# ALIGNED

## Aligned, Quality-centric, Software and Data Engineering

**H2020 – 644055**

# D3.5 – Model Catalogue Tool, Phase 3

| | |
|---|---|
| **Project Number:** | 644055 |
| **Project Title:** | ALIGNED |
| **Document Type: (Deliverable/Internal):** | Deliverable |
| **Deliverable Number:** | 3.5 |
| **Deliverable Type (R/DEM/DEC/OTHER):** | Other |
| **Workpackages Contributing:** | WP3 |
| **Dissemination Level (PU/CO):** | Public |
| **Contractual Delivery Date:** | February 2017 |
| **Actual Delivery Date:** | February 2017 |
| **Version:** | 1.0 |
| **Editor(s)/Lead Authors:** | James Welch, Seyyed Shah |
| **With contributions from:** | Jim Davies, Jeremy Gibbons, Monika Solanki, Steve Harris |
| **Reviewers(s):** | Odhran Gavin |

**Abstract:**

This document describes further development of the Model Catalogue tool to capture, manage and document metadata. The tool will be used to capture independent models of metadata.

**Keyword List:**

Metadata; model-driven; model

*This communication reflects only the author's view and the Commission is not responsible for any use that may be made of the information it contains.*

Trinity College Dublin
The University of Oxford
Universitaet Leipzig
Semantic Web Company GmbH
Wolters Kluwer Deutschland GmbH
Uniwersytet im. Adama Mickiewicza w Poznaniu

# History

| Version | Date | Reason | Revised by |
|---|---|---|---|
| 0.1 | 1/2/2017 | First Draft | James Welch |
| 0.2 | 13/2/2017 | Full Internal Draft for Review | James Welch |
| 0.3 | 14/2/2017 | Revised Internal Draft for Final Review | Odhran Gavin |
| 1.0 | 24/2/2017 | Final Version | Kevin Feeney |

# Author List

| Organisation | Name | Contact Information |
|---|---|---|
| OxSE | Seyyed Shah | Seyyed.shah@cs.ox.ac.uk |
| OxSE | James Welch | James.Welch@cs.ox.ac.uk |

# 1 Executive Summary

This document presents progress on the design and implementation of the ALIGNED Model Catalogue, a tool for capturing, managing and documenting metadata. Once high-quality metadata is documented, the data captured against it may be reused, thus increasing the value of that data. Creating software that captures, manages and analyses the data becomes simpler where detailed metadata is available. This report reports on development progress during the second phase of the ALIGNED project, including updates for usability, integration with the Eclipse Modeling Framework, and tools focussed on the ALIGNED metamodels. This builds on work previously documented in Deliverable 3.2 – Model Catalogue [1].

## Contents

## Table of Figures

## 2 Introduction

This document presents progress on the design and implementation of the ALIGNED Model Catalogue, a tool for capturing, managing and documenting metadata. Careful management of metadata is essential for the effective reuse of data and the correctness of any software designed for processing the data. Metadata may capture best practice in a domain and as such the reuse of metadata can proliferate best practice. Traditionally, most metadata is usually captured implicitly, and embedded in the software or system that use the data. Motivation for developing a tool for managing and curating metadata includes data and software interoperability, documenting metadata as models and building a platform for automatically generating software systems from models.

The Model Catalogue is an online tool that supports the capture and documentation of metadata as generic and reusable models. The tool facilitates collaboration between metadata creators and potential users. The system defines a core language for describing metadata, which enables sharing, documentation and reuse of metadata. The tool uses standards-based concepts for registration and versioning and a standard four-level architecture with an API for interoperability with external tools.

