Research Foundation (DFG) in the program for Scientific Library Services and Information Systems (LIS), within the SFB 833 "The construction of meaning - the dynamics and adaptivity of linguistic structures", also funded by the German Research Foundation, and within the Dutch-Flemish HLT Agency (TST-Centrale), an initiative of and funded by the Dutch Language Union (Nederlandse Taalunie). The HLT Agency is hosted by the Institute for Dutch Lexicology. The tools and models were also discussed and developed with partners from the European project CLARIN and with ISOcat, which is a work item of ISO TC 37.

8 References

R. Barkey, E. Hinrichs, C. Hoppermann, T. Trippel, and C. Zinn. 2011. Trailblaizing through forests of resources in linguistics. In *Digital Humanties*, Stanford. Stanford University.

D. Broeder, M. Kemps-Snijders, D. Van Uytvanck, M. Windhouwer, P. Withers, P. Wittenburg, and C. Zinn. 2010. A data category registry- and component-based metadata framework. In *Proceedings of the 7th conference on International Language Resources and Evaluation*, Malta.

D. Broeder, D. van Uytvanck, M. Gavrilidou, and T. Trippel. 2012. Standardizing a component metadata infrastructure. In *Proceedings of the 8th Conference on International Language Resources and Evaluation (LREC 2012)*, Istanbul.

F. de Vriend, D. Broeder, G. Depoorter, L. Van Eerten, and D. Van Uytvanck. 2010. Creating & testing clarin metadata components. In *Language Resource and Language Technology Standards - State of the Art, Emerging Needs, and Future Developments Workshop, 7th International Conference on Language Resources and Evaluation (LREC)*.

Deutsche Forschungsgemeinschaft. 1998. Vorschläge zur Sicherung guter wissenschaftlicher Praxis: Empfehlungen der Kommission "Selbstkontrolle in der Wissenschaft". Denkschrift. Weinheim: Wiley-VCH. See `http://www.dfg.de/aktuelles_presse/reden_stellungnahmen/download/empfehlung_wiss_praxis_0198.pdf`.

D. Hillmann. 2005. Using dublin core - the elements. Technical report, Dublin Core Metadata Initiative, November. http://dublincore.org/documents/2005/11/07/usageguide/elements.shtml.

IMDI. 2003. Metadata elements for session descriptions, draft proposal version 3.0.4. `http://www.mpi.nl/IMDI/documents/Proposals/IMDI_MetaData_3.0.4.pdf`, October.

IMDI. 2009. Metadata elements for catalogue descriptions, version 3.0.13. `http://www.mpi.nl/IMDI/documents/Proposals/IMDI_Catalogue_3.0.0.pdf`, August.

ISO 12620. 2009. Terminology and other language and content resources - specification of data categories and management of a data category registry for language resource. Technical report, ISO.

G. Simons and S. Bird. 2008. Olac metadata. Technical report, Open Language Archive Community. http://www.language-archives.org/OLAC/metadata-20080531.html.

C. Zinn, C. Hoppermann, and T. Trippel. 2011. Hierarchy of isocat data categories. [Online; based on an ISOcat snapshot of November 2011, accessed 2012-02-14, see `http://www.sfs.uni-tuebingen.de/nalida/en/docu/isocat-hierarchy.html`].

Experiences and Problems creating a CMDI Profile from an Existing Metadata Schema

Hanna Hedeland, Kai Wörner
Hamburg Center for Language Corpora, University of Hamburg
Max-Brauer-Allee 60, D-22765 Hamburg
E-mail: hanna.hedeland@uni-hamburg.de, kai.woerner@uni-hamburg.de

Abstract

To make language resources available through the CLARIN-D infrastructure[1], corpora of spoken discourse at the Hamburg Center for Language Corpora (Hamburger Zentrum für Sprachkorpora, HZSK[2]) have to be described via CMDI compliant metadata[3]. The aim is to create metadata that can be harvested automatically and can then be used in a federated search and browsing environment to facilitate discovery as well as recombination of existing resources.
This paper describes the considerations, efforts and obstacles encountered in the process of creating a CMDI metadata profile for the HZSK. It had—based on an existing metadata format—to encompass most of the existing metadata, share as many existing components and profiles as possible and relate to metadata profiles that are being developed at other CLARIN-D projects that deal with similar resources. Much input has come from the discussion with Florian Schiel and Thorsten Trippel.

Keywords: CMDI, metadata, EXMARaLDA, ISOcat, IMDI, Spoken Language Corpora

1. Coma

The metadata for spoken language corpora at the HZSK is managed through the EXMARaLDA Corpus Manager (Coma, see Wörner 2012). The underlying schema models corpora as a collection of so-called *communications* (distinct events where the recorded communication took place) and *speakers* (the people involved in these communications). These objects relate to one another through *roles* (speakers have roles in one or more communications) and are further described through additional elements (like *recordings* for communications), of which some are fixed (like the names or the sex of the speakers), but most of them expressed in open key-value pairs.

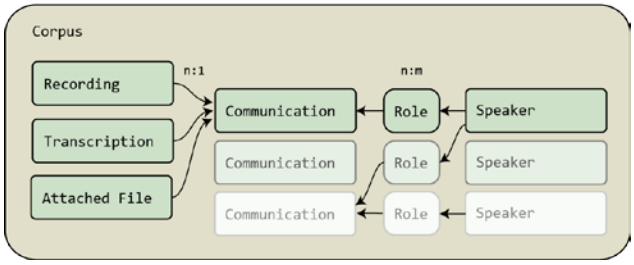


Figure 1: The Coma metadata schema

2. The Corpora at the HZSK

All spoken corpora at the HZSK are similar in that they are multilingual in some sense, but they still differ in many aspects. The metadata provided for each corpus depend on project-specific theories of language and multilingualism, on the research tradition and on the current research question. Although some information on *communications* and *speakers* is indeed common for many corpora, it is often encoded differently, for example as in a speaker's *age* or *birth date*. Other information is highly specific and only relevant for a single corpus, such as the speaker's *command of Polish before emigration to Germany*. Another type of metadata is the parameters closely related to the corpus design, such as various age groups, types of multilingualism or age on arrival in the L2 country for speakers.

3. HZSK Core Metadata Set

When developing a set of basic relevant metadata for the type of spoken corpora we handle at the HZSK, we aimed for a bottom-up approach and departed from the corpora instead of setting up yet another metadata scheme top-down. As the basis for the HZSK core metadata set, we used five corpora with elaborate metadata from the various research groups at the former research centre (*Acquisition of Multilingualism*, *Historical Aspects of Multilingualism and Variance* and *Multilingual Communication*). The result is a small set of obligatory items for the description of corpora, communications, speakers and additional files as well as guidelines for the encoding of further common categories, which remain optional. As a part of the curation process for these five "premium" corpora, we adapted existing metadata and added missing categories.

[1]http://de.clarin.eu/index.php/en/project-summary
[2] http://www.corpora.uni-hamburg.de/
[3]http://www.clarin.eu/cmdi

3.1 Metadata Describing Whole Corpora

To describe a complete corpus, the set relies on the elements provided by the Dublin Core Metadata Element Set (DCES)[4] and the OLAC Metadata Set[5] to provide information in an as-standardized-as-possible way and two additional, self-explanatory elements for HZSK corpora (*keywords* and *shortDescription*).

3.2 Metadata Elements Describing Objects in the Corpus

To describe speakers and communications with metadata, the HZSK core metadata set is based on a relatively small vocabulary that is obligatory for all corpora published by the HZSK and a small number of optional attributes.

Speakers, for example, need to have descriptions about sex, birth date and place as well as their function in the corpus. Communications, for example, require elements that describe the location and time of the event, the related recordings and transcriptions and so on.

The complete, constantly updated list of the HZSK core metadata vocabulary can be examined at http://goo.gl/mxOTV

4. CMDI Implementation Background

Before moving on to the CMDI profiles and components in the following section, we will discuss our aims in using CMDI. The main goal in generating and offering CMDI metadata is to make our resources discoverable to persons without access, and to enable them to decide whether to request access to a particular corpus. The metadata encoded in Coma on the other hand is very detailed and can be correlated with search results using the EXAKT tool once access is granted to a particular corpus. We have metadata in Coma files that is not only irrelevant for the discovery, but also too rich to give away without the data owner's consent, as it would constitute a useful resource on its own.

Since we do not intend to use CMDI as a primary metadata format, there is no obvious reason why we should transform all Coma metadata into CMDI. One reason to still do so, would be the development of the Data Category Registry ISOcat. One could argue that it falls within our responsibility to cover all metadata used in our corpora and extend the DCR accordingly where necessary. In our opinion there are however strong arguments against this: The kind of information encoded by the projects is highly theory-dependent and we might well have two corpora with different definitions of the concept "mother tongue" or "L1". Whereas some Data Categories are still useful even though their definition is rather vague (as in ISOcat DC 2955: "Specifies whether the language is a speakers mother tongue.") because they are commonly used with a more or less shared understanding, we do not believe that this applies for all categories. It also seems advisable to keep the size of ISOcat manageable before it contains every conceivable category, most of them only used in only one corpus, and the effort to relate these categories simply is not feasible anymore.

Another important aspect is the fact that we are not able to gather new metadata for completed corpus projects only to meet the requirements of CMDI. We can only provide information if one of the first two of the following three cases applies:

- the information is explicitly available or can be generated from explicitly available metadata (e.g. speaker age from his/her birth year and the time of the communication).
- the information is not explicitly available, but can be made available automatically (e.g. file size, mime type etc.)
- the information can only be made available through human interpretation, for example by reading the transcription and/or listening to the recordings.

As a result of these initial considerations, we decided to only consider metadata that is available or can be made available automatically and that is relevant for the discovery of our resources.

5. CMDI Implementation

When transforming existing metadata into the CMDI format, the first question is which profile and components to use. Our main criterion is to express our entire HZSK core set in CMDI. According to Broeder et al. (2010:45), the Component Registry will contain recommended components created by CLARIN, but users will also be able to create their own profiles and components in the Component Editor, as long as all contained elements refer to ISOcat or other trusted registries. In Broeder, Van Uytvanck & Wittenburg (2010:10), it is also mentioned that these components will be "based on decomposition of existing metadata sets as OLAC, IMDI and DC". Since we use mainly DC and OLAC for the corpus metadata, we created a new component containing existing DC and OLAC components and four new (not yet published) components. These were two components for the *created* and *rightsHolder* DC categories with Conceptlinks referring directly to the schema and two HZSK specific components, *hzsk:shortDescription* and *hzsk:keywords*, with Conceptlinks referring to ISOcat Data Categories 2520 and 278, respectively. The DC 278 (*keyword*) is labeled *private* and not checked, but at least mentions an ISO standard in the description. For the *hzsk:keywords* component, we would also have needed an open vocabulary that provides the keywords already in use while allowing write-ins, as in the *Genre* element in the *cmdi-content* component. However, the current version of the editor does not seem to have such an option.

The Coma metadata model, on the other hand, is not a widely known standard, which is why the transformation into CMDI became more of a challenge for the metadata describing objects in the corpus. We decided to explore various approaches, ranging from reuse of existing components without changes to the creation of entirely new components.

[4] http://dublincore.org/documents/dces/
[5] http://www.language-archives.org/OLAC/olacms.html

5.1 Reusing IMDI Components

Since IMDI was mentioned in Broeder, Van Uytvanck & Wittenburg (2010:10) and previously known to us, we started out using the existing IMDI components without any modifications or extensions. Obviously, IMDI contains much more information than our HZSK core metadata set. Some basic elements, such as the name, age and languages of speakers or the time and place of the communication they participated in, were already explicitly available from our Coma metadata and could be transformed directly with XSLT. Other, rather technical information, such as the size, and mime type of files, could be made available automatically through a simple java program built around the main XSLT transformation. Many of the additional elements that would require human interpretation and investigation are optional, for example the information on *Communication Context*, which means we simply do not include them. IMDI does however also contain some non-optional elements of this kind, for example the *Family Social Role* or *Ethnic Group* of speakers, or the *Quality* (ranging from 1–5) and *Recording Conditions* of recordings, that has to be set to *Unknown* or *Unspecified*. Since we decided not to gather any new metadata of this kind for CMDI, leaving out optional elements or setting obligatory elements to *Unknown* or *Unspecified* was a rather common solution, resulting in quite many non-useful metadata elements. We also encountered a greater number of errors and mismatches with the existing IMDI XML schema[6] in the IMDI components, which do point to the fact that these components are not really in use by anyone. This made us decide against the existing IMDI profile.

5.2 Creating Own Components

The idea of component metadata seems to get lost if everyone uses one and the same profile but leaves every other field blank. According to Broeder et al. (2010:45) the profile should provide "a blueprint for the personalized metadata schema". We therefore aimed to mainly just implement our HZSK core metadata set as a tailor-made profile. Since CMDI is XML and the profile itself an XML Schema, we could also validate the CMDI export, making sure the HZSK core metadata set is complete and syntactically correct for all corpora.

In a first version for the metadata describing objects in the corpus, we ended up with six new components. The *HZSKCommunication* component, equivalent to the IMDI session, contains components for associated speakers, recordings, transcriptions and attached files. The *HZSKRecording*, *HZSKTranscription* and *HZSKAsocFile* components all contain the *HZSKFile* component. Apart from sub-components, all components contain the respective elements of the HZSK core metadata set.

The main issue with this approach is the question of how to point to similarities or identities between our profile, components and elements and existing ones used in other CLARIN centers. Our main goal should be to arrive at a shared set of basic metadata common to all centers handling similar resources. To arrive at this goal, we would also need to consider the development in other centers, in particular which components and/or DCs are commonly used. It would seem that using Conceptlinks referring to such common DCs could then solve the problem, but there are two remaining issues: Firstly, we would need to know if and how the federated search will consider ISOcat DCs. Secondly, since the semantics of the DCs interact with component structure, we would need some equivalent of simple Conceptlinks to be used systematically for components, through which relations such as equivalence between components could be established. The next section discusses the problems we encountered more in detail.

[6] http://www.mpi.nl/IMDI/schemas/xsd/IMDI_3.0.xsd

6. Problems

6.1 A Plethora of Existing Components and Concepts

Both the CMDI Component Registry and the ISOcat DCR contain plenty of material. It is however difficult to decide on the quality of the various components or categories.

An example to illustrate this: For each Coma corpus, a unique speaker distinction element—usually the element *Sigle*—has to be selected. All speakers, communications, recordings, transcriptions and files are also assigned internal guaranteed unique IDs (GUIDs) not visible to the user. The identity of speakers is highly relevant for the design of our corpora, with bilingual speakers being interviewed in both their languages or children being recorded every two weeks in longitudinal studies. This information is therefore also highly relevant to the potential user of a corpus. In CMDI, there is no built-in option to handle IDs. In the ISOcat DCR, we find DC 2552 (*participant code*), a "[s]hort unique code to identify the person participating in the content of the resource" originating from the *Code* element of IMDI actors. When our *Sigle* acts as unique speaker distinction, this fits perfectly, but since Coma allows other elements to be used as speaker distinction, we should include the GUID too. We could perhaps use the DC 2552 here too, since it is supposed to be unique, but there are also the DCs 1845 (*id*), 3894 (*identifier*) and even 3597 (*speaker ID*). In this case however, we do not only want to consider which definition suits our element best, but also which DC is actually widely used and accepted. This information is however missing.

6.2 Collaborative Component Development

A solution to the first problem would be to discuss the development of profile at the various centers and perhaps to collaboratively develop some basic components that can be reused by several centers. However, since everyone is only allowed to develop their own components in their own private workspace, it is not easy for others to efficiently take part in this development. Making a component visible through publishing in the public space prevents further development, and since it is not possible to define a new component as a version of an already published one, developers are more or less forced to keep their metadata profiles private throughout the entire development process.

6.3 Reusing and Extending Components

Related to the question of different versions of components is the question of how to extend or adapt existing profiles without losing the relation to the source. Even if it would become feasible to create common

source profiles and components, these would still need adaption to meet the requirements of all various centers or "depending upon a particular usage scenario" (Broeder, 2010:45). Ideally, when adapting existing profiles instead of defining entirely new ones, the common elements and components would be recognized as identical. A small initial set of recommended source components could then have been used as the source for new components, keeping the relation to the particular source component explicit in all edited versions. "Save as new" would then imply agreeing on using all elements and components as intended by the source, or deleting them from the new component. Possibly, this would facilitate the federated search, since the equivalence of, for example, components describing speakers, would not have to be established by reading the documentation of all components. The most commonly reused elements and components from the source components could then also be automatically extracted.

This would however require Conceptlinks for components, perhaps using the *collection* type in ISOcat, since many of the DCs in ISOcat can be used within different contexts in a profile with different meanings. The component structure then defines the exact meaning of the DC in use. For example, a language or a location component can be contained within a speaker or the recording component of a communication. The elements in the reused component would of course have the same Conceptlinks, but their meaning would slightly differ. With Conceptlinks for components, it would be possible to automatically compare the context from within which DCs are referred to. If DCs were not allowed to behave this way, but rather had to be defined exactly and then related to one another, the component structure would not carry meaning, but the components would also not be as reusable as it is now the case. It seems that the question of how the exact meaning of metadata elements is constituted needs to be answered if CMDI users are to agree on common metadata.

7. Conclusion/Outlook

Creating a CMDI metadata profile from existing metadata in another format poses different challenges depending on the desired application for the resulting metadata. In the case of the scenario described in this paper, harmonizing the existing metadata vocabulary, reducing its size and creating a custom-fit profile using standardized concepts turned out to be the most fruitful approach. The process of creating the metadata profile itself still leaves some things to be desired: Sharing components between projects dealing with similar data would be especially desirable, but is particularly cumbersome, as is identifying "recommended" components and concepts.

With the (technical) evolution of the CLARIN infrastructure like enhancements to the Component Registry as well as ISOcat, some of the problems encountered will possibly disappear. Especially the introduction of a federated CMDI metadata search will show whether the efforts yield the desired results.

7. References

Broeder, Daan; Kemps-Snijders, Marc; Van Uytvanck, Dieter; Windhouwer, Menzo; Withers, Peter; Wittenburg, Peter; Zinn, Claus (2010). A Data Category Registry- and Component-based Metadata Framework. *Proceedings of the Seventh conference on International Language Resources and Evaluation (LREC10).*Malta, May 19–21, 2010.

Broeder, Daan; Van Uytvanck, Dieter; Wittenburg, Peter (2010).Language Resource and Technology Registry Infrastructure. *Deliverable: D2R-5b, 2010-12-17, Version: 1*.

Wörner, Kai (2012).Finding the balance between strict defaults and total openness: Collecting and managing metadata for spoken language corpora with the EXMARaLDA Corpus Manager. In: Schmidt, Thomas &Wörner, Kai: *Multilingual corpora and multilingual corpus analysis.* Hamburg Studies in Multilingualism (HSM), Amsterdam 2012.

A CMD Core Model for CLARIN Web Services

Menzo Windhouwer, Daan Broeder, Dieter van Uytvanck
Max Planck Institute for Psycholinguistics
Wundtlaan 1, 6525 XD Nijmegen, The Netherlands
Menzo.Windhouwer@mpi.nl, Daan.Broeder@mpi.nl, Dieter.vanUytvanck@mpi.nl

Abstract

In the CLARIN infrastructure various national projects have started initiatives to allow users of the infrastructure to create chains or workflows of web services. The Component Metadata (CMD) core model for web services described in this paper tries to align the metadata descriptions of these various initiatives. This should allow chaining/workflow engines to find matching and invoke services. The paper describes the landscape of web services architectures and the state of the national initiatives. Based on this a CMD core model for CLARIN is proposed, which, within some limits, can be adapted to the specific needs of an initiative by the standard facilities of CMD. The paper closes with the current state and usage of the model and a look into the future.

1. Introduction

In the grand CLARIN[1] (Váradi et al., 2008) vision "the user will have access to repositories of data with standardized descriptions, processing tools ready to operate on standardized data, and all of this will be available on the internet using a service oriented architecture" (CLARIN community, 2008). These processing tools can be dedicated desktop tools but also services hosted by various CLARIN (computing) centers and accessible over the web.

In the preparatory phase CLARIN national projects contributed existing or new initiatives in the domain of web services. In Spain UPF provides various families of services, e.g., statistical and CQP web services (see §4.1). A major result of the German D-SPIN project has been the first version of WebLicht, a chaining engine for linguistic web services (see §4.2). The Dutch and Flemish TTNWW project aims at supporting web service workflows for both textual and multimedia resources (see §4.3).

This means that there is not a single CLARIN web service chaining/workflow engine. However, in the CLARIN infrastructure, which aims at unification instead of fragmentation, it should at least be technically possible for all engines to find matching and invoke all known services within the infrastructure.

One pillar of CLARIN is that all metadata on resources, including web services, are to be specified using the Component MetaData Infrastructure (CMDI). This framework is very flexible and should allow mixing common and engine specific metadata for web services. This paper describes the design and use of an extensible CMD model for common web service metadata.

Sections 2 and 3 give an introduction of the major web service architectures and their impact on metadata descriptions, and the CMDI framework. The next section described how web services are described in various national CLARIN projects. On this basis the CMD core model for CLARIN Web Services and its possible usage will be fleshed out in section 5 and 6. The last section will deal with the current state and usage of the model.

[1] Acronyms can be looked up in §9

2. Web service architectures

In the history of the Internet several ways have been proposed to implement Service Oriented Architectures (SOAs) based on the basic protocol for the World Wide Web HTTP. According to (Richardson et al., 2007) three basic web service architectures can be identified. This classification is based on the differences in how the architectures handle two basic information items:

1. Method information: how does the client convey its intentions to the server, i.e., why should the server do *this* instead of doing *that*?
2. Scoping information: how does the client tell the server which part of the data set to operate on, i.e., why should the server operate on *this* data instead of *that* data?

In the CLARIN landscape all three architectures can be encountered.

2.1 RESTful resource-oriented architectures

A web service architecture is considered RESTful if the method information goes into the verb that determines the nature of the HTTP request, e.g., PUT, GET, POST or DELETE, and resource oriented if the scoping information goes into the URI. Resource orientation means also that this URI does not actually refer to a service but to a resource, where resolving the URI results in a representation of that resource. These architectures are directly build upon the technical foundations that made the World Wide Web successful (Fielding, 2000).

A well-known example of a RESTful resource-oriented architecture is Amazon's Simple Storage Service (Amazon Web Services LLC, 2006). Also services that are exposed by the Atom Publishing protocol (Gregorio et al., 2007) are examples.

2.2 RPC-style architectures

In RPC (Remote Procedure Call) architectures envelopes full of data are sent and received from the services. Both the method and scoping information are kept inside the envelope. The XML-RPC protocol (Winer, 2003) is a prime example of such architecture. It ignores most features of HTTP, i.e., only one URI (the service endpoint) is used and one HTTP method (POST).

Contrary to RESTful architectures this disables a lot of the basic infrastructure, e.g., caching of GET requests, which made the World Wide Web scalable and successful. The same can be said about most usages of SOAP (Simple Object Access Protocol) (W3C XML Protocol Working Group, 2007) on top of HTTP. In this case SOAP is the envelope format in which the method and scoping information is provided.

2.3 REST-RPC hybrid architectures

This group of service architectures have REST-like elements, e.g., they put the scoping information in the URI, but they do that as well for the method information, e.g., have a single endpoint with a query parameter that specifies the service to call. An example of a REST-RPC hybrid is the Flickr REST API (Flickr, 2012).

2.4 Interface Description Language

An Interface Description Language (IDL) is commonly used by RPC architectures to specify the services which are available at an endpoint. In the case of SOAP the IDL is the Web Service Definition Language (WSDL) (Christensen et al., 2001). The WSDL provides information on the input and output of the services.

