DELIVERABLE

Project Acronym: DM2E

Grant Agreement number: ICT-PSP-297274

Project Title: Digitised Manuscripts to Europeana

D3.1 - Initial Specification Report

Revision: 1.1

Authors:

Ewelina Rockenbauer (ONB)
Christian Morbidoni (Net7)
Dov Winer (EAJC)
Steffen Hennicke (UBER)
Klaus Thoden (MPIWG)
Jorge Urzúa (MPIWG)

Project co-funded by the European Commission within the ICT Policy Support Programme		
Dissemination Level		
PP	Restricted to other programme participants (including the Commission Services)	X

Revision history and statement of originality

Revision	Date	Author	Organisation	Description
0.1	02.07.2012	Ewelina Suchorzebska; Dov Winer	ONB; EAJC	Initial draft version
0.2	09.07.2012	Christian Morbidoni / Steffen Hennicke / Klaus Thoden	Net7; UBER; MPIWG	Some additions
Final draft	27.07.2012	Ewelina Rockenbauer	ONB	Some additions
Final 1.0	02.08.2012	Ewelina Rockenbauer	ONB	Some additions and final revision
Final 1.0	09.08.2012	Stefan Gradmann	UBER	Approval Final 1.0
Final 1.1	28.06.2013	Christian Morbidnoni / Violeta Trkulja	Net7 / UBER	Addition of Appendix B: The Pundit Ecosystem

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This deliverable contains original unpublished work except where clearly indicated otherwise.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

List of tables

List of Figures

1 Executive Summary

This is the first deliverable of Digitised Manuscripts to Europeana (DM2E) WP3 –“Initial Specification Report”. It presents the results of Task 3.1 the “Initial functional specifications of the prototyping platform” and parts of Task 3.2 “Building of the prototype platform”. Task 3.1 dealt with the requirements gathering process of functional and non-functional requirements for the development of the experimental content enrichment platform. This document is aimed at participating partners as a resource and guideline for carrying out further work in DM2E.

Task 3.1 has been carried out in two steps, differently from what has been previously agreed in the Description of Work (DoW). Initial functional requirements have been provided by Net7, following the discussion with partners during the project kick-off meeting, and feedback has been collected via email from other WP3 participants.

Then a first prototype has been developed and demonstrated to DM2E partners, during the project meeting and in dedicated virtual meetings, as well as to the Digital Humanities Advisory Board (DHAB). Having a demonstrative prototype facilitated the process of brainstorming and of collecting new requirements in terms of expected functionalities.

Additionally, two online questionnaires have been created and distributed to the online scholarly community. Unfortunately, not enough relevant results have been collected. Due to this, it has been agreed to distribute those questionnaires again in the second half of 2012 and to collect further requirements during relevant gatherings, e.g. the OKFN festival.

Such an “agile” approach has been chosen as considerably more compatible with the timing of the project (the final prototype is due in month 11). It should be noted that this report describes the current state of work in progress. It represents the requirements gathered at month 6 and therefore those requirements may be subject to changes as maybe not all of the requirements can be technically realised or further requirements will emerge during the course of the project.

2 Introduction

This document aims to give an overview of the requirements gathered in Task 3.1 “Initial functional specifications of the prototyping platform” for the future development of the experimental content enrichment and recombination platform KORBO.

Chapter 3 provides an overview of requirements gathered in parallel Digital Humanities projects that are seeking to establish scholarly work environments.

Chapter 4 presents the preliminary requirements which were prepared by Net7 for the first implementation of the prototype.

In chapter 5 the requirements are stated which were gathered during the meetings of the Digital Humanities Advisory Board meetings as well as during WP3 virtual meetings.

Chapter 6 describes the requirements for the next developing phase for Korbo and Pundit.

Finally, in chapter 7, requirements for WP1 and WP2 are given.

This deliverable contains the requirements gathered in the first 6 months of the project. Due to this, during the course of progressing work in DM2E further requirements might emerge.

3 Parallel Digital Humanities projects establishing scholarly work environments

3.1 Bamboo Technology Project (BTP)

The BTP is seeking to advance arts and humanities research through the development of shared technology services. The present phase has two main strands: Research Applications and a Shared Infrastructure.

Research Applications (3.1.2 below) focus on the development of Research Environments and Corpora Space Design processes. Shared Infrastructure (0 below) combines the development of Scholarly Web Services on Services Platform and Collections Interoperability.

Phase 1 ran from October 2010 to March 2012 and was preceded by a planning phase from Spring 2008 to Autumn 2010 in which 600 faculties from 115 institutions participated. This provided an extensive requirements definition for digital humanities that is available in the following documents:

The Final Report; Scholarly Practices Report; The Themes Database, housing quotes about scholarly needs and interests; Scholarly Narratives, describing how scholars perceive technology for their work; The Demonstrator Report, which summarises results from 10 prototype projects.

The research environments being developed are expected to provide easy-to-use, highly scalable environments for digital scholarship that will include core tools for content management, collaboration, and the connection to distributed collections and web services. These research environments can be used by humanities researchers to store and organise sets of digital content, e.g. text, images, video and audio; to create, maintain, and search rich metadata about this content; to annotate and analyse content; and to accomplish these through collaboration with other scholars.

Bamboo distinguishes between “content” and “corpora.” By “content” they mean text, images, video, audio, and associated metadata. By “corpora” they mean structured sets of these materials. Structured texts may themselves include associated images, video, and/or audio. The corpora may be stored in one location or made up of the aggregation of distributed materials across digital collections held in libraries, research centres, museums, and/or archives within and outside of universities.[1]

3.1.1 Requirements

In 2008 Bamboo carried out a round of workshops which provided a corpus of data for an ongoing analysis of scholarly practice. This has been summarised based on the concept of the Scholarly Primitives of John Unsworth and the OCLC “scholarly information activities”. The resulting project Bamboo Scholarly Practice[2] report includes the following summaris-

[1] https://wiki.projectbamboo.org/download/attachments/18382876/BTP-Final-Public.pdf (page 20) [Retrieved 28.06.2012].

[2] http://www.projectbamboo.org/wp-content/uploads/Project-Bamboo-Scholarly-Practices-Report.pdf (page 2-3) [Retrieved 28.06.2012].

ing table that guides the definition of requirements. The themes are operationalised through the Bamboo scholarship services.

Bamboo theme of scholarly practice	Unsworth primitive	OCLC scholarly information Activity
Gathering / Foraging	Discovery	Searching (direct searching, chaining, browsing, probing, accessing)
Synthesising / Filtering Comparing	Sampling	Collecting (gathering, organising)
Contextualising	Referring	Searching (chaining, browsing, probing) Collecting (organising) Cross-cutting (monitoring)
Conceptualising, Refining and Critiquing	Illustrating Representing Comparing	Reading (scanning, assessing, rereading) Cross-cutting (notetaking, translating) Writing (assembling) Collaborating (consulting)
Documenting methods	Representing	Writing (disseminating) Cross-cutting (translating)
Managing data	Discovering Referring Representing	Searching (accessing) Collecting (organising) Collaborating (coordinating, consulting)
Annotating / documenting	Annotating	Writing (assembling) Cross-cutting (notetaking)
Modelling / visualising	Illustrating Representing	Cross-cutting (translating) Writing (assembling)
Overlapping teaching and research	Representing	Collaborating (coordinating) Cross-cutting (translating)
Sharing / dissemination / publishing	Representing	Writing (disseminating)
Funding	Suggested parenthetically	No analogue
Collaborating	Common thread throughout scholarly primitives, not listed separately	Writing (co-authoring) Collaborating (coordinating, networking, consulting)
Citation, credit, peer-review	Referring	Reading (assessing) Writing (dissemination) Collaborating (consulting)

Table 1: Bamboo Requirements Table

3.1.2 Research application

3.1.2.1 Bamboo Work Spaces

The Work Spaces will include basic capabilities for collaboration, content, and scholarly analysis tailored for the humanities. In their early form, the Work Spaces will make it easy for individuals or self-defined groups of scholars to share and comment on digitised materials; to deploy useful gadgets such as calendars, maps, and RSS feeds; and to carry out various forms of mark-up and discussions related to different documents. The Work Spaces will also enable scholars to share and acquire information about tools and services that colleagues at other campuses are using and evaluating.[3]

In phase 1, Work Spaces will be able to ingest and store all content types as digital binaries. This capability – equivalent to storing a file on a hard disk drive, without regard to whether or how that file can be manipulated, analysed, or transformed – is an essential precursor to extended functionality that is useful to a scholar. Examples of extended capabilities include the ability to transform stored content from one format to another (e.g., Word documents to PDF, TIFF images to JPEG, or unstructured text to an indexed object-relational structure à la PhiloLogic); to generate concordances of textual materials; to generate histograms that represent tonal distribution in a digital image; or to collate multiple drafts or editions of a digitised text. In phase 1, Work Spaces will enable annotation, transformation, discussion, and sharing of documents whose principal content is text.

The following capabilities will be available in all Bamboo Work Spaces:

- store and organise sets of any digital content type, including text, images, video, and audio;
- record core metadata about the digitally represented object (e.g., author, title, pertinent copyright information, provenance);
- record technical metadata pertaining to the formats in which objects may be stored (e.g., standard image, video, and audio metadata);
- define custom metadata schemas appropriate to the collected materials and research methods, and record information structured according to schema definitions;
- annotate stored content (e.g., tags, unstructured notes);
- create and use elements that facilitate collaboration (e.g., calendars, RSS feeds, discussion threads, surveys);
- create and use elements that facilitate analysis and consideration (e.g., concordances, tag clouds, maps generated from geolocation metadata, ratings);
- call on remotely deployed web-services to operate over an object or objects (e.g., scholarly services deployed on the Bamboo Services Platform; in phase 1, Bamboo-hosted scholarly services will be applicable only to textual content);
- record information on methods, procedures, and tools used in scholarly research, and report to one or more central aggregation stores;

3.1.2.2 Corpora Space Design

The detailing of the requirements for Corpora Space will be carried out during Phase 1. Examples of ongoing scholarship and scholarly technologies that Bamboo will consult with including Nines, Perseus, ARTFL, TAPoR, and Oxford's JISC-supported VRE-SDM. They will also consider more recent projects, such as Berkeley Prosopography Services and its relationship to the Cuneiform Digital Library.Shared Infrastructure.

[3] Bamboo Technology Proposal (Public), July 2010 https://wiki.projectbamboo.org/download/Uattachments/18382876/BTP-Final-Public.pdf [Retrieved 2012.0628].

Scholarly services to be delivered in Phase 1 include the following:

Names	Description	Scholarly services team
Document Mapping	Map features in document to an object-relational indexing model	Chicago/ARTFL
Concordance	Generate a concordance for one or multiple texts, where each element of the concordance is associated with contextual material drawn from its occurrence(s) in the analysed text(s)	Chicago/ARTFL
Collocation/Cloud	Return word-counts of matches occurring in a text or texts to a specific query term or terms	Chicago/ARTFL
Frequency	Return word-occurrence frequency in a text or texts, grouped by metadata elements (e.g. author) and as a statistic normalised to a specific quantity of text (e.g. 10,000 words)	Chicago/ARTFL
Morphological Analysis	Generate a morphological analysis of an inflected word supplied by the service consumer	Tufts/Perseus
Syntactic Analysis	Generate a syntactic analysis of words in a sentence or set of sentences	Tufts/Perseus
Named Entity Identification	Generate semantic classification of named entities in a text or set of texts (i.e. associate a name in a text with a particular entity in the real world)	Tufts/Perseus
Proxied SEASR Analytics	Configure and initiate analytical services deployed alongside major content: repositories, such as HathiTrust or JSTOR, where co-location of computational power permits analysis of very large data sets or data whose intellectual property constraints prohibit a researcher's direct possession of content.	UIUC/SEASR

Table 2: Scholarly Services Phase 1

Scholarly Web services built and deployed in phase 1 will be derived from existing, proven projects in the humanities, including ARTFL, Perseus, Berkeley Prosopography Services, and SEASR.[4]

Participants in the Bamboo Planning workshops identified categories of scholarly need that map to **curatorial, analytic, semantic, and visualisation services**.

