Project N°: 262608

Acronym: Data without Boundaries

Deliverable D12.1

(Database supporting the full metadata model)

Work Package 12

(Implementing Improved Resource Discovery for OS Data)

Reporting Period:	**From: Month 18**	**To: Month 48**
Project Start Date:	**1st May 2011**	**Duration: 48 Months**
Date of Issue of Deliverable:	**1st October 2013**	
Document Prepared by:	**Partner 6, 8, 16, 18, 19, 25**	**NSD, RODA, MT, UKDA, KNAW-DANS, CED**

Combination of CP & CSA project funded by the European Community
Under the programme "FP7 - SP4 Capacities"
Priority 1.1.3: European Social Science Data Archives and remote access to Official Statistics

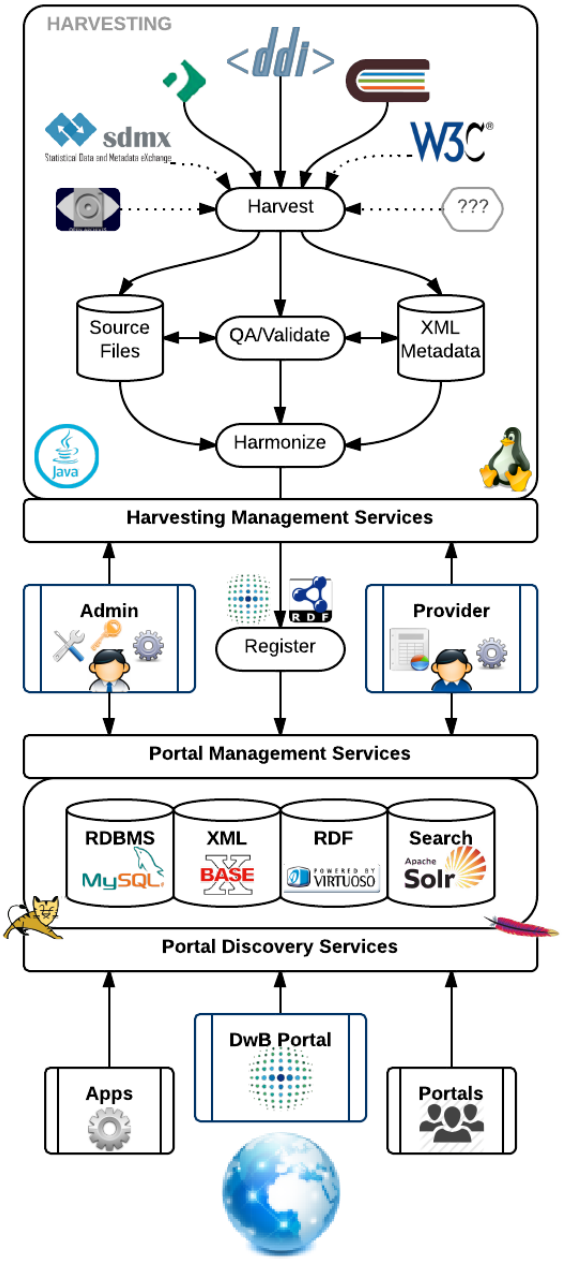
The research leading to these results has received funding from the European Union's Seventh Framework Programme (FP7/2007-2013) under grant agreement n° 262608 (DwB - Data without Boundaries).

TABLE OF CONTENTS

0. OVERVIEW

The diagram on the right provides a high level overview of the architecture of the DwB OS Portal being implemented under WP12. It has been designed based on inputs from other work packages (particularly WP8 and WP5), consultations with domain experts, community trends and best practices, and internal research. The architecture represents a general technical platform that makes it possible to implement a variety of services, in this deliverable we focus on describing the Resource Discovery elements.

Data without Boundaries
WORK PACKAGE 12
DATA DISCOVERY PORTAL



The platform is composed of the following parts:

- The Harvesting Framework and related tools to facilitate the retrieval and ingestion of metadata in various formats from participating providers (see paragraph below).
- An Administrator Dashboard and underlying management services to configure and monitor the infrastructure, and orchestrate the various tasks
- A Provider Portal enabling participating agencies to gain insights on their information present in the system. This includes reports (quality assurance, usage, harvesting) and tools to control their profile, visibility of metadata, or profile information.
- **The Storage Framework (databases) where the information is hosted and indexed in various shapes and form (Relational, XML, RDF) in order to support the search, discovery and other services. The metadata is based on commonly used standards and the DwB Metadata Model.**
- The Discovery Services exposing the platform and information to the outside world, and enabling integration in applications or web site, along with the Discovery Portal user interface

The implementation leverages several open source technologies and packages, chosen for the robustness. The portal prototype implementation is weaved around these in an innovative fashion to deliver an enterprise grade scalable solution.

