Balisage: The Markup Conference 2011
August 2 - 5, 2011

Balisage Paper: A pragmatic approach to XML interoperability – the Component Metadata Infrastructure (CMDI)

Daan Broeder
Max Planck Institute for Psycholinguistics, Nijmegen
<daan.broeder@mpi.nl>

Oliver Schonefeld
Institute for the German Language (IDS), Mannheim
<schonefeld@ids-mannheim.de>

Thorsten Trippel
Tübingen University
<thorsten.trippel@uni-tuebingen.de>

Dieter Van Uytvanck
Max Planck Institute for Psycholinguistics, Nijmegen
<dieter.vanuytvanck@mpi.nl>

Andreas Witt
Institute for the German Language (IDS), Mannheim
<witt@ids-mannheim.de>

How to cite this paper

Broeder, Daan, Oliver Schonefeld, Thorsten Trippel, Dieter Van Uytvanck and Andreas Witt. "A pragmatic approach to XML interoperability – the Component Metadata Infrastructure (CMDI)." Presented at Balisage: The Markup Conference 2011, Montréal, Canada, August 2 - 5, 2011. In *Proceedings of Balisage: The Markup Conference 2011*. Balisage Series on Markup Technologies, vol. 7 (2011).

https://doi.org/10.4242/BalisageVol7.Broeder01.

Abstract

XML has been designed for creating structured documents, but the information that is encoded in these structures are, by definition, out of scope for XML. Additional sources, normally not easily interpretable by computers, such as documentation are needed to determine the intention of specific tags in a tag-set. The Component Metadata Infrastructure (CMDI) takes a rather pragmatic approach to foster interoperability between XML instances in the domain of metadata descriptions for language resources. This paper gives an overview of this approach.

Table of Contents

Introduction

XML documents are commonly used to exchange data. The strict definition of the Markup-Language resulted in a variety of tools and every XML instance (given it is well-formed) can be processed by of-the-shelf tools. XML has been designed for creating structured documents, but the information that is encoded in these structures are, by definition, out of scope for XML. Therefore, generic identifier like `p` can have different meanings, depending on which concrete markup language (or here: XML tag-set) is used. For example, in the case of HTML it denotes a paragraph while it may denote something completely different in another tag-set. An XML schema language, e.g. like DTD, XML Schema or RelaxNG, define a grammar for a given markup language and thus valid XML instances can be told apart from invalid ones, but they do not provide any inherent semantics to "understand" the XML instances. The necessary knowledge to interpret a markup language usually exists in the form of human-readable documentation and is "out of reach" for the computer. However, when trying to exchange (more or less) arbitrary XML instance some form of knowledge is needed to interpret these documents. The topic of the semantics of markup has been mostly discussed from an academic point of view, see e.g. Sperberg-McQueen & Huitfeldt 2011.

The Component Metadata Infrastructure (CMDI) takes a rather pragmatic approach towards adding some semantics to XML to allow exchange of metadata descriptions encoded in various metadata formats in a (slightly adapted) XML encoding. This is done by linking generic identifiers to semantic concepts in a data category registry and thus allow more profound interpretation of the markup. The CMDI approach is set in the domain of metadata descriptions, but may be generalized to be used within other domains.[1]

Component Metadata Infrastructure

The Component Metadata Infrastructure is developed in the context of the CLARIN project (Váradi et al. 2008). CLARIN aims at building an integrated and interoperable research infrastructure for language resources. The goal is to provide a stable, persistent and accessible infrastructure for the eHumanities. One important aspect of CLARIN is to enable easy sharing of

language resources. This will allow researchers to use existing resources as a basis for their work, e.g. by optimizing their existing or new tools, by building derivative resources or expose their resources to a broader audience. Therefore, to make this infrastructure more usable, resources need to be easily accessible, in particular easily findable. The most common approach towards achieving this is to provide descriptive metadata about these resources and use these information to find resources of interest for a particular researcher.