Curatorial services enable scholars to track and organise digital materials they wish to archive and preserve. Digital materials includes (digitised) primary sources, but also records of scholarly methods applied to materials in the course of an inquiry, records of a performance, records of relationships between objects that change over time, geo-spatial data, and visualisations. The constellation of activities involved in tracking and organising digital materials, as described by participants in Bamboo Planning workshops, is documented on the project wiki in Theme Groups "Consider" and "Preserve."[5]

Analytic services enable scholars to perform algorithmic inquiries against corpora of digitised materials. While the majority of analytic software applicable to humanist research enables discovery of pattern and structure in text, this category of technology support for the humanities also includes algorithmic analysis of image, video, and audio material. Scholarly activity in this area, as described by participants in Bamboo Planning workshops,

[4] ARTFL (http://artfl-project.uchicago.edu); Perseus (http://www.perseus.tufts.edu); Berkeley Prosopography Services (http://wikihub.berkeley.edu/x/2YYAAQ); SEASR (http://seasr.org) [Retrieved 30.07.2012].

[5] Theme Groups on the Bamboo Planning wiki: https://wiki.projectbamboo.org/display/BPUB/Theme+Groups [Retrieved 30.07.2012].

is documented in the "Consider" and "Discover" Theme Groups, and their constituent Themes, "Analyze," "Contextualize," and "Filter and Synthesize" on the project wiki.

Semantic services can be viewed as a specialised category of analytic services - enable scholars to algorithmically identify meaning in digital materials, and relationships between materials derived from algorithmically - or manually – associated meaning. Semantic services are categorised distinctly from analytic services because technology that derives. Meaning from digital objects is currently less sophisticated (and more provisional) than analytic technology used to discover structural patterns in digitised corpora.

Visualisation services enable scholars to model and present large or complex sets of information in graphical forms that reveal patterns at a strategic distance, without first requiring a viewer to digest and comprehend each element of the set. This category of scholarly activity, as described by participants in Bamboo Planning workshops, is documented in the "Consider" Theme Group, and its constituent Themes, "Model and Visualize" and "Analyze" on the project wiki.

3.1.3 Bamboo Service Platform

The Bamboo Services Platform (BSP) includes the technology stack on which software services will be deployed; as well as services that address general functionality that underlies more specialised Scholarly Web Services. The *Deployment Stack* includes language, framework, and libraries (e.g., service container, service implementation libraries, authentication framework, logging framework, message mediation). Bamboo will rely on OSGi standards - compliant service container as a cornerstone of the platform's service deployment infrastructure. OSGi permits services based on diverse software dependencies to be co-deployed with minimal effect on implementation code.

3.1.3.1 Collections Interoperability Services

Collections interoperability services deployed on the Bamboo Services Platform will deliver capabilities that are likely to fall in the following categories:

- Metadata interchange using metadata interoperability and complex content description standards, such as RDF, OAI-PMH, OAI-ORE, ATOM Publishing Protocol, OAC, etc.
- Facilities that exploit and support use of persistent resource URIs at appropriate levels of granularity and adhering to the principles and emerging best practices of the W3C Linked Data Initiative
- Proxied access to, aggregation of, or aggregated indexing of content stored in repositories that expose a CMIS-compliant web services interface
- Facilities to support updating and/or augmenting, in place or virtually, existing collections and/or collection metadata as transformations and new scholarly derivatives of content are generated
- Transformation between broadly used content formats, where need exists and opportunity permits, especially with respect to transformations that are prerequisite to consuming scholarly services deployed on the Bamboo Services Platform.

3.2 Research Space

ResearchSpace (RS)[6] aims to support collaborative Internet research, information sharing and web applications for the cultural heritage scholarly community. It is funded by the Mellon Foundation and one of the first datasets to be made available for annotation and other research activity will be the British Museum's collection data (currently 2 million objects).

The ResearchSpace environment intends to provide following integrated elements:

- Data and digital analysis tools
- Collaboration tools
- Semantic RDF data sources
- Data and digital management tools
- Internet design and authoring tools
- Web Publication

ResearchSpace will provide a range of flexible tools to support a wide range of workflows and will develop these tools on an ongoing basis. Semantic technology is at the core of the infrastructure because it provides an effective mechanism for research and collaboration across data provided by different organisations and projects. It aims to reduce the costs of developing and operating new and innovative systems, creating a more sustainable research and production environment. ResearchSpace is an enabling environment that will develop over time with the help of those that use it.

In a recent presentation Dominic Oldman[7], principal investigator of RS, surveys the challenges for digital scholarship and summarizes the RS aims. He stresses that Linked Data, data published using permanent and stable URIs, do not require any changes to underlying database schemas. By agreeing to use a particular publishing schema based on established semantic ontologies the partners are able to develop tools and applications that treat the different datasets as one. These stable applications can be embedded into any web site concerned, for example, with ancient world locations.

The ResearchSpace project seeks to create one such integrated environment. Collaboration applications like wikis overlap with harmonised data repositories and research tools. The social networking tools used so effectively by the Digital Classicists, the use of linked data to align different datasets as illustrated by the Pelagios project, and tools like Zotero will operate in the same integrated environment. It seeks to harmonise complex datasets that often cross subject areas. It wants historians to be able to make connections across different data so that a picture of history can be built up from multiple sources using places, events, themes and people. This is done through the CIDOC-CRM ontology which has detailed support for cultural heritage data. The relationships used in mapping data to the CIDOC CRM ontology are challenging but the results are potentially revolutionary by bringing related fragments of information together to generate new knowledge.

ResearchSpace seeks also to provide tools for non-technical scholars. Such tools will merge and blur the concepts of collaboration, data modelling and academic annotation and all employ an underlying linked data infrastructure linked to institutional data.

[6] http://www.researchspace.org [Retrieved 30.07.2012].

[7] Digital Collaboration, Dominic Oldman, Presentation at the Society for the Preservation of Natural History (SPNHC 2012), Yale University, Peabody Museum, New Haven Connecticut: presentation notes: http://www.researchspace.org/file-cabinet/narrative.pdf?attredirects=0&d=1 [Retrieved 30.07.2012].

3.2.1 Requirements

The latest updates on the specifications for ResearchSpace are available at http://www.researchspace.org/file-cabinet as they evolved from the original ResearchSpace Business Requirements & Specification from 2010 and the 2011 developments. These requirements and specifications details the following System Elements and Research Tools:

System Elements

- Collaborative Content Management System
- Social Networking Tools
- Document / Asset Management
- ResearchSpace and Images and other Digital Assets
- Research Tools
- Collaborative editing tools
- RDF Stores
- CMS Stores
- Mechanism for RDF and CMS data synchronization

Research Tools

- Semantic Search Tool
- Terminology Mapper
- Data Annotation Tool
- Image Annotation Tool
- Image Zoom (and annotation)
- Image Compare
- Relationship / Link Editor
- Version Comparison (Track Changes)
- Geographical and Timeline Mapping

For DM2E purposes the specifications concerning Annotations[8] are of particular relevance. In the following details concerning Image Annotation are provided.

The document ResearchSpace Image Annotation – Additional Detail for Functional Requirement[9] by Dominic Oldman (February 2012) details the following components of image annotation:

A main area in which images are displayed and can be annotated

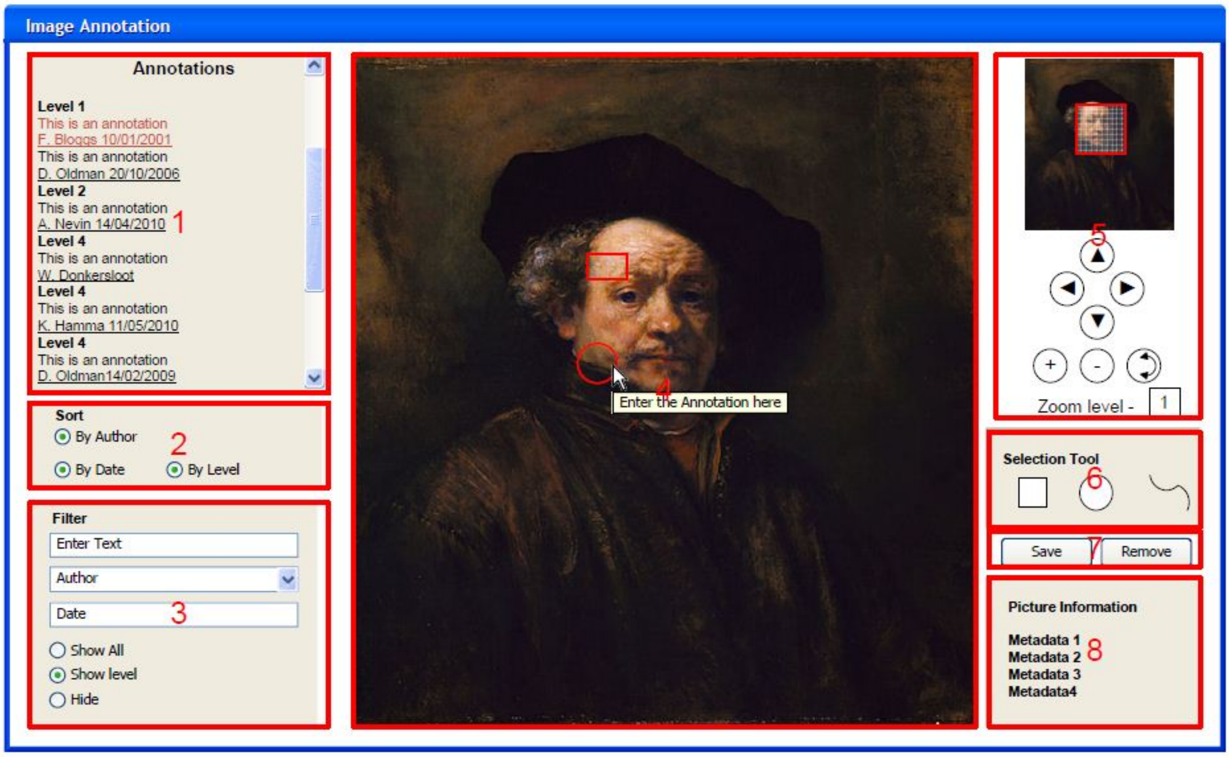
- The annotation text itself
- The ability to sort annotations
- The ability to filter annotations
- The ability to Zoom in and out of the images (particularly high resolution images)
- Tools for selecting regions and points for annotation

[8] Annotation Specifications: http://www.researchspace.org/file-cabinet/ResearchSpace-AnnotationSpec-020612-1355-106.pdf?attredirects=0&d=1 [Retrieved 30.07.2012].

[9] Image Annotation Specifications: http://www.researchspace.org/file-cabinet/ResearchSpace%20Image%20Annotation%200%206.pdf?attredirects=0&d=1 [Retrieved 30.07.2012].

- Information (metadata) about the image
- Related functionality is image overlay. Image overlay allows images to be scaled so that they can be placed on top of each other for precise overlay of different types of image of the same subject or other related images. For example, an xray of an object placed on top of a standard image of the same object. This would allow users to relate surface details to underlying detail which may be the subject of scholarly annotation.

These components are described at the Annex 1 of the ResearchSpace Business Requirements & Specification V2 document (page 60 and 61). Here we copy the details of item 12.6 Image Annotation / Zoom Image Annotation. See also 12.9 Image Overlay.



Level 1
This is an annotation
F. Bloggs 10/01/2001
This is an annotation
D. Oldman 20/10/2006
Level 2
This is an annotation
A. Nevin 14/04/2010
Level 4
This is an annotation
W. Donkersloot
Level 4
This is an annotation
K. Hamma 11/05/2010
Level 4
This is an annotation
D. Oldman14/02/2009

Area 1 – Annotation View

Show previous annotations with date of entry and author. Can be sorted and filtered.
Selecting an annotation should highlight the region on the image. Multiple annotations can be selected.
Selecting an annotation at a particular level of zoom will cause the image to zoom to that level and the area to which the annotation resides.
The user can view an annotation at different zoom levels.
It should be possible to show only project annotations.

Area 2 – Sort

The list of annotations in area 1 can be sorted by author date and level.

Area 3 – Filter and display options

Displayed annotations can be filtered by keyword, author, institution and date range. It should be possible to select multiple authors and institutions.
A calendar control should be provided for the start and end date range.