The process of harvesting metadata will be heavily dependent on both the provider side and the harvesting technology that is used. It requires that we find points that are exposing organized metadata, but it allows these points to vary greatly in which metadata standard - and model - is implemented. In order to develop the most general harvesting system possible, WP12 has contacted a number of potential providers (Italy, France, Denmark, Slovenia, Finland, Switzerland and Sweden) to run tests and organize the information into a common storage, for use with the suggested portal demonstrator.

This document focuses on the Database Model and Storage and is Part 1/5 of the project deliverable. The potential use of the material (incl. metadata retrieval) will be covered by deliverable 12.4

1. METADATA MODEL AND STANDARDS

1.1 Discovery Model

The initial approach for the DwB portal was to design a new metadata model to support the various functionalities and features. Such a task was planned under WP8 and has been completed. Several important developments have also occurred since the project inception, in particular:

- The development of two other parallel practical implementations of derived metadata models under WP5 developed for the storage systems MISSY and CIMES, where hierarchical structures have to be taken into account and handled properly by the Discovery Portal
- The development and release of the General Statistical Information Model (GSIM)
- The emergence and formalization of the DDI-Discovery RDF
- The rising popularity of SKOS (Simple Knowledge Organization System) around statistical data management and efforts around publishing extensions to better support classification management (XKOS)

These have prompted the WP12 team to rethink the model / metadata serialization options, with the objective to adopt a model aligned on community best practices and standards.

After reviewing the various options, we choose to use the DDI Discovery RDF as a starting point as:

- Its mission focuses on discovery, paralleling the work package objectives
- Our initial assessment indicates significant overlap with the portal needs
- It is rapidly gaining in popularity amongst the DDI developer community
- The model is extensible and can be adjusted to cater for specific portal requirements or align with other models
- The WP12 team are in close contact with the lead implementers and have access to expert knowledge
- The specification itself is still under development, which offers the present project an opportunity to contribute to its development

The underlying RDF technology further justifies our choice by:

- Ensuring integration with the open/linked data and semantic web communities
- Offering a powerful and flexible framework that can cater for change and adjustment over time (less rigid than XML)
- Being backed by proven open source packages ensuring long term sustainability and scalability

WP12 is currently in the process of documenting mappings between the latest version of the DDI Discovery RDF specification and the DwB portal requirements to further confirm suitability and identify needs for adjustment to the specifications or for extensions. We also intend to examine its relationships to the other models listed above, in particular the WP8 outputs to ensure that we can meet all identified requirements.

We therefore expect the DwB Discovery model to take the shape of an RDF ontology extending the DDI discovery specification, itself leveraging widely used vocabularies such as Dublin Core, SKOS, and the likes.

1.2 Ingestion standards

In terms of harvesting and registering information into the portal repository, both development lines of the DDI, Codebook and Lifecycle, have been from the beginning identified as the core specifications. DDI is a flexible specifications that can be used in different ways (in particular Lifecycle) and, as a consequence, the XML content can vary significantly depending on the metadata producer, not only in terms of which elements are being used but how they are being populated. We therefore cannot generically support any form of DDI-C or DDI-L but rather need to focus on known tools outputs or flavors. So in terms of structure and content, the development will start by supporting a Nesstar based DDI-C as this is the most widely found flavour. For DDI-L we plan for supporting Colectica and DataForge:SledgeHammer (a tool recently released by Metadata Technology). Known project specific flavors will also be supported on a case by case basis (such as CIMES and MISSY).

We expect other flavors of harvesters to emerge as additional metadata providers are identified. The same is true for supporting other metadata standards such as SDMX, DCAT, or even Dublin Core. SDMX is explicitly mentioned in the DoW. However, SDMX is strong when it comes to aggregate data, whereas DwB is about microdata and this is the kind of data researchers need access to. In addition, SDMX is rarely used as a data maintenance or storage format, its main use is as a transport format. This severely limits its applicability for our present purposes.

The availability of an interface specifications is intended to facilitate such integration, whether within this project timeframe or beyond.

1.3 Minimal Metadata

To be able to register a resource with the portal, a minimal amount of metadata is necessary to ensure that (1) it can be properly indexed, (2) meaningful information can be delivered to the user, and (3) the user can be redirected to the resource location.