For RESTful resource-oriented architectures there has been an on-going debate if an IDL is needed. Patterns are proposed which enable the transition of one service, or resource representation, to another, e.g., Hypermedia as the Engine of Application State (HATEOAS) (Fielding, 2000; Fielding, 2008) where a client basically follows the links between resources just like a browser a does with the links embedded in a HTML page. However, in current practice this style of web services is too free form to automatically determine how to call a service. So relying only on a text document to define the API is naïve and does not scale. For example, parameters can be passed on in many ways, e.g., embedded in the URI path, as query parameters or as part of a multipart POST request. The Web Application Description Language (WADL) (Hadley, 2009) has been submitted to W3C as a possible IDL to describe RESTful web services. But WADL did not make it into a W3C recommendation yet and from time to time competing IDLs are proposed, e.g., ReLL (Alarcón et al., 2010) and the RDF-based RESTdesc (Verborgh, 2012). Also version 2 of WSDL allows describing this RESTful web services. IDLs suitable for RESTful web services can in general also be used for REST-RPC architectures.

3. The Component Metadata Infrastructure

This section introduces CMDI, the metadata infrastructure that is to be used for all metadata describing resources in the CLARIN domain, including web services. The role of and link between descriptions of a web service in an IDL and in CMDI will be described later on in this paper.

In the CLARIN infrastructure CMDI (Broeder et al., 2011) has been developed to be able to better tailor a metadata schema to the needs of a (type of) resource. Previous attempts resulted in either too few metadata elements, e.g., Dublin Core, or in too many, e.g., IMDI. Both cases can result in poor metadata quality as users misuse elements when there are too few or give up when there are too many.

CMDI is based on a registry of reusable components (CLARIN community, 2012). Users can combine suitable components into profiles. These profiles can be transformed into an XML Schema so actual instances of the profiles can be validated. When needed users can create new component and profiles, but they can also copy existing components and adapt them till they suit their specific needs. However, CLARIN will benefit if proliferation of components is kept to the minimum.

Components, elements and values in CMD can be linked to concepts or data categories defined in an external registry. In CLARIN the preferred registry is the ISOcat (Max Planck Institute for Psycholinguistics, 2012) Data Category Registry (DCR), which is an implementation of (ISO 12620, 2009) and as the ISO TC 37 DCR dedicated to the linguistic domain. These links allow establishing semantic interoperability between components, elements or values in different CMD profiles. And even allows for differences in the use of terminology, cases or orthography.

4. CLARIN web service chaining and workflow engines and registries

As stated before various national CLARIN projects have started initiatives in the area of Web Services. In this section these initiatives are sketched with a focus on their support for metadata description of the services.

4.1 Spain

In Spain IULA at UPF provides access to various families of web services (see §4.2.6 in (Funk et al., 2010) and (CLARIN-CAT and -ES community, 2012)):

- Format conversion services: provide different format conversion tools such as PDF, MS Word and HTML to plain text, character conversion tools, etc.;
- Statistical services: provide statistical information on an uploaded corpus, e.g., the "Herdan" index of lexical richness or all the n-grams with its number of occurrences;
- Annotation services: including morphosytactic, syntactic and dependency annotators;
- Corpus management services: deploys a CWB as a web service and allows indexing and further exploitation of an annotated corpus.

Access to the services is provided via SOAP, so the technical, also known as the syntactic, description is given in WSDL. Additional metadata and semantics are provided in a separate semantic description, inspired by the SoapLab2 semantic annotations and the myGrid ontology (Villegas et al., 2010). A CMDI profile[2] has been created for these semantic descriptions. The following fragment[3] is taken from the XSLT processor service description:

[2] See http://catalog.clarin.eu/ds/ComponentRegistry?item=clarin.eu:cr1:p_1295178776924

[3] Due to limited space the XML has been trimmed by abbreviating all end tags to </> and to leave out some content (indicated by ellipses '…').

```
<serviceDescription>
 <serviceName>xsltprocService</>
 …
 <locationURL>…/soaplab2-axis/</>
 <interfaceWSDL>…xsltproc?wsdl</>
 …
 <operations>
  <serviceOperation>
  <operationName>runAndWaitFor</>
  <portName>xsltproc</>
  …
  <operationInputs>
   <MyGridParameter>
    <parameterName>stylesheet</>
    …
    <isConfigurationParameter>false</>
    <semanticType>stylesheet</>
    …
    <XMLSchemaURI>…xsltproc?xsd=1</>
    …
    <formats>
     <formatIdentifier>text/xml</>
     <formatIdentifier>UTF-8</>
   </></>
   …
</></></></>
```

Figure 1: Fragment of an UPF service description

This Spanish initiative is continued in the PANACEA project, a STREP project under EU-FP7 (Bel, 2010). The ELDA PANACEA web service registry (ELDA, 2012) provides the latest usage statistics.

4.2 Germany

The German D-SPIN project created the WebLicht chaining engine for web services (see §1 in (Ogrodniczuk et al., 2011)). Services in WebLicht are REST-based and in current practice a single TCF document is pushed through a pipeline of services, where each service adds a new layer to the TCF document. Around a hundred services, e.g., tokenizers and part-of-speech taggers, for various languages are accessible via WebLicht.
For the syntactic description of services there is no usage of an IDL as the invocation recipe for a service accessible by WebLicht is well known by the chaining engine, i.e., POST the TCF document. The metadata description of services focuses mainly on specifying the required input layers and produced output layers. This description supports profile matching to build a chain. The following fragment illustrates this:

```
<service>
 <name>TreeTagger 117 152</>
 <url>…/tree-tagger3.perl</>
 …
 <replacesinput>false</replacesinput>
 <input type="text/tcf+xml">
  <feature name="lang">
   <value name="de"/>
   <value name="it"/>
   <value name="en"/>
  </>
  <feature name="version">
   <value name="0.3"/>
  </>
  <feature name="layer.tokens"/>
 </>
 <output type="text/tcf+xml">
  <feature name="layer.postags"/>
  <feature name="layer.lemmas"/>
  <feature name="layer.postags.tagset">
   <value refValue="it" refFeature="lang"
     name="stein"/>
   <value refValue="en" refFeature="lang"
     name="penntb"/>
   <value refValue="de" refFeature="lang"
     name="stts"/>
</></></></>
```

Figure 2: Fragment of a WebLicht service description

WebLicht (SfS Tübingen, 2012) can be used by the CLARIN community and development continues in the successor to D-SPIN the CLARIN-D project (CLARIN-D, 2012).

4.3 The Netherlands and Flanders

CLARIN-NL and CLARIN Flanders cooperate in the TTNWW project, which aims at providing access to national services as for example developed in the STEVIN project. Two modalities are being addressed: text and speech. In TTNWW no assumption is made with regard to the web service architecture, i.e., it should be possible to integrate services based on RESTful resource-oriented, RPC-style or REST-RPC hybrid architectures.
Metadata descriptions are based on the data model described in (Kemps-Snijders, 2010). The following example shows a fragment, including a reference to the WSDL via a CMD resource proxy.

```
<CMD>
 <Header>…</>
 <Resources>
  <ResourceProxyList>
   <ResourceProxy>
    <ResourceType>WSDL service</>
    <ResourceRef>…/LangId.asmx</>
   </ResourceProxy>
  </ResourceProxyList>
 </Resources>
 <Components>
  <Service>
   <Type>SOAP</>
   …
   <Name>LangIdWebService</>
   <URL>hdl:service</>
   <Operation>
    <Name>IdentifyLanguage</>
    <Action>…/IdentifyLanguage</>
    <Input>
     <Parameter>
      <Name>IdentifyLanguage.text</>
      …
```

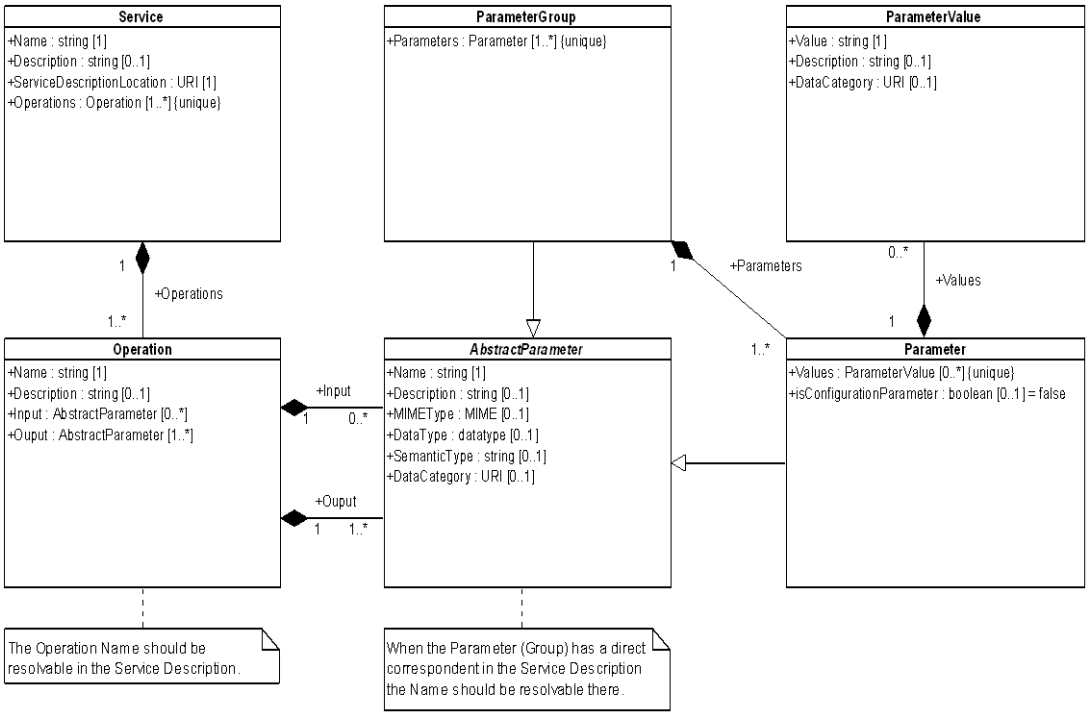


Figure 3: UML model for CLARIN web services

```
      <TechnicalMetadata>
       <MimeType>text/plain</>
       <CharacterEncoding>UTF-8</>
      </TechnicalMetadata>
     </>
     <Parameter>
      <Name>….modern_languages</>
      <DataCategory>…</>
     </Parameter>
     <Parameter>
      <Name>….rare_languages</>
      <DataCategory>…</>
     </Parameter>
    </>
    <Output>
     <Parameter>
      <Name>….Language</>
      <TechnicalMetadata
        parameterRef="IdentifyLanguage.text">
       <MimeType>text/plain</>
       <CharacterEncoding>UTF-8</>
       <PLORK>WAF</PLORK>
       <ContentEncoding>
        <URL>hdl:testSchema</URL>
        <ResourceFormat>PlainTextResource</>
     </></></></>
     <Parameter>
      <Name>….Confidence</>
      <DataCategory>…</>
</></></></></></></></>
```

Figure 4: Fragment of a TTNWW service description

The TTNWW project is on-going and has, at time of writing, not been publically released.

5. A CMD core model for web services

As shown in the previous section the national CLARIN projects support diverse web service architectures including various mechanisms for describing web services on the semantic and syntactic levels. The CMD core model described in this section is an attempt to distil a common core out of these existing descriptions.

5.1 An initial UML model

Discussion on the core model were based on an UML model and after several iterations resulted in the class diagram shown in Figure 3.

In a hierarchical perspective on the diagram, which matches the CMD approach, the *Service* class is taken as the root. A major design decision is that each Service should refer to a service description (see the *Service-DescriptionLocation* attribute), e.g., a reference to a WSDL or WADL instance. Here the core model follows the Spanish approach. The CMD description mainly focuses on semantics and there is an additional syntactic description that provides more technical details. These technical details are needed as the CMD description might be powerful enough to do profile matching, i.e., determine if the output of one service can be used as input to another service, but it does not provide enough information to really invoke these services. This is the penalty for the freedom that REST-style web services allow developers. Take for example an WebLicht service: the WebLicht chaining engine knows its own recipe, i.e., it should POST the TCF document to the URI of a service, but another chaining or workflow engine would not know that. In the syntactical service description for REST-style web services this recipe is made explicit, so any engine can know how to invoke a service.

The core model actually does not state which IDL should be used. For the time being WSDL (2) and WADL seem to

be the most appropriate candidates able to support all the web service architectures described in Section 2.

The *Service* class does not contain any attribute to specify the URI of the service (endpoint) as this is considered technical information, which is provided in the syntactical service description.

The syntactical service description is able to describe a collection of services. In an RPC architecture these are the operations provided by a single endpoint, and also a one WADL document can describe a collection of REST-style web services. A *Service* instance can thus refer to one or more operations.

Each operation is an instance of the *Operation* class which contains the in- and output specifications. As it should be clear how to invoke this operation the name of the operation in the semantic description should be the same as the one used for it in the syntactical description.

Input and output are sets of parameters. As illustrated in the case of the TCF document used by WebLicht, profile matching might actually need to look into the contents of the resource send around in the chain or workflow, i.e., it should be possible to state that a lemmatizer needs an input TCF document containing a token layer. Notice that the syntactical description does not need to specify about layers in the file, it only needs to specify how to ship the TCF document to the service. The UML model deals with this by allowing an in- or output parameter to be either a *ParameterGroup* or a *Parameter*, which are both subclasses of the abstract *AbstractParameter* class. In WebLicht the in- or output TCF document would correspond to a *ParameterGroup* and a layer to a *Parameter* in this group. Both *Parameter* and *ParamaterGroup* share a number of optional attributes that allow providing various levels of profile matching from technical to service specific semantics:

1. *MIMEType*: the technical MIME type of a resource will also reveal its media type, e.g., text/plain;
2. *DataType*: a value domain, in general taken from the well-known XML Schema data types (Biron et al., 2004), e.g., ID;
3. *DataCategory*: a reference to a data category, in general taken from ISOcat, e.g., http://www.isocat.org/datcat/DC-2535 (*/project id/*);
4. *SemanticType*: free form string to indicate service specific types, e.g., 'clam.project.adelheid'.

A profile matching algorithm can use these various levels to prune away semantic mismatches from a list of syntactic matches, e.g., matching an Adelheid (Halteren, 2009) project id with a service that accepts arbitrary plain text would be useless.

The names of parameters or parameter groups, depending on which corresponds to an actual technical parameter, should correspond to names used for the same parameter in the syntactical description.

The lowest level of the hierarchy contains the *ParameterValue* class which is used to capture descriptive information of value enumerations for parameters.

This UML model covers major parts of the various semantic descriptions mentioned in Section 4. The CMD infrastructure will provide the means to add any repository specific information to this common part.

5.2 CMD components for the core model

To be useful in the CLARIN context the UML model has to be instantiated as a set of CMD components. However, CMD does not support any inheritance, i.e., one cannot create an *AbstractParameter* component and describe how *Parameter* and *ParameterGroup* components are related to it, so specific mapping rules between the two models, aimed at maintaining as much of the semantics as possible, have to be followed:

1. Each non-abstract class becomes a component, e.g., *Service* and *Operation* but not *AbstractParameter*;
2. Each attribute, both inherited and local, becomes an element, e.g., *Name* or *Description*, but
3. attributes, both inherited and local, referring to non-abstract classes become components with a child component representing the referred non-abstract class, e.g., *Operations*;
4. Attributes, both inherited and local, referring to abstract classes should become components with optional child components representing all the non-abstract classes lower in the inheritance hierarchy, e.g., *Input* and *Output*;
5. Cardinality constraints are copied where possible, e.g., in the case of the attributes referring to abstract classes these will be lost, e.g., CMD cannot express that an *Output* instance should refer to at least one *Parameter* or *ParameterGroup* instance.

Reusability considerations determine which components related to classes exist on their own in the registry, while others only exist within another component. *ParameterValue*, for example, is only used inside Parameter and is considered unlikely to be reused somewhere else.

The CMD components resulting from this mapping UML model have been created in the Component Registry and combined into a profile[4]. Only in one case the rules described in this section were not followed: the *ServiceDescriptionLocation* attribute was not mapped to a CMD element but to a CMD component. The idea behind this has been to enable the use of a CMD resource proxy for the reference to the syntactic description. This promotes the approach taken in TTNWW as shown in Figure 4.

```
<Resources>
 <ResourceProxyList>
  <ResourceProxy id="h1">
   <ResourceType
     mimetype="application/vnd.sun.wadl+xml">
    Resource
   </>
   <ResourceRef> …/tds-services.wadl</>
 </></>
 …
</Resources>
<Components>
 <ToolService>
  …
```

[4] See http://catalog.clarin.eu/ds/ComponentRegistry?item=clarin.eu:cr1:p_1311927752335&space=public

```
  <Service CoreVersion="1.0">
   <Name> Typological Database System</>
   <ServiceDescriptionLocation ref="h1" />
    <Operations>
     …
</></></></></></>
```

Figure 5 *ServiceDescriptionLocation* uses a resource proxy

6. Usage of the core model

Now that the CMD core model for CLARIN Web Services is available the question arises: how can a web service repository adapt and use it? The core model profile should not be instantiated directly as it functions as a template for profiles specific to the various national initiatives. For CLARIN-NL an extension has been created where a *TechnicalMetaData* component (see also the *TechnicalMetaData* fragments in Figure 4) has been added to the *ParameterGroup* and *Parameter* components. This component contains elements to specify, for example, the character encoding, a reference to an XML schema or the location of an output parameter in a resource. In the following fragment the bold parts of the instance correspond to the core model.

```
<Operation>
 <Name>query</>
 <Description>Query the data section of an IDDF
document.</>
 <Input>
  <Parameter>
   <Name>file</>
   <DataType>string</>
   <SemanticType>iddf.file</>
   <TechnicalMetadata>
    <CharacterEncoding>UTF-8</>
  </></>
  <Parameter>
   <Name>query</>
   <MIMEType>text/xml</>
   <TechnicalMetadata>
    <CharacterEncoding>UTF-8</>
    <ContentEncoding>
     <URL>…/query.rng</>
     <ResourceFormat>IDDF Query XML</>
  </></></></>
  …
 </>
 <Output>
  <ParameterGroup>
   <Name>query-result</>
   <MIMEType>text/xml</>
   <Parameters>
    <Parameter>
     <Name>notion</>
     <DataType>ID</>
     <SemanticType>iddf.notion</>
     <TechnicalMetadata>
      <CharacterEncoding>UTF-8</>
      <ContentEncoding>
       <RelativeLocation>//@iddf:notion</>
     </></></></>
    …
</></></></></>
```

Figure 6 Fragment of a CLARIN-NL service description

The CLARIN-NL tool and services description profile was created by copying the components from the core model and adding the additional components and elements. This need to copy and edit existing components opens up the possibility to also delete components/elements which were mandatory in the core model. Additional components or elements can be freely added but changes to existing components or elements need to follow some rules, so instances are valid both in the core model and the extension:

1. Cardinalities in the extension should be within the boundaries set by the core model, e.g., mandatory elements cannot become optional but optional elements like *Description* can become mandatory;
2. Closed value domains cannot be extended, but open value domains like for *SemanticType* can be turned into closed ones;
3. Data category references in the core model should not be touched as this could imply different semantics.

By following these rules it should be possible to strip of all additional components and elements from an instance and still be left with a valid instance of the core model. Taking the example fragment in Figure 6 only the bold styled elements would be left. This validation process has been implemented and is available to developers at http://www.isocat.org/clarin/ws/cmd-core/. The target audience of the core model consists of developers of web service registries. Web service developers, which want to make their services available to one of the CLARIN chaining/workflow engines, should just use the core model compliant CMD profile of a CLARIN registry.

The fragment in Figure 6 showed part of the semantic description of the TDS IDDF query web service (Dimitriadis, 2009). This fragment has its counterpart in the syntactic, or technical, WSDL description.

```
<method name="POST" id="query">
 <request>
  <representation
    mediaType="multipart/form-data">
   <param name="service" type="xs:string" …
     fixed="query" style="query"
     required="true"/>
   <param name="file" type="xs:string"
     style="query" required="true"/>
   <param name="query" style="query"
     required="true"/>
   …
 </></>
 <response>
  <representation mediaType="text/xml">
   <param name="notion" path="//@iddf:notion"
     repeating="true" style="plain"/>
   …
</></></></>
```

Figure 7 Fragment of a CLARIN-NL WSDL

The TDS IDDF web services use a RPC-REST hybrid architecture, where the method information is passed on in the URI as a query parameter. In Figure 7 this is the first input parameter named *service* with the fixed value 'query', which is the name of the service to be executed

by the RPC endpoint. This parameter does not appear in Figure 6, which shows that this low-level implementation detail is hidden from the semantic description of the web service. Also notice that the names for the operation, i.e., 'query', and the in- and output parameters, e.g., 'file' and 'notion', are the same in the two descriptions, so one can connect the semantic and syntactic information levels.

7. Future work and conclusions

This paper described the development of a CMD core model for CLARIN web service descriptions. At the time of writing only the CLARIN-NL tool and service description profile is compliant with the core model and publically available in the public workspace of the Component Registry. A few Dutch web services have been described, but this profile is not yet in use by the TTNWW project. The German WebLicht project is adopting CMDI and the core model in version 2.0.

It will only be a first step if the various registries use a CMD profile that is compliant with the core model. The next, and most important step to measure uptake, is when the various chaining/workflow engines are able to process both the semantic and syntactic web service descriptions and thus are able to invoke generic services not specifically tailored to their system.

As the construction of the CLARIN infrastructure proceeds more complex use cases are being addressed, also in the area of web services. One of the trends is to incorporate asynchronous web services. In general these are not single services that do an (advanced) operation and return their result 'immediately', but instead various services need to be called in a specific sequence. A common pattern is to call a service to start the operation, then use another service to poll at regular intervals if the operation has finished, and if so to fetch the result by yet another service. In the Netherlands CLAM (Gompel, 2011) is a popular REST-based framework that is based on this pattern. This is in fact a mini workflow and projects like TTNWW are implementing them as such and compose larger workflows out of multiple mini workflows. Users of the infrastructure then call these pre-composed workflows instead of single web services. It remains to be seen if this kind of workflows can be handled in the same way as web services and thus can use the core model, or if another model or adaptions to this model are needed.

Alignment with or reuse of the core model by other (metadata) infrastructure initiatives could enable wider integration. The META-SHARE meta model is also based on components and ISOcat and contains a section on Tools and Services (see §8 in (Desipri et al., 2012)) and would thus be a prime candidate.

8. Acknowledgements

The CMD core model for CLARIN Web Services has profited from feedback from the members of the ISOcat CLARIN Web Services group, especially Marc Kemps-Snijders, Marta Villegas and Thomas Zastrow.

9. Acronyms

API	Application Programming Interface
CLARIN	Common Language Resources and Technology Infrastructure
CMDI	Component Metadata Infrastructure
CWB	Corpus Workbench
DCR	Data Category Registry
D-SPIN	Deutsche Sprachressourcen-Infrastruktur
HATEOAS	Hypertext as the Engine of Application State
HTML	Hypertext Markup Language
HTTP	Hypertext Transfer Protocol
ID	Identifier
IDDF	Integrated Data and Documentation Format
IDL	Interface Description Language
IMDI	ISLE MetaData Initiative
ISO	International Organization for Standardization
IULA	Institut Universitari de Lingüística Aplicada
META	Multilingual Europe Technology Alliance
MIME	Multipurpose Internet Mail Extensions
MS	Microsoft
PANACEA	Platform for Automatic, Normalized Annotation and Cost-Effective Acquisition
PDF	Portable Document Format
RDF	Resource Description Format
REST	Representational State Transfer
RPC	Remote Procedure Call
SOA	Service Oriented Architecture
SOAP	Simple Object Access Protocol
STEVIN	Spraak- en Taaltechnologische Essentiële Voorzieningen In het Nederlands
TC	Technical Committee
TCF	Text Corpus Format
TDS	Typological Database System
TTNWW	TST Tools voor het Nederlands als Webservices in een Workflow
UML	Unified Modeling Language
UPF	Universitat Pompeu Fabra
URI	Uniform Resource Identifier
W3C	World Wide Web Consortium
WADL	Web Application Description Language
WSDL	Web Service Description Language
XML	Extensible Markup Language
XSLT	Extensible Stylesheet Language Transformations

10. References

Alarcón, R. and E. Wilde (2010). RESTler: Crawling RESTful Services. WWW 2010. Raleigh, North Carolina, USA, ACM.