Filter should also include annotations at particular levels. Annotation regions should be proportionate depending upon the zoom level so that, for example, a region defined at zoom level 3 will be proportionately smaller at zoom level 1. When the image is zoomed only those annotations defined within the area should be displayed. It should be possible to hide all annotations. Colour coding of annotations to reflect different authors and institutions.
Area 4 – Image Work Area
The Image itself can be zoomed and moved using the tools (area 5). Annotations can be applied using the annotation tools (area 6). Users can define a region (rectangle, circle or free form) and write an annotation which then appears in the annotation view (area 1). Annotations should be saved or deleted using command buttons (area 7). Removing the actual annotation from the image means that the image can be viewed properly without large amounts of text impairing the view. Regions can also be hidden using the hide feature (area 3). It should be possible to enforce ontology terms for the type of annotation (Open Annotation Project).
Area 5 – Zoom Controls
To control the zoom level of the main image. A zoom control is required based on a streaming tiled method (e.g. Zoomify, IIP). Zoom in and out. Pan vertically and horizontally at any zoom level. Should display the current zoom level and the maximum zoom level.
Area 6 – Region Selection Tools
Rectangular, circle and custom region selection options. User should select one then drag region onto the image. Tool can be deselected by clicking the region option again.
Area 7 – Command Buttons
Save – To save the annotation. Delete – To delete a selected annotation (with authorisation).
Area 8 – Image Information
Display information about the image loaded into the tool (Name, Description, image size, format, owner.)

Table 3: Research Space Image Annotation / Zoom Image Annotation

3.3 Open Annotation Community Group[10]

Unlike previous attempts at annotation interoperability, the Open Annotation Community Group system does not prescribe a protocol for creating, managing and retrieving annotations. Instead it describes a web-centric method, promoting discovery and sharing of annotations without clients or servers having to agree on a particular set of operations on those annotations.

The purpose of the Open Annotation Community Group is to work towards a common, RDF-based, specification for annotating digital resources. The effort started by the reconciliation of two proposals that have emerged over the past two years: the Annotation Ontology[11] and the Open Annotation Model[12]. The editors of these proposals have collaborat-

[10] http://www.w3.org/community/openannotation [Retrieved 30.07.2012].
[11] http://code.google.com/p/annotation-ontology [Retrieved 30.07.2012].
[12] http://www.openannotation.org/spec/beta [Retrieved 30.07.2012].

ed closely to devise a common draft specification that addresses requirements and use cases that were identified in the course of their respective efforts. This draft has become available for public feedback and experimentation in May 2012. The final deliverable of the Open Annotation Community Group will be a specification, published under an appropriate open license that is informed by the existing proposals, the common draft specification, and the community feedback.

The role of annotations in Humanities scholarship was at the centre of the proposal for development of the Open Annotation Model and examples concerning Scholarly Editions and Medieval Manuscripts were included.[13] The following documents are the present draft versions of the recommendations issued on May 2012:

2012-05-04 Core Open Annotation Specification (Community Draft 1)

2012-05-04 Open Annotation Extension Specification (Community Draft 1)

3.3.1 Introduction[14]

Annotating, the act of creating associations between distinct pieces of information, is a pervasive activity online in many guises but lacks a structured approach. Web citizens make comments about online resources using either tools built in to the hosting web site, external web services, or the functionality of an annotation client. Comments about photos on Flickr, videos on YouTube, people's posts on Facebook, or mentions of resources on Twitter could all be considered as annotations associated with the resource being discussed. In addition, there is a plethora of closed and proprietary web-based "sticky note" systems, and stand-alone multimedia annotation systems. The primary complaint about all of these systems is that user created annotations cannot be shared or reused, due to a deliberate "lock-in" strategy within the environments where they were created, or at the very least the lack of a common approach to expressing the annotations.

The Open Annotation data model provides an extensible, interoperable framework for expressing annotations such that they can easily be shared between platforms, with sufficient richness of expression to satisfy complex requirements while remaining simple enough to also allow for the most common use cases, such as attaching a piece of text to a single web resource.

An Annotation is considered to be a set of connected resources, including a body and target, and conveys that the body is somehow about the target. This perspective results in a basic model with three parts, depicted below. The full model supports additional functionality, enabling semantic tagging, embedding content, selecting segments of resources, choosing the appropriate representation of a resource and providing styling hints for consuming clients. Annotations created by or intended for machines are also considered to be in scope, ensuring that the Data Web is not ignored in favour of only considering the human-oriented Document Web.

[13] The Open Annotation Collaboration Phase III: Demonstration and Refinement, Proposal to the Mellon Foundation, October 2010 http://www.openannotation.org/documents/Phase2NarrativeShortForm.pdf [Retrieved 30.07.2012].

[14] http://www.openannotation.org/spec/core/#Introduction [Retrieved 30.07.2012].

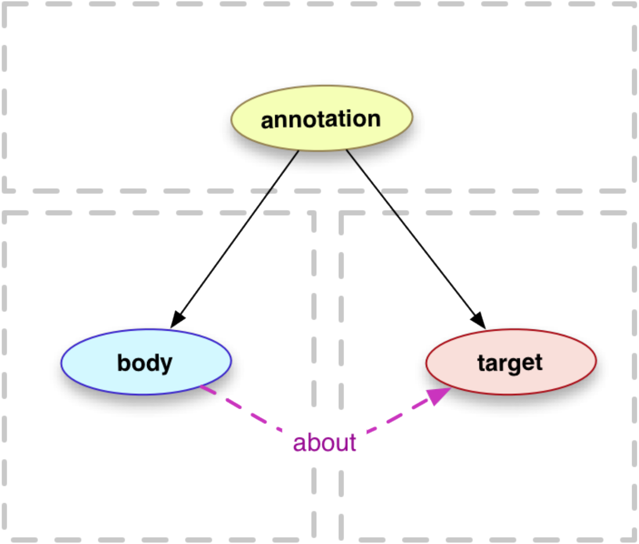


Figure 1: Annotation, Body, and Target

3.3.2 Open Annotation Guiding Principles[15]

1. The effort focuses on **interoperability** for annotations. Its goal is to allow the sharing of annotations across clients, servers, and applications. It will not, in any way, prescribe user interfaces, internal architectures or internal data structures.

2. The interoperability framework focuses on maximising the benefit of annotations with unrestricted access, hence **Open Annotations**. However, it does not preclude restricting access to annotations or their constituent resources. It does not define any authentication or authorisation mechanisms, but does not preclude the use of existing or new techniques.

3. The interoperability framework promotes the use of **existing publish/subscribe techniques** for discovering annotations. It does not specify a client-server protocol, yet does not preclude further specifications from introducing a protocol that builds upon the framework.

4. The interoperability framework is expressed in terms of the **Architecture of the World Wide Web**, and the best practices from the Linked Data effort.

5. The interoperability framework regards an Annotation as a serialisation of a **Graph**. Best practices are recommended for serialisation formats of RDF Graphs.

6. Typically, an Annotation expresses a relationship between one Body, and one or more **Targets**, where the **Body** is somehow "about" the Target(s). However, certain types of Annotations may lack an explicitly specified Body, such as bookmarks or highlights.

7. The Annotation, Body and Targets are individual resources, identified with URIs, which may have **distinct metadata** including especially provenance information such as authorship and date of creation or modification.

[15] http://www.w3.org/community/openannotation/open-annotation-guiding-principles [Retrieved 30.07.2012].

8. The representations obtained by dereferencing the Body or Target URIs may be of **any media type**. A Body may be a video about a Target which is an image.

9. The Body or any Target may be part of a resource, such as an Annotation where the Target is a section of an image or a time range in audio media. The interoperability framework includes solutions for **handling resource segments**, leveraging existing mechanisms where possible.

10. The correct interpretation of an Annotation may be **conditional on additional information** beyond the URIs of the Body and Target(s). Many Annotations have Body or Target resources that change over time, or have different representations available via content negotiation, and only a particular representation is intended by the Annotation. The interoperability framework includes solutions for describing this additional information, leveraging existing mechanisms where possible.

4 Preliminary requirements and draft implementation plan

4.1 Concepts

A list of concepts and definitions has been collected and used as a basic terminology for the prototype.

Basket

Is a (personal) space where users collect items they are interested in.

Item

An entity imported into a Basket. An Item is the representation of an external entity (coming from a LOD dataset, from Europeana, or from a Digital Library). An Item is a "live" copy of the original entity and can be augmented via annotation.

Entity

A Cultural Object, a Person, Place or other types of "things" that might be semantically connected to an Item. For example, the author of a cultural object is an entity and has their own metadata attached as well as possible semantic relations with other items and entities.

Augmentation

Refers to actions performed by the user with the help of appropriate tools to enrich their basket with structured annotations, relations among items and additional metadata.

Semantically structured annotation

An annotation is a piece of additional knowledge expressed by a user "about" an entity or a part of it. An annotation can express high level of semantics, as in the case of a textual comment, in forms (as the natural language) that are very hard to process for computers. A semantically structured annotation expresses often simple but useful semantics in a way that is easily digested by machines. Structured annotations are represented as typed relations among different kind of entities including web pages and data items.

Notebook

A notebook is a collection of annotations usually owned by a user for the purposes of grouping and sharing annotations about a given subject, topic or typology.

4.2 Simple driving use cases

Bob the "Basket curator"

Bob is a scholar and part of his activity consists of collecting both primary and secondary sources and evidences and argumentations about his research idea. For example, he would like to show the connections and paths among specific text passages across different transcriptions of Wittgenstein manuscripts and of other contemporary authors to show and discuss the evolution of a concept.

First, Bob needs a way of finding and collecting sources he wants to include in his argumentation. Then he needs a way of selecting and annotating the text that he collected.

Bob has an account on Korbo (e.g. uses his Google identity) and creates a new personal basket. From the platform web site Bob is able to perform a keyword based search to retrieve items contained in data.europeana.eu and possibly other relevant sources. A simple preview is shown for each result and Bob starts importing items in his basket.

For example Bob could import the item:

http://data.europeana.eu/item/91622/1BF8BC466E65367929379C83FC639F27961AC8,

which corresponds to the following object in Europeana:

http://www.europeana.eu/portal/record/92037/25F9104787668C4B5148BE8E5AB8DBEF5BE5FE03.html.

The basket content is then presented to the user as a browsable collection of items. Bob can:

- Use a faceted browser or other kind of filtering tools to explore the content
- Order items with respect to different criteria (date added, other metadata)
- Full-text search items and entities in the basket
- Go from an item to an other item or to a related entity based on semantic relations among them (e.g. go from Mona Lisa to Leonardo Da Vinci via "creator")

Augmenting by linking to LOD

While visualising an item Bob can create links to LOD entities. For example if the item has a text content, he can select a word in the text, click a "link to LOD" button and a set of LOD entities from dbpedia.org are suggested. Bob chooses one of them and a link is established.

Each time a link to an external LOD entity is found Bob has the possibility of importing additional data related to the LOD entity. For example, he could mark the name "Leonardo Da Vinci" in the text and link it to the entity http://dbpedia.org/page/Leonardo_da_Vinci and then import the relative data (e.g. Leonardo's place of birth, his main works, etc.). This additional data will improve Basket search and browsing.

Alice the DH developer

Alice is a web developer and works at a digital humanities project where scholars have collected a certain amount of structured data by curating baskets and using Pundit to create semantically structured annotations.

In particular the annotated knowledge represents different text passages tagged with one or more philosophical topic they relates to. Structured annotations capture possible influences among texts.

Alice uses the Korbo and Pundit REST APIs to access one or more baskets that curators have granted her access to. She now can query for all the items and show them in a list.

But she also can use the SPARQL endpoint to query for a graph where texts are connected based on their influence links; thus being able to use custom or open source tools to build specialised visualisations. For example Alice is fascinated by a recent development (Edge Maps, http://mariandoerk.de/edgemaps/demo citation: http://mariandoerk.de/edgemaps/ivj2012.pdf) and she decided to build a similar visualisation, e.g. by transforming specific aspects of the RDF data in a JavaScript friendly format as JSON.

Korbo API, however, provides an interesting feature: each relation shown in a visual graph is a structured piece of data and its context (author, date of creation, annotated original text, etc.) is known and retrievable.

