Schema Registry
(M 1.2.1)

Version 03/16/2012
Work Package 1.2
Responsible Partner MInf-BA

DARIAH-DE
Aufbau von Forschungsinfrastrukturen für die e-Humanities

This research and development project is / was funded by the German Federal Ministry of Education and Research (BMBF), fund number 01UG1110A to M, and managed by the Project Management Agency of the German Aerospace Center (Deutsches Zentrum für Luft- und Raumfahrt, PT-DLR).

SPONSORED BY THE

Federal Ministry
of Education
and Research

Project: DARIAH-DE: Aufbau von Forschungsinfrastrukturen für die e-Humanities
BMBF Fund number: 01UG1110A to M
Duration: March 2011 till February 2014

Document status: Final
Dissemination level: public
Authors:
Tobias Gradl (TG), MInf-BA
Andreas Henrich (AH), MInf-BA

Revisions:

Date	Author	Comment
03/01/2012	TG, AH	Initial version
03/16/2012	TG	Minor corrections

Table of Contents:

1. Introduction

The *Schema Registry* is a central component of the DARIAH federation architecture and contains schemas that are required for the interpretation of research data contained in the collections listed in the C*ollection Registry*. To facilitate the federation of collections and their respective research data, the Schema Registry further comprises associations between individual schemas—the crosswalks—that can be used for the transformation[1] of research data from and into associated schemas.

For the management and visualization of schemas and crosswalks, the Schema Registry is based on state-of-the-art concepts in the field of schema and ontology matching and mapping. The application of semi-automatic methods to identify associations between schemas is at the core of the current prototype of the Schema Registry—attempting to allow an automatic definition of candidate-associations followed by the manual validation, correction and documentation by domain experts. One mentionable benefit of automatically identified associations is that they can be considered (typically with a lower weight) in combination with the manually verified associations for recall-oriented queries.

Figure 1 illustrates some of the user-oriented tasks that are related to schema and crosswalk management. Included in the key functionality that the comprehensive, interactive federation architecture in DARIAH should provide, the current implementation is focusing on the semi-automatic identification and modeling of crosswalks between schemas.

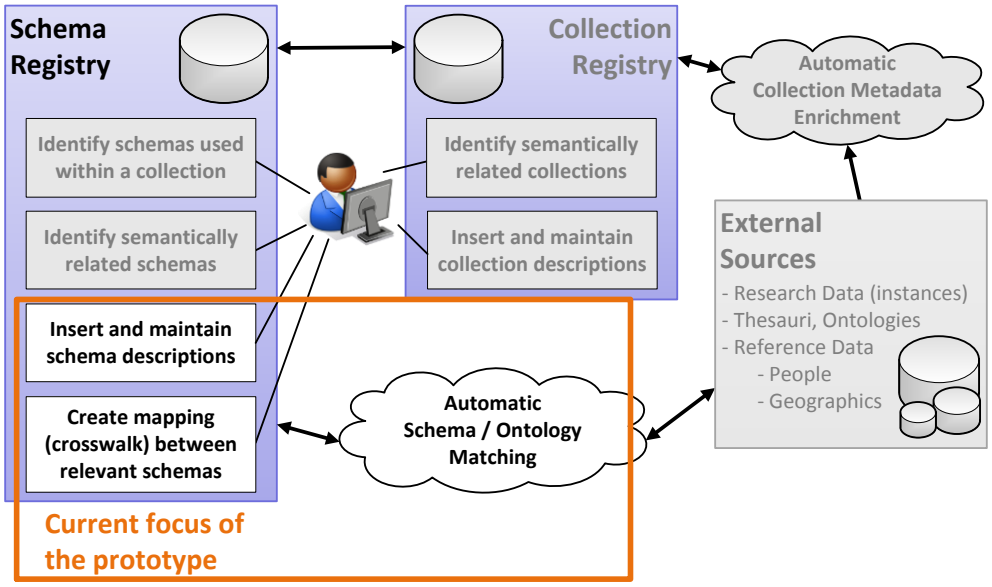

Figure 1: Functionality focus of the current prototype

This document aims to provide an overview of the current implementation state of the Schema Registry as well as the problems and central ideas that form the conceptual base of the prototypic implementation. For this reason, section 2 will first introduce the main problem definition and ambition of the DARIAH federation. Thereafter, section 3 highlights the functionality of the current prototype, concluded by a glance at the next steps and open issues in the development of the Schema Registry in section 4.

[1] A crosswalk is unidirectional and therefore contains only the rules required to transform data defined in schema *A* to its according representation in Schema *B*. The transformation from *B* to *A* requires a distinct crosswalk.