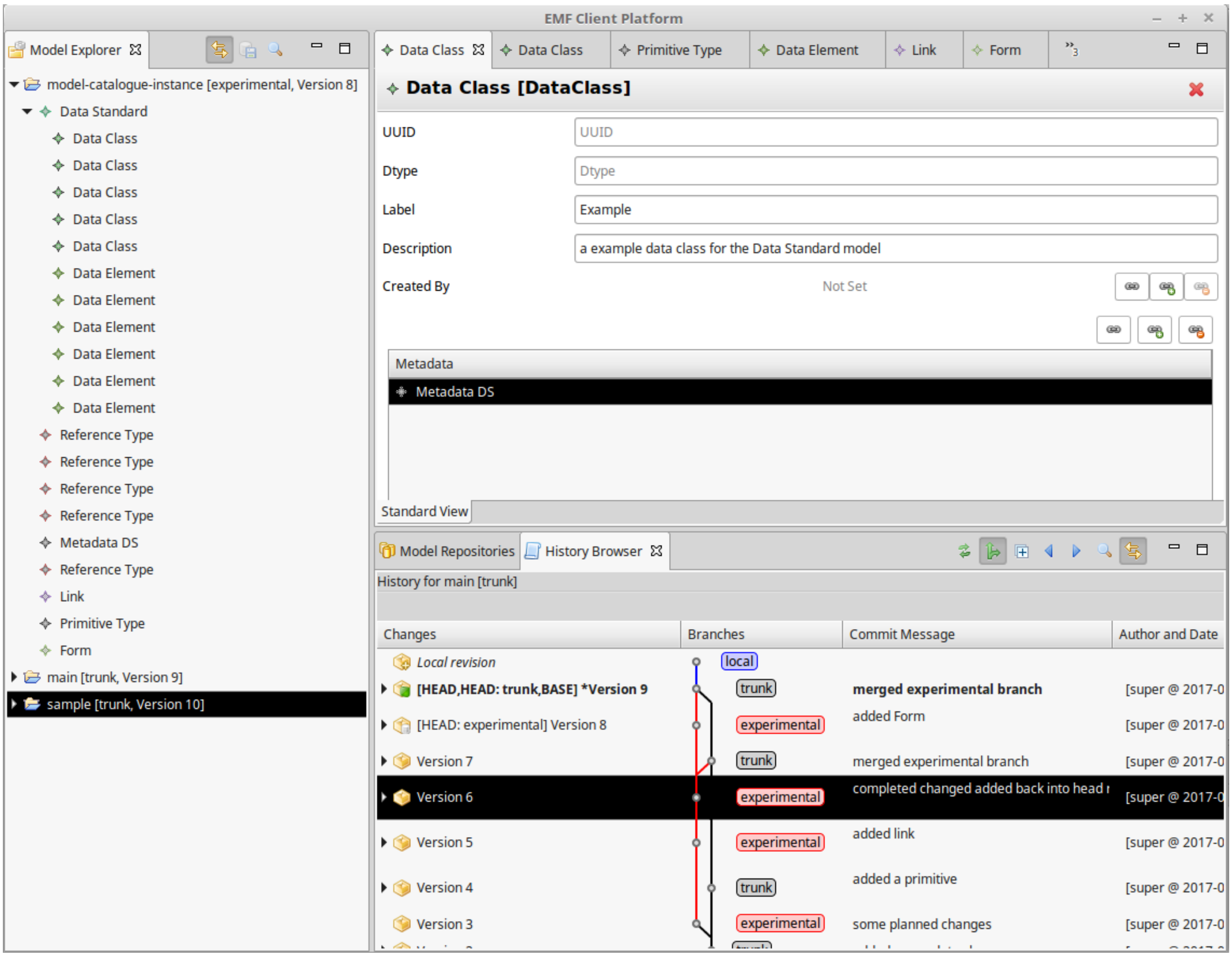
An initial version of the tool was built in Phase 1 of the project, and presented as a prototype for evaluation within the Seshat and Wolters-Kluwer use cases. This work was documented in Deliverable 3.2 – Model Catalogue [1] and the next steps outlined there. This document reports on some of the advancements to the technology during Phase 2 of the project, made in response to user feedback and in anticipation of the tool being used as part of the use cases in Phase 3 of the project. This report highlights progress on three key areas of technological advancement: integration with the Eclipse Modeling Framework, semantic reasoning using an RDF view on the catalogue data, and new features for automation and search. We report on how the design of the catalogue is evolving to account for the particular use-cases within the ALIGNED project, and discuss the remaining development for Phase 3 of the project.

# 3 Eclipse integration and model-driven development

We have begun the integration of core parts of the catalogue functionality with the Eclipse Modeling Framework (EMF). EMF is fundamental to the majority of model-driven development tools within Eclipse, and is also used as the basis for domain-specific languages and transformations. This integration will allow existing model-driven tools within Eclipse to take advantage of the catalogue in order to re-use components of models, increasing the speed of development, and allowing data linking and interoperability between tools built within the framework.