Amazon Web Services LLC. (2006). Amazon Simple Storage Service API Reference. Retrieved 16 February 2012, from http://docs.amazonwebservices.com/AmazonS3/latest/API/APIRest.html.

Bel, N. (2010). Platform for Automatic, Normalized Annotation and Cost-Effective Acquisition of Language Resources for Human Language Technologies: PANACEA. XXVI Congreso de la Sociedad Española para el Procesamiento del Lenguaje Natural (SEPLN-2010). Valencia, Spain.

Biron, P. V., K. Permanente and A. Malhotra. (2004).

XML Schema Part 2: Datatypes Second Edition. W3C recommendation Retrieved 16 February 2012, from http://www.w3.org/TR/xmlschema-2/.

Broeder, D., O. Schonefeld, T. Trippel, D. v. Uytvanck and A. Witt (2011). A pragmatic approach to XML interoperability – the Component Metadata Infrastructure (CMDI). Balisage: The Markup Conference 2011. Montréal, Canada.

Christensen, E., F. Curbera, G. Meredith and S. Weerawarana. (2001). Web Services Description Language. Retrieved 16 February 2012, from http://www.w3.org/TR/wsdl.

CLARIN-CAT and -ES community. (2012). Clarin-Cat-Lab and Clarin-Es-Lab Retrieved 27 Feburary 2012, from http://clarin-cat-lab.org/ and http://clarin-es-lab.org/.

CLARIN-D. (2012). CLARIN-D: a web and centres-based research infrastructure for the social sciences and humanities. Retrieved 16 February 2012, from http://www.clarin-d.de/.

CLARIN community. (2008). About CLARIN » Mission. Retrieved 16 February 2012, from http://www.clarin.eu/external/index.php?page=about-clarin&sub=0.

CLARIN community. (2012). Clarin Component Browser. Retrieved 16 February 2012, from http://catalog.clarin.eu/ds/ComponentRegistry/.

Desipri, E., M. Gavrilidou, P. Labropoulou, S. Piperidis, F. Frontini, M. Monachini, V. Arranz, V. Mapelli, G. Francopoulo and T. Declerck (2012). Documentation and User Manual of the META-SHARE Metadata Model P. Labropoulou and E. Desipri.

Dimitriadis, A. (2009). TDS Curator - A web-services architecture to curate the Typological Database System. CLARIN Call I prohect Retrieved 17 February 2012, from http://www.clarin.nl/node/70#TDS_Curator.

ELDA. (2012). The PANACEA registry. Retrieved 28 March 2012, from http://registry.elda.org/.

Fielding, R. (2000). Architectural Styles and the Design of Network-based Software Architectures. Irvine, University of California.

Fielding, R. T. (2008). REST APIs must be hypertext-driven. Retrieved 16 February 2012, from http://roy.gbiv.com/untangled/2008/rest-apis-must-be-hypertext-driven.

Flickr. (2012). Flickr Services. Retrieved 16 February 2012, from http://www.flickr.com/services/api/request.rest.html.

Funk, A., N. Bel, S. Bel, M. Büchler, D. Cristea, F. Fritzinger, E. Hinrichs, Marie Hinrichs, R. Ion, M. Kemps-Snijders, Y. Panchenko, H. Schmid, P. Wittenburg, U. Quasthoff and T. Zastrow (2010). Requirements Specification Web Services and Workflow Systems. CLARIN deliverable. D2-R6b.

Gompel, M. v. (2011). CLAM: Computational Linguistics Application Mediator. Retrieved 17 February 2012, from http://ilk.uvt.nl/clam/.

Gregorio, J. and B. d. hOra (2007). The Atom Publishing Protocol IETF - Network Working Group. RFC 5023.

Hadley, M. (2009). Web Application Description Language. W3C submission, W3C.

Halteren, H. v. (2009). Adelheid - A Distributed Lemmatizer for Historical Dutch. CLARIN Call 1 project Retrieved 17 February 2012, from http://www.clarin.nl/node/70#Adelheid.

ISO 12620 (2009). Terminology and other language and content resources - Specification of data categories and management of a Data Category Registry for language resources, International Organization for Standardization.

Kemps-Snijders, M. (2010). Web services and workflow creation. CLARIN deliverable. D2R-7b.

Max Planck Institute for Psycholinguistics. (2012). ISOcat - Data Category Registry. Retrieved 16 February 2012, from http://www.isocat.org/.

Ogrodniczuk, M. and A. Przepiórkowski (2011). Integration of Language Resources into Web service infrastructure. CLARIN deliverable. D5R-3b.

Richardson, L. and S. Ruby (2007). RESTful Web Services, O'Reilly.

SfS Tübingen. (2012). WebLicht Web-based Linguistic Chaining Tool. Retrieved 28 March 2012, from https://weblicht.sfs.uni-tuebingen.de/.

Váradi, T., S. Krauwer, P. Wittenburg, M. Wynne and K. Koskenniemi (2008). CLARIN: Common Language Resources and Technology Infrastructure. Sixth International Conference on Language Resources and Evaluation (LREC'08). N. Calzolari, K. Choukri, B. Maegaardet al. Marrakech, Morocco, European Language Resources Association (ELRA).

Verborgh, R. (2012). RESTdesc – Semantic descriptions for RESTful Web APIs. Retrieved 16 February 2012, from http://restdesc.org/.

Villegas, M., N. Bel, S. Bel and V. Rodríguez (2010). A Case Study on Interoperability for Language Resources and Applications. The Seventh International Conference on Language Resources and Evaluation (LREC'10). N. Calzolari, K. Choukri, B. Maegaardet al. Valletta, Malta, European Language Resources Association (ELRA): 3512-3519.

W3C XML Protocol Working Group. (2007). SOAP Specifications. W3C recommendation Retrieved 16 February 2012, from http://www.w3.org/TR/soap/.

Winer, D. (2003). XML-RPC Specification. Retrieved 16 February 2012, from http://xmlrpc.scripting.com/spec.

Towards an ontology of categories for multimodal annotation

Peter Menke and **Philipp Cimiano**

Project X1 "Multimodal Alignment Corpora"
Collaborative Research Centre 673 "Alignment in Communication", Bielefeld University
Peter.Menke@uni-bielefeld.de cimiano@cit-ec.uni-bielefeld.de

Abstract

We examine how multimodal data collections, resulting mainly from (psycho)linguistic experiments, can be expressed in standardized metadata description formats. We summarize how such data collections differ structurally from the traditional concept of corpora, and we list thoughts, problems, and solutions that occurred when we designed and collected ISOcat data categories and CMDI components for the metadata representation of these data collections. As a result we present plans for an ontology of modalities and related concepts and data units, which we consider a more appropriate enviroment for the kind of multimodal data we are dealing with.

1. Introduction

In this paper we originally intended to introduce a software module that automatically generates CMDI[1] metadata instances based on ISOcat data categories[2] for multimodal corpora stored in a corpus management system. During our work, however, we encountered several problems and questions. A majority of these could be traced back to the fact that our data – complex multimodal annotation sets based on mostly (psycho)linguistic experiments – differs from 'traditional' corpus data in many aspects. This had consequences for the selection and the design of data categories as well as metadata components. In the course of this project, those problems and questions became relevant enough for us to decide to rework this paper and to present a summary and discussion dedicated to these problems. We believe that this summary can be useful for other researchers, especially from the area of multimodal communication and complex multimodal annotations.
As a result we present thoughts and plans for the creation of an ontology for modalities and their relations, data structures and data types. This ontology is expected to be a supplement especially for nonlinguistic categories necessary for multimodal research.

1.1. Objects of study

The data structures and collections analyzed in this paper are mainly products of the Collaborative Research Centre (CRC) 673 "Alignment in Communication", located at University of Bielefeld, Germany.
The CRC consists of a group of 13 research projects investigating the communicative concept or phenomenon of *alignment* (Pickering and Garrod, 2004).
One of the goals of the CRC is to observe and analyse this phenomenon in modalities other than speech, and in actual multimodal communication. It also examines communication with non-human interlocutors (avatars or robots). Therefore, current research covers the areas of linguistics (with psycholinguistics and psychology) and computer science (involving artificial intelligence and robotics), thus opening a wide field of theories, methods, software tools and data representation formats used in scientific processes.
This paper focuses on project X1 "Multimodal Alignment Corpora". It provides central services for uniform collection and representation of these heterogeneous data sets, based on

- a generic data model suitable for the representation of multimodal corpus data, and
- a flexible implementation with focus on extensibility for easy integration of additional data representation formats (this also involves export routines to metadata descriptions like CMDI).

The main application being created by X1 is *Ariadne*, a corpus management system. Its overall concept is described in (Menke and Mehler, 2010).

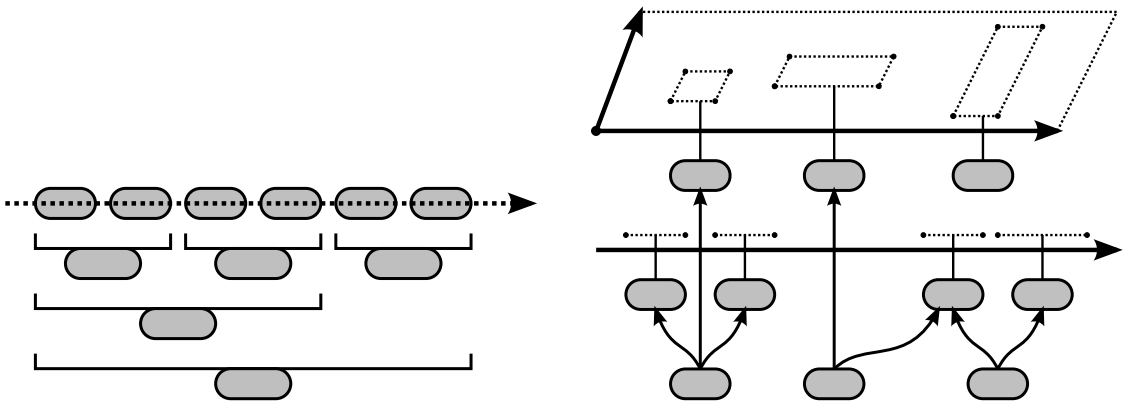
1.2. One of our philosophies: Continual metadata collection

Within our data models and our tools we put a strong focus on automation of recurring tasks. The goal for our tools is to accompany researchers through the process of data generation, from planning to experimenting and creation of transcriptions and annotations. A more detailed explanation of this approach can be found in (Menke and Mehler, 2011).
Starting with the planning phase of experiments, users are encouraged to document their resources in our system by attaching additional information to their data records.
As a result, (meta-)data structures grow and mature during the course of a study. In the end corpora are already equipped with many pieces of information that can then be queried and harvested in order to create data representations in different formats, e.g., metadata descriptions (in CMDI or other formats) suitable for certain harvesters. For the actual metadata generation, only an export routine would be necessary.[3]

[1] CMDI is an abbreviation for "Component MetaData Infrastructure", see `http://www.clarin.eu/cmdi`

[2] ISOcat is maintained at `http://www.isocat.org`; see also (Kemps-Snijders et al., 2008), (Kemps-Snijders et al., 2009).

[3] For CMDI, this routine would hardly require any sophisticated operations because the XML serialization of corpus data already contained in the system generates XML structures that are very similar (in several parts of the document, equal) to the CMDI counterpart.



(a) Schema of traditional corpus models

(b) Schema of our multimodal corpus model

Figure 1: Schema of our approach to modeling multimodal corpora compared to traditional ones. Instead of a single, flat sequence of primary objects (the top row elements connected with the dotted arrow in 1a) we use multiple scales that refer to temporal and spatial segments of reality under observation. Primary objects (dotted) are marks of points, intervals, rectangles, cuboids, and similar regions in the space spanned by those scales. Secondary objects (gray rounded boxes) refer to these.

As a consequence, users only have to *complete* the descriptions and category assignments, but they do not have to *produce them from scratch* at the end of an experiment phase. Therefore, normally they do not use the usual metadata editors (e.g., Arbil[4]), but they use editors and forms provided by the *Ariadne* application instead.
While that last step (the actual generation of metadata descriptions) does not require much effort neither by programmers nor users, problems did occur at earlier stages. Some of the problematic areas will be described and analyzed in the following three sections:

- The common misconception that multimodal corpora are structurally similar to classical corpora, and, thus, can be represented with the same mechanisms, types and categories (section 2.).
- The question whether multimodal corpora, being composed of linguistic as well as nonlinguistic elements, should still be modeled using ISOcat, a data category that is an implementation of an ISO standard explicitly made for linguistics and language resources (subsection 3.1.).
- The question whether emergent data categories still under development and revision should also be published to such data category registries which normally accept only entries that researchers have settled and agreed upon (subsection 3.2.).

2. The relation between the concepts "multimodal corpus" and "corpus"

Intuitively, one assumes that multimodal corpora are special kinds of corpora.
To taxonomists and semanticists the concept "multimodal corpus" is a hyponym to "corpus", and therefore inherits meaning from that concept. To programmers with a background in object-oriented modeling a "multimodal corpus" is a subclass of "corpus". Thus, it inherits all properties of that superclass, and it can optionally define its own properties in addition. Regularly, traditional corpora built around texts or utterances are seen as the general and singular type of corpora, and the (problematic) consequence is to identify those concepts with their superclass "general corpora". People then conclude that multimodal corpora can also be modeled and described in the same way and with the same vocabulary and structure inventory as traditional corpora. As a matter of fact, this holds for many aspects. However, there are caveats and differences that become apparent only gradually. In this section we want to look at the details where the respective types of corpora differ.

2.1. "Primary data" in traditional text corpora

In the past, linguistic corpora used to contain either texts or isolated monologic utterances as their primary data sets (cf. Pei 1966, Bußmann 1996). Due to their linear nature these could be modeled as flat sequences of discrete atomic elements (depending on the theory or model these were characters, tokens, or words). Annotations of these primary texts or utterances were considered secondary data, and references to the primary sequence could be made by giving indices of simple positions or spans. A schematic diagram of this approach is given in Figure 1a.
One of the first representation techniques of this traditional model were transcripts in different formats. In several of these formats, elements from other modalities can be added to some extent, but they always have to be aligned to the primary token stream. This is problematic in cases where events form structures independent from speech (especially, when their granularity is finer or when they occur during speaker pauses).
Later, when computational analysis became more widespread, data representation formats and software tools arose which adopted this paradigm. Today, many annotation systems are still built around such a core model.
One of their advantages is that such data structures are clear and manageable. The linear basic structure also allows for constructions of tree-oriented markup (in XML or in similar markup languages).

[4] see `http://www.lat-mpi.eu/tools/arbil`

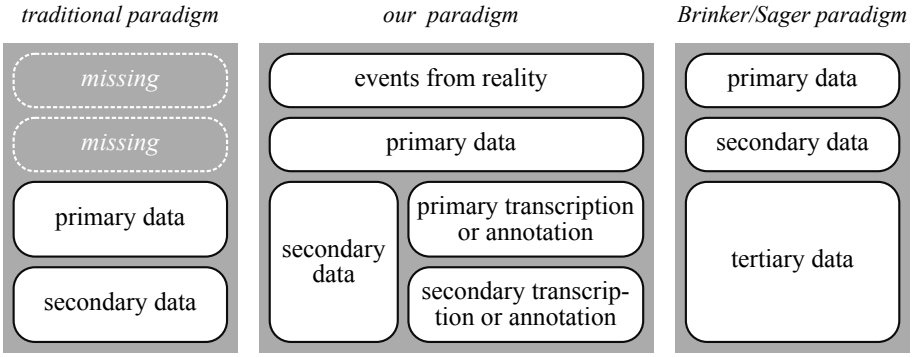


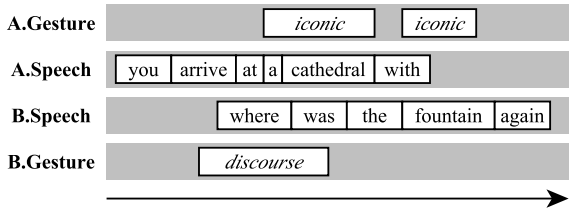
Figure 2: Different readings of concepts like "primary data" and "secondary data" in three major paradigms relevant to this paper. Elements at the same vertical level are equivalent, thus refer to the same concept.

For the analysis of complex multimodal data, however, these data models are insufficient:

1. Multiple independent streams of events cannot be linearized into single primary sequence of tokens without loss of vital information about the complex order of elements (see Figure 3 for an example).

2. Many data models (especially those that are closely based on XML) require strict hierarchies of subsequent annotations (corresponding in complexity to trees and forests in graph theory). In our case, there is need for more complex structures that are structurally equivalent to acyclic graphs. These can also be modeled in XML, but additional mechanisms are required to add the needed expressiveness.

3. By selecting already transcribed utterances as their primary data, the traditional models do not model any data sets which are highly important for complete and sustainable corpus models, especially primary recordings, like video and audio files of experiments. Several disciplines (e.g., phonetics) that base their work on such data sets cannot work with such a corpus representation.



Linearization 1: you · arrive · *discourse* · where · at · a · *iconic* · cathedral · was · the · with · fountain · *iconic* · again

Linearization 2: you · arrive · at · a · cathedral · with · where · was · the · fountain · again · *iconic* · *iconic* · *discourse*

Figure 3: Ficticious example of a set of communicative events: Two speakers produce overlapping speech accompanied by gestures. In addition, two linearization variants are shown. Any linearization would either split up coherent units of speech or disregard their strict temporal order. In both cases, the primary sequence in isolation lacks certain vital aspects of information.

2.2. Our model of primary and secondary data

Compared to this view we have a different understanding of what should be considered primary data. A contrasting juxtaposition of all relevant terms and concepts is given in Figure 2.[5]

According to our definition, primary data are those resources that record or store the relevant events from reality without interpreting actions performed by humans and without taking any other primary data into account. The most important of them are video and audio recordings, along with data dumps and streams from sensors and electrodes (for example, when measuring the potential of facial muscles with an EMG device, or when tracking eye and body movement with respective tracking systems).

In terms of meta-evaluation, all these primary measurings need to be objective and reliable. They should form a valid image of the real events that were the object of study.

Subsequently, secondary data can be created either by automated processes or by human annotators. We call it "secondary" because one or more sets of primary data are required in the creation process. Elements from secondary data themselves can either be *primary* or *secondary annotations*:

- Primary annotations refer to primary data only. Intervals marked on a timeline designating spoken words are a typical case of primary annotation.

- Secondary annotations refer to groups of primary annotations. The assignment of part-of-speech tags to words creates secondary annotations.

For different tasks, very diverse linking structures between annotations can be necessary. While many traditional annotation models restrict allowed structural relations, we do not impose any restrictions in principle. We do, however, offer

[5]There are yet other theories that differ from both our and the traditional way of naming data: For example, (Brinker and Sager, 2010) regard the initial recordings as secondary data because for them primary data are the transient events in reality themselves. In Figure 2 we included a comparison chart of these three naming conventions because we think it is necessary to clarify what reading of "primary" we are using to avoid confusion. For the rest of the document we will adopt the nomenclature of our approach (which we labeled "our paradigm" in the figure).

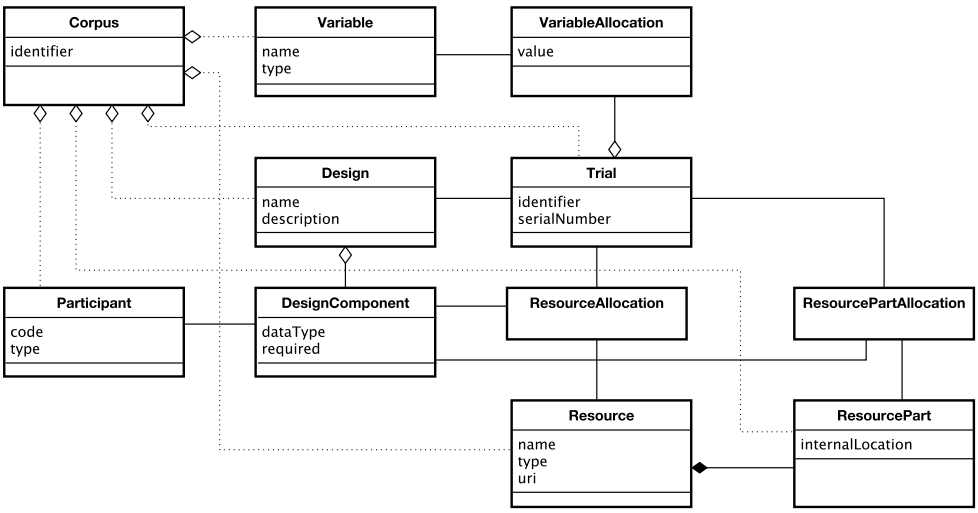


Figure 4: Corpus Class Diagram, simplified: Abstract classes and MetaEntries have been omitted. Associations with the Corpus object are drawn with dotted lines to enhance readability of the remaining edges.

a system of constraints and types that helps to keep track of links between single groups of annotations.

2.3. Our organisational macrostructure

Figure 4 contains a simplified UML class diagram we use for modeling the macrostructure of multimodal corpora. It focuses on corpora resulting from psychological or (psycho-)linguistic experiments. Therefore, a corpus can contain a number of *trials* that represent actual performances (or instances) of an experimental setup. If, for example, 24 pairs of people participated in an experiment, then the corpus contains 24 trial entries.

Each trial is assigned a *design* that describes how the experiment was set up and what kinds of resources have been collected during experimentation or have been created by transcription or annotation.

Such resources are described by *design components*. If the example experiment above called for video recordings from two perspectives, one audio recording and speech transcriptions for both participants, then the corresponding design would consist of five design components: two entries for video resources, one entry for an audio resource, and two entries for speech transcription resources.

A design can be seen as a template for trials. It can be consulted to check, for instance, whether all necessary resources for a trial have already been collected.

The actual data sets are modeled as *resources*. These are simply objects that refer to actual files and other resources via an URI. Resources are connected to the trial and design component they belong to via *resource allocations*. These provide the semantic information needed when corpus objects are to be queried and searched. Without resource allocations it would be difficult to answer popular queries and search commands like "Return all video files that belong to the design component 'video, focus on person A' and that resulted from trial #4". Although the information needed to perform this task is often implicitly provided via file naming conventions, human annotators tend to deviate from naming schemes. In addition, the automated interpretation of such naming schemes requires assistance or additional information. By creating allocation objects users are encouraged to provide the relevant semantics themselves, thus adding the essential pieces of information to the corpus data structure.

In some cases it is necessary to address specific parts of a resource only. The most prominent example are file formats of several annotation tools where distinct groups of transcriptions or annotations are assembled into one single file. Typically, speech transcriptions for all participants of a trial are collected in a single resource file because this approach makes it easier to compare and align temporal information from different groups inside the annotation tool.

This, however, makes it difficult to separate the single groups of annotations when allocating them to different design components. We solve this problem by introducing *resource part* objects. These are associated with a resource and have an internal location descriptor that describes what part of the given resource the object refers to. For annotation documents this can be the name of the layer or group. For XML documents this could be the XML identifier of an element, a set or range of identifiers, or an XPath expression. Resource parts can then also be referred to by allocations in order to link them to design components and trials.

2.4. Discussion: Differences between our model and components from CMDI Registry

One of the steps towards a metadata representation of our model would be the creation of CMDI components. Users are encouraged to reuse existing components, if appropriate. To examine to what extent our model is compatible to already existing entries in the CMDI registry we selected two resources from the CMDI registry[6] on the basis of their names: *BamdesMultimodalCorpus*[7] and *media-corpus-*

[6] `http://catalog.clarin.eu/ds/ComponentRegistry` – all resources have been checked on 23 February 2012.