2. Conceptual context

Thoughts and early ideas of the application domain and core functionality of a Schema Registry in DARIAH have been compiled within various documents[2], of which two are conceptual drafts—provided in the DARIAH-wiki:

- *Einordnung und Abgrenzung der Interoperabilitätsanforderungen an Collection-Registry und Schema-Registry als Grundlage für die weitere Abstimmung* (26 May 2011)

- *Modellierung semantischer Assoziationen in Forschungsdaten der Digital Humanities—Analyse der Anwendbarkeit bestehender Ansätze* (28 November 2011)

There are also two publications—one of which has not been published yet—that focus on heterogeneity aspects in the rich context of the Digital Humanities forming the basic understanding of the federation architecture in DARIAH:

- *IR-Unterstützung für die Digital Humanities: Problemstellungen und erste Lösungsideen* (LWA 2011, 28 - 30 September 2011)

- *DARIAH(-DE): Digital Research Infrastructure for the Arts and Humanities—Concepts and Perspectives* (TELDAP 2012, 21-24 February 2012, not yet published)

The following problem definition and contextual integration of the federation infrastructure are largely based on these four documents and aim to condense their respective key aspects. The purpose of this section is, however, not only to explain the context of the current prototype but also to generate an idea of possible directions for further development.

2.1. Problem definition

Since the domain of DARIAH is not limited to a subset of specific disciplines but consists in the Digital Humanities in its whole diversity, central premises and requirements for the concept of the federation architecture and especially the Schema Registry can be formulated:

- *Collections*: The collection landscape of the Digital Humanities consists in digital libraries, museums and archives that are characterized by their autonomy and heterogeneity in respect of the utilized schemas and provided access protocols. Although DARIAH develops and establishes best-practices and recommendations on the usage of standards and schemas, an appropriate integration strategy cannot depend on the alignment of collections to DARIAH, but needs to consider a flexible adaption of the federation layer to individual collections of the Digital Humanities.

- *Schemas and Ontologies*: Research data of the Digital Humanities is described within numerous different structures that DARIAH needs to generically support. Depending on individual disciplines, research data might be best described within the structures of the TEI header in a literary context, ADeX for archaeological metadata or LIDO for

[2] For links to these documents/conferences please see the *Links* sections at the end of this document

the description of Cultural Heritage objects, which prevents the definition of *one* central schema or ontology.

- *Context*: The description of a collected item is produced with the implicit background knowledge of a scholar, discipline, organization and collection, which is typically not explicated in the metadata of an object. Combining research data from multiple disciplines implies that not only schemas but perspectives on individual items need to be aligned.

- *Schema ambiguities*: There are ambiguities within the fields of metadata schemas: On a syntactical level, the formats for e.g. name and date representations may vary. Semantic heterogeneity occurs if fields are filled with different interpretations of the suggested content. If for instance two collections use Dublin Core to describe cultural heritage objects, the field *dc:creator* could be used to refer to the creator of the real-world object or the creator of the record. The transformation rules in crosswalks are hence not only depending on schemas but also on the individual usage of schemas.

- *Support for specific services*: DARIAH aims to create a shared infrastructure that can easily be extended by domain- or problem-specific services. In order to minimize information loss, the federation needs to respect research data in its original form and allow a query-time transformation into the schema, ontology or vocabulary required to answer a research question.

Considering the full spectrum of the humanities, enabling factors of interdisciplinary research, such as the alignment of metadata schemas and support of access strategies are even more important to DARIAH than in more restricted, discipline-specific contexts.

2.2. Enabling federated services

Collections, schemas and crosswalks define fundamental building blocks for the support of interdisciplinary research in the digital humanities. The efficiency of services that are based on the DARIAH federation layer highly correlates with the quality of the metadata stored in both the Collection and Schema Registry. In particular, the task of answering research questions based on data that is distributed among multiple collections or disciplines is dependent on high quality crosswalks.

Hence, the registration and management of collections in the Collection Registry and of schemas and crosswalks in the Schema Registry does not only answer its self-purpose but aims to enable comprehensive services that are dedicated to the scholar in the humanities and arts. The DARIAH e-Infrastructure distinguishes between:

- *generic questions and tasks* such as breadth-oriented search queries spanning numerous collections and resources, e. g. the generic search, and

- *specific questions* limited to the specific perspective of a research question or methodology (e. g. in specific demonstrators)—requiring the deep analysis of research data.

To support the development of specific services and their depth-oriented perspectives, DARIAH needs to minimize the possibility of technically-induced *information loss* by ac-

knowledging and supporting the diversity of the schemas, research data and the research methodologies and allowing the definition of crosswalks between individual schemas and even individual discipline- or organization-specific interpretations of schemas.

Use case: Generic search

The generic search aims to facilitate search based on the federated data of the Digital Humanities. Figure 2 indicates the current concept of a meta-search that suits the requirements of a loosely coupled collection federation. On the basis of a distributed information retrieval architecture, *the Metasearch Component* delegates queries to the available search APIs of the collections (Collection *X*) or—in case collections only provide metadata harvesting (Collections *F* and *W*)—performs search on indices hosted within DARIAH.