We believe at this point for the following to be the minimal set:

- *Title*: the resource name, to display in search result and for text search
- *Abstract*: A minimum level abstract is needed, to give a shorthand description of the *content* of the data collection.
- *Geography*: the spatial coverage of the resource underlying data (this is a fundamental search criterion). The minimal requirement is to provide a country. Note that if not present in the metadata, this can often be inferred based on the provider.
- *Time*: the time period the resource underlying data covers (a least a year). Note that if not present in a dedicated metadata element, experience shows that this information can very often be extracted from the title.
- *Location*: a url where the user can be redirected to retrieve or find additional information around gaining access to the resource.

Several metadata elements will be automatically tagged by the harvesting and registration mechanisms such as the metadata provider, original format, various timestamps, and others.

In many cases we would be able to ingest a resource solely based on the title. However, that would require that data providers would observe standardized best practices for developing titles, and for

some datasets titles could become very long. It is important to note that while we want to keep this as minimal as possible to encourage and facilitate broad participation in the portal, it is strongly recommended to provide additional information. At the user requests or to ensure quality of search results, we may also choose not to show resources with insufficient metadata to avoid cluttering the output.

1.4 Leveraging ELSST

As an attempt to categorize the resources in a harmonized fashion, we will investigate the option to adopt ELSST as a default terms vocabulary for resource indexing and search purpose.

Language barriers are major obstacles to efficient resource location and utilization across the European Research Area. This is especially so for comparative research that normally requires data and resources from more than one language community. Apart from a handful of significant comparative data collections that are available in several languages, the majority of sources describing European societies are only documented in one language (typically the language of the country from which the data derives). Translation into one or more additional European languages has in most cases not been carried out, due to the costs involved.

However, the language challenge can be attacked by other means than large-scale translations. In the practical implementation of the DDI metadata standard in a multi-language Europe, the thesaurus ELSST stands out as the single most important component of the semantic and content-carrying kind mentioned above. This thesaurus was originally based on the UK Data Archive maintained HASSET-thesaurus, the multi-language idea was developed within the EU-financed LIMBER project (Language Independent Metadata Browsing of European Resources) and has been further enhanced through additional funding from the ESRC and the University of Essex and a subsequent EU grant (MADIERA).

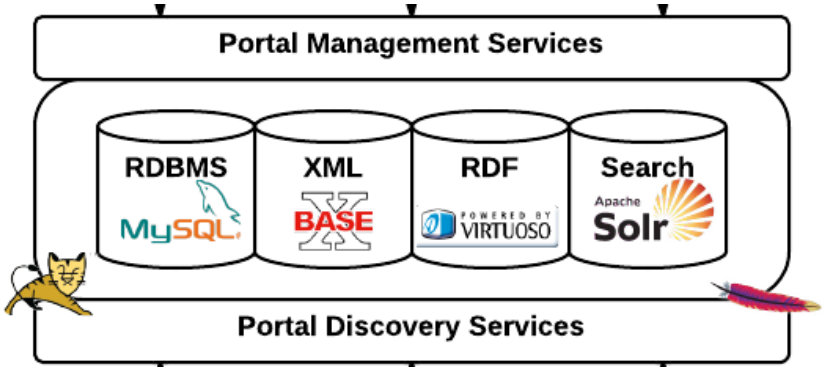
Such a thesaurus is a hierarchically arranged controlled vocabulary, which may be used for indexing and retrieval purposes in the field of information science. If comparative data resources can be efficiently identified across language barriers, the first hurdle is already passed. This can be achieved by the use of language-independent classifications of resources as well as language-independent and thesaurus-supported application of keywords and terms to the relevant parts of the metadata records. If this is done properly a portal user would be able to specify his/her search criteria in any of the supported languages and get a list of hits independent of what language they are described in. The keywords assigned to the metadata from a multilingual thesaurus can be instantly translated back into the supported language of the user.

ELSST provides data users with the means to search across multiple, multi-lingual, multi-cultural repositories and archives to find data relating to a single concept. ELSST is pivotal in the present CESSDA portal, and the plan is that it will continue to be so. It may also be used in data documentation tools to standardize the use and wording of concepts and keywords. The thesaurus is presently being updated and expanded and a version control model has been designed to permit annual releases. This will enable concepts and their relationships and metadata from previous versions to be identifiable.

An ELSST SKOS output is planned, and a concept versioning model has also been designed which will inform the SKOS URI name spaces.

2. METADATA STORAGE

The harvesting, management, and discovery features of the portal will be implemented and orchestrated around a wide variety of metadata that can be serialized and stored in different shapes and forms. Our platform will take advantage of state of the art storage engines that have emerged in the past decade and made possible by astonishing progress in information technology. We have in general moved away from the traditional relational database approach and instead base our solution on XML, semantic, or NOSQL technologies.