[7] CMDI component definition: `http://catalog.clarin.eu/ds/ComponentRegistry/?item=clarin.eu:cr1:p_1288172614021`

profile[8].
Both are compatible to a large extent, but in both cases there are problems and incompatibilites that make a complete metamodeling of our corpus structures difficult.
In *BamdesMultimodalCorpus*, we noticed the peculiarity that only a single value for the attribute *CorpusType* may be given although its range is a union of multiple independent value sets. These sets may indicate

- the number of languages used in the corpus (monolingual, bilingual, multilingual);
- alignment modes for parallel corpora (alignment at document, paragraph, sentence or word level, respectively);
- the nature of underlying data (read, spontaneous, monologue, dialogue).

It was not clear to us why these distinct categories are all merged under one common category, thus rendering it impossible to provide more than one of the values listed above. In addition, we are dealing with corpus types that are not yet included in the domain of *CorpusType* (e.g., corpora based on experiments and studies that can differ from both dialogue and monologue).
The attribute *AnnotationLevelType* is rather complete even for multimodal corpora (with entries such as "facial expression" and "topic annotation"), although there are still some values missing. Also, it would be necessary to replace existing categories with finer-grained values: "gesture" alone, for instance, is insufficient for an appropriate description of the many levels and types of gesture annotation (observable gesture units, morphological information that describes position and movement of fingers, hands, and arms in space, and interpreting annotations about type, meaning and function of gestures) that can be found in some of our corpora and resources, e.g. the SaGA (speech and gesture alignment) corpus (Lücking et al., 2010). This is not the only case where we need to express novel data units and concepts. In Table 1, we collected an overview of all the modalities and their data categories that occur in data sets collected at the CRC 673.
To express all these novel data categories, we are working on an exhaustive ontology describing all modalities along with data units used within them. This ontology will be more complex than a single enumeration of values because there are several kinds of groups and relations to be expressed. However, concepts from this ontology could later be semantically linked to equivalent concepts in other registries, thus making it easier to relate corpus data in our format to other data representations (see section 4.).
Finally, we observed that the *BamdesMultimodalCorpus* component apparently does not allow us to model complex relations to resources (e.g., media files that are part of the corpus): It is possible to refer to additional resources with a reference mechanism, but this only models a simple membership or containment relation We can hardly express the various relations used in our corpus model (assignments of designs, design components, trials to resources and, more importantly, resource parts) – at least not without considerable alterations of the components.
It is unclear to us what would be the best way to use these components that meet the majority, but not all of the requirements of our complex corpus model. Should we alter the existing components by creating copies, or should new components designed from scratch and semantically aligned to the existing ones? We encountered similar questions and problems when it came to the selection of ISOcat data categories for the assignment of types to parts of our components. These will be summarized in the following section.

modality	**data units**
speech	phonemes, morphemes, words, phrases, sentences, utterances, turns, synsets, topics, speech acts
prosody	primary/secondary stress, ToBI prosody annotation
gesture	sequences, phrases, phases, morphological information, practices, topics
head movement	positions, angles, annotated head gestures
gaze	positions, angles, focused objects/areas
facial expression	manual annotations, EMG data collections
augmented reality	objects and events added or altered in augmented reality vision systems
action	object manipulations (identification, movement, conversion, relation)

Table 1: Selection of modalities and related data units and phenomena observed and annotated in various projects of the CRC 673.

3. ISOcat

3.1. ISOcat has been designed for linguistics and language resources

As stated on its website, "ISOcat is an implementation of the ISO 12620:2009 standard"[9], and that standard is dedicated to language resources. We already showed, however, that the scope of our resources significantly exceeds that area – it involves the much more diverse area of human interaction, and also of the interaction in various constellations of human and non-human interlocutors. While many of them are undoubtedly relevant to the area of language resources, we are in doubt whether all of these data categories (e.g., related to raw head or eye tracking data, to mental states of artificial agents, or to the field of augmented reality) should be integrated into the ISOcat system – after all, they are not related to language resources, and even only remotely to the area of linguistics in its traditional sense.
This is another argument for the creation of a separated system that can store such concepts.

[8] CMDI component definition: `http://catalog.clarin.eu/ds/ComponentRegistry/?item=clarin.eu:cr1:p_1324638957739`

[9] `http://www.isocat.org/files/12620.html`

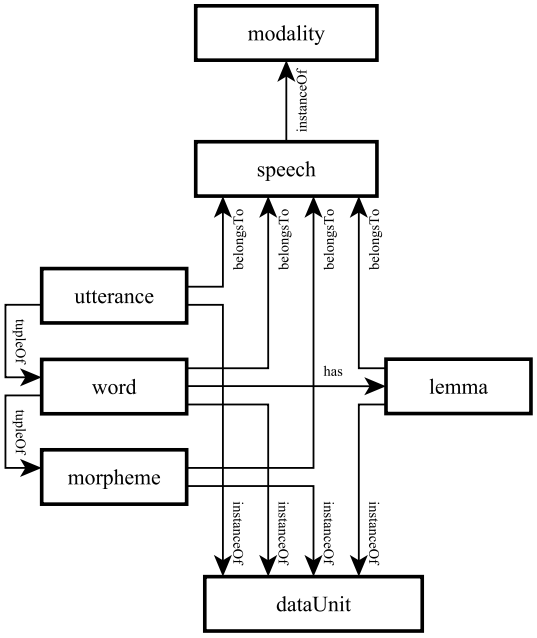


Figure 5: Example of a segment from the ontology of modalities and corresponding data units under development.

3.2. ISOcat is for established data categories

The third argument for such a separated system lies within the fact that a lot of our categories are actively developed in fundamental research. They can be altered and rejected, and data categories can merge and split.

We believe that the ISOcat system is not an appropriate place for data categories of such early stages. Entries in ISOcat should already have been discussed and agreed upon by the people who add them.[10]

An export or integration to ISOcat at later stages in the lifecycle of a category is often reasonable, but for the first unstable phases of a category we consider a separate system more appropriate.

In addition, a more interactive system (e.g., with means for discussions and storage of sources, third-party definitions, and citations) would enhance such a system especially during early category development.

4. Conclusion

The results of our evaluation of both the ISOcat and the CMDI registry systems are:

1. With given categories and CMDI components, an automated export to CMDI instances inside our corpus management system *Ariadne* can easily be achieved.

2. Creation or adoption of CMDI components, and choice of ISOcat entries is not trivial for categories that are less related to language resources, or that are in early production states.

As a consequence we are in the process of creating an ontology of modalities, their relations and their typical data categories. This ontology will be used within the corpus management system to assist users at creating reasonable and consistent relations in their corpus data. It can also be used and queried for the export of RDF data as well as for the generation of metadata descriptions in CMDI and other formats. Categories and concepts of the ontology that also exist in other registries can then be linked with semantic web and linked data technologies. An example of possible classes, structures and instances inside such an ontology is given in Figure 5.

Such a system should also contain means and mechanisms for user interactions of various kinds: Categories under development need to be discussed, and sources from the internet and from publications have to be added and evaluated. We plan to release an early draft of such an interactive ontology in the next weeks. Upon acceptance of the paper that draft can also serve as the basis for a detailed presentation and discussion of the multimodal data types and categories used at our research centre.

[10] In fact, ISOcat does provide support for categories in such early stages, but in combination with other arguments we still consider work in a separate editing platform more advisable.

5. Acknowledgements

Financial support of the German Research Foundation (DFG) through the CRC 673 “Alignment in Communication” (in the project X1 “Multimodal Alignment Corpora: Statistical Modeling and Information Management”) is gratefully acknowledged.

We also thank the anonymous reviewers whose comments on the first version of this paper helped us to improve its quality and to clarify some important parts.

6. References

Klaus Brinker and Sven F. Sager. 2010. *Linguistische Gesprächsanalyse: Eine Einführung*. Erich Schmidt, Berlin.

M. Kemps-Snijders, M. Windhouwer, P. Wittenburg, and S.E. Wright. 2008. ISOcat: Corralling data categories in the wild. In *Proceedings of the International Conference on Language Resources and Evaluation (LREC), Marrakech, Morocco*.

M. Kemps-Snijders, M. Windhouwer, P. Wittenburg, and S.E. Wright. 2009. ISOcat: remodelling metadata for language resources. *International Journal of Metadata, Semantics and Ontologies*, 4(4):261–276.

Andy Lücking, Kirsten Bergmann, Florian Hahn, Stefan Kopp, and Hannes Rieser. 2010. The Bielefeld Speech and Gesture Alignment Corpus (SaGA). In *Proceedings of the LREC 2010 Workshop "Multimodal Corpora - Advances in Capturing, Coding and Analyzing Multimodality"*, pages 92–98.

Peter Menke and Alexander Mehler. 2010. The Ariadne System. A flexible and extensible framework for the modeling and storage of experimental data in the humanities. In *Proceedings of LREC 2010*, Malta. ELDA.

Peter Menke and Alexander Mehler. 2011. From experiments to corpora: The Ariadne Corpus Management System. In *Corpus Linguistics Conference 2011*, Birmingham, UK.

Martin J Pickering and Simon Garrod. 2004. Toward a mechanistic psychology of dialogue. *Behavioral and brain sciences*, 27(2):169–90; discussion 190–226, May.

User Activity Metadata for Reading, Writing and Translation Research

Kristian Tangsgaard Hvelplund and Michael Carl
Copenhagen Business School

Dalgas Have 15
DK-2000 Frederiksberg

E-mail: kthj.isv@cbs.dk, mc.isv@cbs.dk

Abstract

While there exists a large amount of static linguistic resources together with annotation schema and metadata, not much work has been done to describe the processes by which texts are produced. In the mid-1980s, translation process research began to use advanced technologies such as keyboard logging and more recently eye tracking and screen recording to record and study user activity data of human reading, writing and translation processes. There has not, however, been much effort to synchronize and annotate the collected data. This paper suggests a structure for these processes along four dimensions. As the process data depends not only on the human writer/translator, but also on the type of text to be produced and the purpose of the final product, this paper suggests a metadata structure for user activity data which takes into account different dimensions.

1. Introduction

While a large amount of linguistic resources exists for textual (i.e. product) data with a plethora of annotation schema and metadata standards, not much work has been done to describe the process data by which texts are produced. A text document undergoes constant modifications as it develops in time and the final product could be considered a snapshot at a particular moment in time. For instance, wikis may change at any point in time since their content is editable. However, despite their fluctuating nature, wikis are valuable text documents at any one time.

Written translations can be considered a text which is generated based on another text. A translation may be modified or revised, leading to a new version, or another translation may be produced, based on a different understanding of the same source text or they may be targeted towards another audience. The Bible is such an example which, usually perceived as a static text, has been translated many times over many years, into many different languages and into many different versions. While we can compare and study successive versions of a text or a translation, we have only very restricted means to investigate the processes by which a text (or a translation) is produced. At present, few process repositories exist that contain data which allows us to trace and reproduce text production processes in detail. One such repository is Mesa-Lao's (2011) publicly available collection[1] of translation process data which were collected to study explicitation in translation memory mediated environments.

Research into what type of metadata is appropriate and relevant for annotating process data has not received much attention. Carl and Jakobsen (2009) introduce the term "User-Activity Data (UAD) to subsume any kind of process and product data which is consulted or generated by a translator during a translation session". They discuss a data format for the kind of UAD which they produce, but do not include a description of metadata. Göpferich (2009), however, suggests a metadata schema similar to the Dublin Core for translation processes which includes authors, purpose of translation, etc.

Seiner (2001) notes that "Metadata holds the key to understanding and using data. When it is available, metadata enables end users to understand the data and make better decisions based on this understanding." He mentions that "metadata also includes information about knowledge and knowledge-related processes – and knowledge doesn't always begin with data." By describing the contents and context of data files, the quality of the original data/files is improved considerably. It is, however, impossible to categorise metadata only by observing the object data since metadata categorisation is a function of the specific purpose of the object data or the specific usage scenario. Accordingly, various classification schemas have been developed to define and distinguish different types of metadata. One such classification concerns **descriptive metadata**, which involves the description of individual instances of application data, such as title, author, subjects, keywords, publisher, etc. and **structural metadata,** which concerns the description of the design and specification of data structures.[2]

In this paper, we suggest a structure and description of metadata for UAD. We categorize our metadata along four lines:

1. *Experiments metadata* of the reading, writing or translation task describes the purpose of the experiments from which process data is available. A unique ID is given to each experiment.
2. *Stimuli metadata* describes various properties of the source texts such as its language, genre and length. A unique ID is given to each stimulus.
3. *Participants metadata* describes the participants in the experiments with information such as age, native language, preferred translation direction and experience.

[1] http://tradumatica.uab.cat/trace/main,en/

[2] http://www.niso.org/publications/press/UnderstandingMetadata.pdf

A unique ID is assigned to each participant.

4. *Recordings metadata* contains information about the actual recordings of the experiments such as the location of the process log file, recording task type (e.g. translation or reading) target language, etc. A unique recording ID is given to each recording.

The four categories of metadata are explicitly related to each other as illustrated in Figure 1 below:

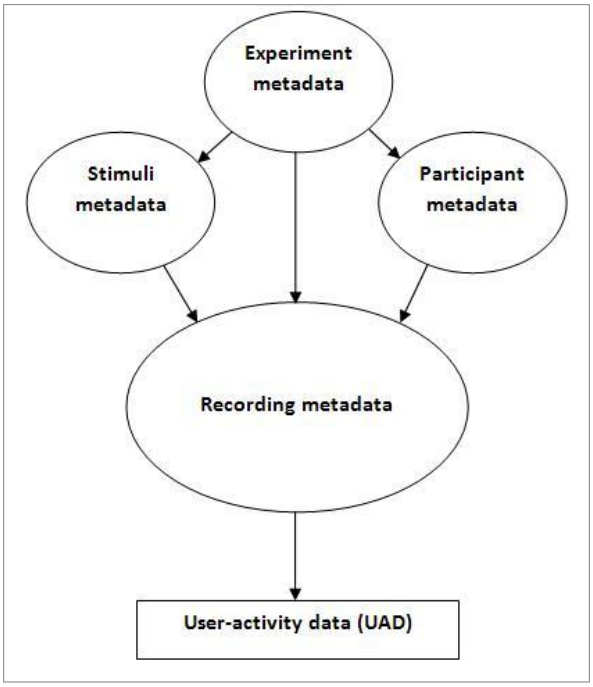


Figure 1: Metadata structure

Experiments metadata is the top-level metadata element. The unique experiment ID is represented as a derived element in the other three metadata categories. Below the experiments metadata element are stimuli metadata and participants metadata. Unique IDs from these two categories are represented in the recordings metadata category, and the recordings metadata category thus contain derived identifier information from the other three categories.

Based on this categorization of metadata, it will be possible, aided by a query interface, to extract UAD based on specific search properties.

2. Representation of metadata

As introduced above, four different categories of metadata describe the process data: experiments metadata, stimuli metadata, participants metadata and recordings metadata. These four categories will be explained in more detail below. For each category, examples of metadata information are provided. The left-most column (in bold) provides a list of examples of the types of metadata contained in each category. The two right-most rows are examples of the metadata content for two experiments.

The process data itself come from four different experiments which are briefly introduced below.

Exp1 contains eye-tracking and key-logging data from 24 translators' translations of four different texts (Hvelplund 2011). The aim of the study was to examine how translators distribute their cognitive effort during the translation process. 12 participants were professional translators and 12 were student translators. Each translator translated four texts that varied with respect to complexity. Two of these four texts were translated under time constraints while two texts were translated under no time constraints. A total of 92 recordings are available from this experiment, as four recordings had to be discarded for various reasons.

Exp2 contains eye-tracking and key-logging data from a translation experiment and eye-tracking data from two reading experiments (Balling *et al*. forthcoming, Jensen *et al*. 2011). The aim of the study was to investigate the matter of parallel processing in translation. In the translation experiment, 19 professional translators each translated two texts. In the two reading experiments, a total of 23 participants each read two texts. Due to problems with the quality of the eye-tracking data the translation experiment, only 16 recordings from 8 translators in are available. 46 recordings are available from the reading experiments.

Exp3 contains eye-tracking and key-logging data from 17 professional translators (Sjørup 2011). The aim of the study was to investigate changes in translators' allocation of cognitive effort when translating different types of metaphorical expressions. Each participant translated two texts and retyped two texts. A total of 68 translation and retyping recordings are available from experiment Exp3.

Exp4 contains eye-tracking and key-logging data from 14 student translators and eight professional translators (Dragsted 2010). The aim of the study was to investigate how translators coordinate comprehension processes and writing processes during the translation processes. Each participant read silently a text and then they translated that same text. A total of 44 translation and reading recordings are available from experiment Exp4.

2.1 Experiments metadata

Experiments metadata contains five pieces of information: an experimentID, an abstract, a list of keywords, the number of participants and the number of recordings, as illustrated in Table 1 below.

Table 1: Experiments metadata

ExperimentID	Exp1	Exp2
Abstract	This study is an empirical investigation of translators' allocation of cognitive resources, and its specific aim is to identify predictable behaviours and patterns of uniformity in translators' alloca-tion of cognitive resources in trans-lation.	The aim of this study is to examine empirically and experimentally the cognitive process by which professional translators translate metaphors from one language to another language.
Keywords	Translation, process data, cognition, distribution, allocation, parallel processing	Translation, process data, cognition, metaphors, conventional metaphors
Participants	24	31
Recordings	92	62

2.2 Stimuli metadata

Stimuli metadata contains information about the source texts that were used in the various experiments. This category contains information such as the domain and language of the stimulus, its length in words and characters and font type and font size. The experiment ID in the top-row of Table 2 below is derived from Experiments metadata.

Table 2: Stimuli metadata

ExperimentID(derived)	Exp1	Exp2
TextID	Exp1_TextA	Exp2_TextB
Domain	Legal	Fiction
SourceLanguage	English	English
LengthWords	145	132
LengthCharacters	845	785
FontType	Tahoma	Tahoma
FontSize	18	18

In addition, stimuli metadata could also contain information about the complexity of the text, as illustrated by readability measures (LIX, SMOG, Flesch-Kincaid, etc.), by word frequency scores and by the ratio of non-literal expressions (idioms, metonyms, metaphors, etc.) to literal expressions. Overall, information which relates specifically to the text stimulus that was presented to the user(s) is described in the stimuli metadata category.

2.3 Participants metadata

Participants metadata contains information about the participants from whom process data have been collected such as the sex of the participant, education, experience working as a translator, L1, L2, L3 etc., cf. Table 3 below. Similar to the stimuli data, the experiment ID in the top-row of the table is derived from Experiments metadata.

Table 3: Participants metadata

ExperimentID(derived)	Exp1	Exp2
ParticipantID	34	34
Sex	F	F
YearOfBirth	1981	1981
FormalTranslator TrainingYears	6	6
Education	cand.ling.merc	cand.ling.merc
DegreeFinishedYear	2008	2008
ExperienceYears	3	5
L1	Danish	Danish
L2	English	English
L3	German	German

In addition, participants metadata contains information about the participant's preferred translation direction (if s/he is a translator), if the person is a touch typist and the participant's eye colour and prescription (if s/he wears contact lenses or glasses).

The same participant may appear multiple times in this participants metadata, as illustrated by the ParticipantID indicator, since a participant may participate in several experiments. Some of the properties of that individual may change over time. For instance, the number of years a participant has worked as a translator may be different for two experiments in case the experiments have been executed at different points in time. A novice translator in one experiment, as indicated by few years of professional experience, may be considered a professional translator in another experiment, which was executed years later, as illustrated by many years of professional experience. It would therefore be undesirable to categorise the participants according to their level of proficiency since this property is not universally static.

Overall, information which relates specifically to the individual participant who took part in an experiment is described in the participants metadata category.

2.4 Recordings metadata

Recordings metadata contains information about the individual recording. The information (ExperimentID, TextID and ParticipantID) in the three top-rows is derived from the experiments, stimuli and participants metadata categories, respectively. Other information that is unique to that very recording is introduced into recordings metadata: the experimental location at which the recording session took place (e.g. a specific university, the participant's office, the participant's home), the type of experiment (translating, reading, copying, revision, post-editing), and the target language of the target text output.

Table 4: Recordings metadata

ExperimentID(derived)	Exp1	Exp2
TextID(derived)	Exp1_TextA	Exp2_TextB
ParticipantID(derived)	34	34
LocationLink	www.website.org/ Exp1_TextA_P34 .xml	www.website.org/ Exp2_TextB_P34 .xml
RecordingDate	20-01-2008	20-03-2010
EyeTrackerType	Tobii1750	TobiiT120
RecordingSoftware	Translog2006+Cl earView	TranslogII
ExperimentalLocation	CBS	CBS
ExperimentType	Translating	Translating
TargetLanguage	Danish	Danish

Recordings metadata also includes information such as the fixation filter used for that specific recording to calculate fixations and the quality of the eye-tracking data. Overall, recordings metadata contains information which is unique to the individual recording.

3. The Structure of UAD

The UAD acquisition software "Translog-II" (Jakobsen, 1999; Carl, 2012) logs keystrokes and mouse activities during text production. It also records gaze fixations and movements over the texts when the participant is working in front of an eye tracker. While Translog-II can also be used for reading and writing research, we present here the logging protocol when used as a translation tool. As described elsewhere (Carl and Jakobsen, 2009), we have developed a six-dimensional representation which divides the data into three *Product data* resources and three *Process data* resources, as explained below.

3.1 Product Data

The product data consist of three resources: i) the source text (ST), ii) its translation, i.e. the target text (TT), and iii) a linkage between linguistic entities of both texts. The location for each character in the ST and TT is identified by its position on the screen and its position in the text. The screen position of each character is identified through a rectangle with its top-left position in terms of X/Y pixels and the width and the height coordinates. The cursor positions give the character position as offset from the beginning of the text. Alignment information indicates which units in the SL text correspond to which units of the TL text.

3.2 Process Data

The process data also consist of three resources, the i) gaze sample observations, ii) fixations and iii) keystroke information. In contrast to the product data, the process data contains a time stamp, indicating in milliseconds when the event took place relative to the beginning of the experiment.

Gaze sample observations consist of screen coordinates as obtained from the eye tracker for the left and right eyes, as well as pupil dilation at a particular time. For each observation, the location of the character which is closest to the gaze sample point is recorded (i.e. the cursor position of the character in the ST or TT) along with information about the location of that character on the screen.

Fixations are computed based on sequences of gaze sample observations. Fixations thus consist of a number of gaze sample observations and represent a time segment in which a word (or symbol) is fixated. In our current representation, fixations have a starting time, a duration, and a cursor position which refers to a position in the ST or TT.

Translog–II also logs keyboard and mouse activities. We distinguish between four different types of keystrokes: insertion, deletion, editing, navigation and the return key.

4. Conclusion and outlook

We intend to build a relational database of translation process data which will be made publicly available in 2012, e.g. through an anonymous SVN (Apache Subversion) server. Users will also be able to query the metadata on a website where an interface will make it possible to select the process data that fit a specific research objective. Currently, almost 300 translation recordings have been collected, as shown in Table 5 below. We expect this number to increase considerably in the near future when recordings from new experiments are added to the repository.

Table 5: Experiment overview

Experiment	**Number of participants**[3]	**Number of recordings**
Exp1	24	92
Exp2	31	62
Exp3	17	68
Exp4	22	44
Total	105	288

Once the repository is available, it will be possible to upload process files according to specific guidelines that will be developed. Online tests that control for object data consistency and metadata consistency will be performed when files are uploaded.

We intend to publicise the existence of the process data repository as well as the metadata through META-SHARE and other appropriate channels.

Due to the multi-dimensional structure of the translation process data, we suggest in this paper to organize the metadata in independent but related sets of information concerning 1) the experimental focus of the text processing activity (Experiments metadata), 2) the origin, domain and difficulty of the source text to be processed (Stimuli metadata), 3) information about the

[3] Only participants from whom there are valid process data available are included in these figures.

participant and his or her experience and background (Participants metadata), and 4) metadata concerning the actual observed data during text production or reception task (Recordings metadata). The goal of organizing the process data into a rigid structure such as the one outlined in this paper is to make process data easily searchable across multiple parameters. The next step will be to implement a query language which will allow us to extract the process data that is related to particular linguistic phenomena, and correlate this with the translators' profile. This information could, for instance, be instrumental to investigating whether novice translators face the same problems as more experienced translators, and which texts passages are particularly difficult for every translator. The clear structuring of the metadata of a given process task is a prerequisite for such an endeavour.

