D3.3 Semantic Digital Archive Prototype

DURAARK

FP7 – ICT – Digital Preservation
Grant agreement No.: 600908

Date: 2014-07-31
Version 1.0
Document id. : duraark/2014/D.3.3/v1.0

Grant agreement number	:	600908
Project acronym	:	DURAARK
Project full title	:	Durable Architectural Knowledge
Project's website	:	www.duraark.eu
Partners	:	LUH – Gottfried Wilhelm Leibniz Universitaet Hannover (Coordinator) [DE] UBO – Rheinische Friedrich-Wilhelms-Universitaet Bonn [DE] FhA – Fraunhofer Austria Research GmbH [AT] TUE – Technische Universiteit Eindhoven [NL] CITA – Kunstakademiets Arkitektskole [DK] LTU – Lulea Tekniska Universitet [SE] Catenda – Catenda AS [NO]
Project instrument	:	EU FP7 Collaborative Project
Project thematic priority	:	Information and Communication Technologies (ICT) Digital Preservation
Project start date	:	2013-02-01
Project duration	:	36 months
Document number	:	duraark/2014/D.3.3
Title of document	:	Semantic Digital Archive Prototype
Deliverable type	:	Software prototype
Contractual date of delivery	:	2014-07-31
Actual date of delivery	:	31.07.2014
Lead beneficiary	:	Catenda
Author(s)	:	Jakob Beetz `<j.beetz@tue.nl>` (TUE) Stefan Dietze `<dietze@l3s.de>` (L3S) Dag Field Edvardsen `<dag.fjeld.edvardsen@catenda.no>` (Catenda) Besnik Fetahu `<fetahu@l3s.de>` (L3S) Ujwal Gadiraju `<gadiraju@l3s.de>` (L3S) Thomas Krijnen `<t.f.krijnen@tue.nl>` (TUE)
Responsible editor(s)	:	Jakob Beetz `<j.beetz@tue.nl>` (TUE) Dag Field Edvardsen `<dag.fjeld.edvardsen@catenda.no>` (Catenda)
Quality assessor(s)	:	Sebastian Ochmann `<ochmann@cs.uni-bonn.de>` Martin Tamke `<Martin.Tamke@kadk.dk>` Östen Jonsson `<osten.jonsson@ldb-centrum.se>`
Approval of this deliverable	:	Marco Fisichella (LUH, Stefan Dietze (LUH)
Distribution	:	Public
Keywords list	:	Semantic Enrichment, Ontologies, Archival, Preservation

Executive Summary

In this report, a description of the first prototype for the Semantic Digital Archive (SDA) is provided. The SDA is split up into a number of separate components that address the Digital Longterm Preservation (DPS) of semantically enriched Building Information Models along with the datasets and vocabularies used for the enrichment. A detailed description of the prototypical implementations of software components, which together form the SDA, is provided that includes a guideline for their use, a description of their integration into the DURAARK workbench prototype and a detailed account of the technical background.

Table of Contents

1 Introduction

In this report a more detailed overview of the M18 prototype implementation of the DURAARK Semantic Digital Archive (SDA) is provided. It builds on the work presented in the preceding deliverables D3.1 - Metadata Schema Extension for Archival Systems[2], and D3.2 - Ontological Framework for a Semantic Digital Archive [1]. In particular, the focus lies on the Semantic Digital Archive (SDA) which aims at enriching and archiving building information models and is composed of three subcomponents:

1. the Semantic Digital Archive Storage **SDAS** component stores semantic enrichments of BIM data as Linked Data. As such, it captures related subgraphs of external Linked Datasets, references to the correlated architectural models as well as the profiles of datasets. Given the evolving nature of Linked Data, it also contains periodic changes occurring in datasets. Although the SDAS is meant to satisfy live queries from users, only most relevant data is provided in the SDAS, while more exhaustive crawls are captured in the Semantic Digital Observatory (SDO).

2. the Semantic Digital Observatory **SDO** component serves as archive of related Linked Datasets which are crawled periodically. This means that the SDO contains all the data from various relevant datasets, and is responsible for tracking evolutionary changes. Such temporal differences in the states of datasets are computed by the SDO, while selected ones are periodically pushed into the SDAS. To facilitate targeted crawling, the SDO profiles relevant datasets (see Section 3) which are stored in the SDAS.

3. **Semantic Enrichment and Metadata Extraction** components which support curation and enrichment activities. ‘Semantic enrichment’ refers to a number of different additions of information to building information handled at different stages and in different processes of the DURAARK framework and its software prototypes. This is explained further in Section 4.

Under the common umbrella term “SDA“ these subcomponents form a part of the DURAARK workbench prototype described in the D2.4 report.
Both, SDAS and SDO target different use cases. The overall vision is to expose Linked Data through the SDA to enable frequent queries through end users and application, while at the same time, enable the archival of all related external datasets. Satisfying both

requirements led us to implement a two-fold storage strategy, where the SDO deploys a relational database to capture periodic and exhaustive crawls of external datasets, while the SDAS deploys a native RDF store, to allow queries through its Linked Data interface (dereferencable URIs, SPARQL endpoint). The exact interaction between the two will be fully realised in the upcoming SDA prototypes.

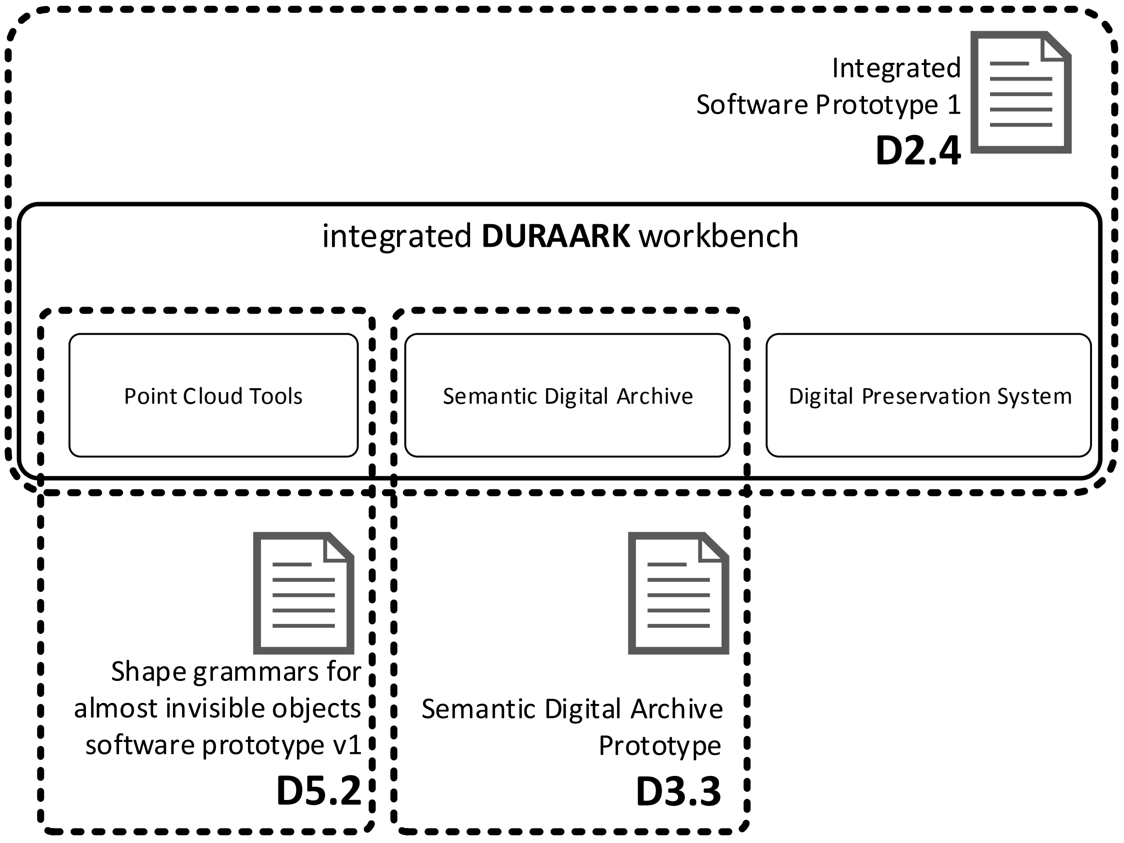
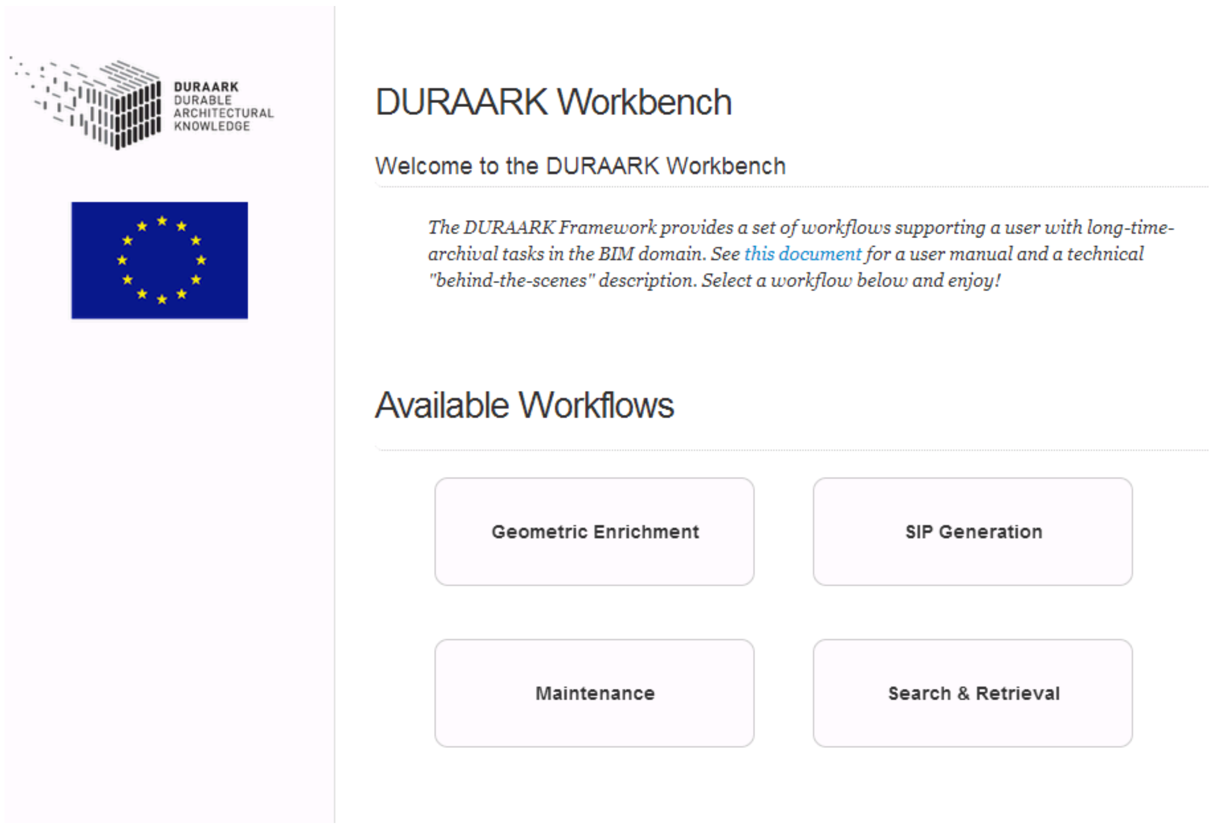


Figure 1: **Schematic overview of the software prototypes of M18/20 milstones.**

As also illustrated in Figure 1, this report documents only aspects of the DURAARK system that are relevant to the semantic enrichment with Linked Data for Building Information Models in Long Term Preservation Systems. The user interfaces, mostly implemented as interactive web pages are integrated into the overall M18 software prototype referred to as the **DURAARK Workbench**. The general system architecture and integration approaches are documented in **D2.4**.


DURAARK Workbench

Welcome to the DURAARK Workbench

The DURAARK Framework provides a set of workflows supporting a user with long-time-archival tasks in the BIM domain. See this document for a user manual and a technical "behind-the-scenes" description. Select a workflow below and enjoy!

Available Workflows

Figure 2: **Screenshot of the workbench web interface of the M18 prototype available at `http://workbench.duraark.eu`**

The SDA consists of a series of subcomponents, each providing in parts their individual interfaces (for instance, SPARQL endpoints allowing to query data programmatically). However, the overall functionality and features are integrated into the DURAARK Workbench `http://workbench.duraark.eu` where functionality can be tested through a Web-based user interface. While the workbench integrates several server-side modules that reside on the Workbench server, other components are running on dedicated machines and provide separate frontends. These are seamlessly integrated into the workbench but can also be accessed individually.

Source code

The source code of the workbench application itself as well as the individual components of the SDA and other tools described in this report are available under Open Source licenses. Additional information such as build and installation guides as well as technical documentation is available in the related wiki pages of the respective repositories as well as the README documentation in the source folders. They can be accessed at the following URLs.

DURAARK Workbench framework: `https://github.com/DURAARK/workbench`

SDO server-side core components: `https://github.com/DURAARK/semantic_digital_observatory`

SDO administration and maintenance User Interfaces: `https://github.com/DURAARK/SDO-WebGUI`

Contextual Enrichment `https://github.com/DURAARK/ifcContextualEnrichment`

IFC metadata and enrichment extractor: `https://github.com/DURAARK/pyIfcExtract`

E57 metadata extractor: `https://github.com/DURAARK/e57Extract`

Scope and use cases

The scope of this initial first prototype is limited to a subset of the use cases that have been defined in the earlier report D2.2.1. Of these, the following use cases are addressed:

- **UC1:** Deposit 3D architectural objects
- **UC3:** Maintain Semantic Digital Archive
- **UC9:** Enrich BIM/IFC model with metadata from a repository

UC1 is addressed by the integrated workbench prototype, for which figure 3 provides an overview of the sequential order of interactions with the individual components. The SDO, SDAS and enrichment components are integrated into the workbench and part of the overall prototype.