Alice uses this feature to allow visitors of her web based visualisation to show what scholar has established a particular influence relation and, more interestingly, look at the specific passage of text that motivates such relation.

4.3 Architecture and information flow

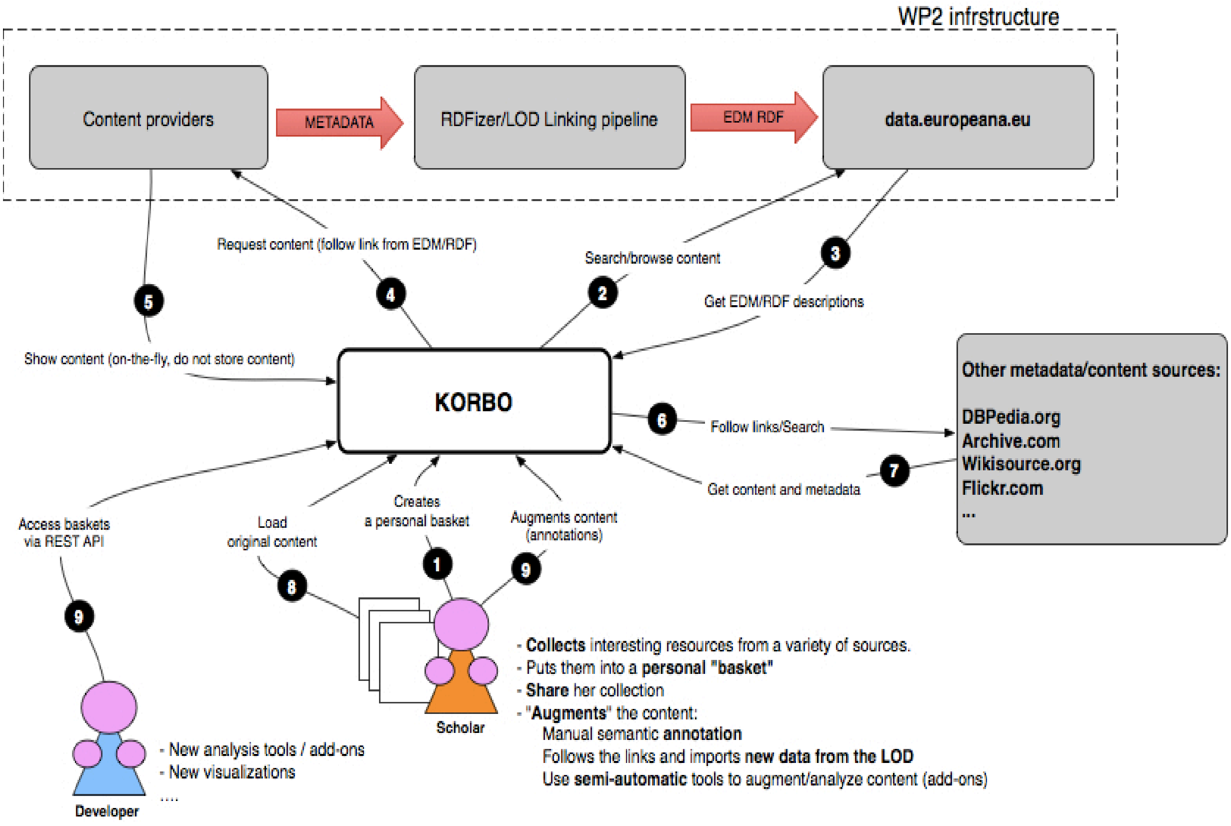

WP2 infrstructure
Content providers
METADATA
RDFizer/LOD Linking pipeline
EDM RDF
data.europeana.eu
Request content (follow link from EDM/RDF)
Search/browse content
3
Get EDM/RDF descriptions
5
4
2
Show content (on-the-fly, do not store content)
KORBO
Other metadata/content sources:
DBPedia.org
Archive.com
Wikisource.org
Flickr.com
...
6
Follow links/Search
7
Get content and metadata
Access baskets via REST API
Load original content
Creates a personal basket
Augments content (annotations)
1
9
8
9
- Collects interesting resources from a variety of sources.
- Puts them into a personal "basket"
- Share her collection
- "Augments" the content:
Manual semantic annotation
Follows the links and imports new data from the LOD
Use semi-automatic tools to augment/analyze content (add-ons)
Scholar
- New analysis tools / add-ons
- New visualizations
....
Developer

Figure 2: Architecture and information flow[16]

As shown in the illustration, Korbo interacts with the EDM stores (that later in the project will contain data from DM2E content partners) and with possible other data sources.

Korbo aggregates metadata from all the sources and store them locally along with pointers (URL) to the actual digital content (e.g. images, files). Such URLs are used to retrieve digital files on the fly from their original location and mix them with metadata and RDF to produce visualisations.

Scholars access the prototype by a simple dashboard where they manage and populate their baskets, while developers have a set of APIs both for writing and for reading basket contents.

[16] The first goal is to enable aggregation of items into baskets and basic augmentation based on LOD links.

4.4 Draft requirements

4.4.1 Functional requirements

Discovering items

It is important that users have appropriate means of discovering items of interest in order to collect them in baskets. While Korbo is not a search engine and the general idea is to expose APIs for external search engines to build on top, the prototype has to provide a minimal search capability.

Users should be able to search primarily in the Europeana LOD portal but also in Muruca Digital Libraries (that already provide RDF data) and other Linked Data compliant datasets (e.g. DBpedia).

Basic GUI to manage baskets and items

The prototype will include management of users and baskets. Each user has to be able to create a basket and populate it with items. Users have to login to be identified. Basket visibility is by default public. The GUI will be a simple backend-like application with the purpose of demonstrating basic functionalities to allow for user driven tests.

Semantically structure annotations of content

Annotating is the act of expressing knowledge about a "resource". As most of the resources of interest (documents, images and any kind of content) are on the web, and this is becoming true even in a domain like the Humanities, where scholars are progressively moving from an analogue to a digital world, being able to annotate web resources and share annotations with others is becoming of primary importance.

The main idea we are exploring is that of enabling users not only to comment, bookmark or tag web pages, but also to create semantically structured data while annotating, thus enriching the so called Web of Data. The ability to express semantically typed relations among resources, relying on ontologies and specific vocabularies, not only enables users to express unambiguous and precise semantics, but also, more interestingly, fosters the reuse of such collaboratively created knowledge within other web applications. For example: provide a powerful semantic search, build innovative ad-hoc data visualisations or ultimately improve the way users explore the web.

Figure 3 (see page 17) might give a better idea of what is meant by semantically structured annotations: the ability for users to create knowledge graphs where web content fragments, concepts and entities are meaningfully connected.

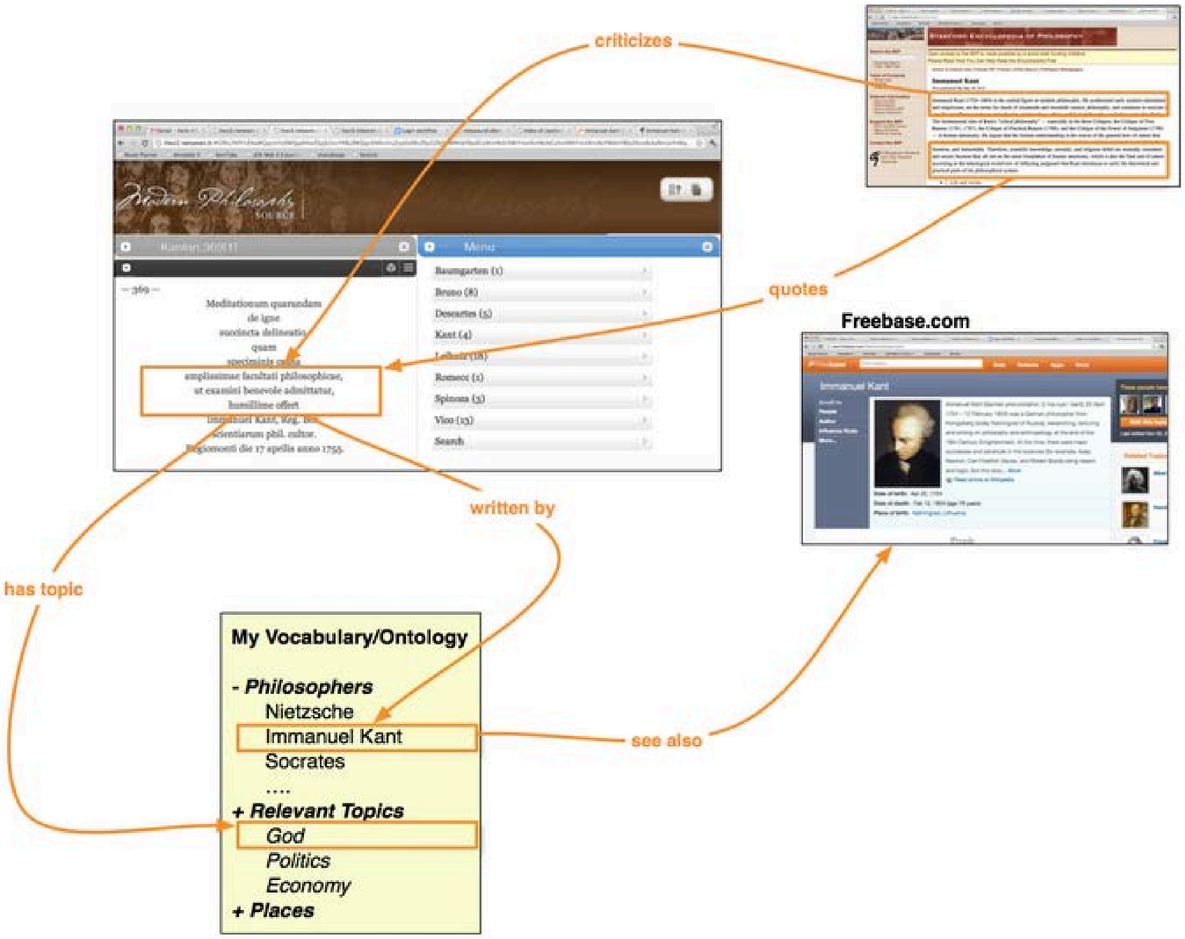


Figure 3: Semantically structured annotation

4.4.2 Non-functional requirements

REST API

The Basket and its data are accessible by a REST API.

The first API of the prototype will be a SPARQL endpoint for each single Basket and will be open (no access control).

EDM data support

- Search data
- Import data

Search in SPARQL endpoints and other APIs

In the case of data.europeana.eu the search has to be performed over a SPARQL endpoint, this has to be made with appropriate queries that restrict the results to given classes of objects.

In the case of Muruca Digital libraries, a web service will be implemented to return results from the built-in Solr index in a normalised format.

Import items

Items imported have RDF representation but might differ in schema and vocabularies used. A set of search drivers has to be implemented that take care of peculiar characteristics of data. For example, the EDM data model splits useful metadata across different LOD resources (Aggregations, Proxies, etc.) and a specific procedure (e.g. multiple calls to LOD interfaces) has to be put in place.

Application level integration with Pundit

Annotation will be based on Pundit that is under development in the SemLib project.

Pundit has to be customised to allow interaction with a basket. This will imply the implementation of on the fly configuration and an API in Pundit to ensure Korbo can instantiate Pundit widgets with appropriate parameters (e.g. the basket ID).

Native RDF data store

As experiments with data are expected to need graph query capabilities, it is considered that using native RDF and triplestore technology to store data is a good choice. It allows having “configurable” SPARQL endpoints without effort.

A SPARQL endpoint has to be provided for each basket by query rewriting to restrict to specific named graphs.

Demonstrative integration with LOD data exploration tools

To demonstrate the possibility of browsing Korbo Linked Data two tools will be integrated:

- Lod Live (http://lodlive.it), to provide a visual proximity graph view on items
- Elda (elda.googlecode.com), to serve different formats for the same data, including HTML with basic browsing features.

4.5 DM2E background tools

As specified in the DoW part of the WP3 activity is to investigate the possibility of integrating existing tools, starting from those one owned by DM2E partners in the following functional areas:

- annotation
- image visualisation
- text mining
- text collation
- content-enriching
- virtual collections
- specialized visualisations

The following list briefly presents the main tools of interest owned by DM2E partners.

