CMDI 1.2: Improvements in the CLARIN Component Metadata Infrastructure

Twan Goosen, Menzo Windhouwer, Oddrun Ohren, Axel Herold, Thomas Eckart, Matej Ďurčo, Oliver Schonefeld

CLARIN ERIC, The Language Archive - DANS, National Library of Norway, Berlin-Brandenburg Academy of Sciences and Humanities, Leipzig University, Institute for Corpus Linguistics and Text Technology, Institute for the German Language

Utrecht, The Netherlands; Nijmegen, The Netherlands; Oslo, Norway; Berlin, Germany; Leipzig, Germany; Viena, Austria; Mannheim, Germany

E-mail: twan@clarin.eu, menzo.windhouwer@dans.knaw.nl, oddrun.ohren@nb.no, herold@bbaw.de, teckart@informatik.uni-leipzig.de, matej.durco@oeaw.ac.at, schonefeld@ids-mannheim.de

Keywords: metadata, infrastructure, centres, compatibility

1. Introduction

Component Metadata Infrastructure (CMDI) has been one of the core pillars of CLARIN since the beginnings of this initiative (for an overview, see Broeder et al., 2012).

It established means for flexible resource descriptions for the domain of language resources with sound provisions for semantic interoperability weaved deeply into the metamodel and the infrastructure to overcome, in a great extent, the rule of metadata schism it set out to combat. Based on this solid grounding, the infrastructure accommodates a growing collection of metadata records. The development of the joint metadata domain both in number of records and diversity of profiles is proof for the success of the model and the infrastructure as such. Currently, at version 1.1 of the CMDI specification, there are 157 public profiles with 872 components defined and the harvester collecting periodically over 680.000 records from some 60 providers in more than 80 different profiles.

However in the first five years of its intensive usage by the CLARIN community naturally a number of design issues have arisen that need further attention. Out of various options to remedy the encountered problems, such as the creation of extensions on top of the existing CMDI implementation or abandoning its component based architecture, a dedicated taskforce eventually chose to work towards a successor to CMDI 1.1 based on the existing paradigm. After careful analysis, the task force worked out a proposal for a number of small but important changes and additions to the CMDI model leading to CMDI version 1.2. In April 2014, the Standing Committee for CLARIN Technical Centres approved the proposal, which meant that work on the implementation could begin.

The changes address the following aspects: lifecycle management, structure of the model and schema sanity (namespace issues, consistency of the meta model, attributes, mandatory/optional elements), use of external vocabularies and cues for tools. They are described in detail in sections 2 and 3.

The work on the model is accompanied by a comprehensive transition plan covering the conversion of existing data and adaptation of existing tools using CMDI data, described in section 4.

Finally, section 5 details still open issues and further plans for the CMDI model and joint metadata domain.

2. New CMDI functionality

2.1 Lifecycle Management

There is no definite metadata representation for any given language resource in terms of single fixed CMDI component or profile. Instead, metadata modellers often encounter situations that make it necessary to adapt or amend existing models. Typically, such situations are caused by needs of data providers that supply more detailed metadata than any existing component caters for. There were no means to track changes between versions of individual metadata components within the current version of CMDI. CMDI 1.2 will provide lifecycle management support for components based on four additional header fields: *Status*, *StatusComment*, *Successor* and *DerivedFrom*. The mandatory *Status* field is used to record the current lifecycle phase of a component (development, production, deprecated). It may be further annotated with a *StatusComment*. For deprecated components, the URI of a *Successor* can optionally be specified to indicate an improved version of the component that should be used instead. The URI specified in the optional *DerivedFrom* field allows for the reconstruction of a component's genesis in relation to other components.

As all published components are kept persistently within the Component Registry[1] (see Broeder et al., 2010), the addition of improved versions of components easily leads to proliferation. Explicit lifecycle management and especially the *Status* field will be useful for constraining the users' and modellers' view to a manageable set of components and help them focus on the most recent versions.