5. References

Balling L. W., Hvelplund, K. T., Sjørup A. C. (under review). Evidence of Parallel Processing During Translation.

Carl M., Jakobsen A. L. (2009). Towards Statistical Modeling of Translators' Activity Data. In International Journal of Speech Technology, Volume 12, Number 4, 125-138.

Dragsted, B. (2010). Coordination of reading and writing processes in translation. An eye on uncharted territory. In G.M. Shreve and E. Angelone (eds). Translation and Cognition. Amsterdam/Philadephia: Benjamins Publishing Company. 41-63.

Göpferich S. (2010). Data Documentation and Data Accessibility in Translation Process Research. In The Translator, Volume 16, Number 1, 93-124.

Hvelplund, K. T. (2011). Allocation of cognitive resources in translation: An eye-tracking and key-logging study. [accessed 23 February 2012 from http://openarchive.cbs.dk/handle/10398/8314] Doctoral dissertation. Copenhagen Business School.

Jensen, K. T. H., Sjørup, A. C., Balling, L. W. 2009. Effects of L1 syntax on L2 translation. In I. M. Mees, F. Alves and S. Göpferich (eds). Methodology, Technology and Innovation in Translation Process Research: A Tribute to Arnt Lykke Jakobsen. (Copenhagen Studies in Language 38). Copenhagen, Samfundslitteratur. 319-336.

Mesa-Lao B. (2011). Explicitation in translation memory-mediated environments. Methodological conclusions from a pilot study. Translation & Interpreting Vol 3, No 1

Seiner R. S. (2001). Meta Data as a Knowledge Management Enabler [accessed 23 February 2012 from http://www.tdan.com/view-articles/4916/].

Sjørup, A. C. (2011). Cognitive effort in metaphor translation: An eye-tracking and key-logging study. Doctoral dissertation. Copenhagen Business School.

Metadata for a Mocoví – Quechua – Spanish parallel corpus

Paula Estrella

FaMAF, Universidad Nacional de Córdoba, Argentina
Haya de la Torre s/n, Ciudad Universitaria
pestrella@famaf.unc.edu.ar

Abstract

In this paper we present the work done to create the metadata associated to a parallel corpus of endangered languages spoken in Argentina, namely Mocoví and Quechua. Creating metadata is of great importance not only to document the resource and the language but also to make it available to the general public through browse or search facilities, given that resources for Amerindian languages are so few and so difficult to find. However, choosing an appropriate schema is not a trivial task if compatibility and interoperability are in mind. Therefore, it was decided to reuse previous work by major initiatives in language archiving and documentation, resulting in the customized IMDI schema described in this article.

1. Introduction

In Argentina, minority and minorized languages have long been in a diglossic relationship with the majority language, Spanish. Moreover, the Argentinian state does not give them any official status. This socio-linguistic situation has led to the near extinction of many indigenous languages, as reported in (Moseley, 2010). In particular, the Mocoví and Quechua languages present a decreasing number of native speakers due to several factors in addition to this diglossic language situation: historically, aboriginal communities were expected to learn Spanish to avoid being isolated, more recently, younger generations migrate to bigger cities where the only language spoken is Spanish, and in some cases the aboriginal languages are reduced to the private sphere (family, religion, folklore) while Spanish is preferred for public scenarios (administration, education, media, etc).

In view of this situation we are working on a project to contribute stop the process in which more indigenous languages get extinct. The goal of the project is to formalize two of the languages spoken in Argentina, namely Mocoví and Quechua, in order to ease their study and preservation. Given the ambitious goals of the project, this is a long-term work and in its initial phase we propose to create a basic linguistic resource, namely a Mocoví – Quechua – Spanish parallel corpus.

A key aspect of linguistic resources is the metadata associated with them, which should help understand the nature of the resource as well as its preservation and usage. Therefore, we want to augment the corpus with relevant metadata to give it more visibility given that resources for Amerindian languages are so few and so difficult to find.

The paper is organized as follows: we first describe the resource we are documenting (Section 2.), including a brief description of the languages involved, in Section 3. we present a set of initiatives studied before selecting a metadata schema that best fits our purposes, then Section 4. presents a customized schema resulting from this study and Section 5. presents some conclusions and future work.

2. Description of the resource

In Argentina there are no less than 18 minority languages, all of them classified as "definitely endangered" (Moseley, 2010). For most of them, no linguistic studies exist to account for their typology, grammar or vocabulary. In most of the cases, they are oral languages and the few resources available, if any, have not been preserved. They are thus in severe need for documentation and, in the best case, for linguistic policies to aid their recovery.

Quechua, as well as Mapundungun and Guaraní, are in a different situation. Since they have the status of official languages in neighbouring countries, they are supported by some linguistic policies for their preservation and they have been formalized and normalized to some extent. Therefore, our purpose is to use Quechua as a testbed for the methodology to be applied to Mocoví, where obtaining resources is very costly. Additionally, Mocoví and Quechua share some features that are not present in Spanish, like the fact that they are agglutinating, as explained in the following subsections.

2.1. The Mocoví language

The Mocoví language is spoken by some 2000 speakers, but is in a clear process of substitution, since from almost 16,000 people who consider themselves Mocoví descendants, only 18% declare Mocoví as their native language. In the province of Santa Fe, our research area, currently only some adults and the elderly still use the language, whereas in El Chaco the Mocoví culture, including the language, retains a more central role in communities.

There are very few studies on the Mocoví language (Grondona, 1998; Gualdieri, 1998; Carrió, 2009), and many of its features are still under discussion, as there is a lack of evidence to support or refute competing hypotheses. Morphologically, Mocoví is clearly an agglutinating language, with grammatical relationships coded in the verbal/nominal nucleus through the pronominal morphology and/or other specific markers. It also presents a very rich grammatical expression of spatial parameters, expressed both in nominal and verbal morphology.

The order of the syntactic constituents seems relatively free, with greater frequency of SVO in transitive clauses, VS in

the intransitive ones, and NA in the nominal phrase.

2.2. The Quechua language

The Quechua language, with all its different varieties, is spoken by 8 to 10 million people in the area of western southamerica, including north-west Argentina, where it is spoken by around 80000 people. It is a co-official language in Bolivia, Colombia, Ecuador and Perú, in all cases together with Spanish, which tends to be mostly used in administrative situations.

Quechua is a highly agglutinating language, and verbs agree with subject and object. This is a feature that may be useful as a bridge between Spanish and Mocoví. There are morphological markers of evidentiality and commitment of the speaker, as well as particles of topicality.

As opposed to Mocoví, Quechua benefits from a number of computational resources, such as dictionaries, bilingual dictionaries (many of them digital or machine-readable), grammar-related resources and digital corpora, some of them parallel or comparable with Spanish, like literary translations or the Wikipedia.

2.3. A three-language parallel corpus

Since obtaining data for Mocoví is far more difficult than for Quechua, we started the compilation of a corpus of Mocoví, and then had it translated to Quechua and Spanish. After the first year of this three-year project, we have located informants for Mocoví, we have established the methodology for corpus collection and triangulation, and we have collected a rather small starting corpus. We expect that in the two years to come the corpus will be enhanced significantly, mainly by integrating this corpus with previous lexica and corpora available from (Gualdieri, 1998; Carrió, 2009).

Mocoví

[natarenataɢanaq loβee ia^{h}antak na laap noɢot]

n-ataren-ataGan-aG	loBe-e	i-ahan-tak
Ind-*heal*-Detr-Nmz	*tooth*-Pc	3sg-*look*-AspProg
na	l-aap	noGot
Det	3Pos-*mouth*	*kid*

Quechua

Khiru-hanpiqqa irqip khirunkunata qhawaykuchkan.

[khiruhan'peχqɒ 'erqɛχ khiruŋku'nata qhɒwaj'kuʃaŋ]

khiru-hanpi-q-qa	irqi-p	khiru-kuna-ta
tooth-heal-AG-TOP	*kid*-GEN	*tooth*-PluN-AC
qhawa-yku-chka-n		
look-IND-PRO-3sg		

Spanish

El dentista revisa la boca del nene.

El	dentista-∅	revisa-∅	la	boca-∅
Det-Msg	*dentist*-sg	*revise*-3sg	Det-Fsg	*mouth*-Fsg
d-el	nene-∅.			
of-Msg	*kid*-3sg			

Figure 1: Example of phonetic, phonological and morphological analyses of the Mocoví-Quechua-Spanish sentence *The dentist is revising the kid's teeth*.

Since Mocoví is an oral language, the corpus was recorded, then transcribed phonetically and phonologically, and morphemes were segmented. A literal as well as a free translation to Spanish were also given. Once this process was complete, all sentences were translated from Spanish into Quechua and the same analyses as for Mocoví was performed. An example of an analyzed sentence from the corpus is shown in Figure 1; the translations plus analyses are given for the sentence "*The dentist is revising the kid's teeth*".

3. Metadata for language documentation

When faced to the issue of designing and implementing the metadata for the corpus, we started reviewing the guidelines provided by several projects given that we want to use an up-to date specification that would also adhere to accepted standards (if any), to avoid generating yet another isolated schema for metadata.

During this process we found out that there are two kinds of initiatives: those dealing specifically with language documentation (e.g. E-MELD, DOBES) and those dealing with language resources in general (e.g. FLaReNet, META-SHARE). In both cases the desiderata for the metadata schema is very similar: it should be as expressive and flexible as possible, it should help achieve visibility and discovery of the data, it should be in an open format (e.g. XML) and it should also be interoperable. However, the practical details and implementation vary widely. Therefore, we decided to focus on specific projects about language documentation as a first step towards the design and implementation of metadata to describe our resource.

The following initiatives are considered the most important in the field of language documentation and were, therefore, studied.

3.1. ELAR

The Hans Rausing Endangered Languages Archive (ELAR) does not suggest any particular formulation of metadata based on the principle that metadata plays an important role in the management and discovery of data by users and, therefore, it should primarily be as expressive and descriptive as possible, regardless of the standard chosen.[1] However, they advise depositors to provide a basic set of metadata records preferably based on the *Best Practice Recommendations for Language Resource Description* produced by the Open Language Archives Community (OLAC) (Simons and Bird, 2003; Bird and Simons, 2003) and because this minimal set of fields will be shared with the OLAC, each field needs to be mapped to an OLAC or ISLE MetaData Initiative (IMDI) (Broeder et al., 2001) field by the depositor.

The following elements make up the suggested minimal set and should be provided for each file at the time of deposit:

- Identifier: unique id for each item in the deposit.
- Format: describes file/mark-up/character encoding format
- Creator: entity primarily responsible for making the content

[1] http://www.hrelp.org/archive/depositors/key_points.html

- Subject.language: the language(s) which is described or documented
- Language: the language in which the content is in.
- Rights: rights held in and over the resource
- Title: short name for the resource
- Description: an account of the content of the resource
- Type: genre of the content in the resource

As for the format in which metadata is stored, they accept a variety of formats (not necessarily open) that include plain text, XML and spreadsheets, leaving this choice to the depositor.
This approach seems very simple to adopt because it does not necessarily involve technical expertise to generate XML content nor does it require gathering a large set of information.

3.2. E-MELD

The Electronic Metastructure for Endangered Languages Data (E-MELD) is one of the few initiatives that hosts resources about the Mocoví language. For example, it contains information about the language, it hosts a lexicon and a dictionary, and the *School of Best Practices* contains a step-by-step guide to migrate Mocoví data stored in a proprietary format into an open format like XML. Other sections such as the *Classroom* section provide general recommendations about metadata for language archiving. Besides providing theoretical background about resource digitalization, archiving and management, E-MELD generated extensive guidelines regarding practical aspects to be considered during resource building (Aristar-Dry, 2008; Boynton et al., 2010).
Regarding metadata, E-MELD recommends that it conforms to recognized standards, specifically to the specialized Open Archives Initiative (OAI) sub-domain, the Open Language Archives Community (OLAC), or to the IMDI Session description. This would cover the minimal set of fields mentioned in the previous sub-section and, as opposed to ELAR, E-MELD already provides a mapping of OLAC to IMDI fields to facilitate the metadata creator's choice of the standard to be used. Moreover, E-MELD strongly encourages creating OLAC metadata using the OLAC Repository Editor (ORE), with which linguists can create XML documents by simply completing a series of online forms; the advantage is that the resulting documents are automatically readable by the OLAC search engine. While this particular service is not available anymore, a new similar service is available (although still being tested) in which a metadata record is created by filling in a form and it is then harvested by the OLAC system.[2] This service is really useful to have our resource appear in the OLAC search engine. In addition to it, we decided to complement the corpus description with a richer metadata specification. From E-MELD we could narrow down the options (to OLAC/IMDI) but the specific details of the metadata set implemented is not available, so other sources were further consulted as described below.

3.3. DOBES

The Dokumentation Bedrohter Sprachen (DoBeS) Programm aims at preserving both the language and culture of indigenous communities and is of particular interest due to its long-standing collaboration with Argentinean researchers documenting endangered languages of the northern region, namely the Chaco region (Golluscio, 2003). There is also a relatively new project in collaboration with DoBeS, which aims at creating a digital archive for resources in the languages tapiete, vilela, wichi and mocoví.[3] Given the closely related work it would be desirable to deposit our work in that archive as it is not yet part of it, and in that case we would have to follow the guidelines provided by DoBeS.
The *Specifications for archival document formats to promote long-term accessibility* states that the IMDI specification is mandatory for cataloging purposes (specifically demanding XML plus IMDI schema). Because most of the resources collected in DoBeS are multimedia, primarily audio and video recordings, usually having accompanying textual material (such as annotations, transcriptions and translations), the main focus of the archive is spoken data. Consequently, DoBeS adapted the IMDI schema to "develop a generic hierarchy of spoken human communication with queryable controlled vocabularies of elements", as explained in (Dwyer and Mosel, 2001).
Browsing the sessions about Argentinean endangered languages in the DoBeS archive reveals that a basic set of fields were completed, instead of taking advantage of IMDI's flexibility to add specific metadata fields deemed useful for the kind of resources we deal with. This issue is taken into consideration in the next Subsection.

3.4. AILLA

The Archive of the Indigenous Languages of Latin America (AILLA) has also adopted the IMDI elements for session description to describe their collections but considering that the resources archived might not be limited to recordings (for instance they could also include grammars, dictionaries, pedagogical materials for language revitalization, etc.) the schema was adapted to describe a broader set of written resources.
Johnson and Dwyer (2002) discuss the IMDI schema version 2.4 (MPI ISLE Team, 2001) and propose a series of modifications, the most important in our view are the following[4]:

- Substitution of the name *Session* for *Bundle*, in order to denote a related set of resources rather than a classical linguistic elicitation session.
- Incorporation of elements *Date Archived* and *Last Modified* to the *Session* group to document the collec-

[2] This new service is available at http://http://talkbank.org/metamaker/ and it makes metadata records visible through the search service at http://search.language-archives.org

[3] http://dobes.caicyt.gov.ar

[4] Following IMDI's format we will denote an element by *Group-X.Element-Y* o equivalently refer to element *Y* in group *X*

tion's activity: the date a bundle is archived is probably different from the date it was created and the modification date indicates the bundle's activity. However, this modifications would not specify what the modification was, for example addition of resources, corrections, etc.

- Add elements *Teaching* and *Analysis* to the *Genre* group (part of the *Content* group), referring to pedagogical materials and to works performed as part of scholarly research, respectively.

- Incorporation of *Project.Funder* element to store information about the funding body of the project in which a collection is produced.

- In the *Participants* subschema two Key-Value pairs are added, namely *Origin* and *Ocupation*, the former potentially useful to identify dialects of the informants and the latter provides additional information on participants.

- Elements *Glossing Type* and *Software* are added to the *Annotation File* subschema used to describe written resources as part of the *Resources* group; values for *Glossing Type* are for instance *morpheme-by-morpheme* while values for *Software* must be names of software products used to generate annotations.

Although this work is based on an older version of IMDI and should therefore be adapted to the newest version, it is a rich source of knowledge about the important metadata fields that should be encoded by adding or discarding certain elements. Therefore, in the next Section we further explore its application to create the metadata for our corpus.

4. Proposed schema

We are well aware that no single schema will fit all purposes but customizing the latest version of IMDI (MPI ISLE Team, 2003) based on AILLA's experience will allow us to create a metadata corpus both compatible with other archive's formats and will also contribute to achieve the goal of reusing as much as possible from previous work. The resulting schema can certainly be mapped to other similar schemas via XSLT stylesheets, other programming scripts or even via manual conversion if the changes needed are minimal.

A comparison of the IMDI version of 2001 to that of 2003 shows that the *Session Description* has been substantially improved and developed, taking into account some of the changes proposed by AILLA, DoBeS and other experts participating in workshops and working groups. For instance, regarding the changes suggested in Section 3.4., the name *Bundle* is considered instead of *Session* and the type of resource *Written Resource* was added along with a large set of pertinent elements.

As a result of reviewing the IMDI schema (version 2003) and considering some of the changes discussed in the previous Section, the following customization is expected to fit our purpose:

Group Name	Elements
Session	Name, Title, Date, Location, Description, Resource Reference, Keys, Project, Content, Resources, Actors, References, **Date Archived**, **Last Modified Changes**
Project	[Name], Title, Id, Contact, Description **Funder**, **Partner Institutions**
Content	Genre, Sub-Genre, Communication Context, [Task], [Modalities], Subject, Languages, Description, [Keys]
Communication Context	Interactivity, Planning Type, Involvement, Social Context, Event Structure, [Channel]
Languages	Language, Description
Actors	Actor, [Description]
Actor	Resource Ref, Role, [Family Social Role], [Name], Full Name, Code, Language, Ethnic Group, Age, Sex, Education, [Anonymous], Contact, [Description], [Keys]
Resources	Media File, Written Resource, [Source], [Anonymous]
Media File	Resource Id, Resource Link, Size, Type, Format, [Quality], Recording Conditions, [Time Position], Access, [Description], [Keys]
Written Resource	Resource Id, Resource Link, Media Resource Link, Date, Type, Sub Type, Format, Size, Derivation, Content Encoding, Character Encoding, Validation, Access, Language Id, [Anonymised], Description, [Keys] **Writing System**
Source	[Resource Ref], [Format], [Quality], [Time/Counter Position], [Access], [Description], [Keys]

Table 1: IMDI schema customized to describe a written parallel Mocoví – Quechua – Spanish corpus. Elements in bold are added to the schema and elements in brackets ([...]) are left with blank content.

- Add the following Key-Value pairs to the *Session* group: *Date Archived*, *Last Modified* and *Changes*, which is intended to specify and keep track of the modifications made, for example what resources were modified, if new resources were added, etc.

- Add *Funder* and *Partner Institutions* to the *Project* group; this is necessary to give the corresponding credits to people working on and funding projects. For instance, all institutions working on a project should be mentioned independently of the institution/person responsible for the resource (denoted by *Project.Contact*). As for *Funder*, in some cases it is mandatory to explicitly acknowledge funding bodies. This addition could be easily implemented using key-

value pairs if *Keys* was available in this group, which was available until version 2.7. Moreover, this would spare us further modifications of the schema.

- In the latest version of IMDI, the *Subject* element in the *Content* group accepts an open vocabulary and the suggestion is to use an existing library classification; instead, we propose to use the subject typology proposed by Sinclair and Ball (1996) as part of the EAGLES guidelines. The reason is that library classifications are too fine-grained and extensive to be easily applicable to each session while the EAGLES guidelines have enough categories to describe a large set of situations and seem more appropriate given that they were generated by studying a set of corpora.

- Like most endangered languages, writing systems are not always standardized and the conventions adopted for written resources must be specified somewhere; for instance, Mocoví does not have a standard writing system but Quechua has more than one. Therefore, a key-value pair *Writing System* is added in the *Keys* group of *Written Resource* element; alternatively, the element *Content.Description* could be used to explain the conventions as prose text.

- Some elements are not deemed important for three reasons: they are redundant (for example *Project.Name* because using *Project.Id* and *Project.Title* is enough to identify the project), they do not apply (an example is *Content.Task*) or the information is not available (an example is *Actor.Family Social Role*). Elements not used will thus be left with blank content.

The resulting schema is shown in Table 1; for space reasons the schema is presented in two columns: following the IMDI terminology the *Group Name* column has the top-level elements of the schema and the *Elements* column contains the elements grouped at each level. In the table elements in bold are those added to the schema as part of the customization process and elements appearing in brackets ([...]) will not be filled in.

4.1. Proposed extensions

Although the archives consulted use the widely accepted IMDI specification, there has been considerable progress in the field, where new metadata infrastructures emerged, making the archives' implementations rather out-dated. In particular, the ISO group ISO TC37/SC42 proposed a central registry of relevant linguistic data categories called the ISO Data Category Registry (DCR) (ISO 12620, 2009), a registry which can be collectively maintained and used by the research community. According to ISO 12620, data categories are "result of the specification of a given data field" and to support the interchange of selections of data categories the standard describes an XML serialization of the data model, the Data Category Interchange Format.
ISOcat3 is the first implementation of ISO 12620 [5] and allows researchers to instantiate the data model by using a web-based interface, where data categories can be created, edited, shared, exported and standardized; ISOcat also provides a web services interface; more information is provided in (Windhouwer and Wright., 2012).
One of the largest initiatives encouraging the use of ISOcat is the Common Language Resources and Technology Infrastructure (CLARIN), which proposes a component-based approach to metadata creation, specifically several sets of metadata elements can be combined into a self-defined scheme for a particular case. This approach is called the Component MetaData Infrastructure (CMDI)[6] and allows compatibility with other specifications, such as IMDI or OLAC. Moreover, they provide the tools (scripts and profiles) to convert an existing IMDI schema to CDMI. Therefore, we believe that providing an IMDI description of our resource is a good starting point to ensure compatibility with other archives, and then we can migrate to this more recent approach.

5. Conclusion and future work

We have presented a customization of IMDI based on previous experiences documenting and archiving endangered language data. From a preliminary study we found out that IMDI is adopted by major language archives, allowing us to thoroughly describe our resources. The advantage of using IMDI is not only that it is designed to document languages but also that it is highly adaptable and has a set of tools that make it easy to implement. These properties are of particular interest to people willing to create and maintain metadata, who are not necessarily experienced in the subject or do not have the computer skills to deal with technical issues (such as XML, conversion scripts, etc); for instance, linguists can benefit from the set of tools offered by IMDI[7]. However, progress in the field led to more recent initiatives that should be adopted; in our particular case, we need IMDI for compatibility with some archives, in view of future sharing, but we can also have a more up-to date description using CDMI by simply converting our IMDI records using a specific profile; this will be studied and implemented in the near future.

6. References

Helen Aristar-Dry. 2008. Preserving Digital Language Materials: Some Considerations for Community Initiatives. *Language and Poverty. Multilingual Matters*, pages 202 – 222.

Steven Bird and Gary Simons. 2003. Extending Dublin Core Metadata to support the description and discovery of language resources. *Computing and the Humanities*, 37:375 – 388.

Jessica Boynton, Steven Moran, Helen Aristar-Dry, and Anthony Rodrigues Aristar. 2010. Using the E-MELD School of Best Practices to create lasting digital documentation. *Language documentation : practice and values, John Benjamins*, pages 133 – 146.

D. Broeder, F. Offenga, D. Willems, and P. Wittenburg. 2001. The IMDI metadata set, its tools and accessible linguistic databases. In *IRCS Workshop*.

[5] Accessible at http://www.isocat.org/

[6] http://www.clarin.eu/cmdi

[7] http: //www.lat-mpi.eu/tools/

Daan Broeder, Marc Kemps-Snijders, Dieter Van Uytvanck, Menzo Windhouwer, Peter Withers, Peter Wittenburg, and Claus Zinn. 2010. A data category registry- and component-based metadata framework. In *Proceedings of the 7th conference on International Language Resources and Evaluation (LREC)*.