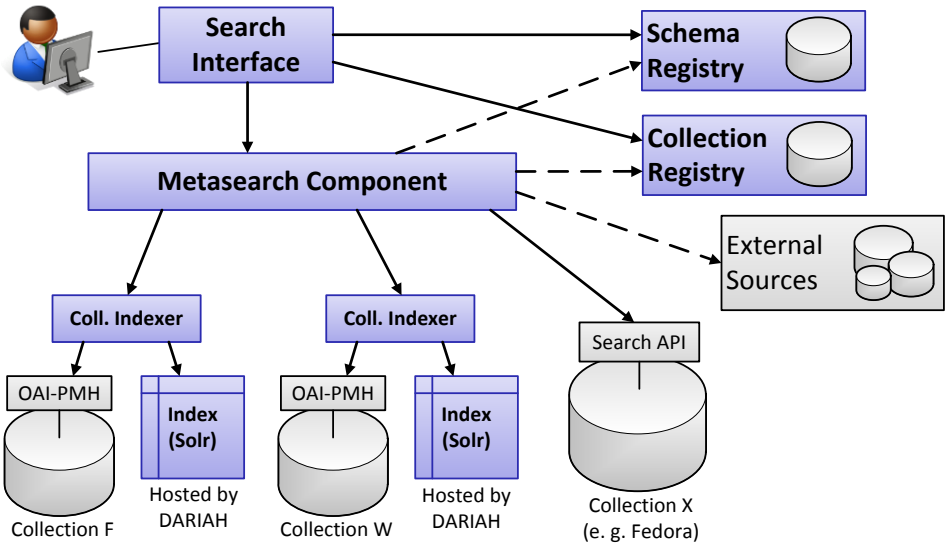

Figure 2: Generic Search in DARIAH

This methodology is usable both for breadth-first and depth-first search and can provide a shared interface for accessing the research data of the digital humanities—independent of the technical details whether the collections only provide distributed querying or harvesting access: The *Search Interface* (1) interacts with the *Schema* and *Collection Registry* and (1a) researchers identify the collections that might contain relevant data. Based on the selected collections the *Search Interface* (1b) interacts with the *Schema Registry* to identify the depth of crosswalks between those collections, allowing the user to adjust facets of the search. (2) The *Search Interface* then assigns the parameters to the *Metasearch Component* which executes a distributed query.

By providing both breadth- and depth-oriented functionality, the generic search can assist the researcher of the digital humanities in finding relevant research data and collections. However, depending on the usage, even the *generic* search can depend on deep associations between—in an extreme case: *individual*—schemas to provide detailed search facets to the user.

3. Prototyping

The first prototype of the Schema Registry has been constructed as a web-application—developed in Java within a Tomcat6/JDK6/Eclipse/Maven development environment.[3] The mapping interface of the Schema Registry is currently based on OpenII Harmony. Functionality for the import and analysis of schemas and associations is based on and extends further infrastructural components of the OpenII. The preliminary choice of Harmony has been taken due to its availability under the Apache 2.0 license allowing for an extensible application to the needs of the Digital Humanities.

In contrary to current state-of-the-art systems in schema and ontology matching, the DARIAH Schema Registry has two main characteristics devoted to the domain of the Digital Humanities and the expected user group—the domain experts in the arts and humanities:

(1) *Web-based access*: The identification and discussion is assumed to be a collaborative task of users with different, non-technical backgrounds. By integrating the mapping interface in a larger web-based application, the Schema Registry (1) can be extended to a collaboration platform, (2) reduce the technical complexity for the end user and (3) leads to an easier extension and maintenance of the logic and interfaces.

(2) *Workflow-orientation*: With the use of algorithms for the identification of relevant correlations in the schemas and research data, the Schema Registry aims to point the scholar to the cognitively demanding tasks that require the interaction with domain experts. The current prototype unravels the first block of complexity that is the creation of crosswalks and reduces the task to (1) the registration of schemas, (2) the nomination of source and target schemas for new crosswalks and (3) the validation of algorithmically generated candidate-associations. The technically complex tasks of transforming external schemas into the data model of the registry—between steps (1) and (2)—and the automatic identification of associations—between steps (2) and (3)—are processed asynchronously and do not expect any feedback from the user.

3.1. Technology Stack

The current implementation of the Schema Registry is based on multiple technical components and frameworks (see figure 3).

- *Spring framework*: Spring 3 has been chosen as base development framework since it assists in the implementation of a Java web-application on all application layers and provides multiple aspects that allow easy extension and maintenance of the registry. Aside from current functionality provided in services and consumed only by the prototypic GUI, Spring 3 eases the integration of the Schema Registry by modules such as Spring Security (for integration with the DARIAH AAI concept) and support for RESTful services.

- *Persistence*: The open-source, object-relational DBMS *PostgreSQL* has been chosen as backend database for the Schema Registry due to the requirements of Harmony.

[3] Web references to used frameworks are provided in the *Links* section at the end of this document

Upon the completed migration of the Harmony model, a variety of databases will be supported. The popular, object-relational mapper (ORM) Hibernate is used to keep the application largely independent of database specifics.

- *Model View Controller (MVC)*: Spring 3 supports web-application development with the Spring MVC framework[4], concentrating the business objects and logic largely in the model, the web-related server activity in the controller and the Java Server Pages (JSPs) in the views of the application. For the resolution of views and the use of view templates, the prototype builds on Apache Tiles.