Part of this context is also an already large installed base of metadata descriptions available using fixed metadata schemas as IMDI and Simons et al. 2008. Although the quality of the metadata is sometimes questionable, it would be unacceptable to put a new framework into place that would lock out these existing metadata resources

Since CLARIN is a rather large, diverse project, different project members have different opinions on how to adequately model the metadata for their types of resources. For a lot of existing resources extensive amounts of metadata descriptions are already available. It seems naïve to assume that agreement on a common metadata schema for a large-scale project like CLARIN can be achieved and will most likely result in the least common denominator, e.g. Dublin Core (Dublin Core, Baker 1998), losing a lot of the express power that is used in existing metadata, as would using a "pivot" schema, both would result in information loss. CLARIN tries to solve the problem by the Component Metadata Infrastructure (CMDI), which is basically a framework to accommodate for different XML-based metadata formats. CMDI provides, supported by various tools, a framework and work flows for creating metadata formats and metadata descriptions as well as semantic foundation for processing metadata descriptions.

Framework overview

CMDI is a framework (see Figure 1) to build component based metadata descriptions. A metadata component is basically a collection of atomic metadata fields or data categories (DatCats) and describes a specific aspect or dimension of a resource, e.g. the title of a document, the creator or the native language of a subject in a video recording. Components can have a recursive

structure, i.e. in addition to atomic fields, the components can also contain other components. Thus, components serve as small building blocks or reusable templates for a specific aspect of a resource. Together with a header, these components are combined into metadata profiles, each of which can be used as a schema for metadata instances. Both, components and profiles can (and should) be stored in a component registry, which is a directory of components to be reused in different contexts. Users can either reuse existing profiles for their metadata descriptions or create new profiles by reusing or creating new components, either manually or with a specialized component editor. Various profiles already exists in the component registry, e.g. for IMDI, OLAC, Dublin Core or the TEI header.[2]

The storage of the schemas in a centralized infrastructure is common practice for metadata schemas, though of course this adds the problem of sustainability to the process, inasmuch as the repository of schemas needs to be constantly available. Though this could be seen as problematic in principle for pragmatic reasons it seems more appropriate than to use local copies with modifications, because it makes sure that tools can operate on the centrally stored files. For a metadata archive, a local store of schema copies could be instantiated, but this would result in the requirement to adjust the schema reference, for interoperability this could cause an additional problem. Hence the use of a central infrastructure is probably the safest solution and in the context of an infrastructure of data and services most likely to be sustainable. This is also consistent with the approaches described by Rehm et al. 2011.

Each metadata field is linked to exactly one data category in a data category registry (DCR) using a persistent identifier. The DCR indicates how the content of the field in a metadata description should be interpreted. If the same data category is used in various metadata schemas, the reference to the DCR will still be the same. This is also independent of the concrete naming of the XML element, including names, cases and orthographic variants. For example, the field `title` in `titleStmt` in the TEI header is linked to the same category as the `title` in Dublin Core.

In the CLARIN project the preferred concept registry is the ISO data category registry ISO-DCR. This registry is an implementation of the ISO 12620 standard model for data categories and offers ample functionality for the needs of the CMDI framework. For the CMDI framework it makes no

essential difference if another registry such as for instance the DCMI is used. However the ISO-DCR does have a tight integration with other CMDI software components such as the component editor, for efficient searching for suitable data categories or even combining metadata modelling with defining new data categories.

The component registry contains CMDI components and profiles. If a metadata creator needs to describe a (for him) new type of resource, he can browse through the available profiles and see if there is one that suits his needs. If there is no suitable profile available he can create a new one, based on existing components or he can create new components and work these with existing ones into a new profile.