Cintia Carrió. 2009. *Mirada Generativa a la Lengua Mocoví (Familia Guaycurú)*. Ph.D. thesis, Universidad Nacional de Crdoba, Argentina.

Arienne Dwyer and Ulrike Mosel. 2001. Metadata Description Recommendations: Content, draft 15.03.01. Technical report.

Lucía Golluscio. 2003. Lenguas en peligro, pueblos en peligro en la Argentina. In Georg y Joachim Born Kremnitz, editor, *Coloquio Internacional sobre Lenguas, literaturas y sociedad en la Argentina. Diálogos sobre la investigación en Argentina, Uruguay y en países germanófonos*, pages 95 – 110.

Verónica María Grondona. 1998. *A Grammar of Mocoví*. Ph.D. thesis, University of Pittsburgh.

Cecilia Beatriz Gualdieri. 1998. *Mocovi (Guaycuru) Fonologia e morfossintaxe*. Ph.D. thesis, Universidade Estadual de Campinas, Brazil.

ISO 12620. 2009. Computer applications in terminology data categories specification of data categories and management of a data category registry for language resources. Technical report.

Heidi Johnson and Arienne Dwyer. 2002. Customizing the IMDI metadata schema for endangered languages. In *3rd International Conference on Language Resources and Evaluation – Workshop on Resources and Tools in Field Linguistics*, pages 51 – 55.

Christopher Moseley, editor. 2010. *Atlas of the Worlds Languages in Danger*. UNESCO Publishing, 3rd edition. Online version: http://www.unesco.org/culture/en/endangeredlanguages/atlas.

MPI ISLE Team. 2001. IMDI Metadata Elements for Session Descriptions, version 2.4. Technical report.

MPI ISLE Team. 2003. IMDI Metadata Elements for Session Descriptions, version 3.0.3. Technical report.

Gary Simons and Steven Bird. 2003. The Open Language Archives Community: An infrastructure for distributed archiving of language resources. *Literary and Linguistic Computing*, 18:117 – 128.

John M. Sinclair and John Ball. 1996. Preliminary recommendations on text typology. Technical Report EAG-TCWG-TTYP/P, EAGLES.

T Váradi, S. Krauwer, P. Wittenburg, M. Wynne, and K. Koskenniemi. 2008. CLARIN: Common language resources and technology infrastructure. In *Proceedings of the 6th International Conference on Language Resources and Evaluation, LREC*.

M. Windhouwer and S. Wright. 2012. Linking to linguistic data categories in isocat. In *Linked Data in Linguistics - Representing and Connecting Language Data and Language Metadata (LDL 2012)*, pages 99 – 107. Springer-Verlag.

Publishing and Exploiting Vocabularies using the OpenSKOS Repository Service

Hennie Brugman [1], Mark Lindeman [2]
1. Meertens Institute
P.O. Box 94264, 1090 GG Amsterdam
E-mail: hennie.brugman@meertens.knaw.nl

2. Pictura Database Publishing
De Hoefsmid 11, 1851 PZ Heiloo
E-mail: M.Lindeman@pictura-dp.nl

Abstract

Many vocabularies in eHumanities and eCulture domains can, and increasingly often are converted to SKOS. The OpenSKOS web service platform provides easy ways to publish, upload, update, harvest, query and distribute SKOS vocabulary data. This has benefits for vocabulary builders, vocabulary consumers and builders of tools that exploit vocabularies. In this paper we present and discuss the OpenSKOS system and a number of its applications, including an application from the domain of linguistic resources and tools.

1 Introduction

The application and relevance of vocabularies for the description of cultural heritage and scientific collections is making a comeback. One of the motivators for this comeback is the emergence of Semantic Web and Linked Open Data. There is much interest in application of data and text mining techniques to disclose collections, but it turns out that many of these techniques also build on vocabulary information.
Recent years have seen forms of standardization for vocabulary data that are consistent with Semantic Web and Linked Data principles. Well known is the W3C SKOS (Simple Knowledge Organization System) recommendation (Miles, 2009). More and more vocabularies, especially in the cultural heritage domain are mapped and converted to the RDF-based SKOS format and data model.

In 2004 the Dutch CATCH research programme started. CATCH (Continuous Access To Cultural Heritage) consists of a number of projects that do research regarding computer science and humanities research questions that are driven by cases from daily practice at large Dutch cultural heritage institutions. CATCHPlus is a partner project of CATCH that does valorization: it has the assignment to turn research prototype systems and demonstrators from the CATCH programme into tools and software services that can actually be used by cultural heritage professionals and users.

CATCHPlus tools and services should, where possible, contribute to the emerging infrastructure for digital cultural heritage. One aspect that many of the tools and services in CATCHPlus have in common is that they deal with or exploit vocabulary data. Therefore CATCHPlus stimulated standardisation of vocabulary formats to SKOS and also started work on a shared service that adds some standardisation to the way these SKOS vocabularies are made available and accessed: OpenSKOS[1], a web service based vocabulary publication platform.

Section 2 will describe requirements and motivations for OpenSKOS. Section 3 will describe the OpenSKOS architecture and components in detail, section 4 will position OpenSKOS in comparison with the ISOcat terminology service and with Linked Open Data. Section 5 describes current and future applications and clients of the OpenSKOS service. We will end the paper with an evaluation and conclusions (section 6).

2 Problem statement

The importance of and interest in vocabulary resources is increasing. These resources are typically created in specialized vocabulary maintenance tools or in modules of collection management systems. They are made available online using interactive web applications or in the form of Linked Data at the most. Over the last couple of years some standardization with respect to format has taken place: many vocabularies are currently mapped to SKOS.
However, it is often still a cumbersome process to locate suitable vocabularies and to (re)use them for one's own resource description tasks, in one's own tool environment. This is especially true when a vocabulary is well maintained and therefore frequently updated. To use a concept that is newly introduced by the vocabulary editors typically requires export and upload/download of the full vocabulary, proprietary format conversions and software adaptation or configuration steps by the producers of several collection management systems.

[1] http://openskos.org

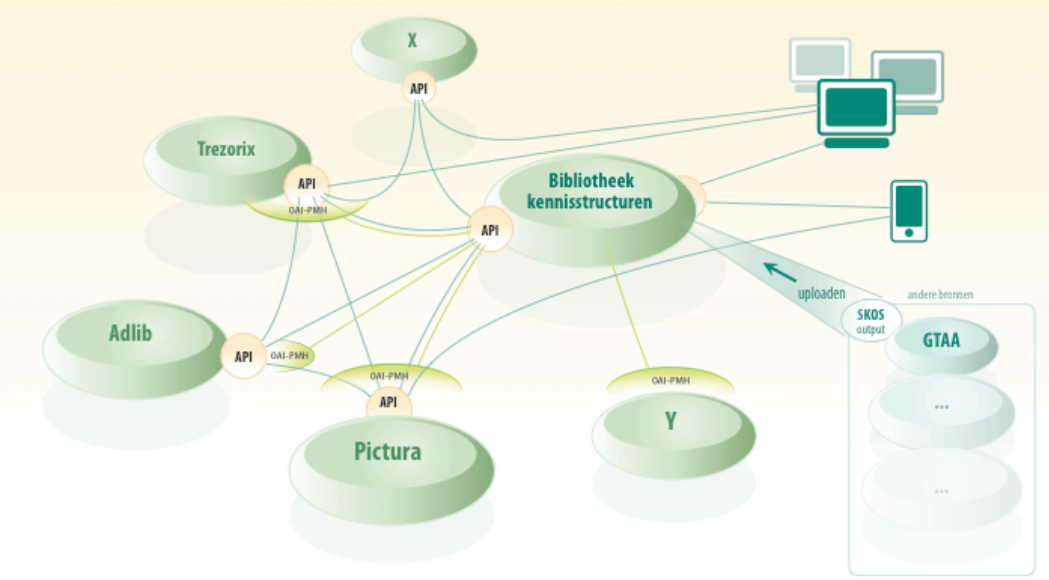


Figure 1: OpenSKOS architecture

Some web service based solutions also provide access to vocabularies as data, but these often have other shortcomings. They do not support periodic and/or incremental updates, they do not support the full underlying data model of the vocabularies (e.g. they are not able to handle relations between concepts), or they are optimized for other use cases than providing concepts for resource description (e.g. they have no proper support for handling long lists of entity names).

The Linked Data movement also imposes additional requirements on vocabulary services: concepts should be identified with stable, resolvable http URIs. Content negotiation is a desirable feature for a Vocabulary service.

Finally, web based (Open) Annotation (Sanderson, 2011) is a new development, that also imposes linked data type of requirements on Vocabulary services. It should be possible to annotate a web resource with URIs of concepts in online repositories.

3 The OpenSKOS service

OpenSKOS is a web service based approach to publication, management and use of vocabulary data that can be mapped to SKOS. The name is not meant to suggest that SKOS is not open; it refers to 'infrastructure and services to provide *open access* to SKOS data'. The main objective is to make it easy for vocabulary producers to publish their vocabularies and updates of it in such a way, that they become available to vocabulary users automatically and instantaneously, and independent of the specific software tools of these vocabulary users.

3.1 Architecture

Figure 1 shows the OpenSKOS architecture, which is a peer-to-peer architecture. Several sites can run instances of the freely available OpenSKOS repository software. Peers with a more centralized role are not technically necessary, although not excluded. Each site can be accessed by means of a RESTful API (Richardson, 2007) that supports a range of queries to retrieve or update SKOS vocabulary information in the repository. Having local copies of vocabularies in a repository instance implies that these can be searched efficiently on basis of locally created indexes.

Different OpenSKOS sites can exchange local copies of vocabularies using the OAI-PMH[2] protocol: OpenSKOS has built-in OAI-PMH data providers and harvesters. New vocabularies can be imported into the system in several ways: they can be harvested from another instance of OpenSKOS, they can be harvested from external OAI data providers, they can be included by implementation of the OpenSKOS API by other parties, or they can be uploaded using a built-in upload module.

Finally, OpenSKOS software contains a Dashboard to support a number of management tasks on each instance of OpenSKOS. This Dashboard can only be accessed after successful authentication.

3.2 The OpenSKOS RESTful API

The system's API is defined in a collaborative effort between the CATCHPlus project office, three major commercial tool providers for the Dutch Cultural Heritage sector (Adlib Systems, Pictura Database Publishing and Trezorix) and the Rijksdienst voor het Cultureel Erfgoed (Dutch department for cultural heritage). The specification is based on previous experiences and known use cases of all partners. The W3C SKOS recommendation was taken as the underlying data model.

2.3.1 Functional scope of the API

To start with, the API can resolve (skos) Concepts and ConceptSchemes ('vocabularies') by URI in a number

[2]http://www.openarchives.org/OAI/openarchivesprotocol.html

of representation formats (JSON, RDF/XML, html). This implies that Linked Data access is a sub set of the web services functional scope. The resolve API has query parameters that allow filtering on language used, and specification of what information is/is not included in the result.
Second, the API has 'find' functionality for Concepts and ConceptSchemes. It supports a query parameter 'q' that takes queries according to the Apache Lucene Query Parser Syntax as values. Searching is possible over all SKOS based fields and over Dublin Core (dcterms) fields, if those are present. The result of a 'find' query is a list of Concepts (represented in the same way as for the concept resolve) and a diagnostics block, for example with number of results that match and number of results on page. Paging and sorting of results is supported.
A specialization of the /find API is the OpenSKOS 'auto complete' function, meant for interactive searching for matching concept labels starting with some characters. The primary use case for this auto complete is supporting resource description tasks in some collection or metadata management system.
The OpenSKOS API namespace contains *Collections* and *Institutions* that are not part of the SKOS model but added for practical reasons. Collections can group a number of conceptschemes together that constitute one resource from an organisational/data management perspective. For example, the thesaurus of the Netherlands Institute for Sound and Vision (archive of the Dutch public broadcast corporations) consists of six sub thesauri but is maintained and published as a whole. *Institutions* are added to make information available on the vocabulary publishers themselves, and to associate authorized vocabulary managers with.
The API explicitly covers SKOS properties that are used to define mappings between concepts, also mappings between concepts belonging to different conceptschemes. The OpenSKOS repository is also a place where mappings across vocabularies can be maintained and exploited.
The OpenSKOS API not only supports HTTP GET operations on the resources described before, but for many of those resources it also supports PUT, POST and DELETE operations. It is therefore possible to perform vocabulary maintenance tasks directly on the repository using the API. For REST examples see openskos.org.

The CATCHPlus project office and Pictura together have built an OpenSKOS implementation that includes an implementation of the API. This implementation is internally based on Apache SOLR. It also includes implementations of other OpenSKOS components: a Dashboard, OAI harvester and data provider (including a job scheduler) and upload module for SKOS uploads.

3.3 OAI-PMH and upload modules

There are in principle three ways to enter vocabulary data into the OpenSKOS repository: create it from scratch using the APIs PUT and POST operations, upload it using the built-in upload module or harvest it using the built-in OAI-PMH harvester and job scheduler.
OpenSKOS repositories are able to harvest vocabulary data or to provide harvesting access to specific vocabularies from other OpenSKOS instances. This harvesting can be done periodically and incrementally. OpenSKOS includes a job scheduler that can be configured to run periodic harvesting jobs.
Reasons to harvest vocabularies to one's own OpenSKOS instance are: it can be used for an initial full download, and it subsequently keeps vocabulary information up to date. Another reason could be to maintain a copy for local indexing and searching. A reason to provide access for harvesting by others: most efficient, flexible and controlled way to allow downloads of potentially large data sets (http could lead to long download times and time outs).

OpenSKOS has a built-in upload module that can only be operated by authorized users using the system's Dashboard.

3.4 Dashboard

For management tasks by authorized users the system has an interactive Dashboard component. After successful authentication a user can access several panes. The "Manage institution" pane allows the user to enter and modify institution metadata, like name, contact information and website. "Manage collections" presents the user with an overview of available collections, and allows the user to create new ones. These collections are associated with the users' Institution. Each collection has associated metadata, like title, description, links to websites, and license information (preferably Open Database licences, of course). Also, for each collection it is possible to specify whether it is harvestable by other OpenSKOS instances and if the associated data is imported by upload or by OAI-PMH harvesting. In the latter case the OAI data providers' base URL can be specified.
Collections are the unit of 'upload' or 'maintenance', and can consist of data for several SKOS ConceptSchemes.
The "Manage users" pane gives an overview of existing users, their email addresses, their access rights (do they have writing access using the API, using the Dashboard or both) and their API key. It also supports creation of new users.
Finally, the "Manage jobs" pane gives an overview of scheduled and finished harvest and upload jobs.

Institution and collection info can not only be inspected and modified using the Dashboard; it is also available to anyone for inspection using the relevant API calls, represented as RDF/XML, JSON or html. The html representation makes it possible to browse over the repository content starting at an Institution, via its Collections and ConceptSchemes to representations of the Concepts themselves.

Figure 2: Snippets of user interfaces of OpenSKOS clients

3.5 Authentication and authorization

Since the main objective of OpenSKOS is to be 'open' we chose not to support authenticated 'read' access to the repository's content, all SKOS information is world-readable. In fact, we actively promote the use of open license forms like the Open Database license by offering this as an optional license form to creators of new vocabulary Collections.

For modification operations (create, update, delete) we support two levels of authorization: access using an API key, and access via the system's Dashboard. At API level modifications to Concepts and ConceptSchemes can be made. Modifications to Institutions, Collections and users all require authentication via the Dashboard.

Users can have either or both of the authorization levels.

4 Related work

OpenSKOS can in terms of genericity be positioned somewhere between a domain- and community-specific terminology repository solution as ISOcat and the generic and general purpose Linked Open Data approach.

ISOcat (Windhouwer, 2010) is an ISO TC 37 registry for Data Categories. These Data Categories are mainly intended for linguistic concepts. ISOcat by definition does not support relations between concepts and relies on separate relation registries for this. Main use cases for ISOcat are registration of concepts and providing a platform for standardisation of linguistic terminology. ISOcat therefore is not the optimal place to maintain or serve large lists of term labels. SKOS and OpenSKOS are less restrictive: they are not restricted to a certain domain, support relations between concepts and support a wider range of use cases. Representing and serving long term lists is normal practice. ISOcat has a RESTful web service that can be and actually is used to feed the OpenSKOS service (see chapter 5.3 about CLAVAS).

Linked Data on the other hand is even more generic: it is not restricted to vocabulary type of data, as SKOS and OpenSKOS are. It can represent any mix of data, metadata and concepts and links between those. The drawback is, that considered as a protocol it is much simpler than the ISOcat and OpenSKOS RESTful APIs. Linked Data access by means of resolvable and stable http URIs and support for content negotiation is a subset of the functionality of the OpenSKOS API.

5 Applications

The OpenSKOS repository service and architecture is the outcome of a process of several years, during which prototypes and experimental tools were built and tested. Over these years several academic, commercial and cultural heritage partners got involved. This section describes a bit of OpenSKOS' history and context, before it discusses current and planned applications of the system.

5.1 OpenSKOS history and context

Previous work in the CATCH research programme and in CATCHPlus resulted in a demonstrator and in a first version of the Vocabulary Repository service. This first version was implemented as a 'thin' Java layer on top of an RDF store (Openlink Virtuoso). Although stable and performant (e.g. online auto completion over the web works fine), this implementation makes a large demand on memory, and we had doubts about its scalability. Furthermore, its API is at best "REST-like", it has limited and incomplete support for modification operations, and there are no provisions for web upload, OAI-PMH harvesting or user authentication.

Nevertheless, this system was and is actually used for

daily collection description work by the triangle Netherlands Institute for Sound and Vision, National Archive, and Pictura and was found an elegant and interesting solution. (S&V is the thesaurus provider, National Archive does collection description with S&V terms using Pictura's Memorix tool).
This relative success led to intensive discussions between CATCHPlus, RCE, Adlib, Pictura, Trezorix that led to refinement of the OpenSKOS concept and a proper RESTful API specification that built on the knowledge, use cases and experience of all partners. Subsequently, the API, infrastructure and Dashboard were implemented by Pictura and CATCHPlus.
Due to this long history with frequent discussions, presentations and experiments in the Dutch cultural heritage context, there is now serious interest to participate. Several large Dutch CH institutions are currently involved in some way.
Recently CLARIN-NL also started a project to apply OpenSKOS for linguistic vocabulary data (see 5.3).

5.2 OpenSKOS clients

Some API clients already exist. A generic browse and search web application was built for CATCHPlus (by Q42, see figure 2). All access to vocabulary data used and shown in this web application is exclusively retrieved via API calls.
Pictura's collection management application Memorix is used on daily basis by National Archive for description of their online image collection. Memorix also functions as an OpenSKOS client.
Sound and Vision has started development of a web based thesaurus management application on top of the OpenSKOS editing APIs to manage their GTAA thesaurus.

5.3 Application by CLARIN(-NL): CLAVAS

Within the Dutch CLARIN context there turned out to be a need for an additional effort to promote uniform terminology. While ISOcat focuses on standardisation of sets of concepts (Datcats) there is an additional need for support of relative simple, but long lists of terms, especially in the context of metadata creation and editing. Therefore CLARIN-NL started the CLAVAS project, which is an application of OpenSKOS. The CLARIN project makes several contributions to OpenSKOS, and CLARIN in turn can benefit from additional efforts done for OpenSKOS. These contributions are three additional SKOS-ified resources (ISO 639-3 language codes, access to public parts of ISOcat through the OpenSKOS API and architecture, and a vocabulary of organisation names relevant for the international domain of linguistic tools and resources. It is explored if this list can be bootstrapped by existing metadata descriptions containing organisation information.
An additional CLAVAS component is a simple web application that supports basic vocabulary curation tasks on simple concept lists.
The CLAVAS project is done by the Meertens Institute, which also hosts the central CATCHPlus project office.

6 Evaluation and conclusions

The OpenSKOS service can be consulted in many use cases where vocabularies play a role. Some examples :

- When defining a metadata component, as for example in the CMDI framework it is possible to associate a metadata field with a ConceptScheme in OpenSKOS simply by associating the field with the URI of the ConceptScheme.
- When creating metadata in a metadata editor values for fields can be selected using the auto complete API of OpenSKOS.
- The service can be exploited in several browse in search scenarios, for example for faceted browsing or for query formulation.
- When Concepts have labels in multiple languages, localized views of metadata records can be displayed.

OpenSKOS supports all SKOS relations between Concepts, both within vocabularies and across vocabularies. SKOS and OpenSKOS also support enrichment of vocabulary concepts with links to other resources on the web (more specifically, in the Linked Data cloud).

Probably the greatest benefit of OpenSKOS is that it provides an easy publication platform for all resources that can be 'SKOS-ified'. This has advantages for vocabulary publishers, for vocabulary consumers and for builders of tools that create or exploit vocabularies.

Advantages for vocabulary publishers are:
- Offering vocabularies to others is as easy as a simple upload action.
- It is easy to use your own vocabulary in the tools of others, if these tools use OpenSKOS.
- Vocabularies can easily and frequently be updated without involvement of others.
- It is easy to link your own vocabulary to vocabularies of others.

Advantages for vocabulary consumers :
- Easy discovery, evaluation and reuse of existing vocabularies (and therefore a reduced need to construct your own).
- New browse and search possibilities.
- Always up to date versions of vocabularies are available

Advantages for tool builders :
- No more periodic updates, no more specific adaptations for specific vocabularies.
- Can benefit from efforts of other tool builders and of vocabulary publishers.

- Can use OpenSKOS API functionality for a range of use cases.

OpenSKOS is available as open source from GitHub, and as installable package. It is implemented on basis Apache SOLR technology in a scalable way. A community of OpenSKOS users is already emerging.

7 Acknowledgements

We would like to thank all people and institutions that contributed to the realization of OpenSKOS by investing time, energy and/or funding. We especially would like to mention RCE, Adlib Systems and Trezorix for their contributions to the definition of the OpenSKOS architecture and API, and the funders of CATCHPlus: the Netherlands Organisation for Scientific Research (NWO), and the Dutch ministeries for Education (OCW) and Economic Affairs.

8 References

Miles, A., Bechhofer, S. (2009). SKOS Simple Knowledge Organisation System Reference. *W3C Recommendation* 18 August 2009.

Richardson, L., Ruby, S. (2007). *RESTful Web Services: Web services for the real world.* O'Reilly Media. May 2007.

Sanderson, R., Van De Sompel, H. (2011). Open Annotation. Beta Data Model Guide. http://www.openannotation.org/spec/. 10 August 2011.

Windhouwer, M.A., Wright, S.E., Kemps-Snijders, M. Referencing ISOcat data categories. *In proceedings of the LRT standards workshop* (LREC 2010), Malta, May 18, 2010

Metadata Management with Arbil

Peter Withers

Max Planck Institute for Psycholinguistics
Wundtlaan 1, 6525 XD, Nijmegen, The Netherlands
peter.withers@mpi.nl

Abstract

Arbil is an application for creating and managing metadata for research data such as audio, video or textual data. The metadata is displayed in tables and trees, which allow an overview of the metadata and the ability to populate and update many metadata sections in bulk. A number of metadata formats are supported and Arbil has been designed as a local application so that it can also be used offline, for instance in remote field sites. The user can view and edit the metadata in tables in the order that the information is available, if the metadata does not comply with the requirements the user will be warned but will not be prevented from entering it in the meantime. It is hoped that the features of the application will lead towards the recording of metadata at an earlier stage resulting in greater detail and better quality of that metadata. If this improvement in workflow is achieved then the metadata will be entered sooner and reassessed during the research process, which will greatly improve the quality of that metadata.