Net7

Pundit

Pundit is a novel semantic annotation and augmentation tool. It enables users to create structured data while annotating web pages.

Annotations span from simple comments to semantic links to web of data entities (as Freebase.com and Dbpedia.org), to fine granular cross-references and citations. Pundit can be configured to include custom controlled vocabularies. In other words, annotations can refer to precise entities and concepts as well as express precise relations among entities and contents. Read more on semantically structured annotations.

Pundit is designed to enable groups of users to share their annotations and collaboratively create structured knowledge.

Pundit has been developed by Semedia at Universita' Politecnica delle Marche within the SemLib EU project.

Net7, that leads the SeLmib project, is further developing the tool to meet the DM2E requirements.

At the moment the project is in its first almost stable release, having its key functionalities up and running. Development is now focused on improving the UI and the collaborative experience.

Link: http://thepund.it

BoxView

BoxView is a suite of Open Source JavaScript libraries designed to handle the simultaneous visualisation of multiple documents and multimedia objects.

BoxView splits the page in multiple “boxes” where each of them is a container in which any type of content can be visualised (a menu, a fragment of text, an entire html page, a video player, an image viewer, and so on).

BoxView handles the boxes’ interactions, including automatically resizing them to fit the container, ordering, letting the user drag them around or collapse some of them.

BoxView has been designed with some sensible defaults in mind (e.g. all boxes will be resized to the same width per default) which one can use to quickly set up a working app, but there are a lot of options and configurations you can tweak to suit your needs.

Each BoxView’s plug-in can be individually configured and has its own CSS for styling. However, if one chooses to use them together, one can use BoxView’s “unique point of configuration” and use themes, a mix of options and CSS stylesheets, to style the whole suite at once.

Link: http://www.muruca.org/boxview

Erato

The Erato project is an attempt to create a simple web API and GUI to remotely edit stored RDF data. Erato uses the standard SPARQL protocol to read the data from a remote RDF store, and SPARQL update protocol to edit the data. As an alternative update protocol, Erato also supports Sesame XML transaction documents (e.g. if the user has an old sesame installed).

Link: http://erato.netseven.it

MPIWG

Language Technology Services

As part of the text mining tools for DM2E, the MPIWG can provide its online language services that morphologically analyse text sent to it. Currently, the supported languages are Arabic, German, Ancient Greek, English, French, Italian, Latin, Dutch, and Chinese.

The service is able to tokenise text sent to it in the above languages and returns a link to online dictionaries for each word. This is done via a morphological analysis of each word form, which results in querying a database for the correct lemma of each token.

Along with it goes a normalisation service, which is able to resolve older spelling features (e. g. 'long s', u/v-disambiguation).

This tool could be paired with Pundit to provide automatic analysis of annotated texts: the integration between the two should happen via HTTP API calls.

Digital Image Library DIGILIB

DIGILIB is an image-viewing environment for the Internet developed by MPIWG and Bern University that can zoom in/out, scale, rotate images, and can modify different properties of them such as contrast, brightness, RGB values, and others.

DIGILIB is a state-less web-based client-server application, where the image content is processed on the fly by a Java Servlet (called Scaler).

The browser's users send an HTTP request specifying parameters such as scaling factor, path of the image inside the server and others; and the mentioned Servlet returns only the portion of the image specified in the HTTP request as an HTTP response.

Currently, there are several fronted implementations for DIGILIB. The most stable frontend comes with the default DIGILIB's distribution and can be downloaded from http://digilib.berlios.de. Another frontend is ZOGILIB and can be downloaded from http://itgroup.mpiwg-berlin.mpg.de/cvs-web/cvsweb.cgi/zogiLib.

The architecture of DIGILIB allows many clients or frontends to access a common backend, even a client can run on remote machines. However, the images must be stored (or mounted) in the same machine where the Servlet runs.

Arboreal

Concerning text collation, the MPIWG can offer the (as of now) stand-alone application Arboreal. This is a content-based XML browser developed by the Archimedes Project. It facilitates the access to XML texts, their annotations, the linking of images and the working with parallel versions of texts.

It also offers morphological functions that support languages like: Latin, Greek, Arabic, Chinese, major western European languages, and languages written in cuneiform.

A binary distribution and the source code of Arboreal can be downloaded from https://it-dev.mpiwg-berlin.mpg.de/tracs/Arboreal/wiki/Download.

Virtual Spaces

Virtual Spaces is a tool for the structured representation of knowledge. Texts, images, and hyperlinks can easily be organised in 2D graphs. These graphs are in turn used to generate a set of HTML files which constitute a virtual tour. Some virtual tours can be found on the Examples page of above site. Additionally, these virtual tours can be exported to PDF and RTF files.

5 Requirements of the Digital Humanities Advisory Board

In the first six project months two meetings of the Digital Humanities Advisory Board (DHAB) were organised at Humboldt University. One took place on the 2nd of March, the second one on the 15th of June. These meetings dealt among other things with initial functional requirements for the prototyping platform.

It was repeatedly stressed that it is important to go well beyond mere annotation functionality in the development of KORBO and Pundit. The following high level requirements and features proposals were collected during both meetings and the WP3 virtual meetings. Due to the limited resources available in DM2E not all the features will be implemented in the prototype, but nevertheless they are important to understand development directions within and outside the DM2E project.

- **Referencing and publishing the basket**: The basket needs to be autonomous and should be reusable in other contexts. For example, in terms of publishing, the basket and the research results based on the work done on the content of the basket with Pundit should be referenceable. In other words, there needs to be an option to publish results of work done in Korbo and Pundit.
- **Data update settings**: The data in a Korbo basket needs to be protectable from updates.
- **Advanced graph analysis** of the contents of Korbo should be possible. Currently, the contents of Korbo can be visualised as a graph. However, the distance or proximity between nodes in the graph is not displayed. Such graph analysis should probably be delegated to external tools and enabled via the Korbo API.
- **Annotating outcomes of queries** (e.g. XPath on TEI-XML to extract certain pattern groups) would be an interesting feature to have. The results of such queries in the form of some kind of triple package would receive an URI and thus be referenceable. One question is if and how scholars work on XML structures and markup text.
- **Privacy settings:** An important issue is the degree of openness/privacy of the data in Korbo and in Pundit. Who is allowed to see my data? Professional scholars tend to define their research problem, track the evolution of their evolving conceptual knowledge representations, interpretations, reasoning and encoding of meaning over a significant period of time and tend not to share their annotations in an open forum. This fact results in the requirement of having privacy settings and to control who can see annotations, add annotations, and comment on annotations. Sophisticated private/public settings need to be implemented.
- **Access control and user groups**: The author of an annotation needs to be visible and recognisable. Generally, authorship is extremely important, as is an option to set up and specify user groups in order to enable collaboration.
- **Tracking versions of objects:** Regarding annotations it is extremely important to be able to track to which object an annotation has been attached to, independent of which specific version it originally had been attached to [Stability of reference].
- **Discourse visualisation**: The possibility of visualising the structures of arguments or discourse (as they appear in the form of annotations) is an important feature.
- **Pattern discovery and visualisation on big data**: Exploit the machines capacity to draw inferences, to “remember things” (for the scholar who has to rely on paper sheets otherwise) and to discover and to visualise patterns (like co-occurrences) on annotated corpora and, generally, on big data. Those are things scholars cannot do

easily “offline” and certainly not in the traditional, analogue world and this might be the key value.

- **Exploit capacity to deal with big data:** Generally, making things faster and more capable regarding big data is crucial. Being able to process large amounts of data is one of the key advantages of computer systems.
- **Enable collaboration**: In addition, working collaboratively must be possible as well.
- **Enable a “definable or customisable linking policy”** let the scholars choose where to look for concepts for annotations (e.g., first in VIAF, then in FOAF etc.). That is something they cannot do otherwise. Also, let scholars share vocabularies (of terms used to annotate and contextualise).
- **Granularity of Annotations**: The level of granularity of source data and the allowed annotations is important. Annotations should be doable on the transcription texts and on the images and on certain sections of a text and an image.
- **Sophisticated publishing options**: Publishing in terms of creating a package or a bundle of what a user did is crucial. This entails to stabilise the results somehow. The user needs to be able to freeze results so they can be referenced, cited, and credited. Stefan Gradmann mentions Jan Velterop’s concept of ‘Nano-Publications’ (cf. here: http://nanopub.org/wordpress. Also, see Sally Chambers thoughts on the topic here: http://www.slideshare.net/schambers3/nanopublications-in-the-arts-and-humanities). The possibility to publish is a crucial incentive for using the tools.
- **Creating scholarly editions of a manuscript**: Some advisory board members envision the Pundit tool as an overlay over the existing structure that makes it possible to re-map the annotations to the document that is in Pundit back to the RDF graph we create and even further back to the original document. This way, users of the tool can create scholarly editions of a manuscript, a "package" that incorporates the annotations made in Pundit, the original manuscript contents and the graph and its enrichments made within WP2. It is then possible to create PDF versions (using XSLT and XSL-FO e.g.) or HTML versions etc.
- **Grouping and filtering options**: Another requirement regarding Korbo would be the possibility of automatically grouping similar baskets or showing/adopting fitting annotations.
- **Enabling “journeys of discovery”**: Users should be able to browse contents / baskets of other users as well. That leads to an easier access to the data and enables a new form of experiencing. New ways of interlinking content could be discovered.
- **Korbo in a “teaching environment”**, i.e. for students collaboratively working on specific documents.
- **Annotations made on different representations of the same conceptual item** (e.g. different transcription, different languages, facsimile and transcription) should be possible. One crucial requirement is that such relations among resources (e.g. isDifferentVersionOf) are present in the EDM data and encoded with a well defined ontology.
- **Allow the creation of basket items from local files** e.g. a paper or an image on the scholar’s computer.
- **Add items while browsing** the web of data, e.g. with lodlive: When the user comes to an interesting resource it should be possible to import it into the basket.
- **Existing annotations** and markup should also be importable to Korbo and Pundit: TEI or other entity markup already present in the original text should be supported by Pundit.

6 Requirements for the next developing phase

6.1 Pundit

The following are the functional requirements collected from different input sources, as the DHAB, the WP3 meetings and Net7 internal user surveys that are considered as inputs to the Pundit development.

6.1.1 Non-functional requirements

Open Annotation compliance

The RDF data model used by Pundit is not yet fully compliant with the recent stable Open Annotation specification (http://www.openannotation.org/spec/core). Matching this requirement is important to enable third party clients to be able to store annotations in the Pundit server.

6.1.2 Functional requirements

Reply annotations

Users should be able to add an annotation as a reply to another annotation.

Edit annotations

Users should be able at least to:

- Edit the free text comment of an existing annotation
- Add/remove triples included in an existing annotation

Image fragments annotation

Users should be able to select a specific region of an image in a web page by drawing a shape on top of it. Such an image fragment should be handled the same way a text fragment is currently handled: allowing users to comment it or to compose triples with the fragment used as subject or object.

This enables also to establish a relation between a fragment of an image and a fragment of text (e.g. linking a transcription excerpt to its corresponding text in the manuscript).

Built-in support for additional entity sources

Additional entity sources (other than generic ones already supported, e.g. Freebase and DBpedia) should be available in the Pundit client. The DM2E consortium needs to agree on a number of relevant entity stores to be supported. This is important regarding the case where scholars establish links to existing, well known works (e.g. books, etc.).

Basic sharing functionalities

Users should be able to set their notebook as private or public. Once one of their notebooks has been made public, users obtain a URL. Sending this URL to other users, via external systems, e.g. e-mail, social networks, recipients of the URL can click on it and possibly decide to set the notebook as active.

Basic annotation filtering

Users should be able to filter annotations shown by Pundit. They should do so by activating/deactivating public notebooks they find. When an annotation is shown contextually to a web page, users can decide to deactivate the public notebook from which it comes from.

Users can freely decide to set a view preference choosing from three options:

- See all public notebooks
- See active notebooks only
- See only own notebook

Semantic expansion

Once a user finds an interesting resource in Pundit (e.g. the target of an annotation: a person, a web page or other kind of resource) it should be possible to see (e.g. in the form of a proximity graph) all the relations that such resource has with other resources.