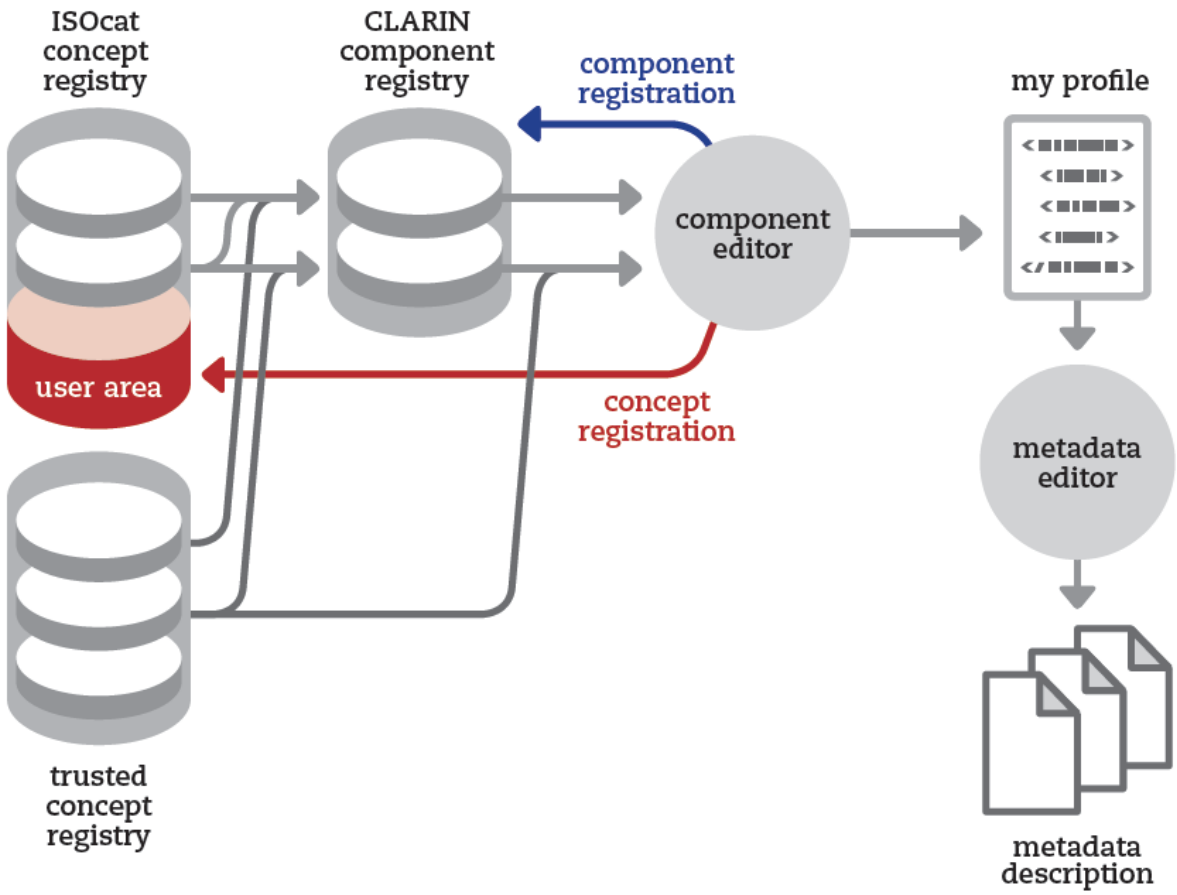
When creating metadata elements in new metadata components users can browse and search for entries in the ISO-DCR to find a concept that matches the semantics of the metadata element. The identifier of the concept is then automatically inserted in the metadata component specification.

To create metadata descriptions users load profiles into the metadata editor, which then can automatically generate forms based on the metadata profile. The user then fills out these forms to enter the data. Of course users may also use an XML editor to create metadata descriptions directly and use the provided XML schemas (see below) to validate the XML documents.

The resulting metadata records are offered for harvesting by OAI-PMH and gathered in one or more central repositories.

Multiple ways to exploit the collected metadata are foreseen ranging from systems doing simple keyword search to those using faceted browsing or structured search. In all of these semantic mapping using the ISO-DCR plays a crucial role. When a user specifies a metadata query, the ISO-DCR then allows to expand this query into set of equivalent ones that will be able to retrieve metadata records where a different terminology than specified in the original query. The identifiers of the terms in the query are used to find equivalent terms and these are then used to generate an additional query. E.g. when a user queries for `titleStmt` an additional query is generated for `title`, since `titleStmt` is linked to `title` via the ISO-DCR.