2.2 Vocabularies

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

The current version of CMDI requires value domains for elements and attributes to be specified locally in the components. CMDI 1.2 on the other hand will support the use of external vocabularies, thereby increasing the possibility to obtain semantic interoperability across metadata.
Metadata modellers will have the opportunity to associate a vocabulary (identified by its URI) with an element in their components and profiles. The metadata creator will then be able to pick values from the specified vocabulary or (for open vocabularies) still choose to use a custom value that does not appear in the vocabulary.
At the model level, the new facilities will be generic, i.e. no assumption about specific services will be made. However, initially the core CMDI infrastructure will be designed to support specifically the OpenSKOS-based CLAVAS vocabulary service (Brugman, 2012), through which vocabularies of languages, organisations and value sets extracted from ISOcat are already available. External vocabularies may either be imported into the CMDI component in question, or be just referenced by the component and be used for dynamic lookup and retrieval of values when editing metadata records.
The above will be facilitated by introducing a new element Vocabulary in ValueScheme elements, with an optional enumeration element for closed vocabularies. At the instance level, an attribute ValueConceptLink (in the CMDI namespace) will be allowed on fields that have a vocabulary linked to hold the URI of the selected value.
Importing the vocabulary as enumeration into the component allows for strict schema validation of the values in the instance data, but does not reflect changes in the vocabulary, on the other hand referencing a vocabulary allows keeping the list of proposed values dynamically up to date, but does not allow for any kind of validation of the element values. Thus the modeller has to decide based on the expected completeness and change rate of the vocabulary which mode to apply.
To make the new functionality available for metadata modellers and creators, both Component Registry and existing metadata editors must be updated accordingly.

2.3 Cues for Tools

Some of the applications in the context of CMDI, especially those directly used by human users, require information that goes beyond formal specification and validation aspects. This includes documentation of meaning and purpose of all content-related elements, which is essential for both metadata creators and human interpreters. CMDI 1.1 already provides an option to document the usage of CMDI elements but lacks this functionality for attributes or components. Therefore CMDI 1.2 expands the existing approach to all kind of metadata entities. This allows schema creators to document their profiles in all necessary details. Furthermore CMDI 1.2 will permit multiple documentation values for different languages, which can be the basis for localised user interfaces.
Also in the context of user-friendly interfaces extensive changes are introduced to augment metadata profiles with information about how the metadata content should be presented to the user. CMDI 1.1 only provided a very simple approach to specify display priorities for elements. This approach is replaced by a new namespace for all kinds of display cues. These cues may contain grouping information to allow merging of dependent components, selection of elements as representatives of their component or explicit visual hints. The set of allowed cues is completely open for future extensions. By using XSLT stylesheets the original component specification can be augmented if necessary. As a consequence a tool or a user can decide if display hints are needed at all or may select between different sets of display cues if available.
A further extension in CMDI 1.2 is the specification of value derivation cues. The experience with CMDI in the last years revealed that a lot of metadata can be automatically derived from other values. This includes the definition of duration as the difference of two timestamps, the specification of language names based on already stated language codes or the support of keywords like "FileSize" that are automatically replaced with their actual value by editor tools. The systematic usage of this feature avoids redundancy, helps metadata creators build consistent metadata and allows an explicit definition of relations between elements. Similar to visual hints there is no fixed set of allowed rules and keywords. Instead a general framework is specified where most information about relations is stored in an external registry and the actual derivation is regarded as an optional functionality of applications.

2.4 Attributes in instances

In CMDI 1.1 attributes on instance elements were always optional. This does not allow for closely mimicking the constraints of some existing models (the TEI Header (TEI Consortium, 2014), for example, has mandatory attributes), and poses a needless restriction. An element 'required' is added to the attribute definition in component specifications in CMDI 1.2 to allow for both optional and mandatory attributes.

3. Fixed CMD functionality

3.1 CMD Namespaces

In CMDI 1.1 a CMD namespace, i.e., *http://www.clarin.eu/cmd/*, was introduced. All CMDI records use this namespace, regardless of the profile, and thus XML Schema. This, although simple, approach has led to problems with the basic assumptions about XML, namespaces and schemas made by tools and standards outside of CLARIN. For example, the OAI-PMH protocol (Lagoze et al, 2002), which is used by CLARIN but specified by the Open Archive Initiative, demands that only one schema is associated with a metadata prefix. But CMDI metadata comes with many schemas, a different one for each profile. Also tools, such as Xerces2-J [2], that perform XML Schema validation, assume (backed by the XML Schema recommendation (Thompson et al, 2004)) that a namespace is associated with a unique schema and base their caching strategy on this. In CMDI 1.2 the single namespace is replaced by a general namespace for the CMDI Envelope, and profile specific namespaces for the payload. This allows binding of the CMDI Envelope schema to the OAI-PMH CMDI

[2] http://xerces.apache.org/xerces2-j

metadata prefix and also supports caching of profiles specific schemas. In principle this touches every resource in the infrastructure. Fortunately many of these tools can use various approaches, e.g., wildcards, to ignore the profile specific namespaces when they access arbitrary CMDI records.

Another namespace related issue is the potential clash between reserved attributes, i.e., *ref* and *componentId*, and use defined attributes. In CMDI 1.2 reserved attributes are moved to the CMD namespace, so the user has the freedom to define attribute with arbitrary names, e.g., including *ref*.