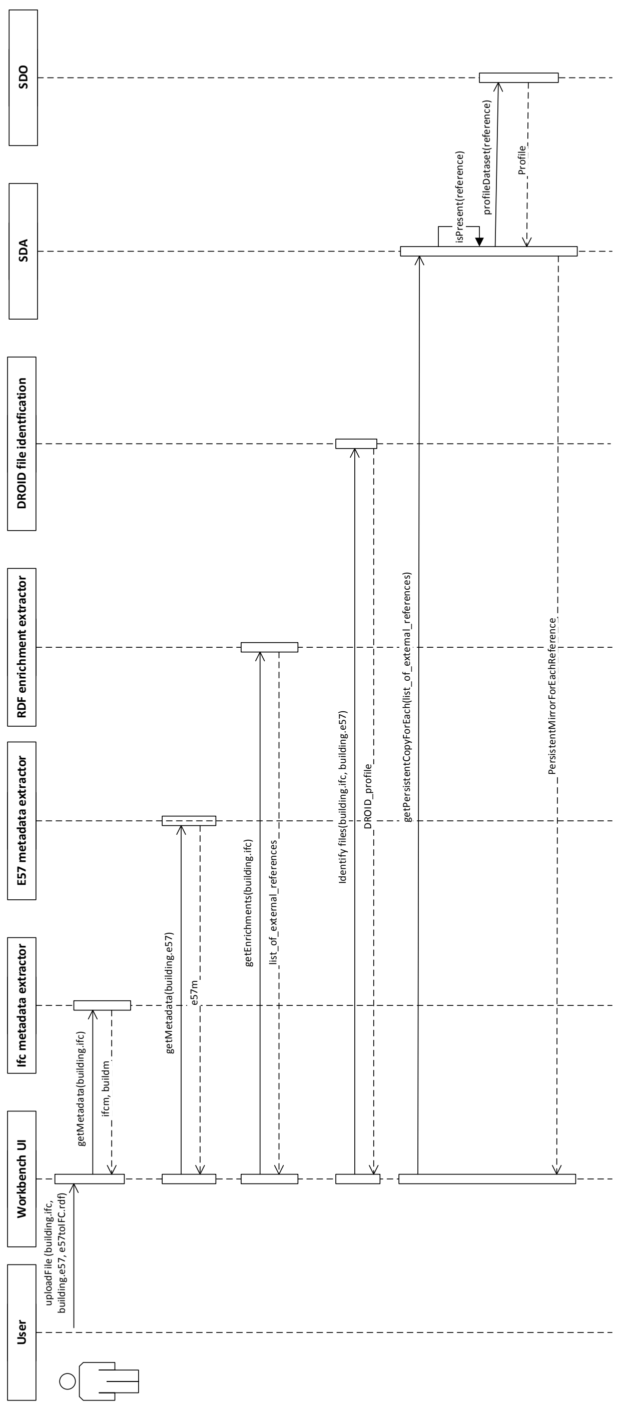

User
Workbench UI
Ifc metadata extractor
E57 metadata extractor
RDF enrichment extractor
DROID file identification
SDA
SDO
uploadFile (building.ifc, building.e57, e57toIFC.rdf)
getMetadata(building.ifc)
ifcm, buildm
getMetadata(building.e57)
e57m
getEnrichments(building.ifc)
list_of_external_references
Identify files(building.ifc, building.e57)
DROID_profile
getPersistentCopyForEach(list_of_external_references)
isPresent(reference)
profileDataset(reference)
Profile
PersistentMirrorForEachReference

Figure 3: **Overview diagram presenting the different components of the DURAARK M18 prototype addressing the main use case UC1 including the WP3 components. The latter are further specified and broken down into their subcomponents in following sections of this document.**

For a general conceptual introduction to the concepts of the SDO, SDAS and underlying metadata, please refer to earlier deliverables, in particular D3.1[2] and D3.2[1] that contain detailed descriptions of the general concepts.

Information and data concepts

Four main distinct concepts are handled in this prototype implementation which are illustrated in figure 4.

- **Physical asset** refers to the building that is preserved in the DURAARK system. For example, the 'Cologne Cathedral' is a building located at a specific geographic location in the city of Cologne, Germany. It is a tangible object that is still in existence. Is has undergone long periods of construction, transformation and rebuilding. It is operated, maintained, used and has been the setting of historic events and contains many well-known objects and building parts of high cultural value.

- **Data objects** are used to represent and preserve many different aspects and states of the building. Next to many common audiovisual representations, two forms of representations are in the focus of the DURAARK project: Building Information Models created by engineers as well as other domain experts. These are captured in the **IFC SPF** format and point cloud data sets measured using different hardware devices and captured in the **E57** file format

- **Metadata** about both the *Physical assets* as well as the *Data objects* are gathered and created in many different ways and captured in the **buildm** metadata schema that can be stored on varying levels of granularity and transformed to e.g. be integrated existing metadata records in archival systems. For a more detailed overview see section 4.3

- **Linked Data** is contained in and makes up part of the metadata to describe and enrich the physical assets and the data objects used for their preservation. Examples for such Linked Data item are references to common vocabularies and ontologies such as DBpedia, Ghetty AAT, the buildingSMART Data Dictionary or GeoNames that are used to describe the assets, data objects and metadata items. Other forms of Linked Data include e.g. product information from manufacturers, the larger context of a buildings' environment or curated information such as related architectural styles.

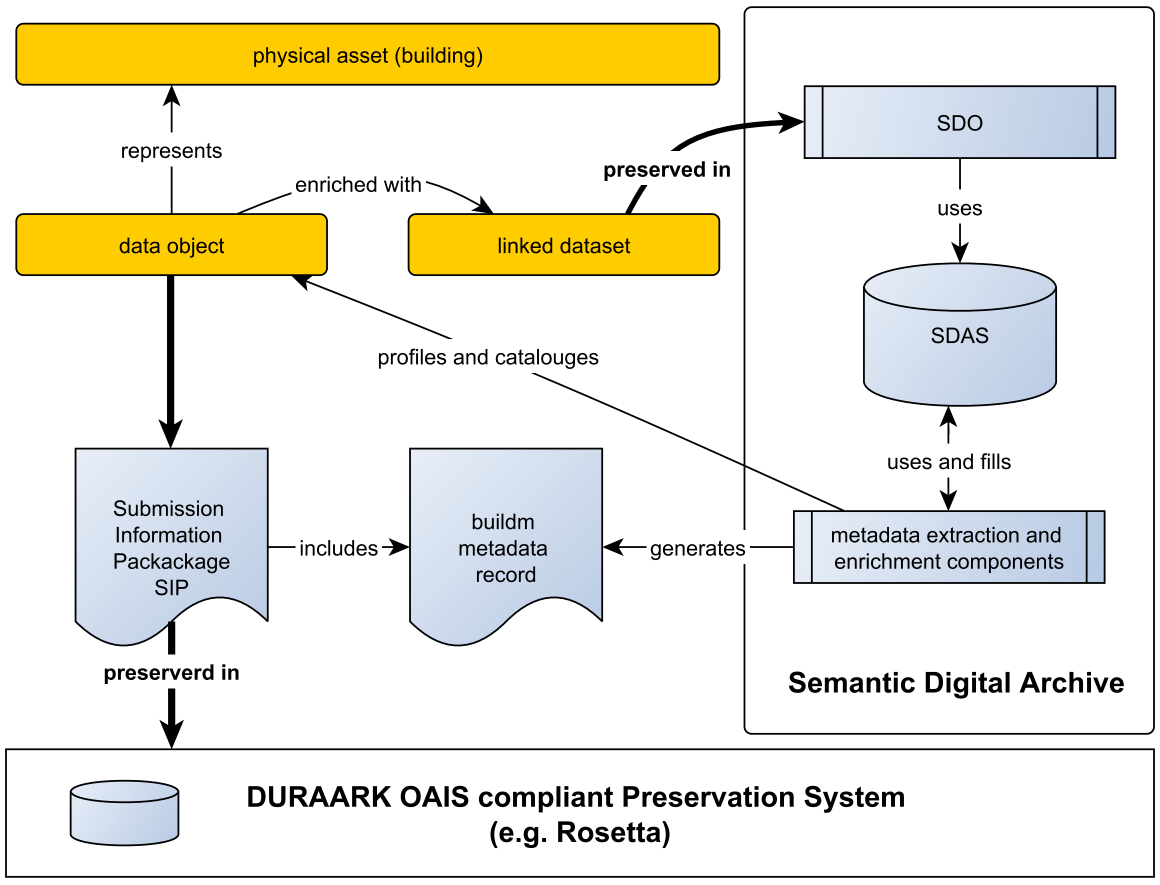


Figure 4: **Conceptual overview of the preservation and harvesting of linked data stemming from digital objects representing a physical asset as well as from extracted, generated and curated metadata record contained in the overall buildm data structure.**

The metadata that is gathered in the `buildm` umbrella structure, can be described on a more granular level. In Figure 5 an overview of the different kinds of metadata is provided that are handled in the context of the DURAARK project.

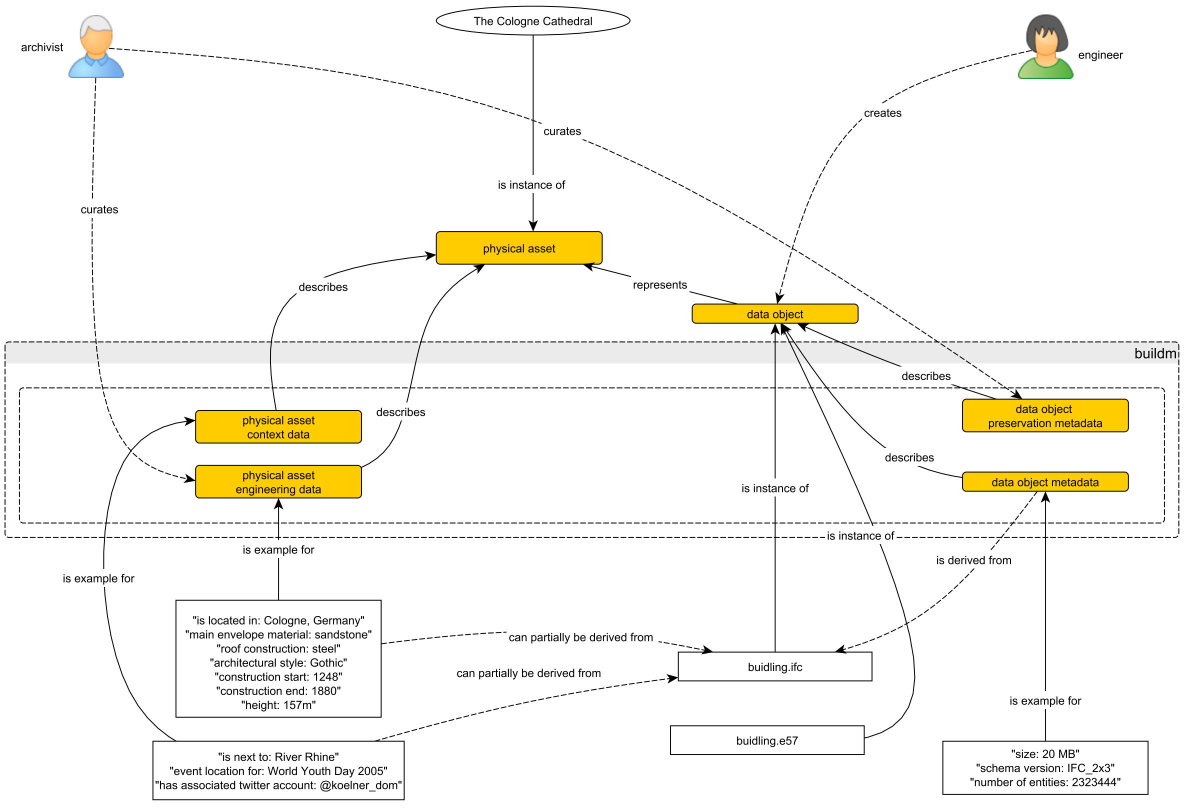

archivist
The Cologne Cathedral
engineer
curates
creates
curates
is instance of
physical asset
describes
represents
data object
buildm
describes
data object preservation metadata
physical asset context data
describes
physical asset engineering data
describes
data object metadata
is instance of
is instance of
is example for
is example for
is derived from
"is located in: Cologne, Germany"
"main envelope material: sandstone"
"roof construction: steel"
"architectural style: Gothic"
"construction start: 1248"
"construction end: 1880"
"height: 157m"
can partially be derived from
can partially be derived from
buidling.ifc
buidling.e57
is example for
"is next to: River Rhine"
"event location for: World Youth Day 2005"
"has associated twitter account: @koelner_dom"
"size: 20 MB"
"schema version: IFC_2x3"
"number of entities: 2323444"

Figure 5: **Overview diagram presenting the different kinds of metadata that are extracted, generated and stored alongside the digital objects representing physical buildings.**

2 SDAS - Semantic Digital Archive Storage

In this section, the main functionalities of the Semantic Digital Archive Storage (SDAS) prototype within the DURAARK system are described. More details with respect to these functionalities within the DURAARK workbench (accessible at `http://workbench.duraark.eu`) can be found in the *Software prototype v1* Deliverable D2.4.
The SDAS serves two main purposes:

1. It is the component where the current and past versions of datasets used for the semantic enrichment of Building Information Models are physically stored for the Long Term Preservation purposes of such semantically rich models.

2. It also serves as a repository of metadata describing the physical assets (buildings) and their context themselves as well as the data object representing them. It is thus also an index and catalogue providing information about the buildings preserved in the DURAARK system as a Linked Data set.