Figure 1: Overview of the CMDI framework.



As mentioned before, a metadata component describes various aspects or dimensions of a resource. Figure 2 shows a schematic representation of a very simple example metadata component "Actor". It contains two atomic fields "firstName" and "lastName" and refers to another component "ActorLanguage", which contains a repeatable[3] atomic field "actorLanguageName". An entity "Actor" therefore consists of a first name, a last name and a list of languages.

In CMDI components are expressed in XML files. Figure 3 displays the "Actor" component in the CMDI component XML specification tag-set. `CMD_Component` elements define new components, `CMD_Element` elements new atomic fields. The `ConceptLink` attribute is the most important aspect in terms of interoperability, because it stores the link to a DCR, or more specific the PID of a data category. Software interpreting the component definitions can use this concept link to draw further conclusions from information, like establishing an equality relation be between different field in different

metadata schemes and use this, e.g. for smart searching. The component descriptions are normally transformed to XML Schema using an XSLT transformation. These XML Schemas are available from the component registry and can e.g. be used in special metadata editors or plain XML editors to aid the user in creating metadata records. Figure 3 shows an example instance of an "Actor" component. In a complete CMDI metadata record the component together with and one or more links to the described resource are wrapped with a header.

Especially in connection with the standardization efforts mentioned in section "Conclusion", TEI ODD will be evaluated as an alternative apparatus for defining metadata components.

Other representation formats such as RDF, OWL and Topic Maps do not seem appropriate for the description of the metadata in comparison to CMDI. It is obvious, that CMDI due the recursive structure of defining components can become rather complex, but the structures are at least assumed to be human readable and structured according to a human prose text on a resource. In contrast to this, the RDF-family is not requiring the linear order, presenting the RDF-triples in arbitrary order. Though CMDI documents can be rendered in RDF (and probably in OWL and Topic Maps), the struture of CMDI is more transparent and usable to human users. CMDI is also not a form of knowledge representation, in which the concepts of a resource are described, but it is intended to provide structured information about a resource for human users.

Figure 2: Schematic representation of the metadata component for "Actor"

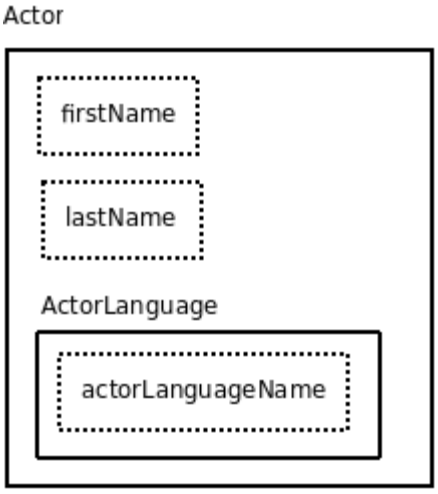


Figure 3: XML representation of a metadata component for “Actor”

```
<CMD_Component name="Actor">
  <CMD_Element name="firstName" ValueScheme="string"
               ConceptLink="http://www.isocat.org/datcat/CMD-
123">
  <CMD_Element name="lastName" ValueScheme="string"
               ConceptLink="http://www.isocat.org/datcat/CMD-
124"/>
  <CMD_Component name="ActorLanguage" id="ActorLanguage"
                 CardinalityMin="0" CardinalityMax="unbounded">
    <CMD_Element name="ActorLanguageName" ValueScheme="string"
                 ConceptLink="http://www.isocat.org/datcat/DC-
1766"/>
  </CMD_Component>
</CMD_Component>
```

Figure 4: XML instance of a metadata description record for “Actor”

```
<Actor>
  <firstName>Foo</firstName>
  <lastName>Bar</lastName>
  <ActorLanguage>
    <ActorLanguageName>Kilivila</ActorLanguageName>
    <ActorLanguageName>French</ActorLanguageName>
  </ActorLanguage>
</Actor>
```

Tools

For the use of the Component Metadata Infrastructure, various tools exist, some being reused from other contexts, others were explicitly developed in this context. Among them are editors, registries and search applications, which will be described briefly.

