DELIVERABLE

Project Acronym:	**CARARE**
Grant Agreement number:	**250445**
Project Title:	***Connecting ARchaeology and ARchitecture in Europeana***

D2.5 White paper on CARARE technical approach

Revision: [final]

Authors:

Dimitris Gavrilis, DCU
Vassilis Tzouvaras, NTUA

With contributions from:

Christian Ertmann-Christriansen, KUAS

Project co-funded by the European Commission within the ICT Policy Support Programme		
Dissemination Level		
P	**Public**	
C	**Confidential, only for members of the consortium and the Commission Services**	**X**

Revision History

Revision	Date	Author	Organisation	Description
1.0	29/07/2010	Dimitris Gavrilis	DCU	Draft
1.1	02/08/2010	Dimitris Gavrilis	DCU	Revised draft
1.2	08/09/2010	Vassilis Tzouvaras	NTUA	Revised draft
1.3	09/09/2010	Dimitris Gavrilis, Vassilis Tzouvaras	DCU & NTUA	Revised draft, Workflow
1.4	10/09/2010	Dimitris Gavrilis	DCU	Technical requirements, References, Annex I,
1.5	22/09/2010	Christian Ertmann-Christiansen, Dimitris Gavrilis	DCU	Major revisions regarding metadata editing policies, Annex I
1.6	16/11/2010	Dimitris Gavrilis, Vassilis Tzouvaras	DCU, NTUA	Major revisions regarding ingest protocol between mapping tool and repository
1.7	26/11/2010	Dimitris Gavrilis	DCU	Revisions regarding ingest protocol

Statement of originality:

This deliverable contains original unpublished work except where clearly indicated otherwise. Acknowledgement of previously published material and of the work of others has been made through appropriate citation, quotation or both.

Contents

1. Executive Summary

This deliverable describes in detail the overall technical architecture of the CARARE project. The goal of the project is to harvest content regarding archaeology and architecture (mostly monuments) from a number of content providers, and deliver it to Europeana. The technical architecture will be implemented and supported by the two technical partners of the project: National Technical University of Athens (NTUA) and Digital Curation Unit of the Athena Research Centre (DCU). The architecture specifies a three stage process, described below.

CARARE content is heterogeneous, as each content provider uses their own schema and catalogues information in different ways, has different coordinate systems to denote location, etc. Thus, the first step was to create a rich and powerful schema (described in another deliverable (D2.2), and known as the CARARE schema) that can encompass most of the information from all the content providers while balancing its complexity. All the individual schemas from all the content providers will be mapped to the CARARE schema and then all their data will be harvested and the actual mapping will take place. This first stage of the process will be handled by the NTUA's mapping tool.

The second stage involves the semantic enrichment of the content; in order to accomplish that, content must be ingested into a repository which will provide the necessary enrichment services. This repository and relevant services will be implemented and managed by the DCU. The enrichment process involves several distinct functions: checking for content quality and notifying providers whose content needs enrichment; adding semantic relations between items among different collections (and content providers); handling geographic coordinates (normalizing different coordinate systems, finding items whose proximity is below a certain threshold); previewing the content, etc. The repository will handle the whole enrichment process by maintaining all changes (versioning) and conforming to established preservation standards.

The third and final stage of the process is the delivery of the transformed content to Europeana in the appropriate format (currently EDM v5.2). For this, a transformation from the CARARE schema to EDM is needed; a specification has been made available through another deliverable (D2.2).

The CARARE system consists of two main systems: the mapping tool and the repository. A crucial point is communication and information flow between these two systems. This has been given special attention and is described in detail in this document. A communication protocol has been designed especially for that; it uses REST based web services for communication, while information is packaged in special packages (submission information packages or SIPs) that focus on information preservation.

2. Introduction

This document presents the technical architecture that will be implemented within the CARARE project. The technical architecture describes two main components: the mapping tool and the CARARE repository as well as the communication mechanism between them. All technologies, functionalities and protocols that will be used throughout the system are presented.

3. Overall Architecture

This section describes the overall architecture of the CARARE system.

Within the CARARE project a system will be developed (CARARE system) that will enable content providers to map their native collection schemas to a unified schema (CARARE schema) and then to a schema supported by Europeana (currently: EDM v5.2). The metadata will be made available for Europeana to harvest and integrate into its database.