Note: Such a feature could be based on an external Linked Data visualisation tool such as LodLive (opportunely customised).

Language preferences

Users should be able to set a preferred language (e.g. for vocabularies, relations, etc.). That means that in the case of multilingual controlled vocabularies or ontology concepts and relations are shown in the preferred language.

6.2 Korbo

The following are the functional requirements collected from different input sources, as the DHAB, the WP3 meetings, and Net7 internal user surveys that are considered as inputs to Korbo development.

Augmentation API

A set of REST API has to be provided for external applications to augment items in a basket by injecting RDF data.

Digital content API

Digital content corresponding to items in baskets (e.g. a transcription or an image) should be accessible via a REST API. Such content will be fetched on the fly from content providers by querying the EDM and fetching content URLs. Such an API is important to allow third party tools to provide text and image analysis and visualisation functionalities.

Upload items from local desktops

Users should be able to add items (text, images) from their local desktops.

Basic baskets access control

An API should be provided for setting a basket private or public.

6.3 Other requirements

The following requirements will be taken into account at a lower priority and will be addressed possibly after the prototype version due in month 11 of the project.

Advanced on the fly configuration

Users should be able to configure the environment (at least vocabularies and entity stores) trough the Pundit GUI. In other words users should be able to:

- add and remove entity stores
- add and remove taxonomies
- add and remove relation lists
- dynamically set the LOD datasets where the semantic expansion is performed

Advanced baskets access control in Korbo

Users should be able to provide access to their baskets to a group of users.

Freeze current status (Korbo and Pundit)

Users should be able to "freeze" the state of a Basket along with annotations made by the user (in Pundit) at a given time. This frozen version should have a stable identifier and URL to be shared or cited.

Support for web content versioning

Annotations should specify at what precise version of content it has been attached to. In the case the web content evolves, it should be possible to understand the annotation has been attached to a previous version of the same content and, under certain conditions, to retrieve such previous version.

At RDF data model level this can be probably addressed supporting the Open Annotation “State” of a given annotation target.

Users Notebooks preference page

An integrated preference page where users can easily manage their basket would effectively allow scholars to use different notebooks and group annotations in meaningful ways.

Users should be able to:

- See all owned notebooks
- Create a new notebook and set basic metadata
- Set the current notebook, the one where annotations are being written
- Set active notebooks, those that the user is interested in (e.g. shared by friends or colleagues)

Visualisation options and graph analysis

Enable the analysis of the graph and display of this analysis. For example, a user should be able to get a kind of timeline of all replies (of a certain type like, agrees, disagrees etc.) to a certain annotation. This would be useful for discourse analysis. Initially, this might be supported by an SPARQL endpoint which allows easy query pattern construction.

7 Requirements for WP1 and WP2

7.1 Named-contents

Pundit, and in general web annotation tools, need a clear way to identify pieces of content to be annotated.

This allows hooking annotations to precisely defined pieces of content instead of to web pages, thus allowing annotation to persist if a web page changes it's URL or if the same piece of content is replicated in multiple web pages.

For this purpose we introduce the notion of "named-content", already supported by Pundit.

Content providers should provide HTML representations of their content including markup to identify named-content (annotable versions).

An HTML representation of an object can include a single named-content (in this case it represents ad single atomic piece of content, e.g. the transcription of a page) or multiple named-contents (e.g. marking up each single paragraph or picture). Deciding the granularity of named-content is up to each content provider.

IMPORTANT: Annotable versions should include only named-contents, that is text of images that are intended to be annotated, and do not include other HTML elements as navigation menus, JavaScript applications, institutional headers, etc. This facilitates Pundit in correctly rendering its JavaScript annotation GUI.

HTML representation including named-contents markup should be available at stable URLs. Such URLs have to be derived from the EDM representation of a CHO. This means there should be a triple in the EDM like the following:

```
@prefix rdf:  <http://purl.org/net7/korbo/vocab#> .
:aggregation-X korbo:hasAnnotableVersionAt <URL>

Markup
Named contents in Pundit are recognised and identified by the follow-
ing markup:

    <div class="pundit-content" about="http://example.org/21345">
        <img src="http://example.org/imgs/21345.jpg"/>
        <p>caption bla bla</p>
    </div>
```

The class attribute of the div element must be "pundit-content"; this informs Pundit that the content is intended to be annotated.

The about attribute content must be an URL and will be used by Pundit to unambiguously identify the content.

An HTTP GET call to such URL should return the HTML itself. In this case:

```
Request
GET http://example.org/21345

Response
<html>
    <body>
        <div class="pundit-content" about="http://example.com/21345">
            <img src="http://example.org/imgs/21345.jpg"/>
            <p>caption bla bla</p>
        </div>
    </body>
</html>
```

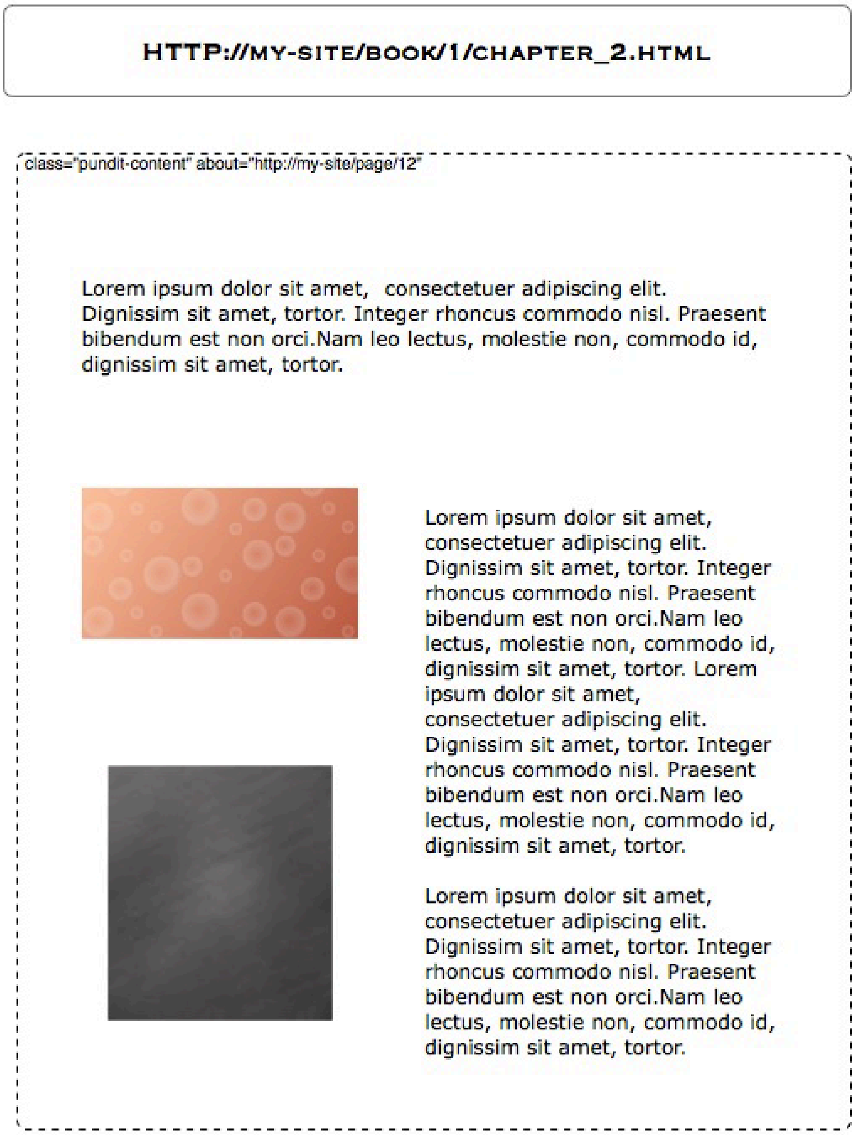
The "div" element may contain a "simple" HTML. The idea is that of including only content that is the representation of a single "providedCHO" (in Europeana), e.g. a Picture with a caption, a transcription, etc.

The guideline is to include only images and text. Flash objects and JavaScript should be avoided.

Read more on named content in the <a href="http://thepund.it/client.php">Pundit web site</a> under the section "Play nice with Pundit".

7.1.1 Graphic examples

The following figure illustrates the situation where an HTML page includes a single named-content. The named content includes pictures as well as text:



HTTP://MY-SITE/BOOK/1/CHAPTER_2.HTML

class="pundit-content" about="http://my-site/page/12"

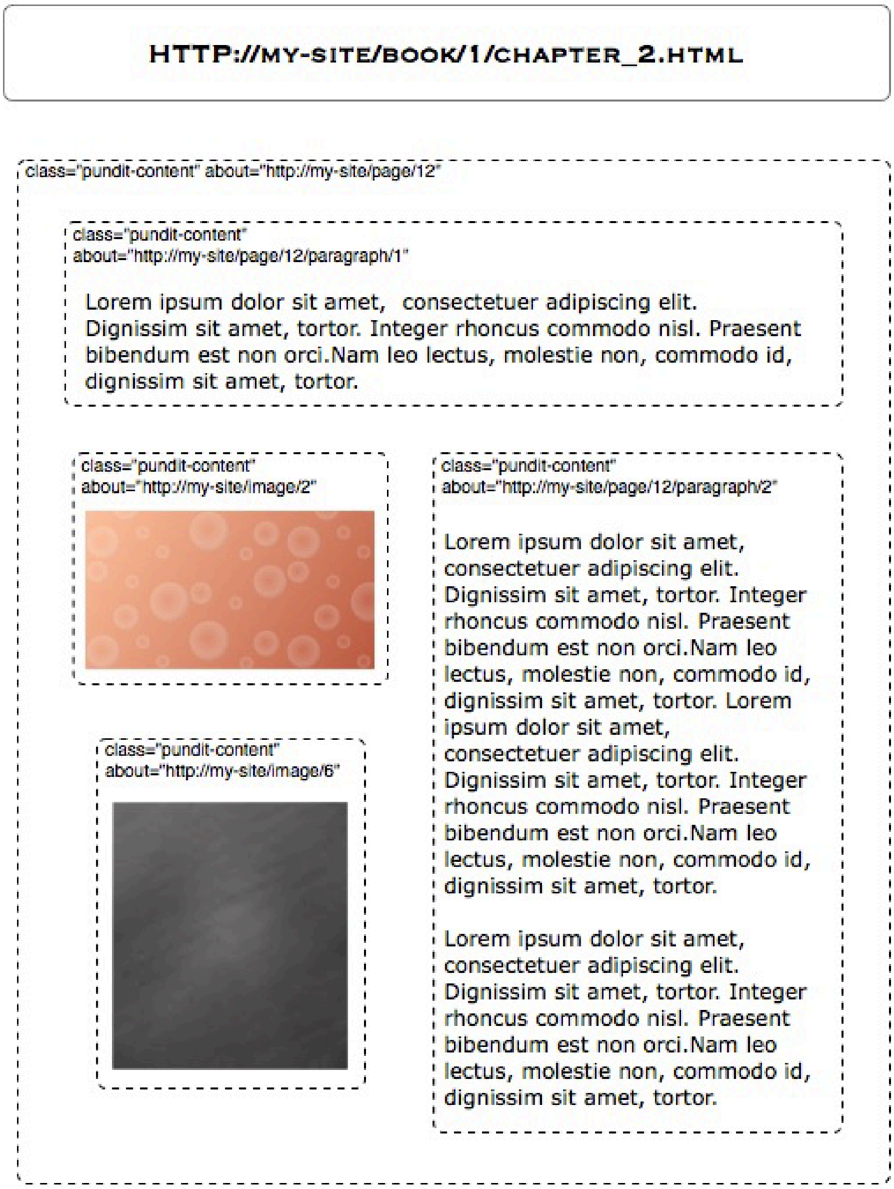
Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Dignissim sit amet, tortor. Integer rhoncus commodo nisl. Praesent bibendum est non orci.Nam leo lectus, molestie non, commodo id, dignissim sit amet, tortor.

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Dignissim sit amet, tortor. Integer rhoncus commodo nisl. Praesent bibendum est non orci.Nam leo lectus, molestie non, commodo id, dignissim sit amet, tortor. Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Dignissim sit amet, tortor. Integer rhoncus commodo nisl. Praesent bibendum est non orci.Nam leo lectus, molestie non, commodo id, dignissim sit amet, tortor.