3.2 Changes in the CMD Envelope

In CMDI 1.1, *IsPartOfList* with its *IsPartOf* elements can be used to link to collections that the described resources and/or metadata are part of. However, the nature of the (implicit) subject of an *IsPartOf* statement has been unclear. While its current position within the Resources element may indicate that any *IsPartOf* relation applies to *all* resources referenced in *ResourceProxyList*, its mere name 'IsPartOf' indicates a single subject.

In CMDI 1.2, this issue will be resolved by moving *IsPartOfList* to the envelope top level alongside Resources, and restricting the semantic of *IsPartOf* to express a partitive relationship between the described resource as a whole and some collection or larger resource.

Other relationships between resources than *IsPartOf* can, broadly speaking, be expressed in one of two ways in the CMDI framework; either using components and elements, or as *ResourceRelation* elements within the *Resource* section of the CMDI envelope. *ResourceRelations* in CMDI 1.1 contain simply a *RelationType* element giving a name for the relation, together with elements *Ref1* and *Ref2* pointing to the related resources.

Existing data shows that the latter method has been very little used. There seems to be a general feeling that the current *ResourceRelation* is too simplistic and underspecified to convey the intended information. Although no fundamental change will be performed in CMDI 1.2, the intention is to clarify the semantics of the current specification, all the while keeping the door open for expressivity extension at a later date.

In CMDI 1.2, *ResourceRelation* elements should always contain exactly two Resource elements (replacing Res1 and Res2), explicitly constraining relationships to be binary. In these elements, a mandatory *ref* attribute (indicating a resource listed in the same CMDI record) and an optional *Role* element with an optional *ConceptLink* attribute is added. Moreover, *RelationType* is extended with an optional *ConceptLink*.

This way, both relationship direction as well as semantic marking of both relation type and resource roles may be defined by metadata creators.

3.3 Component Schema Cleanup

Since the development of CMDI started, multiple developers have worked on the schema that governs how CMDI profiles and components are specified in XML. Different modelling strategies have been applied leading to a mixed bag, e.g., most properties of CMDI elements are specified via XML attributes while similar properties are specified in XML elements for CMDI attributes. In CMDI 1.2 these different approaches are cleaned up by going back to the original approach of using XML attributes whenever applicable.

4. Migration from CMDI 1.1 to 1.2

Centres should upgrade their data and tools if they wish to benefit from the changes in CMDI 1.2 and good integration with the infrastructure as other centres are upgrading as well. CMDI 1.1 will be phased out in the future, but initially the core infrastructure components will support both version 1.1 and 1.2, allowing centres to migrate at their own pace. Centres may choose to keep supporting both versions after upgrading. Migrating to CMDI 1.2 is an active migration process requiring varying degrees of effort from the centres depending on the specifics of the repository and/or tools maintained by the centre involved. Support in the form of upgrade scripts will be supplied by the CMDI taskforce.

4.1 CMDI Toolkit and Component Registry

The CMDI toolkit comprises the definitions (in the form of XML Schema Definition (XSD) and Extensible Stylesheet Language Transformations (XSLT) documents) that define the language for the specification of metadata components and profiles as well as the structure of metadata instances in relation to profiles. The taskforce will produce a new version of this toolkit, which then provides the essential components for creating CMDI 1.2 metadata.

The Component Registry is built on top of this toolkit and will be the first infrastructure component to be adapted to support CMDI 1.2. All existing components and profiles stored in the Component Registry will be converted to CMDI 1.2 once using an XSLT that is part of the toolkit. These components and profiles will become available at a new location in the Component Registry's web service. CMDI 1.1 versions of all components and profiles will be generated on-the-fly by applying a downgrade XSLT and can be requested by tools and users at the existing locations. Therefore, the Component Registry will remain compatible with existing infrastructure components. An analysis has shown out that converting existing components and profiles back to CMDI 1.1 can be carried out losslessly, therefore the validity of existing metadata instances is not affected.

4.2 Conversion of CMD Records

The taskforce will provide an XSLT for upgrading metadata records from CMDI 1.1 to CMDI 1.2. Upgrading a record entails transforming the schema reference into a reference to the schema based on the CMDI 1.2 version of its profile and applying all required changes to make the document compliant with the CMDI 1.2 specification (see sections 2 and 3). In some exceptional cases, an automated transformation cannot be carried out. Specifically, if no profile reference is present in the original record or multiple 'ref' attributes are found on a single element (both of which are schema valid in CMDI 1.1), an error will be yielded and the record will have to be adapted manually.