In the figure below the overall architecture of the CARARE system is presented in figure 1 below:

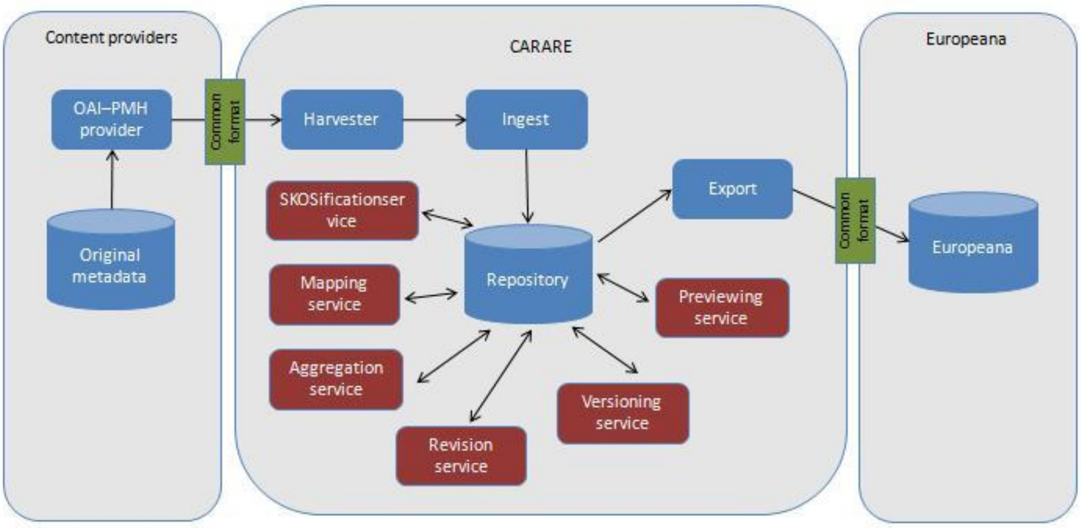

Figure 1: The overall architecture of the CARARE system

In figure 2 below, the workflow that is going to be implemented in the project is presented. The harvester harvests metadata from the content provider and then maps them to the CARARE schema. Afterwards, the metadata are ingested into the CARARE repository. Inside the repository, the content providers can search/browse their digital objects and use the metadata enrichment tool to define any semantic relations (e.g. identity relations) between objects that reside inside the repository). The repository will map the metadata into EDM and allow the content providers to preview them. Finally, the repository will expose the providers' metadata to OAI-PMH allowing Europeana to harvest them.

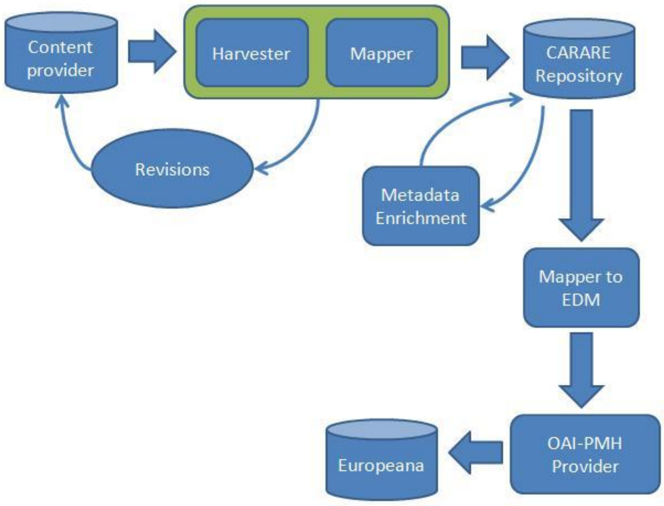

Figure 2: CARARE Workflow

4. Content providers

The content providers must expose their digital objects metadata (to be ingested to Europeana) using the OAI-PMH protocol to the harvester subsystem. Alternatively they can export them to an XML format and upload them either through HTTP or FTP.

There are two main requirements for the content providers' repositories:

- The digital objects must be available online
- It is necessary for the digital objects to have unique identifiers. It is not necessary for the digital objects to have persistent identifiers.