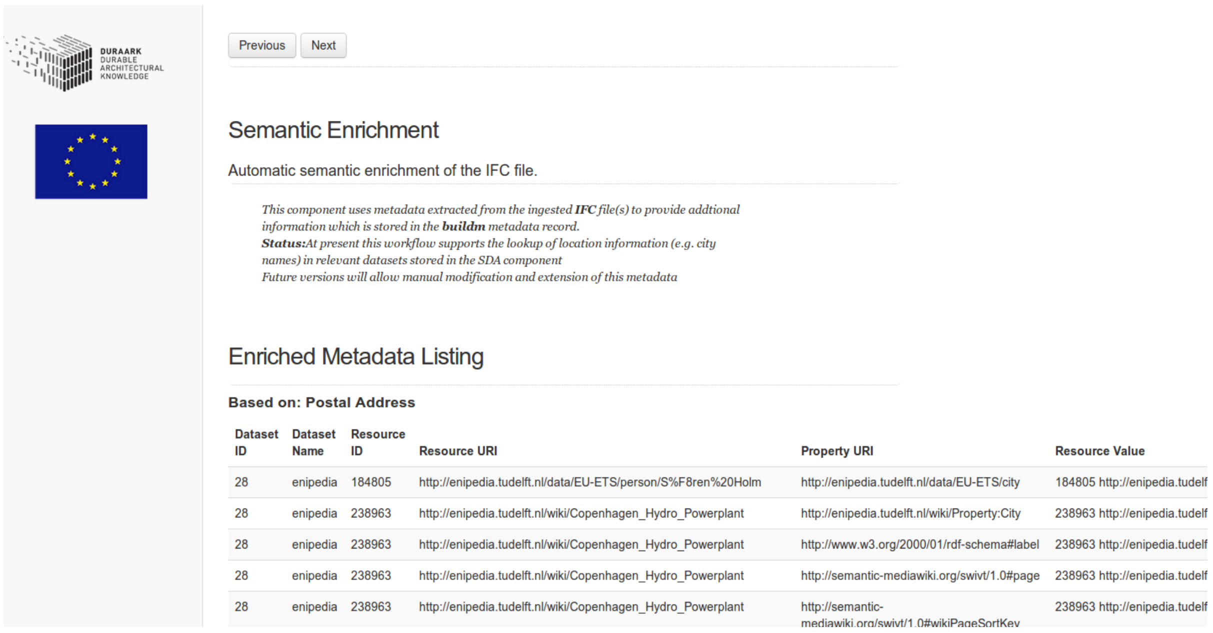
The SDAS is designed as a two-tier architecture:

1. Enrichments of building data through external contextual knowledge, exposed as a ‘Short term‘ access storage of frequently used data in a triple store for responsive access following Linked Data principles.

2. A ‘Long term‘ repository of earlier versions of datasets residing in the SDA in the form of deltas, stored in conventional database as binary blobs. These can be used to revert to earlier states of the evolving datasets at the time of the retrieval. Depending on the size of the original data sets such operations can take a long time.

2.1 Using the Component

At present, the storage component of the SDAS is automatically triggered and managed by the SDO sub-component described in section 3. New delta records of full dumps into the long term storage are triggered by measuring newly profiled datasets stemming from enrichment processes or the scheduled detection of gradual evolutions of known dataset new submissions without user interaction. The ingestion and scheduled maintenance of datasets is invoked manually by the user through profiling an crawling actions further described in Section 3. Figure 6 depicts the workbench UI that can be used in order to trigger and explore the contextual enrichment process.



Semantic Enrichment

Automatic semantic enrichment of the IFC file.

*This component uses metadata extracted from the ingested **IFC** file(s) to provide addtional information which is stored in the **buildm** metadata record.*
***Status:**At present this workflow supports the lookup of location information (e.g. city names) in relevant datasets stored in the SDA component*
Future versions will allow manual modification and extension of this metadata

Enriched Metadata Listing

Based on: Postal Address

Dataset ID	Dataset Name	Resource ID	Resource URI	Property URI	Resource Value
28	enipedia	184805	http://enipedia.tudelft.nl/data/EU-ETS/person/S%F8ren%20Holm	http://enipedia.tudelft.nl/data/EU-ETS/city	184805 http://enipedia.tudelf
28	enipedia	238963	http://enipedia.tudelft.nl/wiki/Copenhagen_Hydro_Powerplant	http://enipedia.tudelft.nl/wiki/Property:City	238963 http://enipedia.tudelf
28	enipedia	238963	http://enipedia.tudelft.nl/wiki/Copenhagen_Hydro_Powerplant	http://www.w3.org/2000/01/rdf-schema#label	238963 http://enipedia.tudelf
28	enipedia	238963	http://enipedia.tudelft.nl/wiki/Copenhagen_Hydro_Powerplant	http://semantic-mediawiki.org/swivt/1.0#page	238963 http://enipedia.tudelf
28	enipedia	238963	http://enipedia.tudelft.nl/wiki/Copenhagen_Hydro_Powerplant	http://semantic-mediawiki.org/swivt/1.0#wikiPageSortKey	238963 http://enipedia.tudelf

Figure 6: **Triples contributing to the contextual enrichment displayed to the user via the DURAARK workbench GUI.**

Additional functionality is implemented at later stage of the project and will be adapted according to the user requirements and measurements from the evaluations carried out in the context of WP7.

2.2 Implementation

The SDAS prototype implementation is primarily based on a *triple store*. A triple store is the general term used to describe a database management system for RDF data[1]. For DURAARK M18 prototype the OpenLink Virtuoso [2] triple store has been chosen as the database backbone for the SDAS. It provides a mechanism for persistent storage and access of RDF graphs. The triple store supports RDF data serializations such as RDF/XML as well as N3/N-triples. Similar well-established and mature triple store implementations

[1]the term 'quad store' on the other hand includes an additional context column per RDF triple that allows the fine grained capturing of context, provenance and named graphs. These however can also be employed using the more common place triple stores

[2]http://virtuoso.openlinksw.com/

such as OpenRDF Sesame, Fuseki, Marmotta etc. provide similar features, performance and tested reliability and provide viable alternatives to the implementation chosen here.

2.3 Data in the SDAS

The triple store is used to store a number of different kinds of data including (i) relevant snapshots and *deltas* (periodic differences in the states of datasets due to temporal evolution) of external linked datasets, (ii) dataset profiles of such datasets (stemming from the SDO component described in section 3), as well as (iii) triples that correspond to the enrichment of individual IFC files. In the following paragraphs, we describe how RDF data enters the triple store based on these triggers.

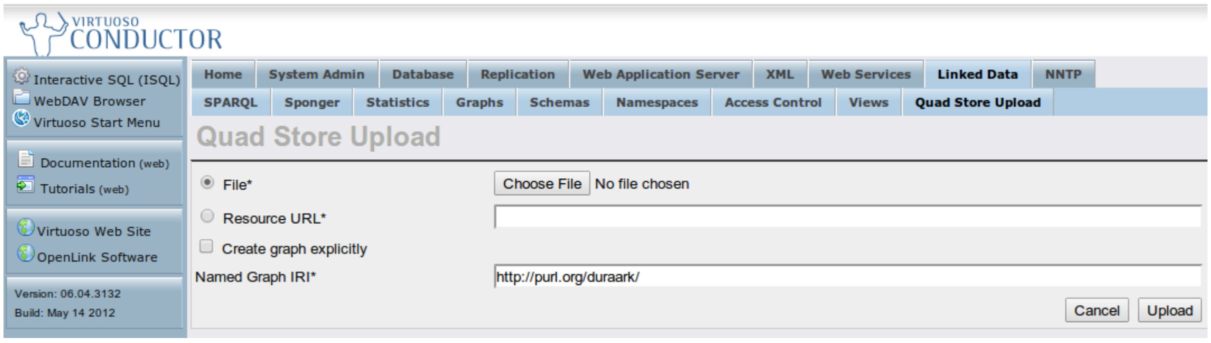
- **Deltas :** The Semantic Digital Observatory (SDO) is responsible for crawling the DURAARK-relevant datasets in the Linked Open Data cloud periodically. At each *crawl-point*[3], the differences in the state of a dataset called *deltas*, are determined on the fly and stored in a corresponding database repository[4]. Further details about *deltas* can be found in Section 3. On the occurrence of possible changes, captured as the difference in the state of a dataset, the triples corresponding to the changes are pushed into the SDAS. For the M18 prototype, this MySQL database repository is located at an external server `db.l3s.uni-hannover.de:3307` which later could be spread out across multiple machines for scalability purposes. The database corresponding to storing the crawled data, `ld_dataset_crawler`, can be accessed using the following credentials.
  Host: `db.l3s.uni-hannover.de:3307`
  User: `duraark_select`
  Password: `RHMTGRxwSz6wYnY5`

- **Dataset Profiles :** The profiles of datasets relevant to DURAARK are captured using methodologies introduced by Fetahu et al [4], and are also stored in the SDAS, where they can be queried. A profile consists of structured dataset metadata describing topics and their relevance. In case new datasets that are relevant to DURAARK are obtained, they are consequently profiled and the profiles are pushed into the SDAS.

[3]The time at which a dataset or a group of datasets are crawled.
[4]using MySQL as the backend implementation in this first prototype

- **IFC Contextual Enrichment :** IFC files can be enriched using data from external datasets and vocabularies as described in detail in section 4. The resulting triples corresponding to the enrichment of an IFC are pushed into the SDAS, where they are stored and can be queried by an end-user.

Currently new data can be added to the SDAS by using the built-in interface to the triple store implementation used in the prototype [5]. Later prototypes of the system will implement user interfaces tailored to the end-users', e.g. archivists' requirements and (limited) technical know-how.



Quad Store Upload

● File*	No file chosen
○ Resource URL*	
☐ Create graph explicitly	
Named Graph IRI*	http://purl.org/duraark/

Figure 7: **The Virtuoso Conductor UI for inserting data into the SDAS.**

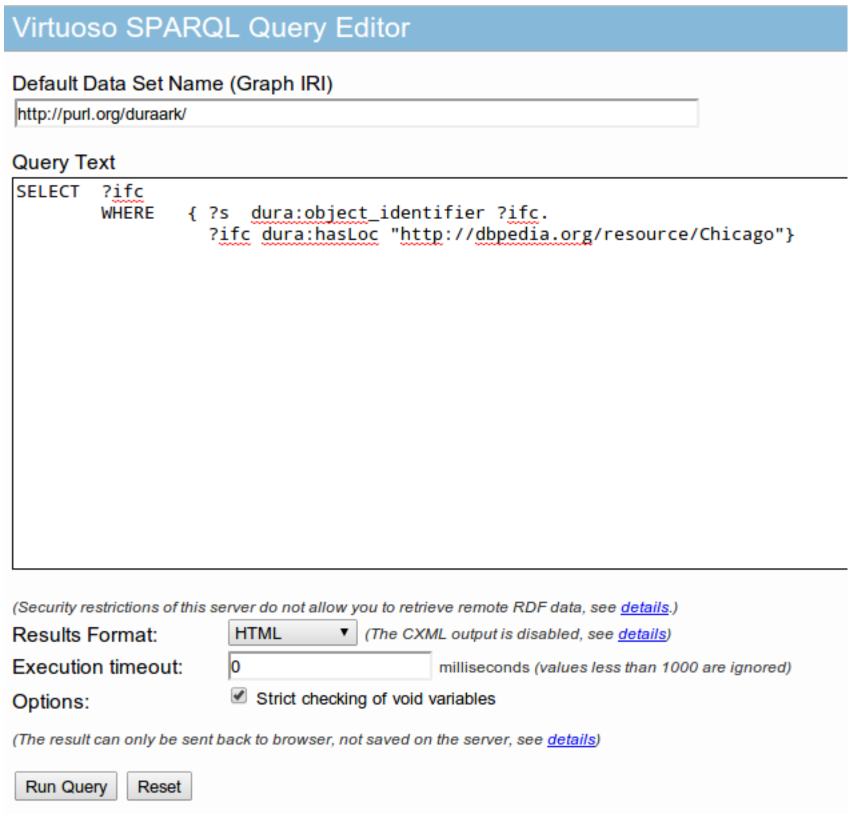
At present, the following steps need to be followed in order to insert triples into the store.

- Log into the Virtuoso UI[6] as an administrator.
- Set destination of the RDF store to insert the triples.
- Set the RDF IRI (Internationalized Resource Identifier)[7].

Querying the Triplestore in the SDAS

The data stored in the SDAS can be queried using the following SPARQL endpoint; `http://duraark.l3s.uni-hannover.de/sparql`. Figure 8 presents the corresponding SPARQL query editor GUI, where the graph IRI can be specified alongside the query itself.

[5] Virtuoso Conductor
[6] `http://meco.l3s.uni-hannover.de:8829/conductor/`
[7] `http://www.w3.org/TR/rdf11-concepts/#section-IRIs`



Default Data Set Name (Graph IRI)

http://purl.org/duraark/

Query Text

```
SELECT  ?ifc
        WHERE   { ?s  dura:object_identifier ?ifc.
                  ?ifc dura:hasLoc "http://dbpedia.org/resource/Chicago"}
```

(Security restrictions of this server do not allow you to retrieve remote RDF data, see details.)

Results Format: HTML (The CXML output is disabled, see details)

Execution timeout: 0 milliseconds (values less than 1000 are ignored)

Options: ✓ Strict checking of void variables

(The result can only be sent back to browser, not saved on the server, see details)

Figure 8: **The Virtuoso SPARQL query editor.**

For example, if one is interested to retrieve the IFC files that describe buildings in *Chicago*, the SDAS can be queried to satisfy such an information need, as depicted in Listing 1.

```
SELECT  ?ifc
        WHERE   { ?s  dura:object_identifier ?ifc.
                  ?ifc dura:hasLoc "http://dbpedia.org/resource/Chicago"}
```

Listing 1: Example query to retrieve IFC file UUIDs that describe buildings in 'Chicago'.

More example queries showcasing the potential of the SDAS prototype, can be found at `data-observatory.org/sdas/`.

2.4 Semantic enrichment of building information with contextual data

The enrichment of IFC files follows a two-fold approach. In the first step, technical descriptive metadata from the IFC file is extracted using the *IFC Metadata Extraction* component. This metadata is captured and passed to the contextual enrichment component. The *Contextual Enrichment* component in turn uses the corresponding terms as a pivot in order to find contextually relevant RDF data from the SDO.