4.3 Tools, Services and Repositories

Since the Component Registry will keep supporting CMDI 1.1, the need to upgrade other tools, services and repositories hosted and maintained by the centres will not be pressing immediately in most cases. To some degree, a chicken-and-egg relation exists between the repositories and the metadata they produce, and exploitation software that processes this metadata. Adding support for CMDI 1.2 to central tools and services that deal with a broad variety of metadata sources and types, such as the Virtual Language Observatory, will be most urgent. As soon as some support exists in the exploitation stack, it makes sense for repositories to start providing CMDI 1.2 metadata. In some cases this can be achieved by applying (additional) transformations. Often, however, this will depend on more thorough modifications in the metadata creation pipeline, including editors and content management systems, especially if the new features of CMDI 1.2 are to be harnessed.

5. Roadmap

Work on the implementation has begun mid 2014, starting with the creation of a new version of the toolkit. Once this has been completed, the Component Registry will be updated, followed by the migration of all registered components and profiles. Finally, the remainder of the infrastructure can be migrated in a distributed fashion. The adoption rate of CMDI 1.2 will have to determine the moment of deprecation of CMDI 1.1.

There are a number of tasks related to CMDI 1.2, some of which are currently being worked on, and some of which are planned for after or in parallel to the implementation of CMDI 1.2. First of all, the taskforce has initiated the process of writing an extensive and formal specification of CMDI. Such a specification does not exist for CMDI 1.1. In addition to this formal description of the technical scope of CMDI, a document describing *best practices,* targeted primarily at the metadata modeller, is also planned for and a first version is expected to get published in the near future.

There is ongoing work - coordinated by the CLARIN Metadata Curation Task Force - on evaluating the quality of the metadata records in the joint metadata domain. The main goal is to provide a service that examines individual records or whole collections, performing a number of basic checks (schema validation, "dead links", etc.), and optionally normalisation of values based on controlled vocabularies, producing a curation report that lists encountered issues. The checks will especially also cover the specifics of the CMD versions, to support the data provider in the transition period. Once completed, this service will be integrated into the basic workflow for harvesting the metadata and filling the VLO.

Finally, it is good to point out that a number of known shortcomings of CMDI 1.1 have been decided not to be addressed in CMDI 1.2, but rather should be investigated further so that a reliable and non-controversial solution can be incorporated in a future version of CMDI. Some features that are often considered to be desirable, but are not present in either CMDI 1.2 or any previous version, are versioning options for metadata instances, the possibility of recursive component hierarchies, a distinction between empty and *nil* field values and component or profile specific limitation of the types of resource references allowed in the instance. It is hoped that the CMDI community will largely and successfully adopt CMDI 1.2 and provide the support required to implement these and other enhancements in the future.

6. Acknowledgements

The authors wish to thank the members of the CLARIN CMDI Taskforce, as well as all participants of the CMDI Future Workshop that was held in Utrecht on October 14, 2013.

7. References

Broeder, D. Kemps-Snijders, M., Van Uytvanck, D., Windhouwer, M., Withers, P., Wittenburg, P., and Zinn, C (2010, May). A Data Category Registry- and Component-based Metadata Framework. In *Proceedings of the Seventh conference on International Language Resources and Evaluation (LREC)*, pages 43–47, Valletta, Malta

Broeder, D., Windhouwer, M., van Uytvanck, D., Goosen, T., and Trippel, T. (2012). CMDI: a Component Metadata Infrastructure. In *Describing LRs with Metadata: Towards Flexibility and Interoperability in the Documentation of LR Workshop Programme*.

Brugman, H., and Lindeman, M. (2012). Publishing and Exploiting Vocabularies using the OpenSKOS Repository Service. In *Describing LRs with Metadata: Towards Flexibility and Interoperability in the Documentation of LR Workshop Programme*.

Lagoze, C., Van de Sompel, H., Nelson, M., and Warner, S. (2002). *The Open Archives Initiative Protocol for Metadata Harvesting*. http://www.openarchives.org/OAI/2.0/openarchivesprotocol.htm. Accessed on 20 June 2014.

TEI Consortium, eds. (2014). *Guidelines for Electronic Text Encoding and Interchange*. 20 January 2014. http://www.tei-c.org/P5/. Accessed on 20 June 2014.

Thomson, H.S., Beech, D., Maloney, M., and Mendelsohn, N. (2004). *XML Schema Part 1: Structures Second Edition*. http://www.w3.org/TR/xmlschema-1/. Accessed on 20 June 2014.

Windhouwer, M., Goosen, T., Schonefeld O, Ohren, O., Eckart, T., Herold, A., Misutka, J., Frankhauser P., Schiel, F., Eckart, K., et al. (2014). *CMDI 1.2 changes - executive summary*. Technical Report CE 2014-0318, CLARIN ERIC, http://www.clarin.eu/content/cmdi-12-changes-executive-summary.