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Dignissim sit amet, tortor. Integer rhoncus commodo nisl. Praesent bibendum est non orci.Nam leo lectus, molestie non, commodo id, dignissim sit amet, tortor.

Figure 4: Example HTML page with single named-content

The following figure graphically illustrates an HTML page where each picture and each paragraph are marked as named-contents.


HTTP://MY-SITE/BOOK/1/CHAPTER_2.HTML
class="pundit-content" about="http://my-site/page/12"
class="pundit-content"
about="http://my-site/page/12/paragraph/1"
class="pundit-content"
about="http://my-site/image/2"
class="pundit-content"
about="http://my-site/page/12/paragraph/2"
class="pundit-content"
about="http://my-site/image/6"

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Dignissim sit amet, tortor. Integer rhoncus commodo nisl. Praesent bibendum est non orci.Nam leo lectus, molestie non, commodo id, dignissim sit amet, tortor.

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Dignissim sit amet, tortor. Integer rhoncus commodo nisl. Praesent bibendum est non orci.Nam leo lectus, molestie non, commodo id, dignissim sit amet, tortor. Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Dignissim sit amet, tortor. Integer rhoncus commodo nisl. Praesent bibendum est non orci.Nam leo lectus, molestie non, commodo id, dignissim sit amet, tortor.

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Dignissim sit amet, tortor. Integer rhoncus commodo nisl. Praesent bibendum est non orci.Nam leo lectus, molestie non, commodo id, dignissim sit amet, tortor.

Figure 5: Example HTML page each picture and paragraph are named-contents

7.1.2 Possible implementation

Content providers, with the assistance of WP3, will develop REST web services to provide stable and well defined URLs to get Annotable Versions of their contents at an appropriate level of granularity.

Example (purely demonstrative):

The following URL

http://example.org/annotable-version/book-213/transcripts/page-34

Could return an HTML annotable version of a single page of a transcription, optionally subdivided in paragraphs and possibly including pictures (e.g. formulas are often represented as pictures enclosed in the text transcript):

```
   <html>
      <body>
          <div class="pundit-content"
about="http://example.org/annotable-version/book-213/transcripts/page-
34">
               <p>
                   Title here
               </p>
               <div class="pundit-content"
about="http://example.org/annotable-version/book-213/transcripts/page-
34/paragraph-1">
                   A paragraph text here.
               </div>
               <div class="pundit-content"
about="http://example.org/annotable-version/book-213/transcripts/page-
34/paragraph-2">
                   A paragraph text here.
               </div>
               <div class="pundit-content"
about="http://example.org/annotable-version/book-213/transcripts/page-
34/paragraph-3">
                   A paragraph text here.
               </div>
               <div class="pundit-content"
about="http://example.org/annotable-version/book-213/pictures/pic-
13.jpg">
                   <img width="100" src="http://example.org/annotable-
version/book-213/pictures/pic-13.jpg" />
                   <p>Image caption here.</p>
               </div>
               <div class="pundit-content"
about="http://example.org/annotable-version/book-213/transcripts/page-
34/paragraph-4">
                   A paragraph text here.
               </div>
               <div class="pundit-content"
about="http://example.org/annotable-version/book-213/transcripts/page-
34/paragraph-5">
                   A paragraph text here.
               </div>
          </div>
      </body>
   </html>
```

The following URL

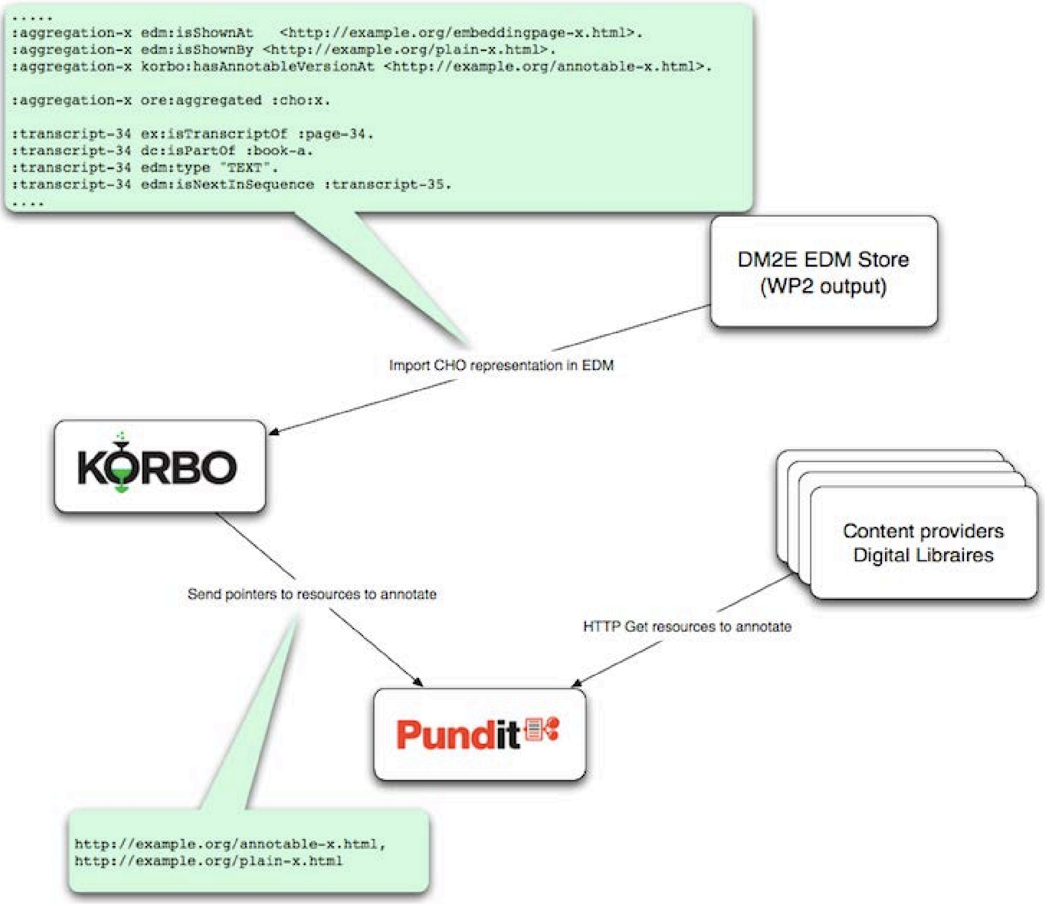
http://example.org/annotable-version/book-213/transcripts/page-34/paragraph-4

Could return an HTML annotable version of a single paragraph of a transcription:

```
      <html>
         <body>
             <div class="pundit-content"
   about="http://example.org/annotable-version/book-213/transcripts/page-
   34/paragraph-4">
                  <p>
                      A paragraph text here.
                  </p>
             </div>
         </body>
      </html>
```

7.2 EDM requirements

The following figure illustrates the high level interaction among the EDM store, Korbo (the content aggregation platform), and Pundit (the annotation tool).



```
.....
:aggregation-x edm:isShownAt   <http://example.org/embeddingpage-x.html>.
:aggregation-x edm:isShownBy <http://example.org/plain-x.html>.
:aggregation-x korbo:hasAnnotableVersionAt <http://example.org/annotable-x.html>.

:aggregation-x ore:aggregated :cho:x.

:transcript-34 ex:isTranscriptOf :page-34.
:transcript-34 dc:isPartOf :book-a.
:transcript-34 edm:type "TEXT".
:transcript-34 edm:isNextInSequence :transcript-35.
....
```

```
http://example.org/annotable-x.html,
http://example.org/plain-x.html
```

DM2E EDM Store
(WP2 output)
Import CHO representation in EDM
KORBO
Content providers
Digital Libraires
Send pointers to resources to annotate
HTTP Get resources to annotate
Pundit

Figure 6: Interaction EDM store, Korbo, and Pundit

Korbo will only import EDM data (it will not import other kinds of metadata providers might have), so it is important that all needed information (in the form of RDF triples) about digital objects are in the EDM.

RDF triples are needed by Korbo to let Pundit load content to be annotated. Such content can be images or text. As described in Named-content markup, the content should be provided in the form of "Annotable Versions" or as plain images.

The following triples are expected to appear in the EDM

A triple indicating where the Annotable Version can be retrieved:

```
@prefix rdf:  <http://purl.org/net7/korbo/vocab#> .
:aggregation-X korbo:hasAnnotableVersionAt <URL>
```

A triple indicating where the "plain" web representation (meaning it contains only the representation of the providedCHO):

```
:aggregation-X edm:isShownBy <URL>
```

A triple pointing to a web page in the original Digital Library where the providedCHO appears:

```
:aggregation-X edm:isShownAt <URL>
```

A triple pointing to a preview picture for the ProvidedCHO

```
:aggregation-X edm:object <URL>
```

Other RDF triples can be useful for Korbo to render basket items, arrange them in meaningful ways or even suggest users to import related items.

Some triples that might be useful from WP3 view point are:

Triples stating an isPartOf relation, for example putting in relation a manuscript page with the entire manuscript

```
:providedCHO-X dc:isPartOf :providedCHO-Y

Back-links are also useful:
:providedCHO-Y dc:hasPart :providedCHO-X
:providedCHO-Y dc:hasPart :providedCHO-Z
```

Triples stating other kinds of semantic relations among ProvidedCHO. In DM2E, as mainly manuscripts are relevant, useful information would be to know what image corresponds to what transcription.

```
:providedCHO-X bibo:transcriptOf providedCHO-Y
```

Back-links are also useful.

Triples indicating the type of content:

```
:providedCHO-X edm:type "TEXT"
:providedCHO-X edm:type "Transcription"
```

In the case of dc:type (or other properties with the same role), it should be useful to agree on a vocabulary (e.g. a SKOS taxonomy) to be supported.

8 Summary

This document describes the results of DM2E's Task 3.1 the "Initial functional specifications of the prototyping platform" and parts of Task 3.2 "Building of the prototype platform".

The main goal of this document is to give an overview of the requirements gathered in the first six DM2E project months. This deliverable, D3.1 "Initial Specifications Report" was designed to provide an overview of the requirements gathering process. Furthermore, it was intended as a guideline for other project partners and ensures the quality and relevance of the development of the experimental content enrichment and recombination platform.

This deliverable lists the requirements gathered in the first two Digital Humanities Advisory Board Meetings (DHAB) and the WP3 virtual meetings. Moreover, this document also contains a summary of other relevant Digital Humanities projects which have an impact on the planned enrichment and recombination platform.

Through this requirements gathering process it was possible for WP3 to identify most relevant requirements which can be now implemented in the prototype which is due in month 11. The last part of the report presents the feasible requirements for the prototype platform as well as requirements towards the content providers which have to make their content available in the new Europeana Data Model (EDM).

9 Appendix A: Korbo alfa API specification

Linked Data API

Korbo exposes resources conforming to the Linked Data principles, using dereferenciable URLs.

Main resources in Korbo are Items and Baskets (collections of items).

Korbo provides Linked Data access to such resources relying on ELDA[17]. Linked Data URLs are composed as follows:

```
{korbo-instance-host}/elda/api/korbo/item/{item-id}
{korbo-instance-host}/elda/api/korbo/basket/{basket-id}
```

Where {korbo-instance-host} is the host of the specific instance of Korbo. For example the demo instance host is http://korbo.netseven.it.

Specific representation formats can be requested using standard HTTP header "Accept:" or by appending appropriate file extension to the URL

Supported representation formats are:

RDF/XML

Accept: "application/rdf+xml"
OR
File extension ".rdf"

Turtle

Accept: "text/turtle"
OR
File extension ".ttl"

JSON

File extension ".json"

XML

Accept: "text/xml"
OR
File extension ".xml"

Error codes

When a request is made to a basket or item that does not exist in Korbo the response is 404.

Using PURLs as URL identifiers

In order to provide stable identifiers to resources in cases where a Korbo instance moves to another server we encourage the use of a redirection layer as Purl.org. Purls are used as identifiers for the resources.

[17] http://elda.googlecode.com/hg/deliver-elda/src/main/docs/index.html [Retrieved 30.07.2012].

TODO: add documentation on how Korbo purl can be configured.