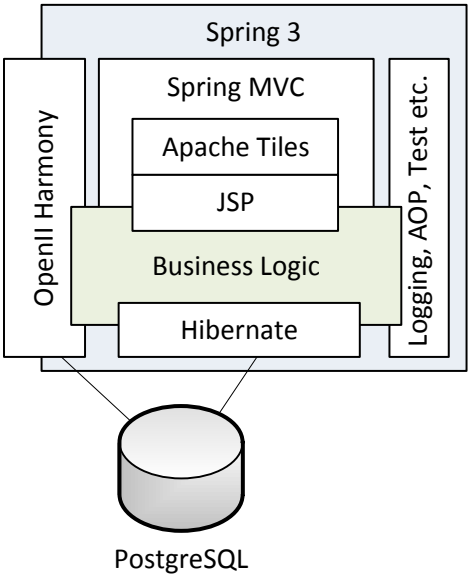

Figure 3: Technology stack used for prototyping

- *Business Logic:* Not all the business functionality is concentrated in the model. Algorithms for the extraction of information from user-supplied schemas and the automatic analysis of associations is realized in a package that the web application references. This package—in the current development state—uses and interacts with the logic and model of OpenII Harmony.

- *Mapping interface:* OpenII Harmony has been integrated as a Java Applet. The schema mapping interface of Harmony communicates with the Schema Registry based on servlet requests—the same interface that is used for JSPs and the RESTful consumption of services.

- *Further modules* of Spring and *other libraries* handling generic functionality such as logging (log4j over slf4j), validation (hibernate-validator) or testing (jUnit, mockito) as used in the prototype.

[4] or Spring Web Flows for that matter.

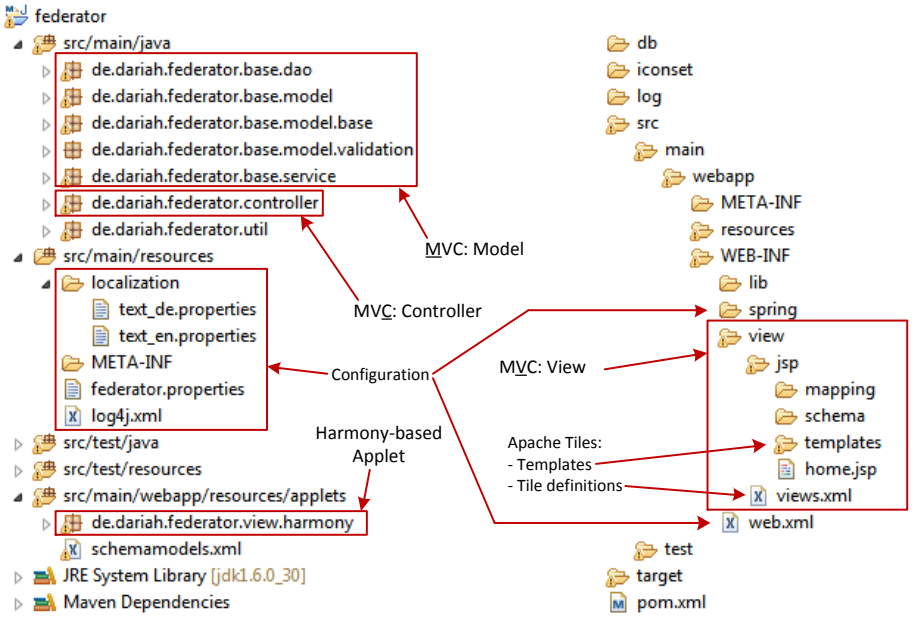

Figure 4: Package structure

The package *de.dariah.federator.base* (see figure 4) contains the elements of the model in an MVC pattern divided in three sub-packages each handling particular aspects of the ***model*** (1) the entities that are persisted (*.model)*, along with their abstract superclasses (*.model.base*) and validation functionality (*.model.validation*), (2) data access objects (*.dao*), which encapsulate the CRUD-operations on individual classes and (3) services (*.service*) spanning transactions over multiple DAOs.

Within *de.dariah.federator.controller* the controllers of the MVC pattern are implemented, the JSP views. Templates and Apache Tiles configuration are managed in a *view* subfolder of the WEB-INF directory. Furthermore, *de.dariah.federator.util* includes infrastructural functionality that is not directly related to the business-logic of the Schema Registry such as—currently—the multilingualism of the interface. The integration of the Harmony GUI in form of an applet is provided within the *de.dariah.federator.view.harmony* package.

3.2. Current functionality

Implemented logic and interfaces of the prototype can be arranged in the three functionality blocks that are visible to the user:

- *Schema Registry*: Includes CRUD-operations on schemas and especially the registration of new schemas by supplying readable[5] sources.

- *Crosswalk Registry*: CRUD-operations on crosswalks—the associations between schemas. By providing automatically generated candidate-associations, the creation of crosswalks is bound to some logical complexity.

- *Mapping interface*: User interface based on two associated tree-widgets for the manual validation and correction of crosswalks.

[5] XML schema supported, other file types are planned

3.2.1. Schema Registry

The screenshot in figure 5 shows an overview of the schemas that are managed by the registry. In the displayed test-setup, four XML Schema definitions[6] have been parsed, analyzed and transformed into the generic data model of the Schema Registry (see figure 6 for the relevant sub-package of the model). The data model should not be interpreted as an integrated schema but as a meta-model that allows a flexible mapping and persistence of schemas and crosswalks in the Schema Registry.