Furthermore, the EMF integration allows new Model Catalogue components to be built in a model-driven engineering fashion – the screenshot in Figure 1 shows an automatically-generated interface for interacting with the catalogue data, including automatic change-management to track multi-user updates. Another auto-generated component stores all versions of every model to disk.

As well as an exercise in 'eating our own dog-food', a model-driven approach to generating Model Catalogue components allows experimentation with the underlying metamodel to support the associated computer science research – finding useful paradigms for scalable descriptions of data and novel mechanisms for reuse.



model-catalogue-instance [experimental, Version 8]
Data Standard
Data Class
Data Class
Data Class
Data Class
Data Class
Data Element
Data Element
Data Element
Data Element
Data Element
Data Element
Reference Type
Reference Type
Reference Type
Reference Type
Metadata DS
Reference Type
Link
Primitive Type
Form
main [trunk, Version 9]
sample [trunk, Version 10]

**Data Class [DataClass]**

| | |
|---|---|
| UUID | UUID |
| Dtype | Dtype |
| Label | Example |
| Description | a example data class for the Data Standard model |
| Created By | Not Set |

| Metadata |
|---|
| Metadata DS |

History for main [trunk]

| Changes | Branches | Commit Message | Author and Date |
|---|---|---|---|
| *Local revision* | local | | |
| **[HEAD,HEAD: trunk,BASE] *Version 9** | trunk | **merged experimental branch** | [super @ 2017-0 |
| [HEAD: experimental] Version 8 | experimental | added Form | [super @ 2017-0 |
| Version 7 | trunk | merged experimental branch | [super @ 2017-0 |
| Version 6 | experimental | completed changed added back into head r | [super @ 2017-0 |
| Version 5 | experimental | added link | [super @ 2017-0 |
| Version 4 | trunk | added a primitive | [super @ 2017-0 |
| Version 3 | experimental | some planned changes | [super @ 2017-0 |

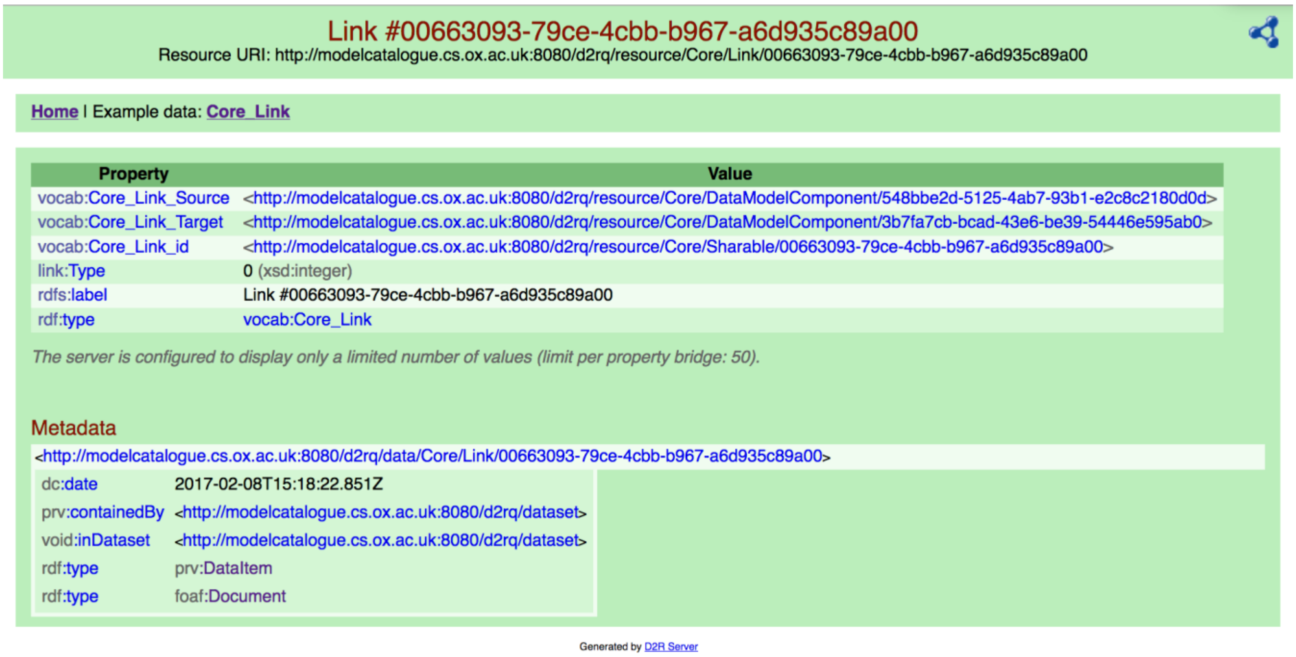
Figure 1: Model Catalogue Eclipse Integration