For example, the demo instance uses the following purls:

- **Korbo baskets:** http://purl.org/net7/korbo/basket/{basket-id}
- **Korbo items:** http://purl.org/net7/korbo/item/{item-id}

Resolving such URLs results in redirects (e.g. 302 FOUND) to {{korbo-instance-host}}/elda/api/korbo/item/{item-id} or {korbo-instance-host}/elda/api/korbo/basket/{basket-id}.

JSONP support

All API calls, except the Linked Data API, support JSONP via the parameter **"jsonp".**

Basket SPARQL endpoint

Each basket has a dedicated SPARQL endpoint. A subset of the SPARQL protocol is supported.

In particular:

- FROM and FROM NAMED clauses are not supported. Such clauses. If present they are removed before executing the query.
- Only GET requests are supported. At the moment POST requests are turned into GET requests with same parameters and headers.

See http://www.w3.org/TR/sparql11-protocol for more info.

Request

```
GET /basket/sparql/{basket-id}
```

1. API: Basket metadata

Each basket has a set of attributes. They specify generic metadata of the basket, e.g. to provide previews or short description of Baskets

Request

```
GET /basket/metadata/{basket-id}
```

Response

```
{
   "result" : [
       {
         "id" : "123jkh",
         "name" : "Disney cartoons",
         "description" : "This is a sample Korbo basket created for demonstration purpose.",
         "image" : "http://en.wikipedia.org/wiki/File:Goofy.svg",
         "owner_id" : "656kjk",
         "owner" :  "Christian Morbidoni"
       }
    ],
    .. space for useful envelope data
 }
```

2. API: Basket domains

Items in a basket come from different data sources. This API call returns a list of all the web domains from where items in the basket have been imported.

Request

```
GET /basket/domains/{basket-id}
```

Response

```
{
  "result": [
        {
           name : "pippo.it",
           number : 24
         },
         {
           name : "data.euroepana.eu",
           number : 12
         }
   ]
   .. space for useful envelope data
}
```

3. API: Items in a basket

This API call returns the complete list of all items present in a basket.

Request

```
GET /basket/items/{basket-id}**?limit=10=&offset=30
```

Response

```
{
    "result" : [
       {
         "id": "454kyk",
         "name" : "Mickey mouse",
         "description" : "One of the more popular characters",
         "image" : "http://upload.wikimedia.org/wikipedia/it/c/c0/Topolino_%28fumetto%29.jpg"
       },
       {
         "id": "673hgu",
         "name" : "Scrooge McDuck",
         "description" : "The The rich uncle of donald Duck",
         "image" : "http://upload.wikimedia.org/wikipedia/it/thumb/7/71/Scroogemcduck.jpg/280px-
Scroogemcduck.jpg"
       }
    ],
    .. space for useful envelope data
 }
```

4. API: Basket reconciliation service

This API provides a reconciliation service compliant with the specification http://code.google.com/p/google-refine/wiki/ReconciliationServiceApi

Implementation: queries are matched against items label and description.

Request

```
GET /basket/reconcile/{basket-id}?query={...json object literal...}
```

Example of query JSON object literal

```
{
   "query" : "Duck",
   "limit" : 10,
   "type" : "????", // TODO: discuss possible values of this field
   "type_strict" : "any", // TODO: discuss possible values of this field, if to be supported
   "properties" : [
     { "p" : "creator", "v" : "Carl Barks" }, // TODO: discuss use of properties. A possibility would be to
associate a property to a graph pattern in the triplestore. e.g. creator = dc:creator |
edm:proxyFor.dc:creator
   ]
}
```

Response

```
{
    "result" : [
       {
         "id": "588ewo",
         "name" : "Donald Duck",
         "description" : "Called Paperino in italy is the most famous Duck in the Duck city",
         "image" : "http://upload.wikimedia.org/wikipedia/it/c/c0/Topolino_%28fumetto%29.jpg",
         "score" : 4
         // Possible score is (1 + N) IFF label matches, N IFF description matches, (2 + N) IFF description
and label matches
         // Where N is the number of total occurrences.
       },
       {
         "id": "673hgu",
         "name" : "Scrooge McDuck",
         "description" : "The The rich uncle of donald Duck",
         "image" : "http://upload.wikimedia.org/wikipedia/it/thumb/7/71/Scroogemcduck.jpg/280px-
Scroogemcduck.jpg",
         "score" : 3
       }
    ],
    .. space for useful envelope data
 }
```

5. API: Item Metadata

Every item imported in the KORBO platform has a set of „imported“ metadata that are derived form the original item metadata at the moment of importing it to the KORBO. They are a limited set of metadata (source url, label, description, type, domain, thumbnail)

```
GET /item/metadata/{item-id}
```

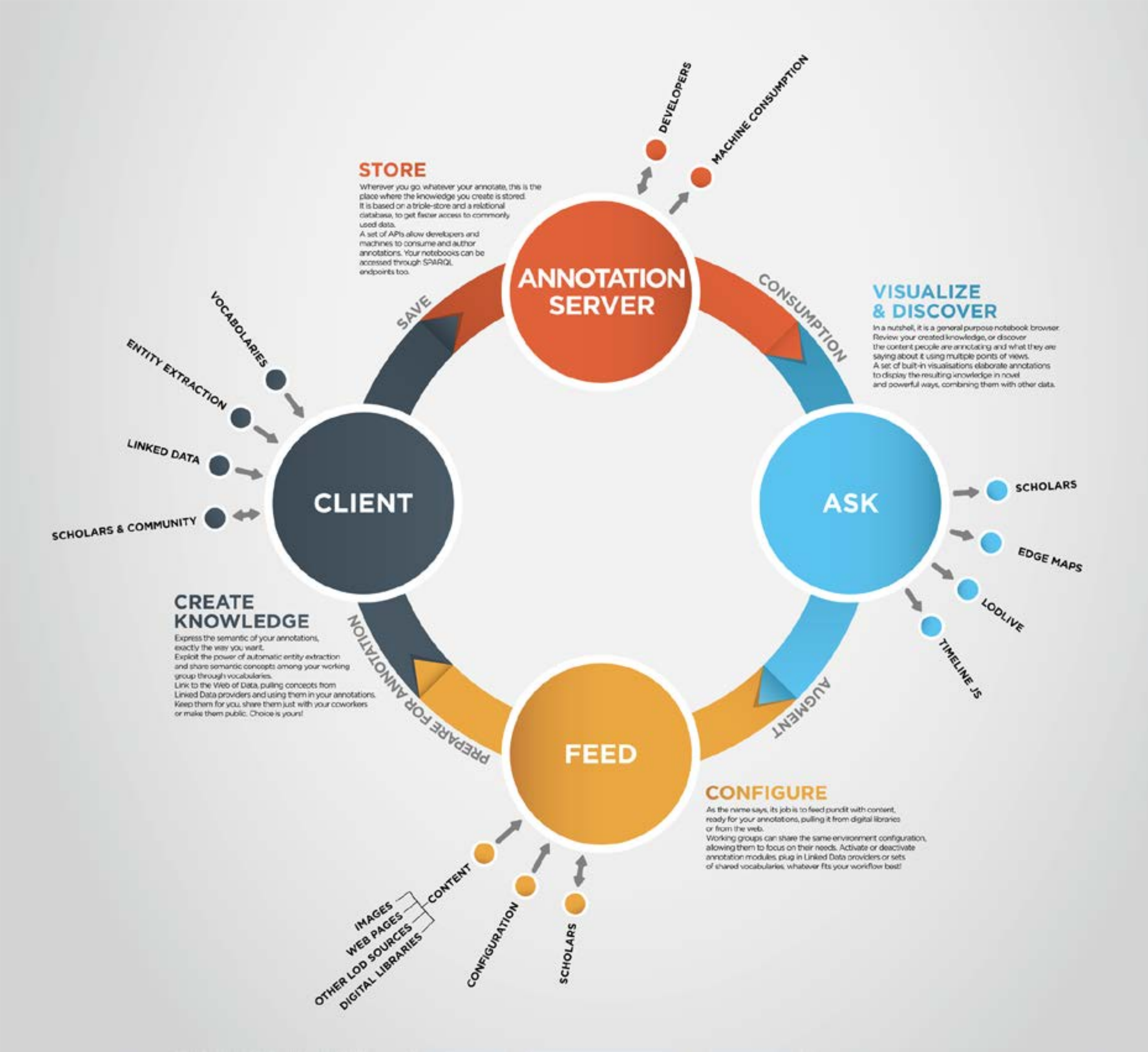
Response

```
{
    "result" : [
      {
         "id": "673hgu",
         "name" : "Scrooge McDuck",
         "description" : "The The rich uncle of donald Duck",
         "image" : "http://upload.wikimedia.org/wikipedia/it/thumb/7/71/Scroogemcduck.jpg/280px-
Scroogemcduck.jpg",
         "source_url" : "http://disney.com/characters/1234"
      }
   ],
   .. space for useful envelope data
  }
```

10 Appendix B: The Pundit Ecosystem

This appendix clarifies the software components that are discussed in the framework of the DM2E scholarship environment.

The main components that constitute the Pundit system are depicted in the following figure:



STORE

Wherever you go whatever your annotate, this is the place where the knowledge you create is stored. It is based on a triple-store and a relational database, to get faster access to commonly used data. A set of APIs allow developers and machines to consume and author annotations. Your notebooks can be accessed through SPARQL endpoints too.

VISUALIZE & DISCOVER

In a nutshell, it is a general purpose notebook browser. Review your created knowledge, or discover the content people are annotating and what they are saying about it using multiple points of views. A set of built-in visualisations elaborate annotations to display the resulting knowledge in novel and powerful ways, combining them with other data.

CREATE KNOWLEDGE

Express the semantic of your annotations, exactly the way you want. Exploit the power of automatic entity extraction and share semantic concepts among your working group through vocabularies. Link to the Web of Data, pulling concepts from Linked Data providers and using them in your annotations. Keep them for you, share them just with your coworkers or make them public. Choice is yours!

CONFIGURE

As the name says, its job is to feed pundit with content, ready for your annotations, pulling it from digital libraries or from the web. Working groups can share the same environment configuration, allowing them to focus on their needs. Activate or deactivate annotation modules, plug in Linked Data providers or sets of shared vocabularies, whatever fits your workflow best!

Figure 7: Main components of the Pundit system

The following table summarises the components, shortly describing their roles in the overall system.

Grey rows indicate the components under development in the DM2E project, the white ones are the additional software tools that are being integrated and customised to build showcases and conduct experiments.

Name	Role	Technology	Notes	Demo/URL
Annotation	Provides storage for annotations	Sesame API compliant	Provides REST APIs to consume and	Public release in summer

Server	and related RDF data	triple store, MySQL database	create annotations	2013
Pundit client	Provides the user interface to annotate a web page.	Javascript + Dojo framework	Can be configured to host specific vocabularies and to add/remove GUI components	http://goo.gl/BWWTe
Ask the Pundit	Provides a web portal for managing personal notebooks and search public ones.	NodeJs + Javascript + Dojo 1.8	The portal also includes links to vertical external visualizations	http://ask.as.thepund.it
Feed the Pundit	Provides a point of access to Pundit as-a-service. Given a web resource URL, Pundit is instantiated to annotate the resource.	PHP + Javascript	Provides a Web GUI and a REST API that can be used to integrate the Pundit client with generic web applications. Takes as input parameter the desired configuration of the Pundit client.	http://feed.thepund.it
Muruca DL	Legacy Net7 solution for Linked Data Digital Libraries	PHP + Sinfony	It has been adopted by the Wittgenstein Archive in Bergen. Works well with Pundit as it provides stable dereferenceable URLs for the content in different formats.	http://wittgensteinsource.org
BoxView	A content visualization javascript library. Allows different contents to be dynamically displayed in the same browser tab.	Javascript + JQuery	It is used as a default template in MURUCA DL	http://wittgensteinsource.org
LodLive	Open source Linked Data browsing tool	Javascript	Used as demonstrative annotation visualizer and browser	http://goo.gl/K03MQ
Edgemaps	Open source graph visualization tool	Raphael JS + SVG	Used as a demonstrative vertical visualization of annotations (citations among philosophers)	http://goo.gl/h72ku