5. Harvester

The harvester subsystem harvests metadata from the content providers using the OAI-PMH protocol. Although the OAI-PMH is the preferred way for harvesting the CARARE metadata, in case content providers cannot setup an OAI repository, they can upload metadata either through HTTP or FTP. In order to speed up the uploading process, the XML files can be packaged into a ZIP format.

6. Schema mapper

The metadata schema mapper subsystem allows the content providers to create templates for mapping the metadata from their proprietary schemas to the CARARE metadata schema that will be defined within the project.

The tool will allow to manually map providers fields to the CARARE schema. The mapping transformations will be stored and the provider will be able to edit them at any point if necessary. It will support value concatenation (e.g. many-to-one mappings), conditional mappings and string manipulation services. Providers that have metadata in known formats will be able to omit this step (they have also to provide the XSLT transformation to the CARARE schema). Finally, the tool will provide XML previewing and quality control services.

7. Repository

The CARARE repository will hold the content providers' metadata and perform a number of functions that will aim at enriching these metadata. The repository consists of a set of storage spaces and of a set of services that perform various operations on the metadata. A graphical representation of the repository can be seen in Figure 3 below.

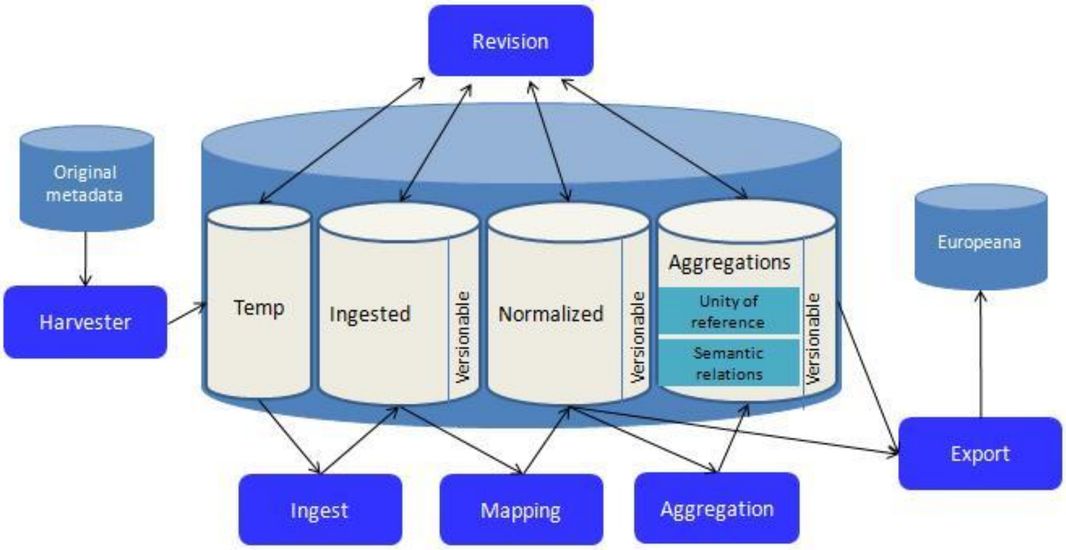

Figure 3: Graphical representation of the CARARE repository

In order to access the repository services, the content providers must obtain a username and password. With these credentials, they can login and search/browse through their collections. After locating a digital object, the system will be able to display the various metadata schemas:

- Native metadata
- CARARE schema
- EDM schema

Allowing the content provider to inspect the digital object and locate any errors or make improvements if necessary. Any modification must be made at the content provider's source repository and re-ingested into the CARARE repository following the ingest process described above.

The CARARE repository will keep track of the different versions of the metadata schemas (both the Native and CARARE). So for every ingest that is performed, the mapping tool will push to the repository 2 files: the Native schema and the CARARE schema.

The EDM document will not be stored in the repository. Instead it will be generated on the fly using the appropriate transformation from the CARARE schema to EDM.

When an item is withdrawn in the repository, during the successive harvest from Europeana, the specific item will be marked as deleted (status attribute at the record level of the OAI-PMH protocol - http://www.openarchives.org/OAI/openarchivesprotocol.html#deletion)

In Figure 4 below the content provider's basic workflow is presented.