The core Java implementation of the Model Catalogue has also been enhanced with a view to extensibility and re-use. A new *plugin architecture* has been built, providing extension points through which new functionality can be built and dynamically integrated. For example, two key extension points are those of *Importer* and *Exporter*: developers can write their own importers and exporters using the Model Catalogue Java API to automatically document models in their own language, or to use the catalogue as an interface for compiling new models from existing ones. For example, a developer may choose to write a data model importer that documents the usage of a no-SQL database, and an exporter that generates queries to retrieve that data and insert it into an SQL database.

Further plugins are being developed for bespoke types of data model, and custom interfaces that can be used to display and edit particular types of model in a more familiar fashion. For example, a graphical editor for UML diagrams, or a builder tool for designing data entry forms. This plugin architecture also allows custom configurations of the catalogue to deployed – using just those plugins necessary for the context: providing a better user experience and requiring minimal system resources.

# 4 Semantic reasoning

*Semantic links* are created in the catalogue to associate parts of different models – typically Data Elements – to assert that they are similar in meaning or use. This allows descriptions of meaning to be re-used: by asserting: "this element is the same as that one" a modeller may take advantage of definitions in other models, reducing the effort in documentation. These links also give us the formal notion of 'semantic interoperability': that data from two sources may be combined for a particular purpose.



Link #00663093-79ce-4cbb-b967-a6d935c89a00

Resource URI: http://modelcatalogue.cs.ox.ac.uk:8080/d2rq/resource/Core/Link/00663093-79ce-4cbb-b967-a6d935c89a00

Home | Example data: Core_Link

| Property | Value |
|---|---|
| vocab:Core_Link_Source | <http://modelcatalogue.cs.ox.ac.uk:8080/d2rq/resource/Core/DataModelComponent/548bbe2d-5125-4ab7-93b1-e2c8c2180d0d> |
| vocab:Core_Link_Target | <http://modelcatalogue.cs.ox.ac.uk:8080/d2rq/resource/Core/DataModelComponent/3b7fa7cb-bcad-43e6-be39-54446e595ab0> |
| vocab:Core_Link_id | <http://modelcatalogue.cs.ox.ac.uk:8080/d2rq/resource/Core/Sharable/00663093-79ce-4cbb-b967-a6d935c89a00> |
| link:Type | 0 (xsd:integer) |
| rdfs:label | Link #00663093-79ce-4cbb-b967-a6d935c89a00 |
| rdf:type | vocab:Core_Link |

*The server is configured to display only a limited number of values (limit per property bridge: 50).*

Metadata

<http://modelcatalogue.cs.ox.ac.uk:8080/d2rq/data/Core/Link/00663093-79ce-4cbb-b967-a6d935c89a00>

| | |
|---|---|
| dc:date | 2017-02-08T15:18:22.851Z |
| prv:containedBy | <http://modelcatalogue.cs.ox.ac.uk:8080/d2rq/dataset> |
| void:inDataset | <http://modelcatalogue.cs.ox.ac.uk:8080/d2rq/dataset> |
| rdf:type | prv:DataItem |
| rdf:type | foaf:Document |

Generated by D2R Server

Figure 2: Screenshot showing RDF representation of catalogue contents

In order to reason about this semantic interoperability property, we have found it useful to view the data in terms of triples, such that we can apply off-the-shelf reasoning tools. In order to do this, we have deployed the D2RQ tool (http://d2rq.org) to expose our internal relational data as RDF triples (See screenshot in Figure 2). Although the mapping requires some further customisation for easier use, the mapping is sufficient for us to reason about key properties of the semantic links: circularities in the transitive 'same-as' link, and contradictions in definitions using the 'same-as' and 'not-same-as' links.

A side effect of making this representation available is that the catalogue contents can be linked to other open datasets. Of particular interest to the ALIGNED project, we are interested in how catalogue metadata may be linked to other published artefacts with Dublin Core, how provenance information may be attached with PROV, linking to existing tagging and folksonomies through the Modular and Unified Tagging Ontology (MUTO), and how design intent may be linked to model components using DIO [3].