One important aspect of this approach is that most of these database engines rely on external schema and models to enforce the information structural and referential integrity. For example, XML documents are validated by a XSD schema or meaning and semantics of RDF triples expressed through ontologies. This results in a very flexible and scalable storage framework that can meet today's demand and adjust to future needs.

The remaining parts of this section describe the specific technologies that will be used for the portal project.

2.1 File Storage

While large metadata collections are best managed and processed in specialized database engines, it is very common for such information to be stored or exchanged as electronic files (that can be compressed for effective transfer). This will often be the case during the portal harvesting, processing, and registration processes. We therefore anticipate and plan for thousands of files to be living behind the scenes of the portal central databases.

As all interactions with these resources will be taking places through automated or semi-automated processes, the risk of human error is minimized and we will simply rely on traditional file systems for storing files. The management processes will read/write these resources and import/export them into databases as relevant.

For harvesting or other situations where maintaining different versions of file resources is important, we will leverage source code version/revision control technology to capture and manage changes over time. This innovative approach can be particularly powerful, for example to report what has changed in a DDI document between two harvests or point in time. For the prototype, we will be implementing this using Git.

2.2 XML Database

While the discovery metadata will ultimately be serialized and managed into an RDF format, most if not all of the portal incoming metadata will be in some XML format. These will initially exist in the file system but need to find their way into an XML storage for bulk processing or fast retrieval. We therefore need a database engine capable of efficiently storing and querying this information.

For the portal, we elected to use the open source BaseX package. This native XML database engine initially developed at the University of Konstanz has come to maturity in the past few years and offers a wide range of features, including the ability to run as a server, on the desktop, or be embedded in Java applications, support for XQuery 3.0, and accessibility over REST. While other open source options are available, in particular the popular eXist package, our experience shows that BaseX, besides offering a broad set of functionality, performs and scales well, even on large database collections (in the hundreds of Mb or Gb range). Note that, as an alternate option, we will evaluate the native XML capabilities of the OpenLink Virtuoso platform used for managing RDF (see below), as this could provide an all in one solution.

Pushing XML files into native XML database is a simple task that does not require the design of structures or schema ahead of time. The engine automatically indexes all elements as needed. Likewise retrieval and querying is performed using the XQuery language. This therefore overall significantly minimizes development efforts.

2.3 RDF Database

As the discovery metadata will be based on RDF technology, we need a high performance storage engine that can support the prototype and scale up in production environment. Initial estimates indicate that the portal database will likely store hundreds of millions of RDF triples (primarily due to variable level metadata, in particular codes/categories). While such numbers would have been mind boggling a few years ago, both open source and commercial technologies are now available to handle such a collection.

One such solution often cited for its record breaking achievements is OpenLink Virtuoso. Available as both open source and commercial versions, Virtuoso is a hybrid platform designed to meet various database needs (relational, XML, RDF) and deliver web application and services. It caters particularly well for the RDF community as a high performance quad-store engine. The recently released version 7 introduced a column store back-end and improved compression which further improved performance. This led to a demonstrated use of the server on a 150 billion triple collection in the context of the LOD2 EU funded project.

It is to be noted that the open source version of Virtuoso is only available on single server instances. A clustered environment would require the acquisition of a commercial license. We however do not anticipate such need in the prototype implementation or even production mode as the metadata could be distributed across multiple single server instances if necessary as the primary search functionality are not the responsibility of the RDF store.

While we will adopt Virtuoso as our RDF storage platform of choice, we will ensure loose coupling through the use of the Apache Jena library as abstraction between our applications and the triple store. This would enable us to switch to other Jena friendly database engine in the future should such need arise.

2.4 Search Engine Database

Apache Solr is one of the gold standards when it comes to search engines, and is one of the most popular, blazing fast open source enterprise search platform. Its major features include powerful full-text search, hit highlighting, faceted search, near real-time indexing, dynamic clustering, database integration, rich document (e.g., Word, PDF) handling, and geospatial search. Solr is highly reliable, scalable and fault tolerant, providing distributed indexing, replication and load-balanced querying, automated failover and recovery, centralized configuration and more. Solr powers the search and navigation features of many of the world's largest Internet sites.

For these many reasons, we have selected Solr as the core solution to deliver the portal search and discovery functionality. Upon ingestion, relevant elements from the DwB discovery RDF metadata will be submitted for indexing by the Solr engine. The portal user interface and search services will directly rely on the Solr REST API, which will not only deliver powerful capabilities but expose well documented search protocols that developers are familiar with.

2.5 Other Databases

We do not anticipate at this point additional data/metadata storage requirements. If the need arise for a more traditional relational database engine, we will have the option to leverage the SQL server available in Virtuoso, or alternatively use the open source MySql package. NOSQL packages would also be evaluated if applicable.