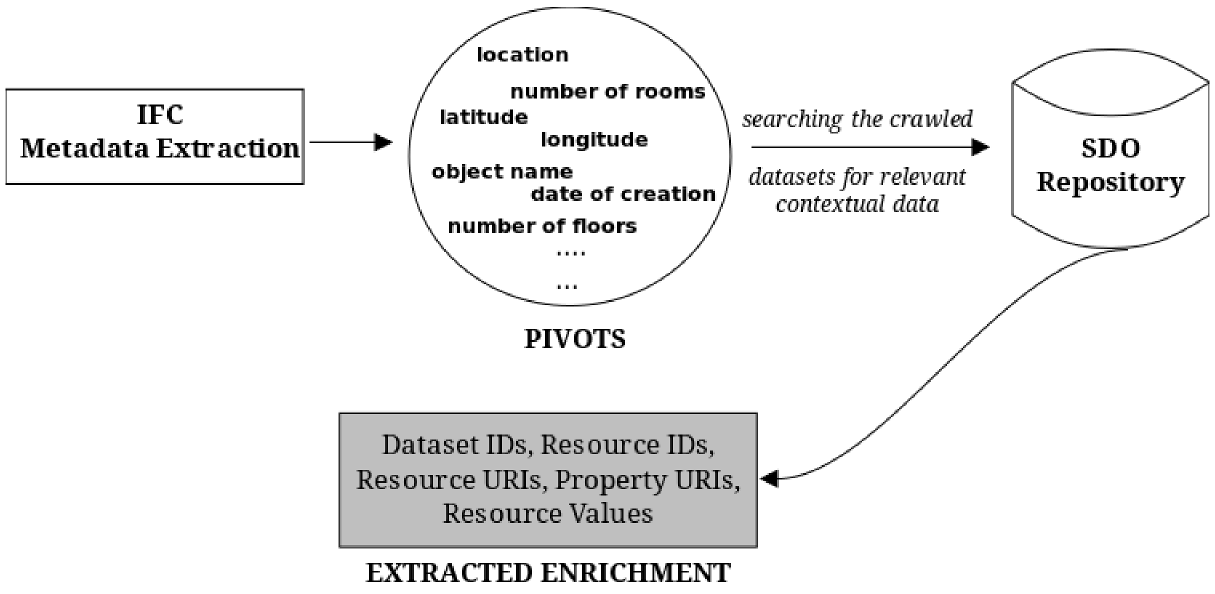


Figure 9: **IFC Contextual Enrichment Process.**

For example, consider the following snippet to depict the metadata that is extracted from an IFC file.

```
@prefix dbpedia-owl: <http://dbpedia.org/ontology/> .
@prefix dbp-prop: <http://dbpedia.org/property/> .
@prefix dct: <http://purl.org/dc/terms/> .
@prefix duraark: <http://duraark.eu/voabularies/buildm#> .
@prefix foaf: <http://xmlns.com/foaf/0.1/> .
@prefix geo-pos: <http://www.w3.org/2003/01/geo/wgs84_pos#> .
@prefix xsd: <http://www.w3.org/2001/XMLSchema#> .
@prefix dc: <http://purl.org/dc/elements/1.1/> .
@prefix qudt: <http://www.qudt.org/qudt/owl/1.0.0/unit/> .

<project_a8346b197cc64f00a35f228897de4f71> duraark:object_identifier
    "2eD6iPVCPF0ADV8eYNtazn"^^xsd:string .
```

```
<project_a8346b197cc64f00a35f228897de4f71> foaf:name "DTU 127"^^xsd:string .
<project_a8346b197cc64f00a35f228897de4f71> dbp-prop:startDate "1970-01-01
    01:00:00"^^xsd:date .
<project_a8346b197cc64f00a35f228897de4f71> dbpedia-owl:buildingStartYear
    "1970-01-01 01:00:00"^^xsd:date .
<project_a8346b197cc64f00a35f228897de4f71> duraark:length_unit
    "MILLIMETRE"^^xsd:string .
<project_a8346b197cc64f00a35f228897de4f71> duraark:authoring_tool "Autodesk Revit
    2013 Autodesk Revit 2013 2013"^^xsd:string .
<project_a8346b197cc64f00a35f228897de4f71> duraark:authoring_tool "Eindhoven
    University of Technology ifcspfrdfcat 0.01a"^^xsd:string .
<project_a8346b197cc64f00a35f228897de4f71> foaf:based_near [ geo-pos:lat
    "55.68300000" ; geo-pos:lon "12.55000000" ] .
<project_a8346b197cc64f00a35f228897de4f71> duraark:floor_count "8"^^xsd:integer .
<project_a8346b197cc64f00a35f228897de4f71> duraark:room_count "55"^^xsd:integer .
<project_a8346b197cc64f00a35f228897de4f71> dbpedia-owl:address
    "Lyngby"^^xsd:string .
<project_a8346b197cc64f00a35f228897de4f71> dc:creator "Morten Jensen"^^xsd:string .
<project_a8346b197cc64f00a35f228897de4f71> duraark:enrichment_vocabulary
    "http://dbpedia.org/property"^^xsd:string .
<project_a8346b197cc64f00a35f228897de4f71> duraark:enrichment_vocabulary
    "http://sws.geonames.org"^^xsd:string .
<project_a8346b197cc64f00a35f228897de4f71> duraark:enrichment_vocabulary
    "http://vocab.getty.edu/aat"^^xsd:string .
<project_a8346b197cc64f00a35f228897de4f71> duraark:enrichment_vocabulary
    "http://vocab.getty.edu/ontology"^^xsd:string .
```

Listing 2: Metadata extracted from an example IFC file.

Here, properties such as `dbpedia-owl:floorCount` for instance, can be enriched by using contextual triples from the Semantic Digital Observatory (SDO) repository, which stores all the data from crawled datasets and vocabularies. In addition, existing vocabularies can be found through the enrichment process to represent some relevant properties (for example, latitude and longitude using the *WGS84 Geo Positioning* vocabulary[8]). Linked Data, offering structured data about, for instance, energy-efficiency policies, geodata or traffic and environmental information can thus be leveraged in order to enrich building information and add further context to the archival data [3].
Such location tagging using common vocabularies and schemas allows the inference of many additional facts, e.g. census and cadastre data etc. The exposure of such data to the archive search front-end (Probado) allows advanced search and retrieval based e.g. on population specifics, neighbourhood properties etc. However most of these Linked Data sets are still under heavy constructions. For the particular case of the DURAARK example data, only limited data is available at present. Most of the combined data objects of a

[8] http://www.w3.org/2003/01/geo/

physical assets that contain *both* an IFC SPF as well as a point cloud e75 are stemming from the consortium partners in Denmark and Norway. Unfortunately, Open Data similar to the UK initiative is still under development and cannot be used for demonstration purposes at present[9].

Existing vocabularies can be found through the enrichment process to represent some relevant properties (for example, latitude and longitude using the *WGS84 Geo Positioning* vocabulary[10]). Such location tagging using common vocabularies and schemas allows the inference of many additional facts, e.g. census data etc. The exposure of such data to the archive search front-end (Probado) allows advanced search and retrieval based e.g. on population specifics, neighbourhood properties etc. However most of these Linked Datasets are still under heavy construction. For the particular case of the DURAARK example data, only limited data is available at present. Most of the combined data objects of a physical assets that contain *both* an IFC SPF as well as a pointcloud e75 are stemming from the consortium partners in Denmark and Norway. Unfortunately, Open Data similar to the UK initiative is still under development and cannot be used for demonstration purposes at present[11].

Within this prototype, enrichment of the IFC files is achieved by using the datasets and external vocabularies which are stored in the SDO. For example, the IFC file that contains a building that is located in Chicago, is enriched with the following data. The snippet provided in the Listing 3, shows the triples contributing to the enrichment, through relevant DBpeida[12] properties describing the country, population, area, and so forth.

```
dbpedia:Chicago dbpprop:hasPhotoCollection  ns174:Chicago ;
    dbpedia-owl:type    dbpedia:City ;
    dbpedia-owl:elevation   181.966 ;
    dbpedia-owl:country dbpedia:United_States ;
    ns173:areaTotal "606.057217818624"^^ns170:squareKilometre ,
        "606.1"^^ns170:squareKilometre ;
    dbpedia-owl:areaTotal   6.06057e+08 ,
        6.061e+08 ;
    dbpedia-owl:populationUrban 8711000 ;
    dbpprop:populationDensitySqMi   11864.4 ;
```

[9] http://uk.fm.dk/publications/2012/good-basic-data-for-everyone/
[10] http://www.w3.org/2003/01/geo/
[11] http://uk.fm.dk/publications/2012/good-basic-data-for-everyone/
[12] http://wiki.dbpedia.org/

```
.
```

Listing 3: Example snippet providing contextual enrichment of the IFC file.

Consider the following example that depicts the pipeline of the enrichment process. Again, let us assume an IFC file describing a structure situated in 'Chicago'.

- In the first stage of the enrichment process, *Chicago* is extracted as a location pivot.
- Using the location pivot, relevant information is retrieved from the datasets crawled and stored in the MySQL repository of the Semantic Digital Observatory(SDO). All resource instances that contain information related to the pivot, in this case *Chicago*, are retrieved. Therefore, depending on the nature of datasets crawled and stored in the SDO, corresponding aspects are enriched.
- This contextually relevant information is then converted into the form of triples fit for consumption by the Semantic Digital Archive. A snippet of such an instance is presented in the Listing 4.

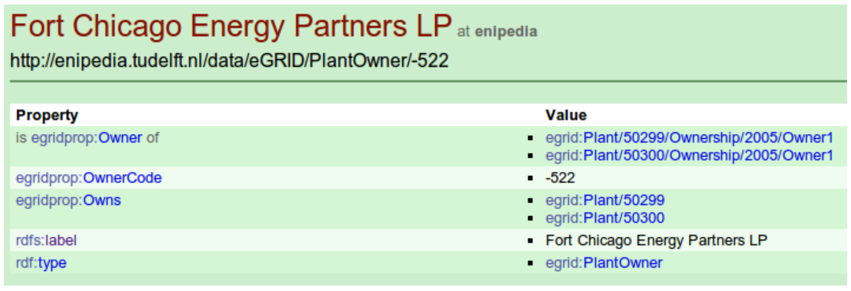
```
:enipedia a void:Dataset;
    rdfs:label "dataset_id 28,resource_id 211751"^^xsd:string;
    <http://enipedia.tudelft.nl/data/eGRID/PlantOwner/-522>
        <http://www.w3.org/2000/01/rdf-schema#label> "Fort Chicago Energy Partners
        LP"^^xsd:string;
.
```

Listing 4: Example triple providing contextual enrichment of the IFC file.

In this example, we see that the resource instance providing the contextual enrichment comes from the `enipedia` dataset[13], crawled and stored in the Semantic Digital Observatory (SDO). Enipedia contains information about energy and industry related topics, ranging from natural gas infrastructure in various parts of the world to electricity production, that is fit for the consumption by the semantic web. The resource instance presented as a source of enrichment describes the owner of an electricity plan in Chicago. eGRID[14] contains data relevant to the environmental characteristics of almost all electric power generated in the United States. Although, one may argue that a power plant that is located in the same city, may not directly stimulate a sense of semantic enrichment, this

[13] http://enipedia.tudelft.nl/wiki/Main_Page
[14] http://www.epa.gov/cleanenergy/energy-resources/egrid/

is a first step in the direction of attaining vicinity-based enrichment. The product of the dereference-able URI `http://enipedia.tudelft.nl/data/eGRID/PlantOwner/-522` is presented in the Figure



Fort Chicago Energy Partners LP at enipedia
http://enipedia.tudelft.nl/data/eGRID/PlantOwner/-522

Property	Value
is egridprop:Owner of	▪ egrid:Plant/50299/Ownership/2005/Owner1 ▪ egrid:Plant/50300/Ownership/2005/Owner1
egridprop:OwnerCode	▪ -522
egridprop:Owns	▪ egrid:Plant/50299 ▪ egrid:Plant/50300
rdfs:label	▪ Fort Chicago Energy Partners LP
rdf:type	▪ egrid:PlantOwner

Figure 10: **Example of a dereference-able enrichment URI.**

Note that the current prototype includes possible enrichments from the datasets present in the SDO[15]. By crawling further datasets in the Linked Open Data cloud (currently in progress), we can achieve a wider range of contextual enrichment.

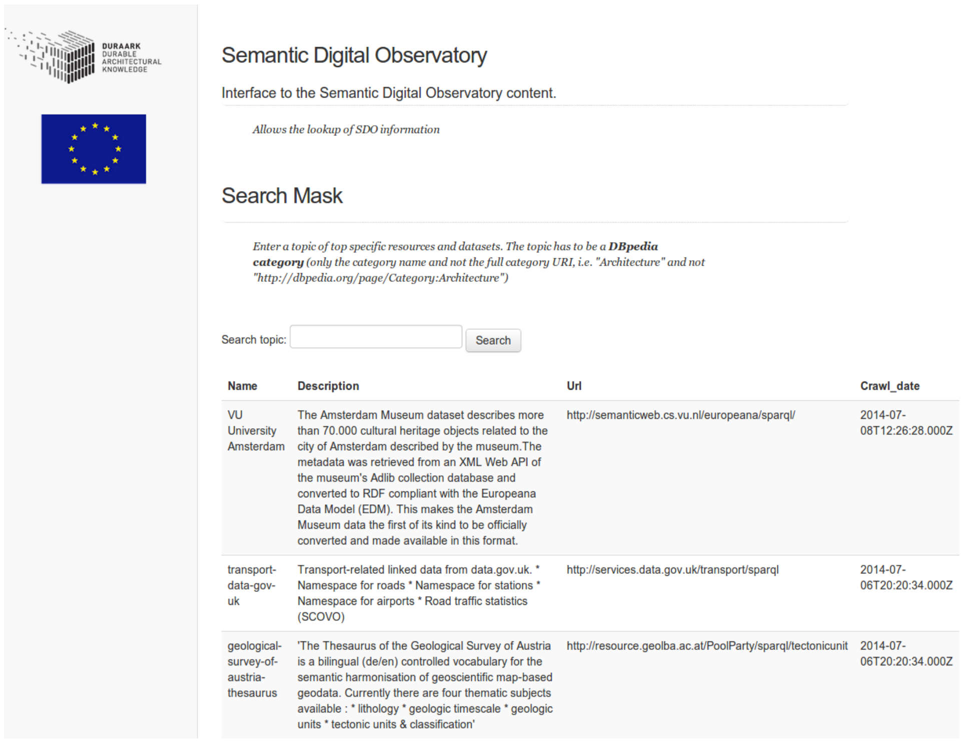
Adding the Enrichment Triples to SDAS

On the completion of the enrichment process as described above, the triples contributing to the enrichment are presented to the user through the DURAARK workbench GUI as shown in Figure 6.
Apart from presenting such enrichment-triples to the user, they are also written to an the buildm metadata record (an example is shown in the snippet, 4 , and consequently inserted into the SDAS. The triples associated with the enrichment can be added to the triplestore by using either the Virtuoso UI or by using HTTP/WebDAV GET and PUT[16].

[15] `http://data-observatory.org/dataset_crawler/index.php`
[16] `http://www.w3.org/Protocols/rfc2616/rfc2616-sec9.html#sec9.6`

3 SDO - Semantic Digital Observatory

The Semantic Digital Observatory(SDO) component of the DURAARK system aims to facilitate the discovery and retrieval of suitable architecture-relevant datasets and to provide structured metadata about available datasets. The two main modules that support the SDO in achieving these objectives are (i) *Dataset Crawler Module*, and (ii) *Dataset Profiling Module*. The data stored within the SDO itself can be explored within the DURAARK WorkbenchUI, as depicted in Figure 11.