To assist with this linking, we have also added the ability to add namespaces to metadata elements within the catalogue. These can be used to indicate fields for linking in the RDF representation, or can be used by plugins (as described in Section 3) to configure generated artefacts, such as adding constraints to systems generated with Semantic Booster [2], or shaping XSD outputs to match existing specifications.

## 5 Automation and search

Our experience with previous implementations of metadata registries is that it can be difficult to encourage users to carefully document the whole data model to a level that is sufficient for potential users of the data. In particular, when dealing with models at scale, even simple tasks like finding a data element in another model to link to, or comparing multiple versions, can be complicated and time-consuming. In order to improve usability, we are building a number of features to assist modellers to use the tool effectively and efficiently.

At the heart of this effort is greater power in searching across many hundreds or thousands of data elements, in order to find related items, create semantic links between items, and import or re-use whole model components. We've integrated the popular Lucene and Solr tools (http://lucene.apache.org), which help with indexing, and allow us faster and more flexible searching using keywords, related terms and intelligent suggestions. The speed improvements offered by these tools make the Model Catalogue as a whole scalable for domains with large numbers of complex data models.

Finding similar elements to link to can reduce the time it takes to document a data element. To allow users to find similar items, we've implemented an autosuggest feature which will find potential matches across all models, or a particular model, based on data types, element names, and text matching in the description.

Using the semantic reasoning described in Section 4 can also assist users in creating semantic links, using the transitive properties to help them find related data elements not already explicitly linked. Such reasoning can also help find relations between larger model components: for example linking two data classes where the component data elements of each class are already linked.

Comparing different models is also something that users need extra assistance with – especially comparing multiple incremental versions of the same model. To aid users in this activity, we are implementing a specific web interface, supported by back-end API methods. Viewing two models side-by-side, with differences highlighted, provides a user-friendly experience that will be familiar to those with experience using traditional 'diff' style tools.

# 6 ALIGNED model repository

As part of our commitment to cataloguing the models used in the ALIGNED project, we are building additional functionality to deal with the range of formats used. Plugins to import RDF Schema and the latest version of the Shapes Constraint Language (SHACL) are under construction, and improvements are being made to our existing OWL importer.

In the semantic web community, the visualisation of ontology models is important to understanding how concepts can be related, and it is standard practice to include such visualisations when documenting the model. Our plan is to use VOWL (http://vowl.visualdataweb.org/v2/) to enable the ALIGNED catalogue contents to be visualised in a range of standard off-the-shelf tools.

Work is also underway to provide export formats for the ALIGNED tools that would allow the ontologies to be converted for use with other technologies and tool-chains. Current export formats under construction include RDF/XML, Turtle and JSON-LD.

Our aim is to produce a publically accessible version of the catalogue to persist the ALIGNED metamodels beyond the life of the project.

# 7 Next Steps

In Deliverable 3.2, we planned to perform some qualitative researh into the functionality and usability of the catalogue. To date this has largely been speculative or anecdotal, and so in the final phase of the project we plan to apply a more formal process to ensure that the catalogue meets the needs of our users. This will begin with a basic System Usability Scale (SUS) questionnaire to get a high-level feel for the usability of the catalogue across three core user groups: software developers; those involved in editing and maintaining datasets; and those who would be interested in linking to or re-using datasets described in the catalogue.

The use of the catalogue in the ALIGNED use-cases and trials will be the primary focus for phase 3 of the project. The tool will be used to document the SESHAT data model, and used as a development environment for development and iteration of the Wolters-Kluwer IPG system. The completion and finalisation of the features outlined in this document will support the ongoing use of the tool as part of the ALIGNED methodology, may allow integration

The impact of the catalogue in these use-cases will be reported in Deliverable 6.6 – Evaluation Report on Trials, Phase 3.

Further integration with the Semantic Booster tool is planned, and will be reported in Deliverable 3.6 – Semantic Booster, Phase 3. This will include custom configurations through annotations and namespaced metadata (as described in Section 4).

# 8 References

[1] ALIGNED consortium members, D3.2 – Model Catalogue, ALIGNED Deliverable, 2016

[2] ALIGNED consortium members, D3.3 – Semantic Booster, ALIGNED Deliverable, 2016

[3] ALIGNED consortium members, D2.7 – Metamodel Phase 2, ALIGNED Deliverable, 2016