ISOcat: the Data category registry for ISO TC 37

The data category ISOcat ISO 12620 stores data categories and implements ISO 12620:2009. It is a specialized concept registry, historically developed for data categories used in terminology exchange. However, the concept was so flexible and useful that it was extended to further areas, including linguistic resource management with all required metadata categories.

As a web-based registry for data categories and concepts, ISOcat can be extended by additional data categories as required by users to cater for the individual project needs. Data categories can be defined privately or publicly, submitted for ISO-standardization or not.

Each data category in ISOcat receives a Persistent Identifier (PID) which is used to reference to it, especially suited to be included in metadata and schemata of linguistic resources to foster semantic interoperability. Some schema languages, e.g., TBX XCS and TEI ODD, have built-in support to embed these PIDs into the schema. However, more generic schema languages such as Relax NG and W3C XML Schema do not, but with the definition of attributes schemas in these languages can easily be extended to include them.

The CMDI Component Registry

The Component Registry is also a web-based service, but currently not part of an ISO standard. Within the Component Registry, users of CMDI can store their metadata components and profiles. But it not only allows storage, but also contains editing functionalities.

In the CMDI Component Registry each component is also assigned an identifier that is unique in the context of the component registry, in order for other components to integrate it. Additionally this component identifier can be used as a reference for the profiles, that is, the instances document type declaration and namespaces can point to the component registry for their XML Schema.

Arbil: The CMDI supporting metadata editor

A special challenge for any metadata framework is the creation of instances, which needs to be easy and user friendly. As CMDI is highly adjustable and flexible, this poses additional complications. With the metadata editor Arbil, there is an XML-Editor that is aware of CMDI-structures and connects to the component registry downloading the available (schematized) CMDI profiles. Since there can be very many, the user can limit the number of CMDI profiles that are actually shown in the user interface

Relation Registry

The CMDI Relation Registry (RR) is designed to augment a limitation in ISO-DCR and allow the metadata search user to create (temporary) simple relations between different data categories in the ISO-DCR. The ISO-DCR can overcome “accidental” semantic overlap between different terms, i.e. two metdata developers used different terms but agree on the same definitions. The RR can be used by users searching the metadata to overcome intentional semantic overlap, i.e. the metadata modelers decided that two terms actually mean different things, but where the user decides that this difference is irrelevant for him. He would specify the relation “Term1” == “Term2” and the semantic mapping machinery of the metadata search would expand every query with “Term1” with one that also uses “Term2”.

Joint Metadata Repository

The joint metadata repository (JMDR) is the place where all the harvested CMDI metadata records are stored. The harvesting method is the well-known OAI-PMH, currently there is not yet a registry where the CMDI metadata providers are registered, but such a registry is under consideration.

There may be several of such joint metadata repositories, each specializing in one type of metadata search service. Considering the (expected) great variety of metadata schemas, it was thought advantageous to use native XML database to allow searching through the collected CMDI records.

Currently no semantic normalization is done when the records are stored in the JMDR, this is to allow a query to retrieve only those records that actually use a profile specific terminology.

Searching over structured CMDI data

Added value of highly structured and rich metadata descriptions can be achieved if the search process is more elaborated, leading to more precise and fast results than a full-text search, without lowering the recall. Two examples of such search interfaces are the Virtual Language Observatory and the NaLiDa Faceted Search. Both harvest the CMDI metadata from data providers, but they have a different functionality.

The Virtual Language Observatory started of with earlier metadata versions. It presents a number of different facets, from which a user selects the interesting data categories.