Semantic Digital Observatory

Interface to the Semantic Digital Observatory content.

Allows the lookup of SDO information

Search Mask

Enter a topic of top specific resources and datasets. The topic has to be a ***DBpedia category*** *(only the category name and not the full category URI, i.e. "Architecture" and not "http://dbpedia.org/page/Category:Architecture")*

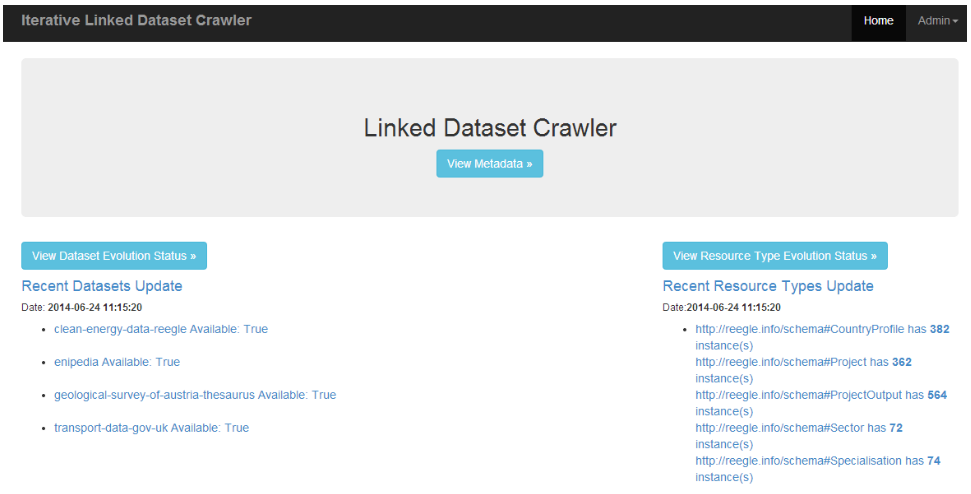
Search topic: Search

Name	Description	Url	Crawl_date
VU University Amsterdam	The Amsterdam Museum dataset describes more than 70.000 cultural heritage objects related to the city of Amsterdam described by the museum.The metadata was retrieved from an XML Web API of the museum's Adlib collection database and converted to RDF compliant with the Europeana Data Model (EDM). This makes the Amsterdam Museum data the first of its kind to be officially converted and made available in this format.	http://semanticweb.cs.vu.nl/europeana/sparql/	2014-07-08T12:26:28.000Z
transport-data-gov-uk	Transport-related linked data from data.gov.uk. * Namespace for roads * Namespace for stations * Namespace for airports * Road traffic statistics (SCOVO)	http://services.data.gov.uk/transport/sparql	2014-07-06T20:20:34.000Z
geological-survey-of-austria-thesaurus	'The Thesaurus of the Geological Survey of Austria is a bilingual (de/en) controlled vocabulary for the semantic harmonisation of geoscientific map-based geodata. Currently there are four thematic subjects available : * lithology * geologic timescale * geologic units * tectonic units & classification'	http://resource.geolba.ac.at/PoolParty/sparql/tectonicunit	2014-07-06T20:20:34.000Z

Figure 11: **The WorkbenchUI to explore the data stored within the SDO.**

3.1 Using the component UIs

The SDO prototype includes an interface prototype of the dataset crawler. This interface provides means to access the crawled resources, given specific crawl-points of interest from the periodical crawls. Furthermore, the interface provides a means to filter for specific datasets and resource types. This can be accessed through the Maintenance component of the DURAARK Workbench at `http://bw-dssv18.bwk.tue.nl:9000/modules/contrib/sdo/`.


Iterative Linked Dataset Crawler
Home
Admin
Linked Dataset Crawler
View Metadata »
View Dataset Evolution Status »
View Resource Type Evolution Status »

Recent Datasets Update

Date: **2014-06-24 11:15:20**

- clean-energy-data-reegle Available: True
- enipedia Available: True
- geological-survey-of-austria-thesaurus Available: True
- transport-data-gov-uk Available: True

Recent Resource Types Update

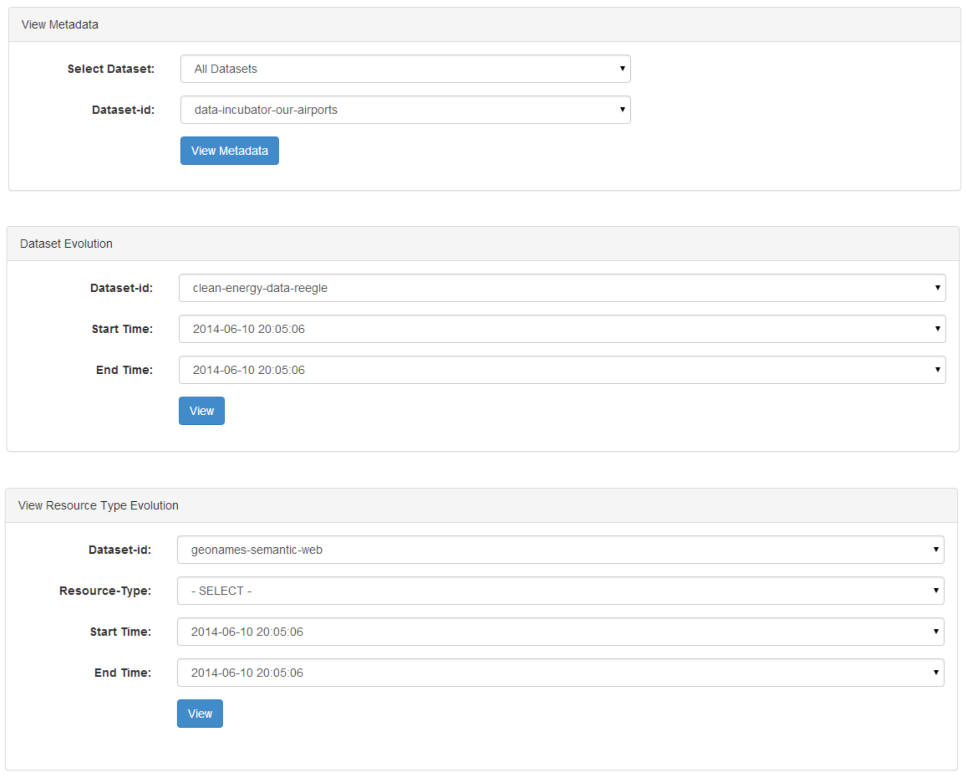
Date:**2014-06-24 11:15:20**

- http://reegle.info/schema#CountryProfile has **382** instance(s)
  http://reegle.info/schema#Project has **362** instance(s)
  http://reegle.info/schema#ProjectOutput has **564** instance(s)
  http://reegle.info/schema#Sector has **72** instance(s)
  http://reegle.info/schema#Specialisation has **74** instance(s)

Figure 12: **The entry page for the dataset's crawler interface.**

Furthermore, we provide a web interface, which offers basic query functionalities. Figure 12 shows the entry page of the web interface, where information regarding the latest crawls, updated datasets and their availability is shown. The web application has three main components (see Figure 13):

1. displaying of the dataset's metadata,
2. dataset evolution, showing summaries of added/updated/deleted resources for the different types, and
3. dataset type specific evolution, showing a summary for a specific resource type the number of added / updated/deleted resource instances for specific crawl time-points.


View Metadata

Select Dataset: All Datasets

Dataset-id: data-incubator-our-airports

Dataset Evolution

Dataset-id: clean-energy-data-reegle

Start Time: 2014-06-10 20:05:06

End Time: 2014-06-10 20:05:06

View Resource Type Evolution

Dataset-id: geonames-semantic-web

Resource-Type: - SELECT -

Start Time: 2014-06-10 20:05:06

End Time: 2014-06-10 20:05:06

Figure 13: **The different functionalities of the dataset's crawler interface.**

Beyond the currently implemented graphical user interfaces, the SDO can also be interacted with using service endpoints. Examples and a manual can be found in Appendix A.

3.2 Dataset Crawler Module

The dataset crawler module handles the responsibility of extracting resources from linked datasets, relevant for the DURAARK project and thereby used for enrichment of archival data. We store the crawled data in a relational database. The database schema used to store data as well as metadata from the crawls helps us to ensure easy, meaningful, and accurate storage and retrieval of the crawled data.
The crawler is designed with a pre-set intent to accommodate methods for assessing the temporal evolution of linked datasets. A dataset which has not been crawled before will thereby be crawled completely and all corresponding data will be stored in a relational database[17]. This would thereby correspond to a dump of that dataset, stored according to the database schema. However, in case a dataset has already been crawled, the differences

[17]using the MySQL implementation for the current prototype

between the previously crawled state of the dataset and the current state are determined on-the-fly. These differences alone, called *deltas* or *diffs*, are then stored. Therefore, for any dataset that has been crawled multiple times at different crawl-points[18], by storing all data in the first crawl and consequently storing only the *deltas*, we will be able to reconstruct the state of the dataset at any of the given crawl-points.

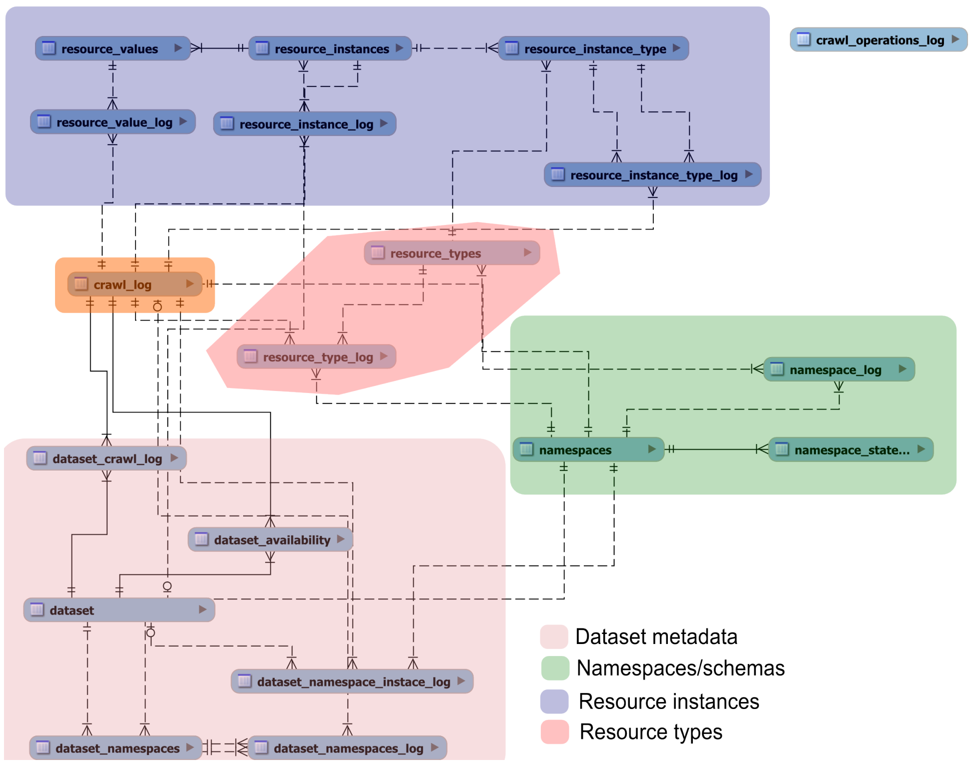


Figure 14: **Schema for storing crawled datasets.**

Figure 14 depicts the extracted data from crawls and their decomposition into coherent sub-parts (e.g. dataset metadata, resources etc.) as per the database schema. The schema helps us to store data corresponding to a crawled dataset, in a coherent manner and provides an elegant way to query the data at various granularities and with varying crawl features.

[18]The time at which a given *crawl* operation is triggered.

3.3 Computation of Diffs

As elucidated in the previous Deliverable D3.3.2[19], when we rely on external datasets for providing a means of enriching IFC files and archival data, we need to ensure persistent correctness.

We ensure that changes in the states of a dataset are captured via periodic crawls of the dataset. Consider the following example; let us assume that a part of the enrichment of a particular IFC file, involves the location entity, 'Bombay'. However, the city of Bombay was formerly called with several other names such as 'Mombayn', 'Bombain', 'Monbaym', to name a few. About two decades ago, the name of the city was officially changed to 'Mumbai'. When such circumstances trigger changes in external datasets, then these changes should be reflected in the data that is being used to enrich the architectural data for archival. So, the IFC file should consequently be enriched with the location entity 'Mumbai'.

The differences between the state of a dataset at different crawl-points can be captured efficiently using the dataset crawler module. Note that it is possible to infer changes at different levels in a dataset, for example, schema-level changes, or resource instance level changes. The following possibilities with respect to changes in a dataset are handled by the dataset crawler.

- **Insertions.** New triples may be added to a dataset. Such additions introduced in the dataset correspond to *insertions*.
- **Deletions.** Over time, triples may be deleted from a dataset due to various reasons ranging from persisting correctness to detection of errors. These correspond to *deletions*.
- **Updates.** The information represented in a dataset may be dictated by external factors, which can result in changes in the literal values or properties of existing triples in the dataset. For example, the population of a city changes constantly and thus has to be updated in order to ensure persistent correctness of the literals. Such changes to existing triples in a dataset, correspond to *updates*.