Keywords: Metadata, Editor, Resources, Corpus, Linguistics, IMDI, Clarin, XML, Schema, Archiving

1. Introduction

Arbil is an application designed to create and manage metadata for research data and to arrange this data into a structure appropriate for archiving. The metadata is displayed in tables and trees, which allow an overview of the metadata and the ability to populate and update many metadata sections in bulk. A number of metadata formats are supported and Arbil has been designed as a local application so that it can also be used offline, for instance in remote field sites. The metadata can be entered in any order or at any stage during the process and then exported with the data files for use in the archive or as a backup of the current work. Once the metadata and its data are ready for archiving and an Internet connection is available it can be exported from Arbil and in the case of IMDI it can then be transferred to the main archive via LAMUS (Broeder et al., 2006) (archive management and upload system). In this paper we discuss why the use of a dedicated metadata editor is of benefit and why Arbil was written, we also discuss how this application can be used to create and edit metadata.

2. Why use a metadata editor

There are many reasons to provide metadata, yet if the process of creating and managing that metadata is difficult, the quality and completeness will suffer. It is a reasonable assumption that from the point of view of the researcher collecting the primary data, that this data is considered valuable and worth preserving with metadata so that it can be subsequently found, understood and referenced in future publications. In many cases there can also be an obligation to provide to the speakers of the language being researched and their descendants access to the collected material, and this would not be complete without metadata. From the point of view of the archivist, the task is not just to preserve the data, but also to organise the material in a structured way such that it can be identified, searched for and accessed when required. For these reasons it is important that we provide a tool that makes the process of creating metadata simple and transparent, reducing repetitive tasks whenever possible.

A metadata tool must have at very least all the functions that a basic text editor provides, such as copy, paste, find and undo. A simple text editor at first glance has the advantage of being simple to use and very flexible. However, it does not enforce any structure on to the metadata being edited. Conversely if a structured metadata editor is confusing, or not reliable, or does not have the basic set of functionality to which the user is accustomed, then the users may end up resorting to an unstructured tool instead. Which can lead to inconsistency of the metadata produced, hence it is crucial that an easy to use and reliable tool is available for the task. Arbil is designed to fill this need by providing an intuitive modern interface in which to create and manage metadata for the data files being archived. Features like drag and drop are used extensively both for constructing a hierarchical corpus tree structure and for adding nodes to tables for viewing and editing. Bulk editing of metadata can be done for instance via copy and paste, which allows a string of text to be pasted into multiple fields of multiple rows, or to paste multiple fields into the matching fields of multiple rows. Whenever a field is edited the changes are stored in an undo / redo buffer which allows all the changes made since the last save to be undone or redone. Arbil supports both IMDI (Broeder and Wittenburg, 2006) and Clarin (Váradi et al., 2008) formats and through the use of XML schema files additional formats can potentially be supported. Both the IMDI and Clarin formats allow the metadata to be arranged into corpus tree structures. Arbil displays trees of metadata in its user interface 'remote corpus', 'local corpus', 'favourites' and the directories containing the data files. Snippets of frequently used sections of metadata can be collected in the favourites and then easily utilised in the process of constructing new metadata, greatly reducing the amount of repetitive data entry.

3. Why Arbil was created

Arbil came into existence as a result of a meeting between members of the DOBES (Wittenburg et al., 2002) commu-

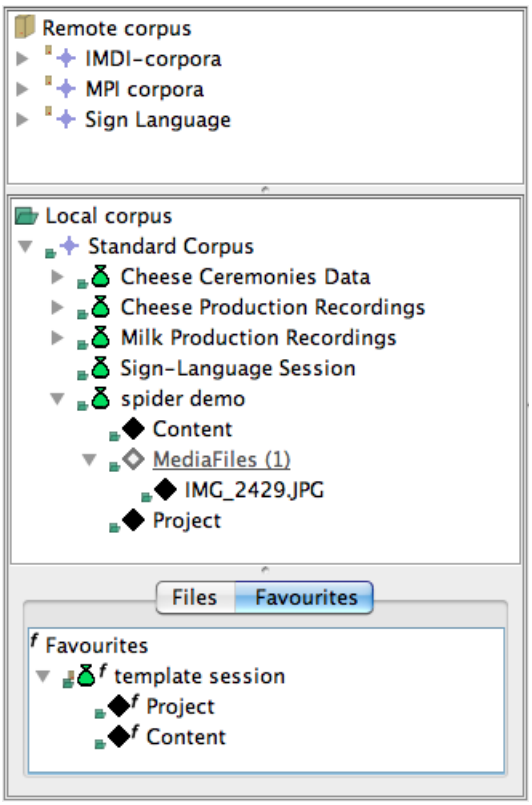


Figure 1: Tree display of the metadata.

nity and members of the MPI developers team where the need was recognised for an offline metadata editor with tabular functionality, something the previous IMDI Editor lacked. A prototype application called Linorg was developed by the author, and based on the feedback given during discussion with members of the DOBES community; from that prototype he subsequently developed what is now Arbil. The development of Arbil has also benefited from discussions with many linguists at the MPI and the experience gained from the previous metadata applications developed at the MPI. Arbil contains many features in order to fulfil the wide-ranging needs expressed while maintaining the functionality of the previous IMDI Editor tool. While Arbil is primarily designed to create metadata, it also has functions to help organise the collected material and create a local well-organised corpus before it is archived. These functions include the ability to search for and compare metadata, and to search for and open the data files in the relevant application. Arbil continues to be actively developed to extend these features further.

Often researchers are working in a field site where there is limited or no Internet connection. For this reason it is important that a tool such as Arbil is able to work correctly when offline. Arbil achieves this by keeping a local copy of all the required files such as controlled vocabularies and will update them if required from the server when an Internet connection is available. One of the most network intensive activities is browsing the remote archive; clearly this will not be possible without a network. However, for this reason, Arbil has the ability to mirror branches from the main archive so that they can still be referred to offline and in the field.

Figure 2: Metadata node view.

4. Entering metadata

Some metadata editors, for instance the IMDI Editor, requires that the user enters the metadata in a predefined order making it impossible to move forward until a value is entered. While this is useful when the data to be entered is minimal and or the required information is completely available at the time of entry, in reality this is likely to result in a situation when the data is not fully available and the user is forced to either fragment the metadata by recording some of it outside the system or by entering dummy metadata with the good intention of fixing it later. Both of these workarounds can lead to inconsistency of the metadata recorded. This issue is addressed in Arbil by allowing the metadata fields to be completed when the information is available and to simply warn a user when something is missing or is not in the required format. At the point of exporting the metadata, all files are checked for inconsistencies and warnings are given if there are issues. Only at the point of pushing the metadata into the archive will the user be blocked if they have not correctly completed all the required fields.

In Arbil the metadata is viewed in tables, which can contain a single node of metadata as a list of fields, or many different nodes, each with its fields as a separate row in the table, or all the nodes of one metadata file inline. This tabular view of the data allows multiple metadata nodes to be quickly viewed across the rows of the table. The metadata can be can edited in any table in which it is viewed, for instance in the search results table or a table of individually selected metdata nodes. These manually constructed tables can be assembled by selecting metadata nodes of interest and drag-and-dropping them into a table.

In many metadata sets the number of fields required to describe the data and its context can be extensive; this can make it difficult for a user to see their relevant information at a glance. In order to accommodate this the table columns in Arbil are customisable, so that only those relevant to a particular user need be displayed. These selected sets of columns viewed in a table can be saved and then easily applied to any table, and if required, a default combination of columns can be selected so that new tables show only the required information. In order to further visualise the metadata in the table, the columns can be resized or sorted

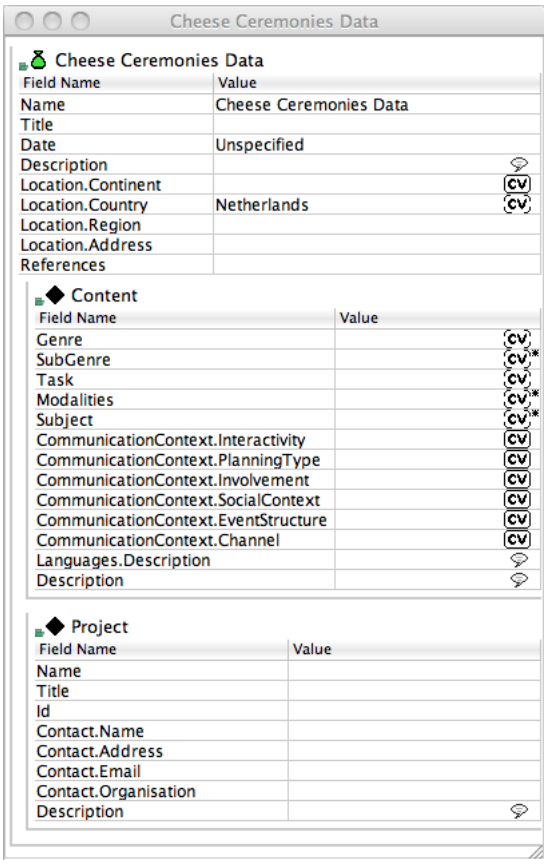


Figure 3: Metadata file view.

on any column and reordered. Rows can easily be added and then dragged from one table to another, and the cells can be highlighted based on matching text. Because much of the metadata is hierarchical with multiple sub nodes in a single file, this cannot always be displayed in a single row of a table. For instance in the IMDI metadata format actors, written resources and media files are sub nodes within a single session file. However, in this case additional columns can be displayed where the name and icons of the sub elements are displayed in a single cell of the row.

When there are many fields to fill in for a given metadata set, it is important to clearly see what each fields is intended for and which fields are of a higher priority than others. For this reason a description can be provided (in the metadata format specification) for each field explaining the intended usage and this is displayed in the tooltip of that field. When a field is set as mandatory it will be given a colour highlight if the metadata is not filled in. Likewise in the case of fields requiring specific formatting, such as date fields, the text will be highlighted when the formatting is incorrect.

Creating and editing of the metadata is only one part of a much larger workflow, hence it is necessary to both import and export this metadata in Arbil. Any valid IMDI or CMDI metadata file can be imported into Arbil. The data files that the metadata describes can optionally be imported at the same time, for instance when migrating or merging from one computer to another. If a backup is required then all the metadata and data files within Arbil can be exported into a self-contained directory, for instance onto a USB hard

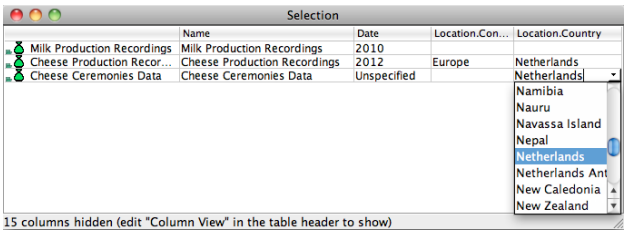


Figure 4: Multiple metadata files view.

drive. In the case of IMDI the resulting export can then be uploaded into LAMUS and from there inserted in the archive. During both the import and export processes, all of the metadata is checked for errors and a list of warnings given if any are found. The metadata in Arbil can be exported in other formats via XSLT transforms and one such transform is provided with the application that converts from IMDI into HTML. In addition, when any metadata is displayed in a table the contents can be copied and then pasted into a text editor or into spreadsheet.

5. Conclusion

Arbil has been developed with a strong focus on workflow and usability. It allows the user to view and edit the metadata in tables without mandating any particular order of metadata entry while warning if the metadata does not comply with the requirements. It is hoped that the features of the application will lead towards the recording of metadata at an earlier stage resulting in greater detail and better quality of that metadata. It is also hoped that this metadata will prove useful for the linguists during the process of their research. Creating metadata at the time the data is collected can assist workflow by helping to keep track of the collected data files. By providing a way to organise these data files and utilise the metadata for searching the collected data and to backup the current data with its metadata, it is hoped that Arbil will assist the workflow of the researcher. If this improvement in workflow is achieved then the metadata will be entered sooner and reassessed during the research process, which will greatly improve the quality of that metadata. Hence, if the chore of entering of metadata at the end of a project is replaced by useful metadata throughout the life of the project it is likely to be of benefit to the process as a whole.

6. References

D. Broeder and P. Wittenburg. 2006. The IMDI metadata framework, its current application and future direction. *International Journal of Metadata, Semantics and Ontologies*, 1(2):119–132.

D. Broeder, A. Claus, F. Offenga, R. Skiba, P. Trilsbeek, and P. Wittenburg. 2006. LAMUS : the Language Archive Management and Upload System. In *Proceedings of the Fifth International Conference on Language Resources and Evaluation (LREC'06)*, pages 2291–2294, Genoa. European Language Resources Association (ELRA). www.lat-mpi.eu/papers/papers-2006/lamus-paper-final2.pdf.

T. Váradi, S. Krauwer, P. Wittenburg, M. Wynne, and K. Koskenniemi. 2008. Clarin: Common language resources and technology infrastructure. In *Proceedings of the Sixth International Conference on Language Resources and Evaluation (LREC'08)*, pages 1244–1248, Marrakech. European Language Resources Association (ELRA). www.lrec-conf.org/proceedings/lrec2008/pdf/317_paper.pdf.

P. Wittenburg, U. Mosel, and A. Dwyer. 2002. Methods of Language Documentation in the DOBES project. In *Proceedings of the Third International Conference on Language Resources and Evaluation (LREC'02)*, pages 36–42, Las Palmas. www.lrec-conf.org/proceedings/lrec2002/pdf/221.pdf.

Semantic Mapping – groundwork for query expansion and semantic search in LR metadata

Matej Ďurčo[1], **Daan Broeder**[2], **Menzo Windhouwer**[2]

[1] Institute for Corpus Linguistics and Text Technology (ICLTT), Vienna, Austria
[2] Max Planck Institute for Psycholinguistics, Nijmegen, The Netherlands
[1] matej.durco@assoc.oeaw.ac.at, [2] {daan.broeder, menzo.windhouwer}@mpi.nl

Abstract

This paper describes a module of the Component Metadata Infrastructure, that allows query expansion by providing mappings between search indexes. This enables semantic search, ultimately increasing the recall when searching in metadata collections. The module builds on the Data Category Registry and Component Metadata Framework that are part of CMDI.

1. Introduction

In recent years, multiple large-scale initiatives have been set out to combat the fragmented nature of the language resources landscape in general and the metadata interoperability problems in particular. A comprehensive architecture for harmonized handling of metadata – the Component Metadata Infrastructure (CMDI)[1] (Broeder et al., 2011) – is being implemented within the CLARIN project[2]. This service-oriented architecture consisting of a number of interacting software modules allows metadata creation and provision based on a flexible meta model, the *Component Metadata Framework*, that facilitates creation of customized metadata schemas – acknowledging that no one metadata schema can cover the large variety of language resources and usage scenarios – however at the same time equipped with well-defined methods to ground their semantic interpretation in a community-wide controlled vocabulary – the data category registry. (Kemps-Snijders et al., 2009; Broeder et al., 2010)

This approach of integrating prerequisites for semantic interoperability directly into the process of metadata creation differs from the traditional methods of schema matching that try to establish pairwise alignments between schemas only after they were created and published, algorithm-based or employing explicit manually defined crosswalks. (Rahm and Bernstein, 2001; Shvaiko and Euzenat, 2005; Shvaiko and Euzenat, 2008)

Consequently, the infrastructure also foresees a dedicated module, *Semantic Mapping*, that exploits this novel mechanism to deliver correspondences between different metadata schemas. In this article we describe this module.

2. Underlying infrastructure

As mentioned, the proposed module is part of CMDI and interacts with multiple modules of the infrastructure. Before we describe the interaction itself in chapter 4., we introduce in short these modules and the data they provide:

The *Data Category Registry* (DCR) is a central registry that enables the community to collectively define and maintain a set of relevant linguistic data categories. The resulting commonly agreed controlled vocabulary is the cornerstone for grounding the semantic interpretation within the CMD framework. The data model and the procedures of the DCR are defined by the ISO standard (ISO12620:2009, 2009), and is implemented in *ISOcat*[3].

The *Component Metadata Framework* (CMD) is built on top of the DCR and complements it. While the DCR defines the atomic concepts, within CMD the metadata schemas can be constructed out of reusable components - collections of metadata fields. The components can contain other components, and they can be reused in multiple profiles as long as each field "refers via a PID to exactly one data category in the ISO DCR, thus indicating unambiguously how the content of the field in a metadata description should be interpreted" (Broeder et al., 2010). This allows to trivially infer equivalencies between metadata fields in different CMD-based schemas. While the primary registry used in CMD is the ISOcat DCR, other authoritative sources for data categories ("trusted registries") are accepted, especially Dublin Core Metadata Initiative. (Powell et al., 2005)

The framework as described so far provides a sound mechanism for binding the semantic interpretation of the metadata descriptions. However there needs to be an additional means to capture information about relations between data categories. This information was deliberately not included in the DCR, because relations often depend on the context in which they are used, making global agreement unfeasible. CMDI proposes a separate module – the *Relation Registry* (RR) (Kemps-Snijders et al., 2008) –, where arbitrary relations between data categories can be stored and maintained. We expect that the RR should be under control of the metadata user whereas the DCR is under control of the metadata modeler.

There is a prototypical implementation of such a relation registry called *RELcat* being developed at MPI, Nijmegen. (Windhouwer, 2011; Schuurman and Windhouwer., 2011), that already hosts a few relation sets. There is no user interface to it yet, but it is accessible as a REST-webservice[4]. This implementation stores the individual relations as RDF-

[1] `http://www.clarin.eu/cmdi`
[2] `http://clarin.eu`
[3] `http://www.isocat.org/`
[4] sample relation set: `http://lux13.mpi.nl/relcat/rest/set/cmdi`

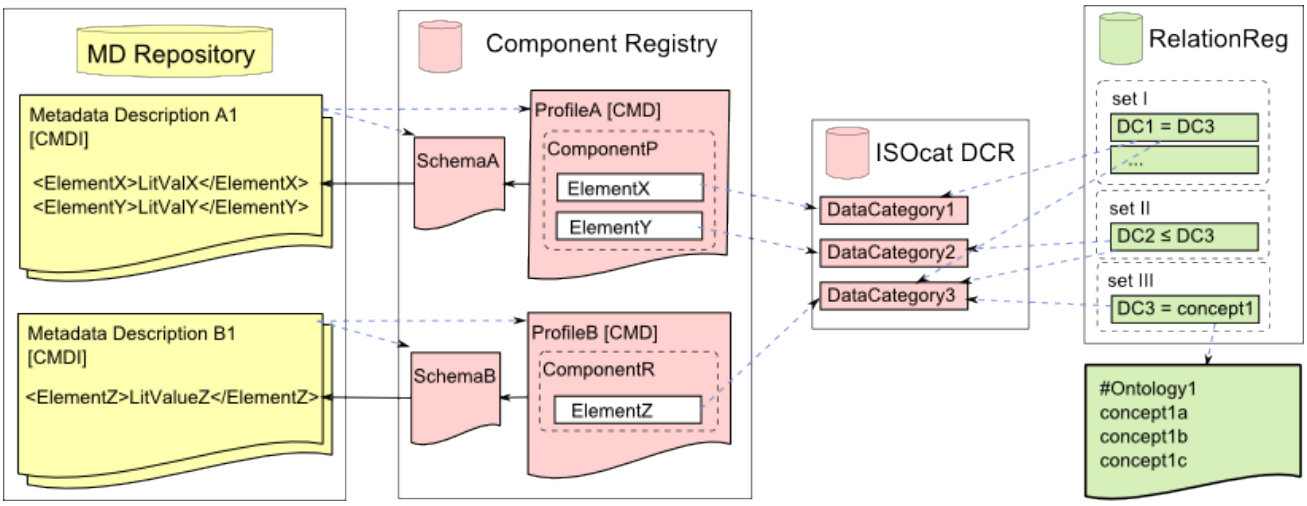


Figure 1: The diagram depicts the links between pieces of data in the individual registries that serve as basis for semantic mapping

triples

$$< subjectDatacat, relationPredicate, objectDatcat >$$

allowing typed relations, like equivalency (`rel:sameAs`) and subsumption (`rel:subClassOf`). The relations are grouped into relation sets that can be used independently.
And finally, there is the *Metadata Repository* aimed to collect all the harvested metadata descriptions from CLARIN centers, and *Metadata Service* that provides search access to this body of data. As such, Metadata Service is the primary application to use Semantic Mapping, to optionally expand user queries before issuing a search in the Metadata Repository. (Ďurčo and Olsson, 2011)

3. smcIndex

In this section we describe *smcIndex* – the data type for input and output of the proposed application. An smcIndex is a human-readable string adhering to a specific syntax, denoting some search index. The generic syntax is:

$$smcIndex ::= context\ contextSep\ conceptLabel$$

We distinguish two types of smcIndexes: (i) *dcrIndex* referring to data categories and (ii) *cmdIndex* denoting a specific "CMD-entity", i.e. a metadata field, component or whole profile defined within CMD. The *cmdIndex* can be interpreted as a XPath into the instances of CMD-profiles. In contrast to it, the *dcrIndexes* are generally not directly applicable on existing data, but can be understood as abstract indexes referring to well-defined concepts – the data categories – and for actual search they need to be resolved to the metadata fields they are referred by. In return one can expect to match more metadata fields from multiple profiles, all referring to the same data category.
These two types of smcIndex also follow different construction patterns:

$$\begin{aligned} smcIndex &::= dcrIndex \mid cmdIndex \\ dcrIndex &::= dcrID\ contextSep\ datcatLabel \end{aligned}$$

$$\begin{aligned} cmdIndex &::= profile \\ &\quad \mid [\, profile\ contextSep \,]\ dotPath \\ dotPath &::= [\, dotPath\ pathSep \,]\ elemName \\ contextSep &::= \texttt{`.`} \mid \texttt{`:`} \\ pathSep &::= \texttt{`.`} \\ dcrId &::= \texttt{`isocat`} \mid \texttt{`dc`} \end{aligned}$$

The grammar is based on the way indices are referenced in CQL-syntax[5] (`dc.title`) and on the dot-notation used in IMDI-browser[6] (`Session.Location.Country`).
dcrID is a shortcut referring to a data category registry similar to the namespace-mechanism in XML-documents. *datcatLabel* is the verbose Identifier- (e.g. `telephoneNumber`) or the Name-attribute (in any available translation, e.g. `numero di telefono@it`) of the data category. *profile* is the name of the profile. *dotPath* allows to address a leaf element (`Session.Actor.Role`), or any intermediary XML-element corresponding to a CMD-component (`Session.Actor`) within a metadata description. This enables the search in whole components, instead of having to list all elements of given component.
Generally, smcIndexes can be ambiguous, meaning they can refer to multiple concepts, or entities (CMD-elements). This is due to the fact that the names of the data categories, and CMD-entities are not guaranteed unique. The module will have to cope with this, by providing on demand the list of identifiers corresponding to a given smcIndex.

4. Function

In this section, we describe the actual task of the module – **mapping indexes to indexes**. The returned mappings can be used by other applications to expand or translate the original user query, to match elements in other schemas. [7]

[5] Context Query Language, `http://www.loc.gov/standards/sru/specs/cql.html`

[6] `http://www.lat-mpi.eu/tools/imdi`

[7] Though tightly related, mapping of terms and query expansion are to be seen as two separate functions.

Initialization

First there is an initialization phase, in which the application fetches the information from the source modules (cf. 2.). All profiles and components from the Component Registry are read and all the URIs to data categories are extracted to construct an inverted map of data categories:

datcatURI $\mapsto$ *profile.component.element[]*

The collected data categories are enriched with information from corresponding registries (DCRs), adding the verbose identifier, the description and available translations into other working languages.
Finally, relation sets defined in the Relation Registry are fetched and matched with the data categories in the map to create sets of semantically equivalent (or otherwise related) data categories.