The NaLiDa faceted browser is slightly more elaborated as it implements conditional facets, i.e. additional facets appear based on earlier selections. For example the facet “corpus type” is irrelevant for non-corpora, hence is only shown if the resource type “corpus” is selected. However, the NaLiDa faceted browser focuses on resources in a national context.

Conclusion

XML encoding is a solid foundation to encode metadata descriptions. In the past, various different metadata schemas emerged based on XML technology, like IMDI, Simons et al. 2008 and DCMI. Several technologies, like OAI-PMH have been created for the easy dissemination of XML encoded metadata descriptions. However, XML is not sufficient to exchange data. One either has to agree on a common schema or transform their data into pivot formats. Especially with rich and elaborated metadata schemas these approaches are cumbersome and most often lead to loss of information. A level beyond XML is needed to convey semantic information about the markup, which allows to draw further conclusions on the information encoded in XML documents. The CLARIN project takes a rather pragmatic

approach towards this problem by adding registries for metadata components and data categories. The information of both registries combined allow to perform, at least to some extend, reasoning about the information encoded in metadata descriptions. For example, more sophisticated searches are possible. To some extend, this approach can be generalized and applied to other scenarios to foster XML document interchange without nervelessly requiring to agree on a common XML markup schema. At the time of this writing, CMDI has been proposed as a work item in ISO/TC 37/SC 4.

References

[Baker 1998] Baker, T. *"Languages for Dublin Core"*. D-Lib Magazine, 4:12. 1998. doi:https://doi.org/10.1045/december98-baker

[DCMI] Dublin Core Metadata Initiative *"DCMI Metadata Terms"*, this Version of 2010-10-11, http://dublincore.org/documents/2010/10/11/dcmi-terms/, latest version http://dublincore.org/documents/dcmi-terms/

[Dublin Core] Dublin Core Metadate Initiative *"Metadata Basics"*. 1995–2011, see http://dublincore.org/metadata-basics/

[IMDI] ISLE Metadata Initiative (IMDI) *"Metadata Elements for Catalogue Descriptions"*. Part 1 B, Version 3.0.13, August 2009 http://www.mpi.nl/IMDI/documents/Proposals/IMDI_Catalogue_3.0.0.pdf

[ISO 12620] ISO 12620. *"Computer Applications in Terminology – Data Categories – Specification of Data Categories and Management of a Data Category Registry for Language Resources"*. ISO, Geneva, Switzerland, 2009.

[Simons et al. 2008] Simons, G. and Bird, S. *"OLAC Metadata"*. 2008, cited version http://www.language-archives.org/OLAC/metadata-20080531.html, latest version http://www.language-archives.org/OLAC/metadata.html

[Rehm et al. 2011] Rehm, G., Schonefeld, O., Trippel, T., Witt, A. *"Sustainability of Linguistic Resources Revisited"*. In: Proceedings of the International Symposium on XML for the Long Haul: Issues in the Long-term Preservation of XML. Balisage Series on Markup

Technologies, Vol. 6, 2010.
doi:https://doi.org/10.4242/BalisageVol6.Witt01

[Sperberg-McQueen & Huitfeldt 2011] Sperberg-McQueen, C. M. and Huitfeldt, C. *"Ten Problems in the Interpretation of XML Documents"*. In: Proceedings of the Conference of Processing Text-Technological Resources 2008, Bielefeld (to appear).

[Váradi et al. 2008] Váradi, T., Krauwer, S., Wittenburg, P., Wynne, M. and Koskenniemi, M. *"CLARIN: Common language resources and technology infrastructure"*. In: Proceedings of LREC 2008, Marrakech, Morocco, 2008. pp. 1244–1248. http://www.lrec-conf.org/proceedings/lrec2008/pdf/317_paper.pdf

---

[1] The authors are well aware, that providing a sound semantic foundation, e.g. an ontology or alike, for less closed domain will be, at least, a challenging task.

[2] The metadata schemas of these sets have been decomposed into components and then recomposed into profiles, while as many components were reused.

[3] This is not yet marked clearly in the figure, we'll find a better graphical notation for the final paper.