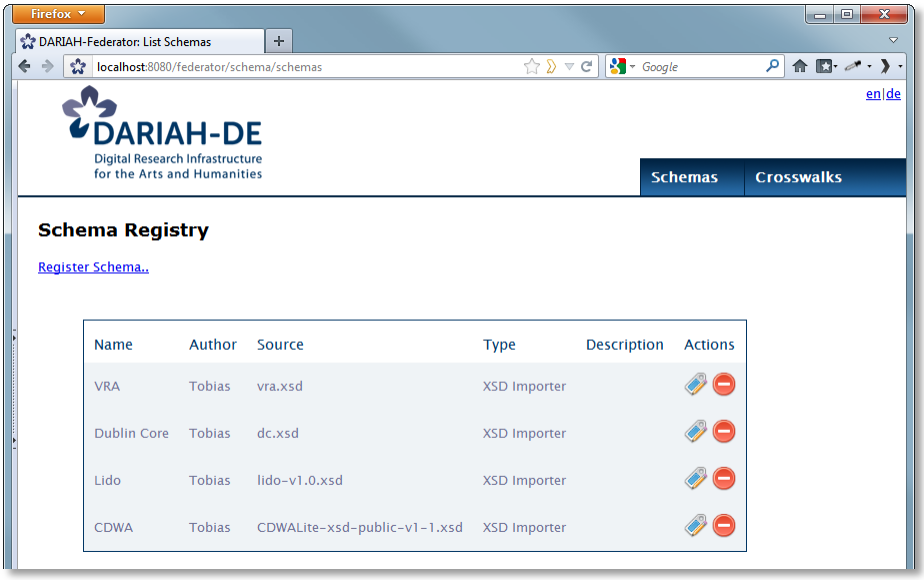

Figure 5: View on registered schemas

Read, update and delete operations on schemas are currently implemented to operate on the metadata on the schema, not on individual elements.

Consistency needs to be ensured on two levels:

- *Standards*: write operations on the element-level should—even in further iterations of the registry—only be possible with a user providing a new schema source, e. g. by pointing to the URL, where the schema can be found.

- *Crosswalks*: the deletion of schemas and their modification on an element level should only be allowed if the schema is not bound to its usage in a crosswalk or collection. Write operations should lead to new versions of schemas inheriting relevant crosswalks from their respective predecessors. Schema versioning and inheritance is, however, not implemented in the current version.

Despite the simplicity of the current interface and the read, update and delete functionality provided, the registration of a new schema implies a detailed analysis of an external file, the transformation of the schema into the data model of the registry and the persistence within that model.

[6] see the „Links“ section of this document for further information on VRA, Dublin Core, Lido and CDWA.