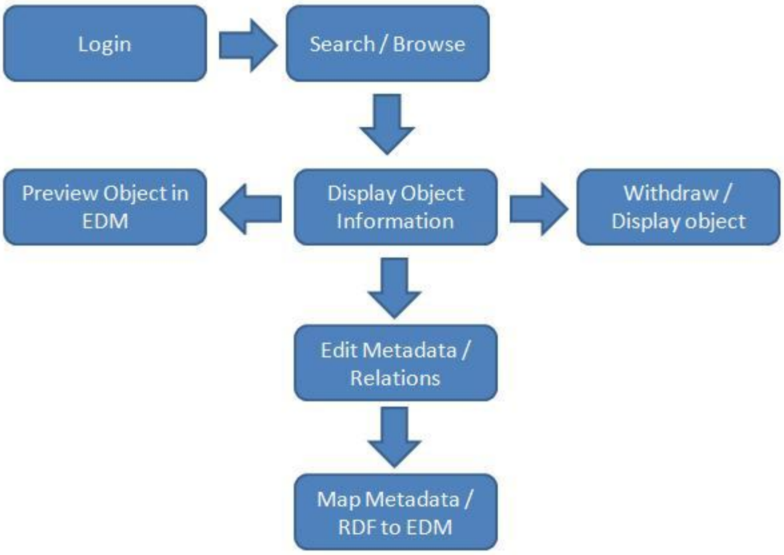
Login
Search / Browse
Preview Object in EDM
Display Object Information
Withdraw / Display object
Edit Metadata / Relations
Map Metadata / RDF to EDM

Figure 4: Content providers' workflow

Digital object outline

Each digital object inside the repository will consist of the following datastreams:

Datastream	Description
DC	The basic Dublin Core standard 15 element set metadata schema.
NATIVE	The content provider's original native metadata schema.
CARARE	The transformed metadata schema from the original one to the CARARE schema (performed by the mapper).
EDM	The transformed metadata schema from CARARE to the latest stable version of the EDM schema.
RELS-EXT	A set of RDF statements describing the relations between the specific object and other objects (or containers).
RELS-INT	A set of RDF statements describing the internal datastreams of the specific digital object. (*optional*)
PREMIS	This metadata schema contains a log of the operations performed on the digital object in PREMIS format.

All datastreams for each digital object support versioning.

Repository communication with schema mapper

After the content providers have finished mapping their metadata to the CARARE schema, the metadata for all digital objects will be mapped and ingested into the repository.

Method of communication

The ingestion mechanism will rely on a set of REST-based web services. The web service will allow the mapping tool to ingest information into the CARARE repository. The web service will require one parameter which is the URL of the submission package. The web service will download the submission package, verify it and ingest it into the repository.

The REST-based web service will be available at http://store.carare.eu and will have the following specifications:

- Will implement the GET method of HTTP request
- Will accept the one variable (GET type) with name package. This variable will contain a proper URL of the location of the package to ingest (e.g.: http://store.carare.eu/ingest/index.php?package=http://194.177.192.14/carare/package1.zip)

After downloading and processing the submission package, the ingest service will return an XML formatted response (see below).

Submission package

The ingest process will utilize a REST based web service which will be provided by the CARARE repository. The service will recognize submission packages that will be verified and ingested. Each submission package will correspond to a unique item. These submission packages will have the following features:

File name:	[item_id].zip
File type:	Compressed file in zip format
Contents:	3 XML files • info.xml • native.xml • carare.xml • mapping.xsl
info.xml	Contains the following information regarding an item: • content provider id *[mandatory]* • content provider name *[mandatory]* • user id (user who published the item) *[optional]* • user name (user who published the item) *[optional]* • native item identifier (unique) *[mandatory]* • native item name *[optional]*
info.xml	Contains the following information regarding a package: • package timestamp (created) *[mandatory]* • package size (total size of package) *[mandatory]* • items list (a listing of all items included in the package. Each item in

	the items list will contain the id, name attrivutes (as in the item level info.xml format) plus the filename of the item *[mandatory]*
native.xml	Contains the native record (well formed xml)
carare.xml	Contains the carare record (well formed xml)
mapping.xsl	Contains the xslt document that was used to produce the transformation

The service can handle either **single package submissions** or **bundles of submission packages**. A submission package is a compressed (.zip format) file which contains multiple submission packages (as described above).