[19] Ontological Framework for a Semantic Digital Archive

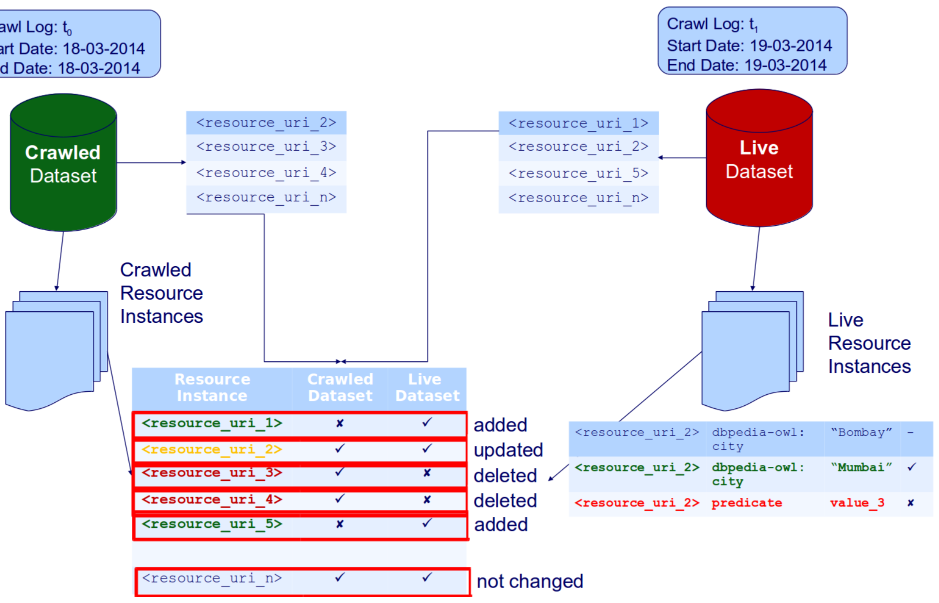


Figure 15: **Consistency in the Computation of Diffs.**

Figure 15 presents the example discussed earlier, depicting the computation of *diffs* between a previously crawled dataset at crawl-point t_0 and a fresh crawl at crawl-point t_1. First, a change in the 'live dataset' in the form of an insertion of the triple corresponding to the URI `<resource_uri_2>` is observed. Thus, the triple describing the city `Bombay` is added. Consequently, if the value of the property `dbpedia-owl:city` is updated, then a consequent crawl would capture this difference in the literal value of the property as an update to `Mumbai`. Similarly, deletions made are also detected during the computation of *diffs*.

The workflow of the dataset crawler is described in detail in the Deliverable D2.2.4[20]. When significant changes emerge within a dataset over a period of time, then the *diffs* are rendered as RDF triples and pushed into the triplestore in the SDA.

At the time of ingestion during UC1, the user will be informed about the uptodateness of

3.4 Dataset Profiling Module

The contents of external linked datasets are not very apparent. In order to maintain a catalog describing the contents of a dataset that is relevant to the DURAARK system, and how the contents change and evolve over time, there is a need for additional techniques.

[20]D2.2.4: Software prototype

The dataset profiling module as part of the Semantic Digital Observatory, plays a vital role in generating structured metadata from crawled datasets (resulting from the crawler module described above). The generated metadata consists of topics (as DBpedia categories). Topics are extracted from the textual literals from the crawled resources, by adopting a named entity disambiguation (NED) process [?]. Furthermore, the topics are ranked based on their relevance given a dataset from which they are extracted. In details, the process of generating the dataset profiles corresponds to the steps in the pipeline depicted in Figure 16.

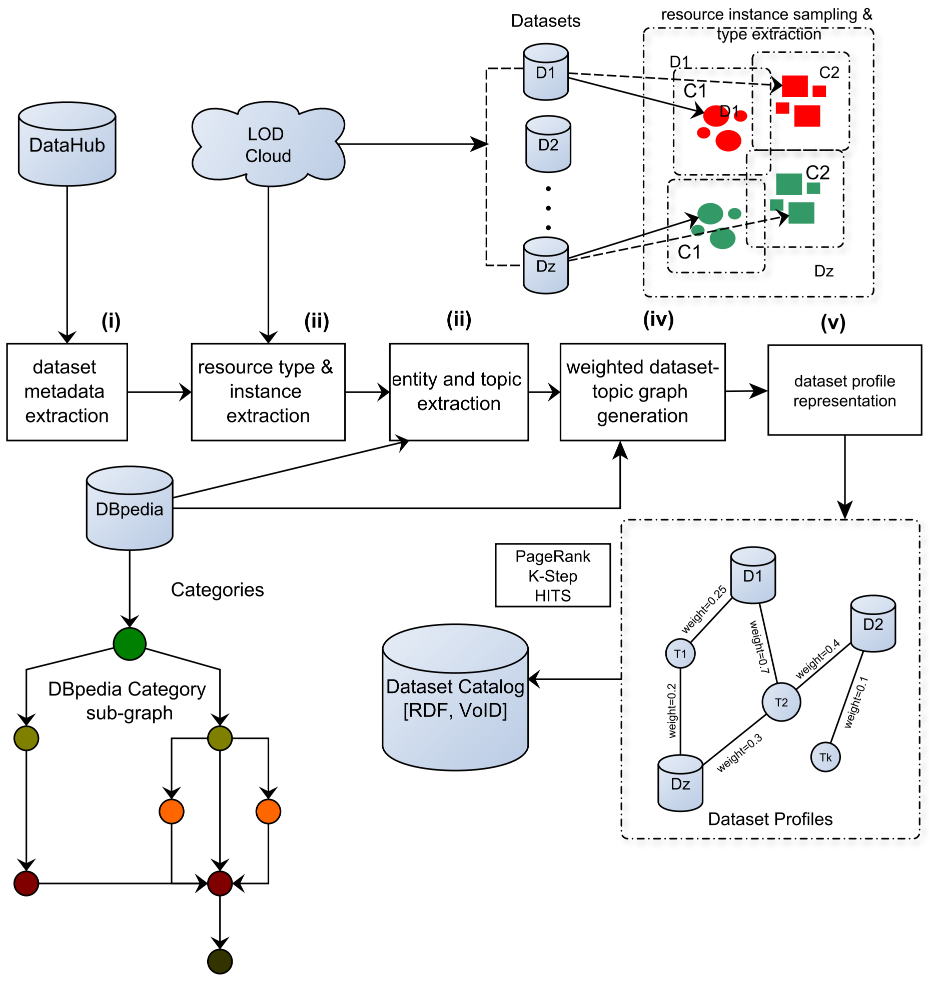

Datasets
resource instance sampling & type extraction
D1
D2
Dz
C1
C2
DataHub
LOD Cloud
(i)
(ii)
(ii)
(iv)
(v)
dataset metadata extraction
resource type & instance extraction
entity and topic extraction
weighted dataset-topic graph generation
dataset profile representation
DBpedia
Categories
DBpedia Category sub-graph
PageRank K-Step HITS
Dataset Catalog [RDF, VoID]
T1
T2
Tk
weight=0.25
weight=0.7
weight=0.4
weight=0.2
weight=0.3
weight=0.1
Dataset Profiles

Figure 16: **Dataset profiling pipeline.**

The profiling of datasets can be directed to specific datasets, and a subset of crawled resources depending on the suitability and availability of the datasets. A detailed description

of the workflow in dataset profiling is presented in the Deliverable D2.4[21].
On generating profiles of relevant datasets, the triples corresponding to the profiles are pushed into the triple store within the SDA. Here, the dataset profiles can be queried using the corresponding SPARQL end-point.
The dataset profiling module, which is constructed based on [?], as part of the DURAARK workbench. It can be freely downloaded[22].

[21]D2.4: Software Prototype
[22]https://github.com/bfetahu/ldp_tool

4 Semantic enrichment

The term 'Semantic enrichment' refers to a number of different additions of information to building information handled at different stages and in different processes of the DURAARK framework and its software prototypes. These stages can be categorized into:

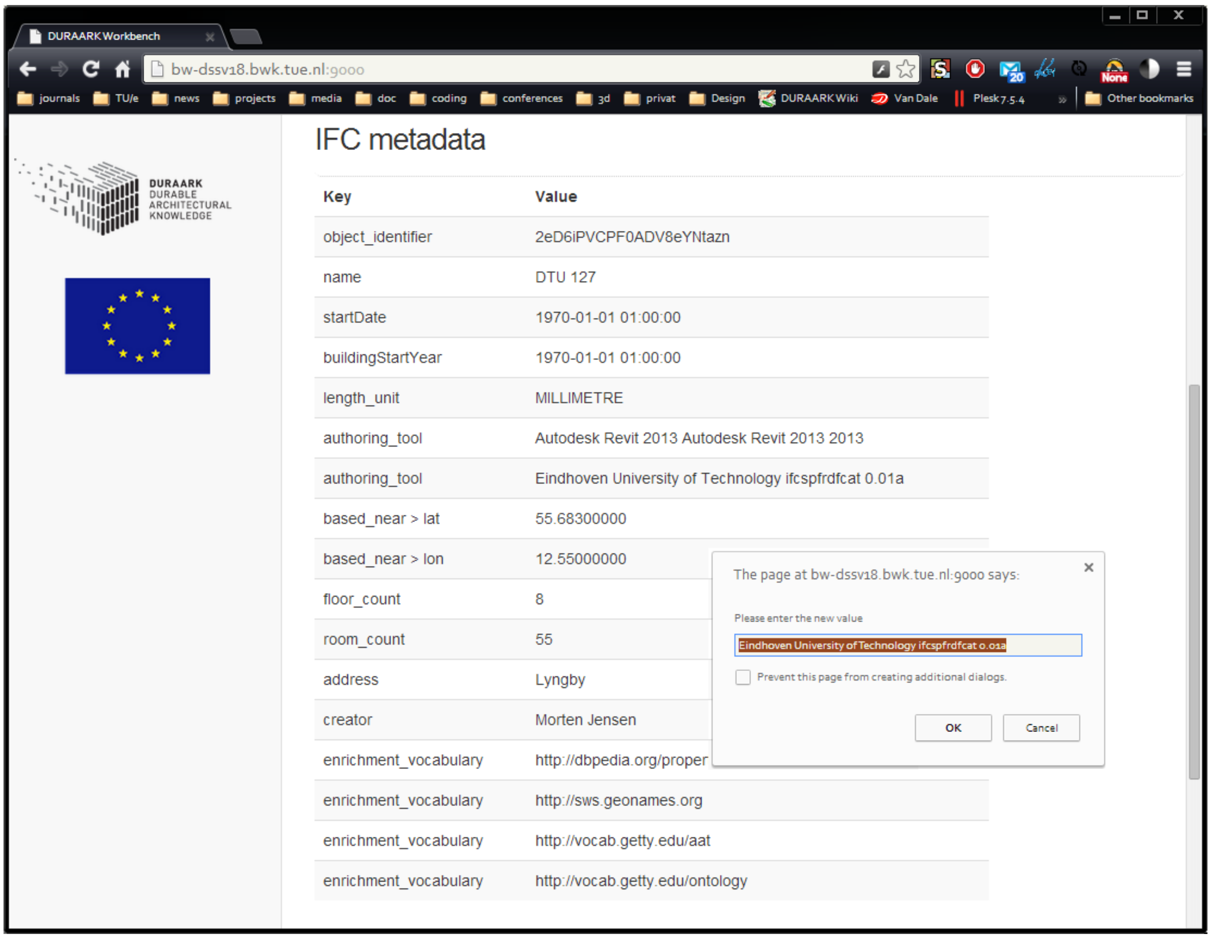
1. **Semantic enrichment of engineering objects**. *Stored in the IFC SPF*. Enrichment takes place **before ingestion**

2. **Semantic enrichment of building information with contextual data**. *Captured in the metadata record stored along-side the data object in the SIP*. Enrichment **during ingestion**

3. Semantic enrichment of archival metadata. *Stored in the archive catalogues and the SDAS*. Enrichment **during** and **after ingestion**.

In the following sections all three of these forms of enrichment and their implementation into the M18 are described. The scope of the M18 prototype however starts with the **metadata extraction** of information already present when the ingested data enters the DURAARK system.

4.1 Using the component UIs



IFC metadata

Key	Value
object_identifier	2eD6iPVCPF0ADV8eYNtazn
name	DTU 127
startDate	1970-01-01 01:00:00
buildingStartYear	1970-01-01 01:00:00
length_unit	MILLIMETRE
authoring_tool	Autodesk Revit 2013 Autodesk Revit 2013 2013
authoring_tool	Eindhoven University of Technology ifcspfrdfcat 0.01a
based_near > lat	55.68300000
based_near > lon	12.55000000
floor_count	8
room_count	55
address	Lyngby
creator	Morten Jensen
enrichment_vocabulary	http://dbpedia.org/proper
enrichment_vocabulary	http://sws.geonames.org
enrichment_vocabulary	http://vocab.getty.edu/aat
enrichment_vocabulary	http://vocab.getty.edu/ontology

The page at bw-dssv18.bwk.tue.nl:9000 says:

Please enter the new value

Eindhoven University of Technology ifcspfrdfcat 0.01a

☐ Prevent this page from creating additional dialogs.

OK Cancel

Figure 17: **Screenshot of the metadata that is semi-automatically extracted from the ingested files and presented to the user. Selecting one of the entries allows the user to modify the data values and store it back into the `buildm` record**

In the M18 prototype, the metadata extraction is automatically invoked during the ingest process in the "SIP generation" component wizard. Two command line tools are invoked on the uploaded IFC and E57 files respectively, which extract relevant metadata according to the revised specifications in earlier DURAARK deliverables (D3.2, D7.1). All extracted metadata is presented in a tabular overview with key-value pairs displaying the respective properties of the **buildm** schema. As illustrated in figure 17, upon clicking a value of the extracted metadata, the user is allowed to modify the respective entry.
More details with respect to these functionalities within the DURAARK workbench (accessible at `http://workbench.duraark.eu`) can be found in the *Software prototype*

v1 Deliverable D2.4. Later prototypes will allow additional manual enrichment of metadata with external vocabularies from configurable datasets residing the SDA (see also sections 3 and 2)

4.2 Semantic enrichment of engineering objects

In this stage, Linked Data is used to enrich objects in IFC files based on the `IfcSinglePropertyValue` mechanism proposed in D3.2. During the use case UC1 ingestion process, this data is extracted and catalogued to be send to the SDAS/SDO components. This data is generated by the engineer **before** the ingestion process and is already present at the time of ingestion. The primary use case is the reference to structured vocabularies like the bSDD (see D3.1) and to information of e.g. individual product manufacturers that have produced e.g. a door which is **attached to individual object within the IFC model**. This type of enrichments is **stored in the IFC SPF**. Even though independent software prototype implementations exist that allow such enrichment of the IFC files these are considered out of scope for the M18 prototype implementation. All enrichments from this stage are extracted during the Metadata extraction stage (see the following Section 4.3). All vocabularies that have been found

4.3 Metadata extraction

In figure 5 an overview of the different kinds of metadata is provided that are handled in the context of the DURAARK project. Metadata is both automatically extracted from the ingested contents as well as generated and edited by the curator during the ingest process. The different sorts of metadata are bundled together into the 'buildm' structure. In `PREMIS` terms, this metadata record is divided into "descriptive metadata" (left hand side of fig 5) and "technical metadata" (right hand side of fig 5). The metadata extracted and derived from all ingested data objects as well as the enrichment that are suggested by the context enrichment component are rendered into the UI of the prototype that is integrated into the workbench. Figure 17 illustrates this. The user is enabled to modify and add this information that is then stored in the buildm metadata record. For M18, only a subset of all envisaged information is captured and only of a subset of this is transferred into the appropriate locations of the METS/PREMIS record that will be stored to describe the SIP in the Rosetta archival system. The details of the SIP generation can be found in the M18 report D2.4.