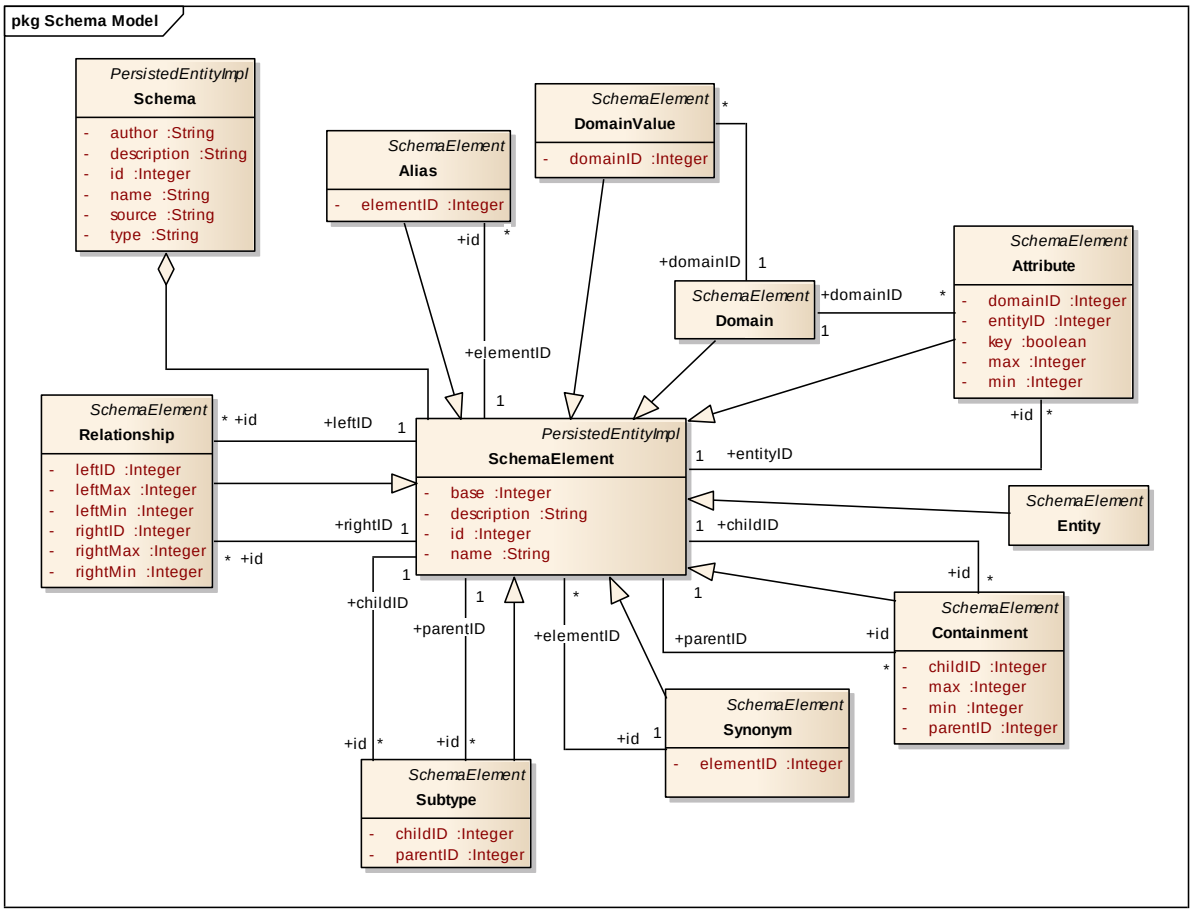

Figure 6: Generic schema persistence meta-model (simplified)

The class diagram in figure 6 shows the simplified classes of the meta-model that is relevant to the management of schemas:

- *Schema*: Serves mainly as container class for schema elements but contains some metadata on the schema as a whole, such as name, author, description and source.

- *Schema Elements*: Implementations of the abstract *SchemaElement* include multiple types of elements that describe the content (e. g. Alias, Attribute) and structure (Containment, Relationship, Subtype) of the schema in a detailed but generic fashion.

3.2.2. Crosswalk Registry

The Schema Registry serves as container for (1) the schemas used by any of the collections registered in the Collection Registry and (2) the associations (crosswalks) between the schemas. The definition of a crosswalk between schemas in DARIAH is compatible with and can be based on the state-of-the-art in schema and ontology matching, in which individual mappings between (groups of) fields are identified and modeled on the schema level.

Figure 7 provides a screenshot of the crosswalk overview within the current prototype. Like the schema overview, the interface is currently held simple and provides only basic information on the crosswalk, such as the source and target schema and a count (used primarily as an indicator for the success of algorithmic calculations) of individual associations that have been identified by the system or the user.

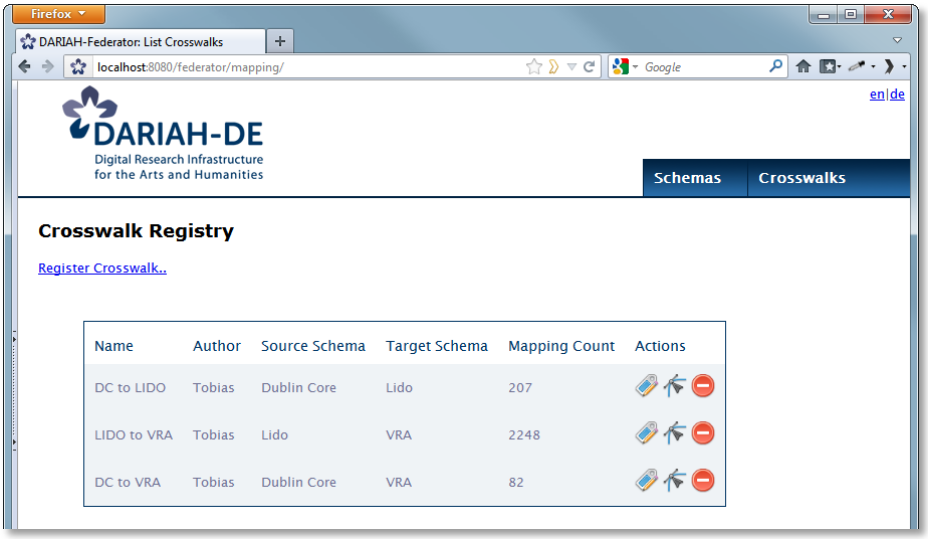

Figure 7: View on registered crosswalks

To create a crosswalk, the user needs to select the source and target schema and provide metadata on the crosswalk. With the submission of this information, the registry initiates the algorithmic analysis of the selected schemas and their potential associations based on the name similarity (exact and edit-distance) of schema elements. Multiple matchers are provided with Harmony, which provide some simple algorithms. For a semantics-driven analysis of schema associations, however, information (1) on the schemas (labels, structure), (2) the research data (instances) existing within these schemas and (3) external sources such as thesauri, ontologies and reference data should be combined—the latter being provided by work package 1.2 in the form of *Reference Data: Persons* and *Reference Data: Geo*.

The class diagram in figure 8 shows the (simplified) subset of the class model that is relevant to the persistence of crosswalks and their usage in end-user services that build upon the federation layer of DARIAH:

- *Mapping* symbolizes the crosswalk as a whole, which in itself does not contain any information about correlations between the schemas but metadata such as the name and author of the mapping as well as the attached source and target schemas. Furthermore, a *Mapping* serves the purpose of a container for instances of *MappingCell*.