Operation

In the operation mode, the application accepts any index (*smcIndex*, cf. 3.) and returns a list of corresponding indexes (or only the input index, if no correspondences were found):

smcIndex $\mapsto$ *smcIndex[ ]*

We can distinguish following levels for this function:
(1) *data category identity* – for the resolution only the basic data category map derived from Component Registry is employed. Accordingly, only indexes denoting CMD-elements (*cmdIndexes)* bound to a given data category are returned:

```
isocat.size ↦
   [teiHeader.extent,
    TextCorpusProfile.Number]
```

cmdIndex as input is also possible. It is translated to a corresponding data category, proceeding as above:

```
imdi-corpus.Name ↦
   (isocat.resourceName) ↦
   TextCorpusProfile.GeneralInfo.Name
```

(2) *relations between data categories* – employing also information from the Relation Registry, related (equivalent) data categories are retrieved and subsequently both the input and the related data categories resolved to cmdIndexes:

```
isocat.resourceTitle ↦
   (+ dc.title) ↦
   [imdi-corpus.Title,
    TextCorpusProfile.GeneralInfo.Title,
    teiHeader.titleStmt.title,
    teiHeader.monogr.title]
```

(3) *container data categories* – further expansions will be possible once the container data categories (Schuurman and Windhouwer., 2011) will be used. Currently only fields (leaf nodes) in metadata descriptions are linked to data categories. However, at times, there is a need to conceptually bind also the components, meaning that besides the "atomic" data category for `actorName`, there would be also a data category for the concept `Actor`. Having concept links also on components will require a compositional approach to the task of semantic mapping, resulting in:

```
Actor.Name ↦
    [Actor.Name, Actor.FullName,
     Person.Name, Person.FullName]
```

Extensions

A useful supplementary function of the module would be to provide a list of existing indexes. That would allow the search user-interface to equip the query-input with auto-completion. Also the application should deliver additional information about the indexes like description and a link to the definition of the underlying entity in the source registry.
Once there will be overlapping[8] user-defined relation sets in the Relation Registry an additional input parameter will be required to *explicitly restrict the selection of relation sets* to apply in the mapping function.
Also, use of *other than equivalency relations* will necessitate more complex logic in the query expansion and accordingly also more complex response of the SMC, either returning the relation types themselves as well or equip the list of indexes with some similarity ratio.

Usage example

The practical usage of this module is primarily within a service allowing search in LR metadata. The search application using the SMC module can provide the user with a (localized) list of data categories to search in. After the user issued a query request, each of the data categories used in the query is translated by the SMC module - based on the mappings - into a list of corresponding *cmdIndexes*, that can be used to search directly in the metadata records.
So, if we take the example mapping (2) from subsection Operation: the user presented with a list of isocat data categories to search in, selected `isocat.resourceTitle`, typed some search `term` and submitted the query. Internally, SMC module translates the `isocat.resourceTitle` data category in the query returning a list of *cmdIndexes*: `[imdi-corpus.Title, TextCorpusProfile.GeneralInfo.Title, teiHeader.titleStmt.title, teiHeader.monogr.title]`. Based on this list, the query `isocat.resourceTitle = search-term` is expanded[9] into a union, with the term being searched in every *cmdIndex* in the list:

```
imdi-corpus.Title = term OR
TCP.GeneralInfo.Title = term OR
teiHeader.titleStmt.title = term OR
teiHeader.monogr.title = term
```

[8] i.e. different relations may be defined for one data category in different relation sets

[9] The query expansion is not the task of SMC and a separate module should be responsible for this.

Thus, the result returned to the user contains metadata records with any of the above fields matching the search term.

Another possible usage scenario is a faceted browser using the mappings from the SMC module to map the data from different fields in different schemas to corresponding data categories, presenting to the user the data categories as facets to browse in the heterogeneous dataset.

Implementation

The core function of the SMC is implemented as a set of XSL-stylesheets, with auxiliary functionality (like caching or a wrapping web service) provided by a wrapping application implemented in Java. There is also a plan to provide an XQuery implementation. The SMC module is maintained in the CMDI code repository[10].

5. Summary and Outlook

In this article, we described a module of the Component Metadata Infrastructure performing semantic mapping on search indexes. This builds the base for query expansion to facilitate semantic search and enhance recall when querying the Metadata Repository.

The Semantic Mapping module is based on the DCR and CMD framework and is being developed as a separate service on the side of CLARIN Metadata Service, its primary consuming service, but shall be equally usable by other applications.

Further work is needed on more complex types of response (similarity ratio, relation types) and also on the interaction with Metadata Service to find the optimal way of providing the features of semantic mapping and query expansion as semantic search within the search user-interface.

And finally, grounding the metadata fields by linking them with the data categories, as globally identified semantically defined concepts, is also one step towards expressing the metadata records as Linked Open Data enabling the integration into the Semantic Web.

Acknowledgement

This work was produced as a part of the master thesis "SMC4LRT - Semantic Mapping Component for Language Resources" being written by Matej Ďurčo supervised by Prof. Andreas Rauber at Technical University Vienna.

6. References

Daan Broeder, Marc Kemps-Snijders, Dieter Van Uytvanck, Menzo Windhouwer, Peter Withers, Peter Wittenburg, and Claus Zinn. 2010. A data category registry- and component-based metadata framework. In Nicoletta Calzolari, Khalid Choukri, Bente Maegaard, Joseph Mariani, Jan Odjik, Stelios Piperidis, Mike Rosner, and Daniel Tapias, editors, *Proceedings of the Seventh conference on International Language Resources and Evaluation (LREC'10)*, Valletta, Malta, May. European Language Resources Association (ELRA).

Daan Broeder, Oliver Schonefeld, Thorsten Trippel, Dieter Van Uytvanck, and Andreas Witt. 2011. A pragmatic approach to XML interoperability - the Component Metadata Infrastructure (CMDI). In *Balisage: The Markup Conference 2011*, volume 7. citeulike:9861691.

ISO12620:2009. 2009. Computer applications in terminology data categories specification of data categories and management of a data category registry for language resources.

Marc Kemps-Snijders, Menzo Windhouwer, and Sue Ellen Wright. 2008. Putting data categories in their semantic context. In *Proceedings of the IEEE e-Humanities Workshop (e-Humanities)*, Indianapolis, Indiana, USA, December.

Marc Kemps-Snijders, Menzo Windhouwer, and Peter Wittenburg. 2009. Isocat: Remodeling metadata for language resources. In *International Journal of Metadata, Semantics and Ontologies (IJMSO)*, volume 4 (4), pages 261–276.

A. Powell, M. Nilsson, A. Naeve, and P. Johnston. 2005. DCMI Abstract Model. Technical report, March.

Erhard Rahm and Philip A. Bernstein. 2001. A survey of approaches to automatic schema matching. *VLDB JOURNAL*, 10:2001.

I. Schuurman and M.A. Windhouwer. 2011. Explicit semantics for enriched documents. what do isocat, relcat and schemacat have to offer? In *2nd Supporting Digital Humanities conference (SDH 2011), 17-18 November 2011, Copenhagen, Denmark*, Copenhagen, Denmark.

Pavel Shvaiko and Jerome Euzenat. 2005. A classification of schema-based matching approaches. Technical report.

Pavel Shvaiko and Jérôme Euzenat. 2008. Ten challenges for ontology matching. In Robert Meersman and Zahir Tari, editors, *On the Move to Meaningful Internet Systems: OTM 2008*, volume 5332 of *Lecture Notes in Computer Science*, pages 1164–1182. Springer Berlin / Heidelberg. 10.1007/978-3-540-88873-4_18.

Matej Ďurčo and Leif-Jöran Olsson. 2011. CMDRSB - CLARIN Metadata Repository/Service/Browser. In *Presentation at CMDI Workshop, Nijmegen*, Nijmegen, 01. MPI for Psycholinguistics.

Menzo Windhouwer. 2011. Relcat and friends. In *Presentation at CLARIN-NL ISOcat workshop*, Nijmegen, 05. MPI for Psycholinguistics.

[10] `http://svn.clarin.eu/SMC`

Applying CMDI in real life: the Meertens case

Martine de Bruin, Marc Kemps-Snijders, Jan Pieter Kunst, Maarten van der Peet, Rob Zeeman, Junte Zhang
Meertens Institute
Joan Muyskensweg 25
1096 CJ Amsterdam
The Netherlands
E-mail: martine.de.bruin@meertens.knaw.nl, marc.kemps.snijders@meertens.knaw.nl , janpieter.kunst@meertens.knaw.nl, maarten.van.der.peet@meertens.knaw.nl , rob.zeeman@meertens.knaw.nl, junte.zhang@meertens.knaw.nl

Abstract

The CMDI (Component Metadata Infrastructure) has gained widespread acceptance across multiple projects and organizations. To incorporate this approach many organizations need to adjust their organizational and technological structure to unlock the potential of the CMDI approach. The Meertens Institute has applied the CMDI approach to a large number of projects covering the full life cycle of the CMDI process, including metadata creation, ingest, publication and search processes. This paper covers our experiences with the CMDI approach and describes various aspects of our work process and projects in which the CMDI approach was adopted.

Keywords: CMDI, metadata, Meertens

1. Introduction

Metadata management remains a crucial aspect in the life cycle of (language) resources. Within several projects such as CLARIN and METASHARE, the Component Metadata Infrastructure (CMDI, ISO TC 37 SC 4 work item for ISO 24622) has been adopted as the basis for creating and publishing metadata descriptions. With the formal metadata models in place much work is devoted to creating the technical infrastructure capable of supporting the full metadata life cycle process and unlocking the envisaged potential of this approach. This involves creating the appropriate tools for user communities involved in the metadata creation process, metadata repository management environments, publication capabilities and search engines. The Meertens Institute made also some organizational changes to accommodate for the new approach resulting in a newly created position of 'Coordinator Research Collections'. This paper highlights several aspects of the life cycle management process from a both a practical and a technological perspective at the Meertens Institute to illustrate current experiences with the CMDI approach.

2. Metadata creation

The Component Metadata Infrastructure consists of a flexible approach towards creating metadata descriptions whilst maintaining semantic interoperability. Metadata profiles are constructed from reusable components in which data fields are linked to ISOcat[1] data categories. Data categories are defined as the 'result of the specification of a given data field(ISO 12620, 2009)'. Metadata profiles and components are stored in the CLARIN Component Registry[2] and serve as the basis for schema generation associated with each metadata document. A basic CMDI document thus consists of a generic section mainly describing relations to associated resources and metadata documents and a flexible section containing resource specific descriptions.

From a creation scenario perspective CMDI creation falls into two categories: bulk conversion of existing metadata records and user creation of new metadata descriptions. Both find a common basis in CMDI profile creation and linking of data categories. Tools that provide support for the two scenarios differ significantly. End user creation of metadata descriptions almost always requires a user interface for interaction with the end users. The Gekaapte Brieven project is an example of a project where a large group of volunteers are adding metadata and transcribing over 8,000 letters from the 17th and 18th century recovered from National Archives in London. These letters were originally captured during raids by English Capers and now provide a unique insight into the daily use of language during that time period. For the purpose of this project an integrated metadata editor and transcription environment has been created providing tailored support for the volunteers' tasks. While there are several CMDI editors available, ARBIL to name one, these are generally either too generic in nature or tailored for specific profile support (the NaLiDa editor from Tuebingen) to be adequately used by novice users. From a technical perspective creation of a custom editor proved to be straightforward and resulted in an easy to use, intuitive user interface for end users. The backend synchronization logic of preparing the batches for annotation by the end users and

[1] http://www.isocat.org

[2] http://catalog.clarin.eu/ds/ComponentRegistry/#

sending the resulting CMDI files back to the server is more complex in nature but can be reused for similar annotation tasks. As a result of this project more than 8,000 letters have been described in a raw CMDI format within a month. The transcription process is currently underway and will be completed before the end of this year. A small team of reviewers performs quality control checks on the metadata as part of the project. After completion, the metadata and transcriptions will be ingested into the archive and made available through the Meertens Repository.

The Meertens Institute also houses a large number of other digital collections for which metadata is generally available in formats other than CMDI. For each collection bulk conversion processes are employed that convert the currently available metadata information into CMDI format. Several custom scripts are employed for bulk creation processes that can be run without end user intervention and generally do not require any user interface. One example of a bulk conversion process was carried out in the context of the CLARIN-NL C-DSD. In this project the Meertens Liederenbank (Song Database) was largely converted into CMDI descriptions based on the metadata information gathered during previous projects. The level of granularity for the CMDI profiles was chosen such that it is possible to identify individual resources such as books scans and audio files, but also

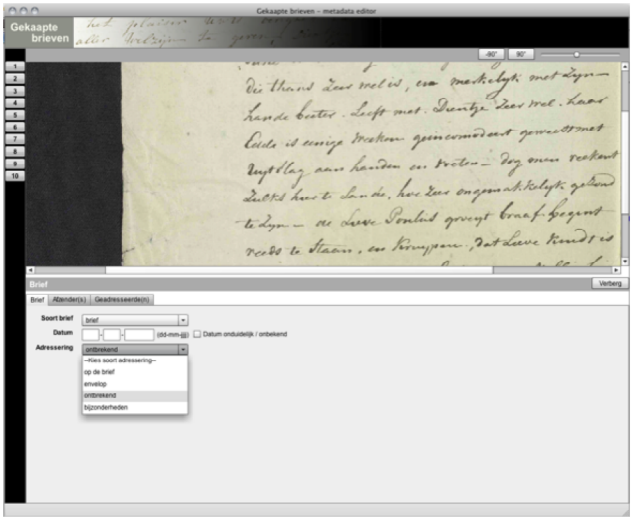


Figure 1: *Gekaapte Brieven* metadata and transcription environment

capture various relations between them such as song and singer. Each of the resources can thus be located and reused independently of the higher level constructs and largely reflects the organizational structure that was already present in the Liederenbank. This resulted in over 250,000 CMDI records being produced describing the database at various levels of granularity: songs, song scans, book scans, recordings, singers, symbolic music notations and related photographic materials. All metadata descriptions have been ingested and are made available through the Meertens Repository.

3. Ingest and publication process

The publication process of CMDI records requires a number of steps which all can be automated using the basic CMDI structure and number of standard available components and services. This step assumes that all records can at least be validated against the specified CMDI profiles and ISOcat references are available in the CMDI specifications. Content specific checks must have been performed before the ingest process takes place and usually involves manual intervention after completion of the metadata creation process. This type of quality assessment tasks lies within the realm of responsibilities of the newly created position of 'Coordinator Research Collections' who serves as a gatekeeper to the published collections to ensure that the descriptive metadata meets the envisaged quality levels. An outline of the automated part of the ingest process is depicted in the figure below.

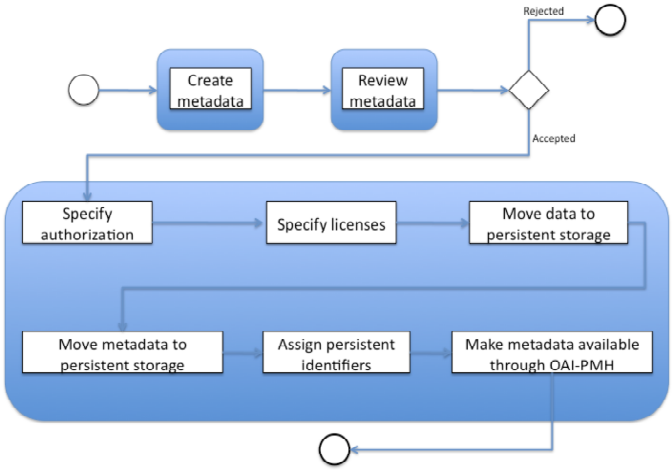


Figure 2: Outline of CMDI ingest and publication process

The automated ingest process involves recursive analysis of the CMDI records and assignment of persistent identifiers for both the resources and CMDI records. Persistent identifiers are obtained from SARA, the Dutch representative in the EPIC consortium using a REST web service. This provides a stable, highly available platform through which Handles with a Meertens specific handle prefix (10744) may be created and managed. The service platform provided by SARA is expected to find support within both the Dutch scientific as well as the Cultural Heritage community. As a next step in the ingest process all CMDI records are transformed to yield a DCMI (Dublin Core Metadata Initiative) representation before the metadata is published to a modified OAI-PMH server in line with CLARIN guidelines. The server is configured to be able to deliver both the mandatory DCMI descriptions as well as the CMDI descriptions. These are publicly harvestable and are available, for example, in the CLARIN-EU VLO[3]. Metadata records are directly accessible through the PID specified in the CMDI MdSelfLink of each document and are served

[3] http://www.clarin.eu/vlo/

directly from the same database that feeds the OAI-PMH server. Web based access to underlying resources is provided and is subject to authorization and authentication procedures. While most of the Meertens' resources are directly available from the institute's servers, it is the intention of the Meertens Institute to subcontract long term archiving to specialized archive repositories such as DANS (Data Archiving and Networked Services) and TLA (The Language Archive) in line with archiving guidelines. To achieve this, the Meertens Institute actively participates in the TLA agreement between the Max Planck Gesellschaft, Berlin-Brandenburg Academy of Sciences and Humanities and the Royal Netherlands Academy of Arts and Sciences under which both DANS and the Meertens reside.

4. Authorization and authentication

Authorization and authentication are of primary concern to researchers working with sensitive data. The Meertens Institute for example houses a large collection of digitized questionnaires spanning several decades that cannot be made publically available as a result of the prevailing privacy regulations. The Questionnaire Collection consists of scans of hand written questionnaire responses and contains responders' names and addresses. As a result, these must either be anonymized or can only be available to selected end users. The Meertens developed a proprietary authorization system that is capable of protecting its resources in various manners. It is connected to the Surfnet and CLARIN federations allowing federation users to access the materials. Each resource can be individually protected and it is possible to associate several licenses to each resource that must be accepted by the end user before access is granted. Although the Meertens Institute applies a general 'open access' policy situations may occur where additional licenses apply to the data. When access to a resource is requested the owner (usually the researcher) of the resource is contacted. The owner then decides whether or not to add the user to the access control list.

5. Making resources available to end users

While publication of metadata is important to disseminate information on the available research data it is equally important to provide end users with the necessary means to search through the metadata and locate the information of interest. The CLARIN VLO provides general means to access the data harvested across multiple organizations institutes. For individual institutes, like the Meertens, it also makes sense to only make metadata descriptions and associated resources available that are relevant to the specific community it intends to serve. To accommodate for both scenarios all metadata records are indexed automatically using Lucene/SOLR[4], an open source search platform in a separate step after the ingest process has completed. The approach taken is applicable to any CMDI description and poses no limitations on the metadata structure other than the basic CMDI format. It has been tested on over 49,000 metadata records gathered from various institutes in Europe that have adopted the CMDI approach. The index is created automatically using the information obtained from the CMDI profile descriptions and ISOcat references. ISOcat references are used as fields for the indexing schema to dynamically add an index and efficiently store the contents of the metadata elements. All content is also full text indexed. If no ISOcat references are available, then there is at least an index containing the full text. In cases where the CMDI structure needs to be disambiguated the contextual information is taken into account. A reference to /description/, for example, may refer to a description of a resource, a person or an organization. Provided that the higher level resource, person and organization elements are annotated using ISOcat references, the indexing process is able to take the context of the description

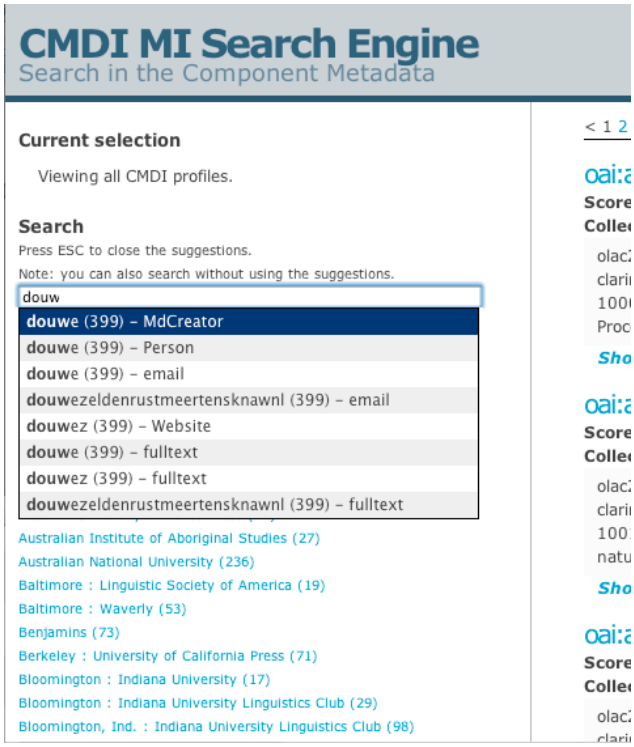


Figure 3: Use of ISOcat data categories in search demonstrator

element into account. Each of the resource/description, person/description and organization/description is indexed separately and may be searched (see figure 3). An easy to use front end provides end users with the features such as results lists based on relevance ranking, auto completion, feedback on the categories in which information was found, facetted browsing and suggestions for related metadata records. The search engine is also made available as a web service supporting the SRU/CQL dialect currently developed in CLARIN-EU and the CLARIN-NL Search & Develop

[4] http://lucene.apache.org/solr/

project allowing it to be incorporated in a combined, federated metadata and content search scenario. Here, users may construct queries that not only allow location of resource using the metadata descriptions but can also search directly in the content of associated content search engines. A prototype of this has been developed in the CLARIN-NL Search & Develop project combining the CMDI metadata search with content search engines from the Meertens Institute, MPI Nijmegen, INL(Institute for Dutch Lexicology) and DANS(Data Archiving and Networked Services). The Meertens content search engine was developed during an earlier CLARIN-NL project MIMORE and makes three of its language variation databases (MAND, DynaSand and DiDDD) available through a combined search engine.

6. Conclusion

The Meertens Institute has adopted the CMDI metadata approach and is actively developing the necessary technical and organization support to transform its resources into the technical landscape of CLARIN. Our experiences show that adoption of the main principles of the CMDI and ISOcat models is capable of providing a manageable environment. Heterogeneous metadata descriptions can be created reflecting the different natures of the resource types available within our institute and metadata creation scenarios. These can be made available to a large user community using standard protocols and in more user-friendly manners. It is our intention to further participate in the further development of a true e-science infrastructure towards Virtual Research Environments (VREs). Here, not only data and metadata management tasks become part of a VRE but also data enrichment services, such as NLP processing pipelines, are integrated to provide a workbench tailored towards a targeted end user community. In the Netherlands the Meertens Institute is an active participant in pending project proposals such as Nederlab and CLARIAH.

7. Acknowledgements

The projects described in this paper have been made possible through projects grants form the Prins Bernhard Cultuur Fonds and CLARIN-NL.

8. References

Broeder, D., Schonefeld, O., Trippel, T., Van Uytvanck, D., & Witt, A. (2011).A pragmatic approach to XML interoperability — the Component Metadata Infrastructure (CMDI). *Proceedings of Balisage: The Markup Conference 2011. Balisage Series on Markup Technologies, 7.*

Broeder, D., Kemps-Snijders, M., Van Uytvanck, D., Windhouwer, M., Withers, P., Wittenburg, P., & Zinn, C. (2010). A data category registry- and component-based metadata framework. *In N. Calzolari, B. Maegaard, J. Mariani, J. Odjik, K. Choukri, S. Piperidis, M. Rosner, & D. Tapias (Eds.), Proceedings of the Seventh conference on International Language Resources and Evaluation (LREC'10) (pp. 43-47). European Language Resources Association (ELRA*).

CMDI, ISO TC 37 SC 4 work item for ISO 24622)

Dima, Emanuel, Christina Hoppermann, Thorsten Trippel and Claus Zinn (forthcoming): "A Metadata Editor to Support the Description of Linguistic Resources". *Accepted to LREC 2012, the 8th International Conference on Language Resources and Evaluation*.

M. Gavrilidou, P. Labropoulou, S. Piperisid, M. Monachini, F. Frontini, G. Francopoulo, V. Arranz, V. Mapelli. A Metadata Schema for the Description of Language Resources (LRs), *International Joint Conference on Natural Language Processing, Chiang Mai / Thailand*

Marc Kemps-Snijders, Menzo Windhouwer, Peter Wittenburg, Sue Ellen Wright (2009) ISOcat: remodelling metadata for language resources. nternational *Journal of Metadata, Semantics and Ontologies*

Zeldenrust, D.A. & M. Kemps-Snijders. (2011) "Establishing connections: Making resources available through the CLARIN infrastructure.'. *In: Supporting Digital Humanities 2011, Answering the Unaskable. Kopenhagen : [s.n.], 2011*