4.4 IFC Metadata extraction implementation

The IFC metadata extraction utility extracts entity instance attributes out of an IFC-SPF file. The tool is written in Python and relies on a C++ module for parsing IFC-SPF files[23]. By heavily relying on operator overloading in Python the utility mimics a Domain Specific Language for IFC extraction. For end-users, this allows for a concise way to specify the attributes to be extracted out of the IFC file, separating out the technicalities associated with parsing the IFC file.

The definition of the extracted attributes consists of a list of statements, each of which defines an attribute (or combination thereof) to be extracted. Such a statement starts with an IFC keyword datatype to populate a set of interrogated instances with elements of the same (super)type. Data curators can successively descend further into the graph by specifying entity instance attributes or inverse attribute names. The utility emits RDF as its output format.

As an example, one might write the following line to emit a triple that signifies the unique identifier for the project in the IFC file:

```
file.IfcProject.GlobalId >> "duraark:object_identifier"
```

This results in the following triple being emitted, based on the IFC-SPF contents:

```
<project_a8346b197cc64f00a35f228897de4f71> duraark:object_identifier
    "2eD6iPVCPF0ADV8eYNtazn"^^xsd:string .
```

Note that attributes that that refer to another entity instance (or collection of them) can not be directly emitted. In fact, emitted attributes should be of the one the elementary datatypes, such as STRING, INTEGER, ENUMERATION or any of the simple types redeclaring any of these, such as `IfcLabel`, which simply redeclares a STRING datatype. Therefore

```
file.IfcProject.OwnerHistory
```

cannot be directly emitted since it would refer to an entity instance of type `IfcOwnerHistory`, but rather,

```
file.IfcProject.OwnerHistory.CreationDate
```

can be emitted since it is an attribute of type `IfcTimeStamp`, which is in fact of type INTEGER. The metadata extraction utility comes with a set of convenience function that can map input fields to more meaningful or human readable output. For example, in the buildm extraction definition there is a statement

```
file.IfcProject.OwnerHistory.CreationDate >> formatters.time >>
    ("dbp-prop:startDate", "dbpedia-owl:buildingStartYear")
```

[23] https://github.com/aothms/IfcOpenShell

which would convert, for example, the UNIX timestamp 1403774711 into the string "2014-06-26 11:25:11". In addition, once emitted as RDF, the xsd:date type is associated with the string so that the two triples generated on account of this statement are:

```
<project_a8346b197cc64f00a35f228897de4f71> dbp-prop:startDate "2014-06-26
    11:25:11"^^xsd:date .
<project_a8346b197cc64f00a35f228897de4f71> dbpedia-owl:buildingStartYear
    "2014-06-26 11:25:11"^^xsd:date .
```

The architecture of the IFC metadata extraction utility, outlined in the paragraphs above, separates the extraction schema from the IFC parsing implementation and therefore provides a succinct and maintainable way for data curators to provide extraction schema statements. Maintaining the set of attributes extracted by this utility does not need in-depth knowledge of the programming languages used to write the tool.

5 Decisions & Risks

5.1 Technical decisions and impact

Umbrella structure of metadata as RDF

Throughout most of the archival world, XML based schemas are still predominant. This becomes apparent when investigating the currently most widely accepted metadata description format such as `METS` and `PREMIS`. Both are supported by a range of tools and (commercial) DPS solutions. Yet, in the context of the DURAARK project and related software prototypes we have opted to base the core data structures on the Resource Description Format (RDF) instead. The reasoning behind this decision was driven by two main considerations:

Seamless integration into Linked Data environments as well as the immediate reuse of common vocabularies. By using RDF for the `buildm` metadata records and publishing them in accordance with the Linked Data principles, information about buildings that is preserved according to the DURAARK recommendations and using the implemented tools can be found and reused. Connections (links) to other Linked Data sets can be created in much straight format manners and the creation of yet another information silo is avoided.

Trends in the industry to move to RDF. While at the time (in M12 of the project) the initial decision to e.g. use an OWL/RDF form of `PREMIS` was still based on academic experimental ports of the established vocabularies (see e.g. D3.1 and D3.2), a trend of standardization and market dissemination of shared data and vocabularies as RDF can be observed.
By the time of writing, not only the OWL/RDF version of `PREMIS` has become an official standard of the Library of Congress[24]. Also the extensive Getty AAT vocabulary has been published as linked data in the in the last half year since deciding for RDF[25]

[24] http://www.loc.gov/standards/premis/ontology-announcement.html
[25] http://www.getty.edu/research/tools/vocabularies/lod/

Separate Storage for SDAS and SDO

Both, SDAS and SDO target different use cases. The overall vision is to expose Linked Data through the SDA to enable frequent queries through end users and application, while at the same time, enable the archival of all related external datasets. Satisfying both requirements led us to implement a two-fold storage strategy, where the SDO deploys a relational database to capture periodic and exhaustive crawls of external datasets, while the SDAS deploys a native RDF store, to allow queries through its Linked Data interface (dereferencable URIs, SPARQL endpoint). The exact interaction between the two will be fully realised in the upcoming SDA prototypes.

Exposing Linked Data

At present, a number of different approaches have been proposed to enrich Building Information Models with additional information. The main example of such suggestions is the Norwegian Technical Recommendation SN/TS 8328 that uses existing modelling mechanisms of the Industry Foundation Classes model to link concepts from the buildingSMART Data Dictionary to individual objects. Even though the technical infrastructure is fully operational, its uptake in industry can hardly be predicted. The decision taken in the DURAARK implementation to focus on the use of Linked Data standards and best practises instead is based on its much wider uptake and market acceptance: For the bSDD and its dedicated interfaces, only software vendors relevant to the building industry are likely candidates to some day probably implement software support into their products and services. For Linked Data based on the Resource Description Framework however, a much wider tapestry of industry stakeholders is already very actively support these technological approaches. The building industry related software vendors will eventually follow.

Web-based Architecture

The decision to base the user interfaces of the Semantic Digital Archive and other components on a web-based architecture can be justified and explained by a number of points:

- **Scalability** and independence from individual machines: Depending on the extend of the required storage, the persistency of the stored datasets and delta can be distributed over an arbitrary number of physical machines

- **Separation of concerns**. Using a distributed web-based architecture in the DURAARK context enables the distribution of the individual components. Components with special hardware requirements, such as the graphics-intensive point cloud-related components can be run on dedicated hardware, while other less demanding components can be hosted on regular terminals that can be expected in e.g. libraries and archives.
- **Shared archives** Institution and industrial stakeholders may want to share, synchronize or mirror their Semantic Digital Archives. Using web-based architectures makes such future scenarios easier.

5.2 Risk assessment

This section describes technical risks, consequence and contingency actions for potential risks emerging from the current technological design:

Risk ID 0

Risk Description Scalability issues emerging from a growing number of data sets in the SDA long term storage of evolving datasets.

Risk Assessment :

Impact Medium

Probability Medium

Description Linked Data used for the enrichment of newly ingested assets cannot be mirrored in the SDAS anymore. Deltas of present datasets cannot be stored and the 'evolution path' of datasets present is lost.

Contingency Solution Additional database hardware and storage space can be added. The SDAS can be distributed over several machines and should scale almost linearly.

Risk ID 1

Risk Description Scalability issues emerging from a growing number of users querying datasets the SDAS with free form queries.

Risk Assessment :

Impact High

Probability Low

Description Querying the triple store of the SDAS for profiles and records in the SDAS results in sluggish query responds, time outs or memory issues.

Contingency Solution Next to additional hardware resources, the querying possibilities are restricted to default query templates similar to the ones documented in the appendix. Permissions for arbitrary queries are granted to e.g. scholars only upon request.

Risk ID 2

Risk Description Formulation of SPARQL queries too demanding for end users (e.g. archivists) who are interacting with the data stores (as opposed to the Workbench).

Risk Assessment :

Impact Low

Probability Low (Explanation: end users are likely to use the workbench for interacting with DURAARK data)

Description Retrieving meaningful information and records impossible for end-users. Faulty queries threaten system overall performance

Contingency Solution Provide dedicated (REST) query API encapsulating SPARQL queries in 'error proof' calls including time outs, LIMITs etc. Create dedicated UI for SPARQL query compilation e.g. using forms.

6 Licenses

IPR Type	IP used or generated	Software name	License	Information
software	generated	DURAARK Semantic Digital Archive (SDA) storage server	MIT	sub-component of D3.3, implemented as a standalone service
software	generated	DURAARK Semantic Digital Observatory (SDO)	MIT	sub-component of D3.3, implemented as a standalone service
software	generated	Semantic context enrichment	MIT	sub-component of D3.3 used for the semantic enrichment of ingested models with contextual LD based on NER
software	generated	Metadata extraction from IFC SPF	LGPL	sub-component of D3.3 implemented as python command line tool
software	generated	Metadata extraction from E57	MIT	sub-component of D3.3 command line tool
software	used	Stanford NLP	GPL v2	Named Entity Recognition (NER) and Information Extraction (IE) http://nlp.stanford.edu/ner/
software	used	Apache Jena	Apache License, 2.0	RDF processing framework https://jena.apache.org
software	used	IfcOpenShell	LGPL	IFC processing and geometry engine http://ifcopenshell.org/
software	used	libE57	other	Software Tools for Managing E57 Files http://www.libe57.org//
software	used	MySQL	GPL v2	Database backend http://www.mysql.com/
software	used	Virtuoso Open-Source Edition	GPL v2	Triple Store engine http://virtuoso.openlinksw.com

7 Conclusions & Impact

The “Semantic Digital Archive“ prototype consists of a collection of components which provide logic for the archival, enrichment and linking of data as well as storage for the resulting datasets. As such, these form a central part of the overall DURAARK system (described in D2.4) and are of significant importance for the project as a whole. Both, this deliverable as well as the integration of WP3 components into a coherent prototype in WP2 are important steps towards the DURAARK work plan and final system. The components described in this document implement the use cases UC1, UC3, UC8, UC9 and provide core of the system functionality.
Next to this, the datasets generated and stored within WP3 and related components facilitate dissemination of DURAARK results as open public data - for instance the crawled Linked Datasets or the enriched building models - constituting outcomes in their own right. As such, individual components described in this deliverable will be published and disseminated as individual project results and made available as self-contained resources. This includes the enrichment and crawling functionalities as well as the provided datasets in the SDAS and SDO.
This being the first prototype of the SDA functionalities and data are neither complete nor final at this stage and will be incrementally added and improved. Along with progress in WP2, integration will be refined and functionalities will be adjusted in line with the improvement of architecture and use cases. This will be further facilitated through evaluation activities in WP7.

Glossary

buildingSMART Data Dictionary bSDD The international reference repository of building related concepts governed by the buildingSMART organization. Based on the International Framework for Dictionaries (IFD).. 39

Building Information Model (BIM) Object-oriented, parametric and process-oriented data structures to organize information relevant to buildings. 39

Industry Foundation Classes (IFC) The Industry Foundation Classes is an open interoperability model for the exchange of information reltated to building and construction. 39

International Framework for Dictionaries (IFD) Conceptual framework and data model to organize building-related information. Formally specified in the ISO 12006 parts 2 and 3. Basis for the reference vocabulary. 39

Linked Open Data (LOD) Set of accessible information published openly for reuse. The envisioned 'global graph of data' or 'cloud' employs Semantic Web technologies to cross-reference data across networks. Among its most important nuclei is the DBPedia data set that is derived from the WikiPedia corpus long. 39

LOD Linked Open Data long. 39

Long Term Preservation (LTP) The internationally widely accepted Open Archival Information System (OAIS) framework, defines preservation as

> "The act of maintaining information, Independently Understandable by a Designated Community, and with evidence supporting its Authenticity, over the Long Term."

(source: OAIS "Pink Book"). 39

Open Archival Information System (OAIS) Framework defining general concepts and best practises for Digital Long Term Preservation. 39

Resource Description Framework (RDF) Conceptual approach to modelling information. It is based on triples containing a 'subject' 'predicate' and 'object' that are described with Uniform Resource Identifiers (URI), most often reachable via web protocols such as HTTP. RDF models form directed graphs that can span across networked locations. Popular clear-text serialization formats include RDF/XML, N3 and Turtle. long. 39

Semantic Digital Archive (SDA) a part of the DURAARK framework that stores snapshots of linked data sets referenced from archives and their descriptions. 2, 39

Semantic Digital Observatory (SDO) a part of the DURAARK system that crawls, fetches, monitores and updates external data sets stored for preservation in the SDA. 39

STandard for the Exchange of Product data (STEP) A group of information standards covering a wide spectrum of engineering domains grouped under the ISO 10303 series of standards. 39

STEP Physical File (SPF) A clear-text file format defined in the STandard for the Exchange of Product Data (STEP). Specified in the ISO 10303, part 21 and often referred to as "Part 21 file" or "SPFF" (STEP Physical File Format). 39

References

[1] Jakob Beetz, Michelle Lindlar, Stefan Dietze, Ujwal Gadiraju, Dag Field Edvardsen, and Lars Bjørkhaug. D3.2- ontological framework for a semantic digital archive. 2014.