- *MappingCell* represents an individual logical association between schemas and is assigned a confidence score and an implementation of a function for the execution of a transformation. A *MappingCell* is bound to an element of the target schema but has the capability of combining the values of multiple input values (*MappingCellInput*) from the source schema. Logically, an N:M mapping between source and target schemas can therefore be represented.

- *MappingCellInput* finally represents either a schema element of the source schema or a constant that is used within the transformation specified by the function associated with the particular *MappingCell*.

- *Function* is the actual transformation rule, that uses the *MappingCellInput* instances attached to a particular *MappingCell* and determines the respective output value for the target schema. Exemplary functions can consist in simple numeric calculations or regular expressions (e. g. combining first name and last name values of the source to a full name value in the target schema)[7]

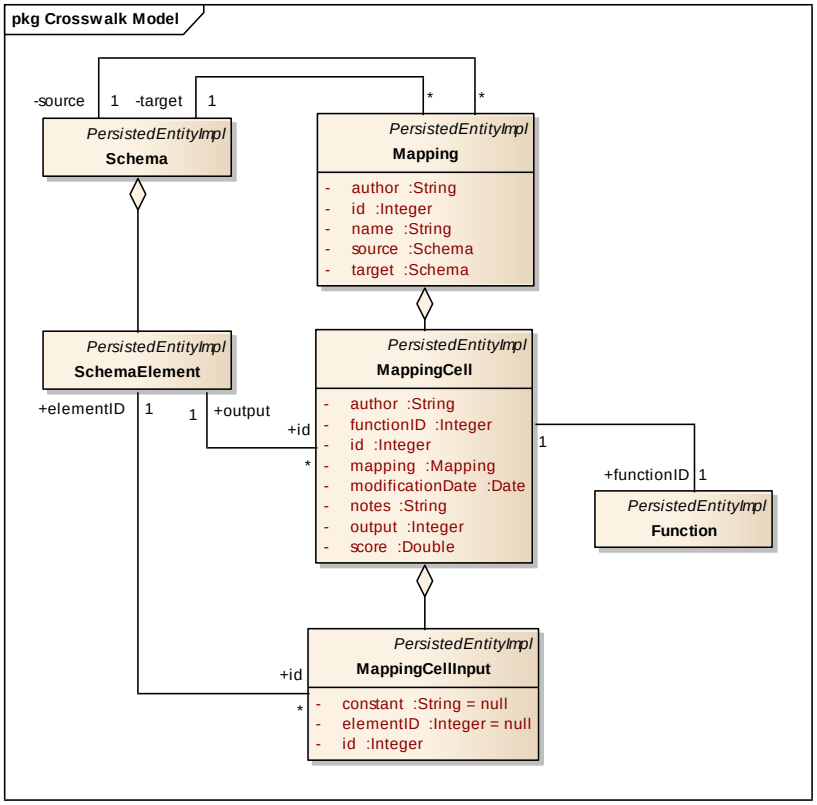

Figure 8: Crosswalk-related classes of the meta-model (simplified)

3.2.3. Mapping Interface

After the manual steps of selecting the source and target schemas of a crosswalk and the automatic detection of associations, initial crosswalks with candidate-associations are prepared in the registry. Independent of the quality of individual associations, domain experts need to be able to perform a manual validation of a crosswalk, including the correction of false positives and the identification of associations that algorithms might have missed.

The screenshot in figure 9 shows the schema alignment functionality currently implemented in the prototype. A Java applet that is based on the user interface of OpenII Harmony is integrated in the web interface of the Schema Registry. The applet interacts with a controller (compare figure 4) of the Schema Registry to access the schema and crosswalk information of the registry.

[7] Note: Functions can depend solely on the schema or the interpretation of a schema by an institution, discipline or individual. For instance, the *dc:creator* field could be filled with various name representations or even with a pointer to a concept in an ontology. The transformation therefore can depend on the schema, the collection and the individual instance. The current state-of-the-art, however, is only focusing on the schemas.