File hierarchy example of a bundle:

- upload_1.zip
    - info.xml
    - item_1.zip
        - info.xml
        - native.xml
        - carare.xml
        - mapping.xsl
    - item_2.zip
        - info.xml
        - native.xml
        - carare.xml
        - mapping.xsl
    - …

There are two different kinds of info.xml files: a) info.xml at the package level and b) info.xml at the item level. The info.xml files contain important information regarding either the package or the item. Details about the structure of the info.xml files and their contents are shown below:

Example of info.xml file [package level]:

```
<?xml version="1.0" encoding="UTF-8"?>
<package timestamp="" size="" >
<items>
        <item id="4" name="test 4" filename="item1" />
</items>
</package>
```

Example of info.xml file [item level]:

```
<?xml version="1.0" encoding="UTF-8"?>
<provider id=”2” name=”DCU” />
<user id=”13” name=”Dimitris Gavrilis” />
<item id=”51” name=”test” />
```

Error codes

After processing the submission package, the service will provide a list with all errors that were recognized by the system (if any). The errors will be structured in XML format and will contain the following information:

- Content provider identifier
- Item identifier
- Error code
- Error description

Example of a response:

```
<response>
<error provider_id=”2” item_id=”51” id=”1”>Not Well Formed</error>
</response>
```

Error codes:

Error code	**Description**
1	Invalid submission package (files missing, empty files)
2	Not well formed (at least one of the xml files is not well formed)
3	Content provider not found
4	Invalid item identifier
5	Submission file download error

This approach will enable the mapping tool to submit a bundle of items to the carare repository, after the verification and ingestion process if the system returns an error for e.g. 5 items, it is possible to re-ingest them separately (in another bundle of 5 or in 5 individual submissions).

Repository export to Europeana

The repository will allow for its content to be mapped to EDM and exposed to Europeana. For this, there will be a built in metadata mapping mechanism that will employ an XSLT schema that will perform the necessary transformation. A set of related web services will trigger the transformation for:

- a unique digital object
- an entire collection

Each transformed EDM schema will be stored in the **EDM** datastream inside the repository. This way, the processing load on the server will be reduced since the server will only need to create the EDM datastreams when:

1. A new CARARE datastream is ingested (either from the interface with the schema mapper or directly from the repository).
2. An existing CARARE schema is edited directly on the repository.
3. The CARARE ➔ EDM mapping description is modified.

An OAI-PMH provider will expose the contents of the repository to Europeana (only the EDMv5 datastreams) making possible for Europeana to harvest the provider’s collections on demand.

Metadata enrichment tool

The metadata enrichment tool will allow the content providers to enrich their metadata before they are ingested into Europeana. This tool will provide the providers to enrich their records with semantic relations (populated from an ontology). This relation enrichment functionality will be implemented using and html based RDF editor will allow the content providers to add/edit the RDF relations with other objects.

Metadata revision policy

In order to minimize errors and reduce complexity, any modifications the content providers make into their metadata must be made using their own repositories and then re-ingest the respective digital objects into the CARARE repository following the ingest procedure (through the harvester and mapper subsystems).

EDM previewing

The EDM previewing service will allow the content providers to browse anyone of their digital objects into EDM. For this, an XSLT transformation will transform the contents of the CARARE schema to EDM (**without storing them as a new version**) and display them to the content provider through the item previewing service.

8. Content provider repository workflow

The content providers after uploading their metadata into the mapping service (or making them available for harvesting), the metadata will be automatically transformed into the CARARE schema and be ingested into the repository. After that, the content provider will be given a user account through which access to the repository services will be made available.