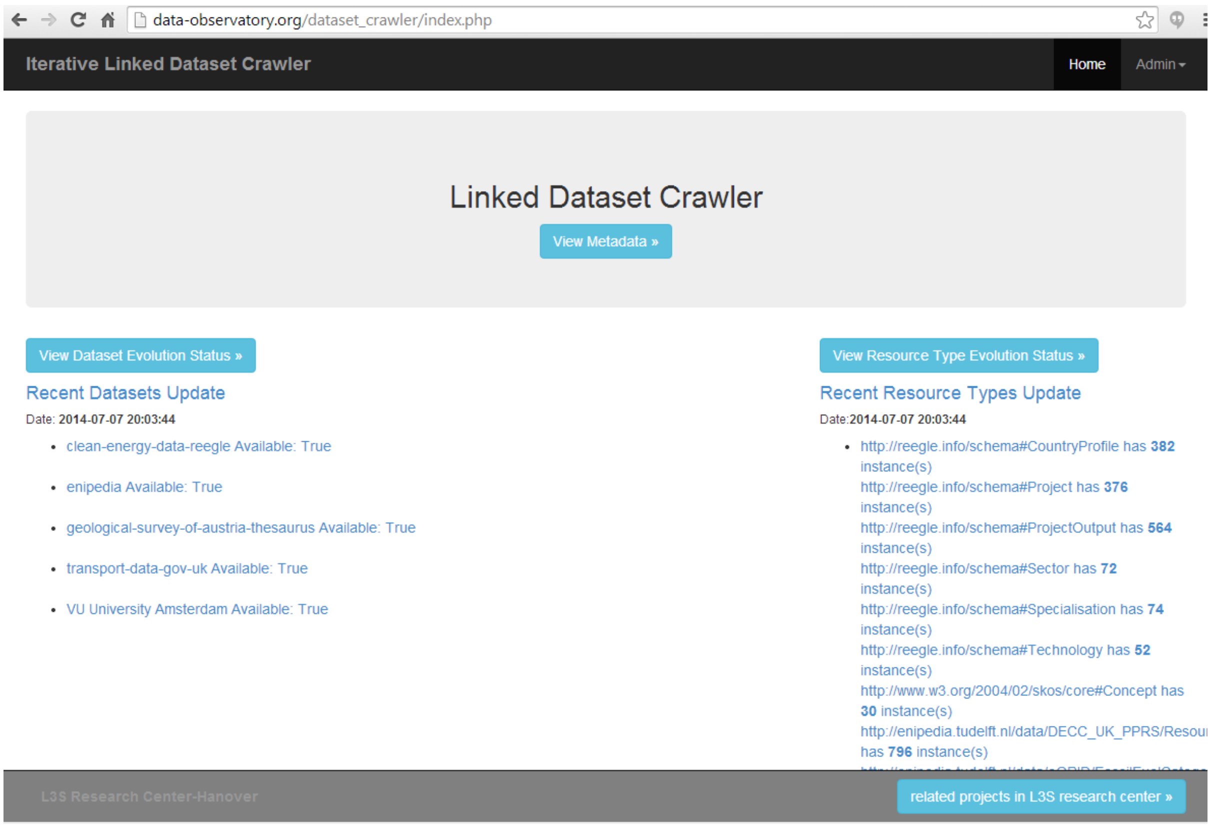
[2] Jakob Beetz, Michelle Lindlar, Stefan Dietze, Ujwal Gadiraju, and Qinqin Long. D3.1- metadata schema extension for archival systems. 2014.

[3] Stefan Dietze, Jakob Beetz, Ujwal Gadiraju, Georgios Katsimpras, Raoul Wessel, and René Berndt. Towards preservation of semantically enriched architectural knowledge. In *3rd International Workshop on Semantic Digital Archives (SDA 2013)*, September 2013.

[4] Besnik Fetahu, Stefan Dietze, Bernardo Pereira Nunes, Davide Taibi, and Marco Antonio Casanova. Generating structured profiles of linked data graphs. In *Proceedings of the 12th International Semantic Web Conference*. Springer, 2013.

A Example queries for exploring the SDO

In this section the prototypical web interface for browsing the current contents of the Semantic Digital Observatory (SDO) is elucidated by providing a number of concrete examples. The interface is made available at the following URL: `http://data-observatory.org/dataset_crawler/index.php`. A screen-shot of the interface is provided in Figure 18.


data-observatory.org/dataset_crawler/index.php
Iterative Linked Dataset Crawler
Home
Admin
View Metadata »
View Dataset Evolution Status »
View Resource Type Evolution Status »
related projects in L3S research center »

Linked Dataset Crawler

Recent Datasets Update

Date: 2014-07-07 20:03:44

- clean-energy-data-reegle Available: True
- enipedia Available: True
- geological-survey-of-austria-thesaurus Available: True
- transport-data-gov-uk Available: True
- VU University Amsterdam Available: True

Recent Resource Types Update

Date:2014-07-07 20:03:44

- http://reegle.info/schema#CountryProfile has **382** instance(s)
  http://reegle.info/schema#Project has **376** instance(s)
  http://reegle.info/schema#ProjectOutput has **564** instance(s)
  http://reegle.info/schema#Sector has **72** instance(s)
  http://reegle.info/schema#Specialisation has **74** instance(s)
  http://reegle.info/schema#Technology has **52** instance(s)
  http://www.w3.org/2004/02/skos/core#Concept has **30** instance(s)
  http://enipedia.tudelft.nl/data/DECC_UK_PPRS/Resou has **796** instance(s)

L3S Research Center-Hanover

Figure 18: **The Semantic Digital Observatory lists datasets that has been crawled**

In general, the components, as described in the SDA (see 2) and SDO (see 3) sections of this report, can be queried using the SPARQL and REST endpoints. In this section a number of examples are provided that allow the exploration of the SDO implementation. All queries are mere examples and results on the live data depends on the current state of the repository and may vary considerably.

A.1 Contents of the archive and its profiles

In order to get an overview and a list of all currently crawled datasets in the SDO, the following url can be entered unchanged into the address-field of a web browser.
`http://asev.l3s.uni-hannover.de:3000/sdoinfo`
This provides a JSON record listing all currently available datasets that have been profiled by the SDO
The SDO features a SPARQL endpoint that can be used to explore their contents. Beyond the examples that are rendered into the corresponding HTML pages in the DURAARK workbench, the following listings provide examples of exploring the contents of the profiler. In later implementation stages of the DURAARK toolset, this will be integrated in more user-friendly ways that will help librarians and archivists to maintain the data in accordance to the requirements stated in UC9

A.1.1 SPARQL request for a list of dataset IDs from the profiles dataset

The following query lists all datasets with their IDs currently covered by the SDO

```
SELECT DISTINCT ?dataset
WHERE {
    ?dataset a <http://rdfs.org/ns/void#Dataset>
}
```

Listing 5: Example query to list all available datasets

This link URL-encodes the above SPARQL query and executes it on the SDO instance Notice that the URL above returns a set of dataset ids that can be used in the URL-templates listed below.

A.1.2 SPARQL request for Top Scored Dataset Topics

Replace the `[[DATASET_ID]]` and the square brackets around it in the url below below with one acquired above to use this url in a web browser.

```
SELECT * WHERE {
    ?linkset void:target
        <http://data-observatory.org/lod-profiles/dataset/[[DATASET_ID]]>.
    ?linkset vol:hasLink ?link.
    ?link vol:linksResource ?category.
    ?link vol:hasScore ?score.
}
ORDER BY DESC (?score)
LIMIT 100
```

Listing 6: Example query to list all links from a particular dataset

This link URL-encodes the above SPARQL query and returns the top scored dataset topics. In combination with the dataset ids, the returned topics can be used to construct new URLs based on the following URL-templates .

A.1.3 SPARQL request for Top Specific Datasets for a Topic

The following query lists all datasets that are have been tagged with a specific topic based on the relevance score. The place holder `[[CATEGORY_NAME]]` can be replaced by valid categories such as "Architecture", "Construction" "Preservation" etc.

```
SELECT * WHERE {
    ?dataset a void:Linkset.
    ?dataset vol:hasLink ?link.
    ?link vol:linksResource
        <http://dbpedia.org/resource/Category:[[CATEGORY_NAME]]>.
    ?link vol:hasScore ?score.
}
ORDER BY DESC (?score)
LIMIT 100&
```

Listing 7: Example query to list all links from a particular dataset

This link lists all datasets relevant to the category "Architecture"

A.1.4 Top Specific Resources and Datasets for a Topic

```
SELECT ?dataset ?link ?score ?link_1 ?entity ?resource WHERE {
    ?dataset a void:Linkset.
    ?dataset vol:hasLink ?link.
    ?link vol:linksResource
        <http://dbpedia.org/resource/Category:[[CATEGORY_NAME]]>.
    ?link vol:derivedFrom ?entity.
    ?link vol:hasScore ?score.
    ?link_1 vol:linksResource ?entity.
    ?dataset vol:hasLink ?link_1.
    ?link_1 vol:derivedFrom ?resource
}
ORDER BY DESC(?score)
```

```
LIMIT 100
```

Listing 8: SPARQL query for top datasets relevant for particular topic sorted by relevance score

This link URL encodes the above query and and lists all Top Specific Resources and Datasets for the topic "Architecture".

A.1.5 Top Specific Entities for a Topic

```
SELECT DISTINCT ?link_1 ?entity ?score
WHERE {
    ?dataset a void:Linkset.
    ?dataset vol:hasLink ?link.
    ?link vol:linksResource
        <http://dbpedia.org/resource/Category:[[CATEGORY_NAME]]>.
    ?link vol:derivedFrom ?entity.
    ?link vol:hasScore ?score.
    ?link_1 vol:linksResource ?entity.
    ?dataset vol:hasLink ?link_1.
    ?link_1 vol:derivedFrom ?resource
}
ORDER BY DESC(?score)
LIMIT 100
```

Listing 9: SPARQL query for top datasets relevant for particular topic sorted by relevance score

This link provides the top top specific entities for a topic.

A.2 Examples exploring the datasets with JSON

A.2.1 Query Dataset by ID

First url gives us a list of dataset-ids:
executes the above query in the SDO. The request results in a JSON answer similar to the one provided in following
Using that url in a webbrowser gives us a json return, and the first item in the json array is:

```
{
    "dataset": {
        "type": "uri",
        "value":
            "http://data-observatory.org/lod-profiles/dataset/clean-energy-data-reegle"
    }
}
```

Listing 10: JSON structure returned from dataset ID query.

The first dataset_id from here is (looking away from the standard part of the dataset.value): "clean-energy-data-reegle"

```
SELECT * WHERE {
    ?linkset void:target
        <http://data-observatory.org/lod-profiles/dataset/[[DATASET_ID]]>.
    ?linkset vol:hasLink ?link.
    ?link vol:linksResource ?category.
    ?link vol:hasScore ?score.
}
ORDER BY DESC (?score)
LIMIT 100
```

Listing 11: JSON structure returned from dataset ID query.

This URL-encoded query link can be used to - "clean-energy-data-reegle" - we get this link.
The result from the list above is (in json format) a list of "Top scored dataset topics". The second (more interesting than the first) real item from this list is this json-part:

```
{
    "linkset": {
        "type": "uri",
        "value":
            "http://data-observatory.org/lod-profiles/linkset/clean-energy-data-reegle"
    },
    "link": {
        "type": "uri",
        "value":
            "http://data-observatory.org/lod-profiles/link/469e960f-cc1b-4b61-85f9-143dd30952ab"
    },
    "category": {
        "type": "uri",
        "value": "http://dbpedia.org/resource/Category:Countries"
    },
    "score": {
        "type": "typed-literal",
        "datatype": "http://www.w3.org/2001/XMLSchema\#double",
        "value": "2.71687e-09"
    }
}
```

Listing 12: JSON reokt tot top scored dataset topics.

This says that the category Countries is the one with the second highest score and the score value is 2.71687e-09 [[Besnik: what does that mean .. how is that value calculated]]

Replacing the place holder `[[CATEGORY_NAME]]` in in this link with the string "Countries" as in this link yields the following result which is a json-formatted list of top datasets for that category (Country). The first real entry is the following one:

```
{
    "dataset": {
        "type": "uri",
y
        "value":
            "http://data-observatory.org/lod-profiles/linkset/open-data-thesaurus"
    },
    "link": {
        "type": "uri",
        "value":
            "http://data-observatory.org/lod-profiles/link/b18fd79f-400a-4597-8d96-477643b1ecd0"
    },
    "score": {
        "type": "typed-literal",
        "datatype": "http://www.w3.org/2001/XMLSchema\#double",
        "value": "4.52637e-05"
    }
}
```

Listing 13: top datasets for that category.

This tells us that the top dataset for the topic "Countries" is "open-data-thesaurus" (if we remove the boilerplate part of the uri).
We can also ask for the Top Specific Resources and Datasets for a Topic. This is related to the fourth url template: http://data-observatory.org/lod-profiles/sparql?default-graph-uri=http
If we continue to use our topic "Countries" the url becomes: http://data-observatory.org/lod-profiles/sparql?default-graph-uri=http
When entering that in a webbrowser we get another json result. The first real items in the json array is:

```
{
    "dataset": {
        "type": "uri",
        "value":
            "http://data-observatory.org/lod-profiles/linkset/open-data-thesaurus"
    },
    "link": {
        "type": "uri",
        "value":
            "http://data-observatory.org/lod-profiles/link/b18fd79f-400a-4597-8d96-477643b1ecd0"
    },
    "score": {
        "type": "typed-literal",
        "datatype": "http://www.w3.org/2001/XMLSchema\#double",
        "value": "4.52637e-05"
    },
    "link\_1": {
        "type": "uri",
        "value":
            "http://data-observatory.org/lod-profiles/link/98871260-efd0-46ba-9051-398781384f48"
    },
    "entity": {
        "type": "uri",
        "value": "http://dbpedia.org/resource/Austria"
    },
    "resource": {
        "type": "uri",
        "value": "http://vocabulary.semantic-web.at/OpenData/Austria"
    }
}
```

Listing 14: top datasets for that category.

So the top dataset for the topic contries is "open-data-thesaurus" and the top resource is "Austria".
Another example is to construct a SPARQL query that return the "Top Specific Entities for a Topic". If the placeholder `[[CATEGORY_NAME]]` in this URL-template, is replaced with the string "Countries" (which results in this link) yield the following result (as JSON)

```
    From that url we get back some json, and the first element in the json array is
        the following:

    {
        "link\_1": {
            "type": "uri",
            "value":
                "http://data-observatory.org/lod-profiles/link/98871260-efd0-46ba-9051-398781384f48"
        },
        "entity": {
            "type": "uri",
            "value": "http://dbpedia.org/resource/Austria"
        },
        "score": {
            "type": "typed-literal",
            "datatype": "http://www.w3.org/2001/XMLSchema\#double",
            "value": "4.52637e-05"
        }
```

```
}
```

Listing 15: top datasets for that category.

This tells us that the top specific entity for the topic "Countries" is Austria, and that the score is 4.52637e-05.