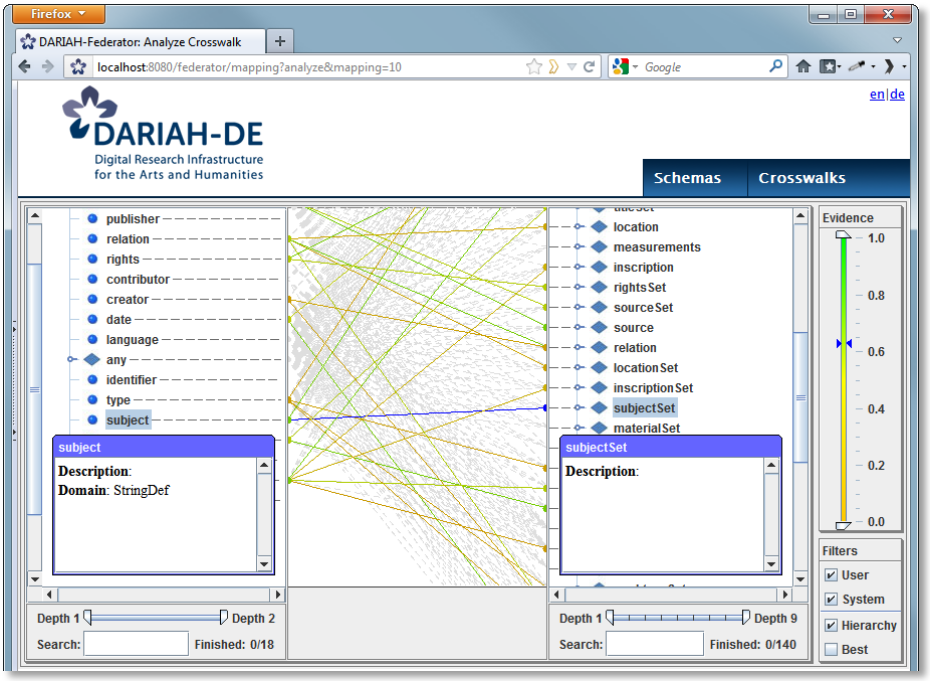

Figure 9: Manual validation and correction of crosswalks

4. Next Steps

The current version of the prototype implements basic functionality to manage schemas and crosswalks in a central registry. In addition to the registry in its common sense the current implementation contains a web-based interface for the management of schemas and crosswalks—integrating the schema mapping interface of OpenII Harmony.

Along with improvements on the stability and interactivity of the prototype—especially when handling large schemas within the interface of Harmony, the tasks that should be considered as possible next steps of development, can be grouped in:

- *Algorithmic support*: By separating algorithmic and interactive phases of the alignment workflow, the base for the integration of more powerful algorithms has been provided—enabling not only a more effective detection of associations but sophisticated functionality such as the clustering of schemas and ontologies resulting e. g. in recommendations which collections and/or schemas should be associated.

- *Visualization techniques*: Along with richer algorithms, the Schema Registry needs to provide interfaces that facilitate collaboration e. g. by highlighting schemas and crosswalks that need the attention of the domain expert that is currently logged in.

- *Execution of crosswalks*: Schemas and crosswalks are part of the federation strategy of DARIAH. However, they get utilizable by higher-level services that consume the services and transformation rules. The generic search, for instance will use the mappings that are defined on the schema level to translate between schemas at query-time to ensure the greatest flexibility regarding the generic/specific character of posed questions.

Links

Links were last accessed on 1 March 2012

Documents

DARIAH(-DE): Digital Research Infrastructure for the Arts and Humanities—Concepts and Perspectives;
http://collab.teldap.tw/teldap2012/

Einordnung und Abgrenzung der Interoperabilitätsanforderungen an Collection-Registry und Schema-Registry als Grundlage für die weitere Abstimmung (conceptual draft);
https://dev2.dariah.eu/wiki/download/attachments/2295602/AP1_2_Interoperabilit%C3%A4t.pdf?version=1&modificationDate=1326219424816

IR-Unterstützung für die Digital Humanities: Problemstellungen und erste Lösungsideen;
https://dev2.dariah.eu/wiki/download/attachments/2295727/Gradl_Henrich_LWA11_Papier.pdf?version=1&modificationDate=1326219474322 and http://lwa2011.cs.uni-magdeburg.de

Modellierung semantischer Assoziationen in Forschungsdaten der Digital Humanities – Analyse der Anwendbarkeit bestehender Ansätze;
https://dev2.dariah.eu/wiki/download/attachments/2295602/Modellierung_semantischer_Assoziationen.pdf?version=1&modificationDate=1326219424715

Mentioned and used standards (test-case)

Archaeological DataeXport-Standard (ADeX);
http://www.landesarchaeologen.de/verband/kommissionen/archaeologie-und-informationssysteme/projektearbeitsgruppen/adex/

Categories for the Description of Works of Art (CDWA): the test-setup uses CDWA lite, which is based on the Categories for the Description of Works of Art (CDWA) and Cataloging Cultural Objects: A Guide to Describing Cultural Works and Their Images (CCO);
http://www.getty.edu/research/publications/electronic_publications/cdwa/cdwalite.html

Dublin Core (DC): the test-setup uses the simple *Dublin Core Metadata Element Set, Version 1.1*; http://dublincore.org/documents/dces/

Lightweight Information Describing Objects (LIDO): schema that combines the efforts of multiple standardization initiatives, including the CDWA lite schema;
http://cidoc.mediahost.org/WG_Data_Harvesting%28en%29%28E1%29.xml

VRA Core: schema for describing works of visual culture and their documenting images;
http://www.loc.gov/standards/vracore/

Frameworks and Tools

Apache Maven;
http://maven.apache.org/

Apache Tiles;
http://tiles.apache.org/

Apache Tomcat;
http://tomcat.apache.org/

Eclipse;
http://www.eclipse.org/

Hibernate;
http://www.hibernate.org/

Java Development Kit (JDK);
http://www.oracle.com/technetwork/java/javase/overview/index.html

Java Server Pages (JSP);
http://www.oracle.com/technetwork/java/javaee/jsp/index.html

Open Information Integration (OpenII);
http://openii.sourceforge.net

PostgreSQL;
http://www.postgresql.org/

Spring 3;
http://www.springsource.org/