A schematic of the user workflow is presented in figure 5 below:

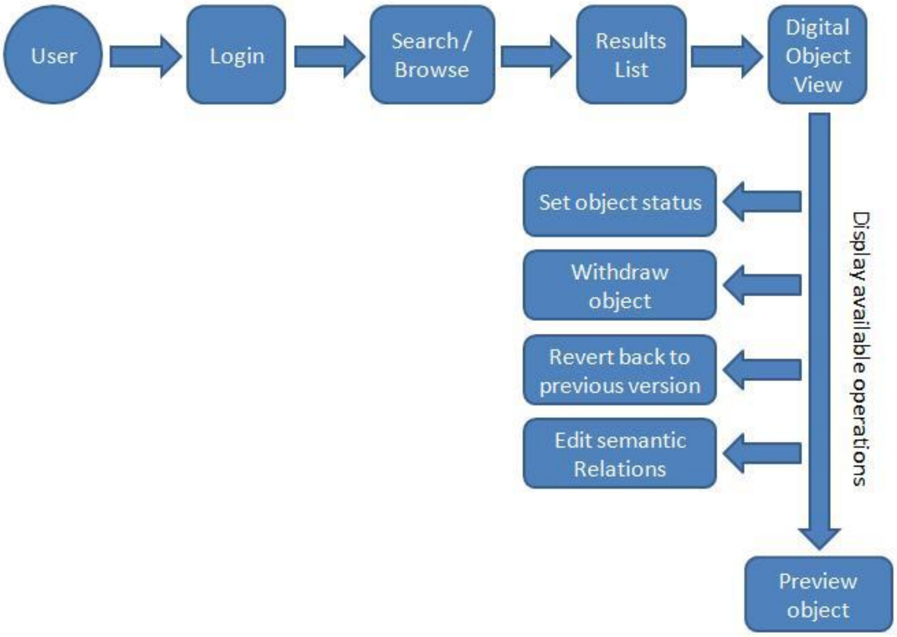

Figure 5: User workflow

When user visits the web site the repository they will need to login into the system. The repository services are digital object driven. This means that after logging in, the user will have to locate a digital object to work with. This will be accomplished with the digital object search/browse service. The service will search using a specific set of indexes (small subset of the CARARE schema). After the search request, the results list is presented and the user locates an object to work with. The digital object view comes up through which all available operations on the specific object are presented. These operations are:

- **Set object status**. The user can decide whether to enable or withdraw the item. This only affects the objects visibility to Europeana (no deletions allowed in the repository).
- **Edit RDF relations**. The RDF editing service is invoked through which the user can add/edit/delete RDF relations between the selected object and other digital objects or containers.
- **Preview datastream**. The user can preview any of the datastreams in XML format or in HTML (if the associated XSLT exists). HTML previewing will made available for: EDM and CARARE.
- **Preview Versions.** The user will also be able to see all the changes made to each record and preview all the changes made through time. This will be accomplished through the versioning mechanism that the CARARE repository provides.

The RDF relations that the users can edit will serve the purpose of allowing the content providers to enrich their collection with relations with objects that belong to other content providers (although it will be possible to create relations between items that belong to the same provider). This has the advantage that it does not require persistent URIs for all items that are linked with each other since all items reside in a single repository.

9. Technical requirements and details

Regarding the technical requirements/details two main technologies will be used for constructing the CARARE system:

<u>Application development:</u> Java and PHP.

Java (SE >= 1.5.X) will be used for the mapping tool and the repository main web services (the services developed will be incorporated into fedora-commons). PHP (>=5.x) will be used for the construction of the user web based system (the repository's frontend to the content providers) .

<u>Databases:</u> MySQL, PostgreSQL, Low level storage

The MySQL relational database (>=5.x) will be used for indexing the repository's metadata. The metadata will be stored into the repository using Fedora-common's file system based low-level storage. The mapping service will employ a PostgreSQL database.

<u>Web services:</u> REST/SOAP

Most web services will be REST based. However some of the repository services that communicate with Fedora-commons are based on SOAP.

10. References

In this section you may find links to the services and technologies that are used or described throughout this document.

- MySQL. http://www.mysql.com
- PostgreSQL. http://www.postgresql.org/
- Fedora-commons. http://www.fedora-commons.org/
- Java. http://www.java.com/
- PHP. http://www.php.net/
- Web services. http://www.w3.org/TR/ws-arch/
- XSLT. http://www.w3.org/TR/xslt
- XML Schema. http://www.w3.org/XML/Schema
- Extensible Markup Language (XML). http://www.w3.org/XML/
- Resource Description Framework (RDF). http://www.w3.org/RDF/
- OAI-PMH. http://www.openarchives.org/