DELIVERABLE

Project Acronym: DM2E

Grant Agreement number: ICT-PSP-297274

Project Title: Digitised Manuscripts to Europeana

D3.3 - E-Learning Courses published

Revision: Final 1.1

Authors:

Christian Morbidoni (NET7)
Francesca Di Donato (NET7)
Simone Fonda (NET7)

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

Revision history and statement of originality

Revision	Date	Author	Organisation	Description
0.1	06.02.14	Christian Morbidoni, Francesca Di Donato, Simone Fonda	Net7	Initial draft version
0.2	09.02.14	Lena Stanley-Clamp	EAJC	Proofreading
0.3	11.02.14	Steffen Hennicke	UBER	Some additions and revision
0.4	12.02.14	Violeta Trkulja	UBER	Final revision
0.5	13.02.14	Christian Morbidoni	Net7	Some additions
1.0	14.02.14	Violeta Trkulja	UBER	Approval of Final 1.0
1.1	21.02.14	Violeta Trkulja	UBER	Fixed some broken links

Contents

1 Role and scope of this deliverable

The Work Package 3 of the DM2E project is working to develop and constantly improve a software platform, which is composed of some core blocks developed within this project, and which makes use of open-source tools to build applications on top of the Linked Data produced in the DM2E project.

Such data consists of:

- the Linked Data produced by the ingestion and transformation of the data provided by DM2E content partners thanks to the modeling and technical work done in WP1 and 2. The data can be consumed via a standard Linked Data interface - and by SPARQL in controlled environments - and is represented in the DM2E RDF data model;
- the annotations created by scholars using Pundit (one of the core components of the platform). Annotations are actual structured data (represented in RDF) that enrich the online content with connections among texts, images and external Linked Data entities. This data can be consumed via the Pundit Server REST API or via SPARQL endpoints.

The platform has a two-fold target:

- To provide support for common scholarly activities such as commenting, annotating, collecting, and sharing. Exploring benefits that the Semantic Web technologies can bring to users;
- To provide a framework for developers to build on, by reusing data and annotations, deploying the tools for specific communities and by providing specialised visualisations/explorations of data.

In this deliverable we first provide the context by presenting an overview of the main components of the platform, documented in the previous Deliverable 3.2, we then present a set of learning materials that we are publishing on the Wiki of the project.

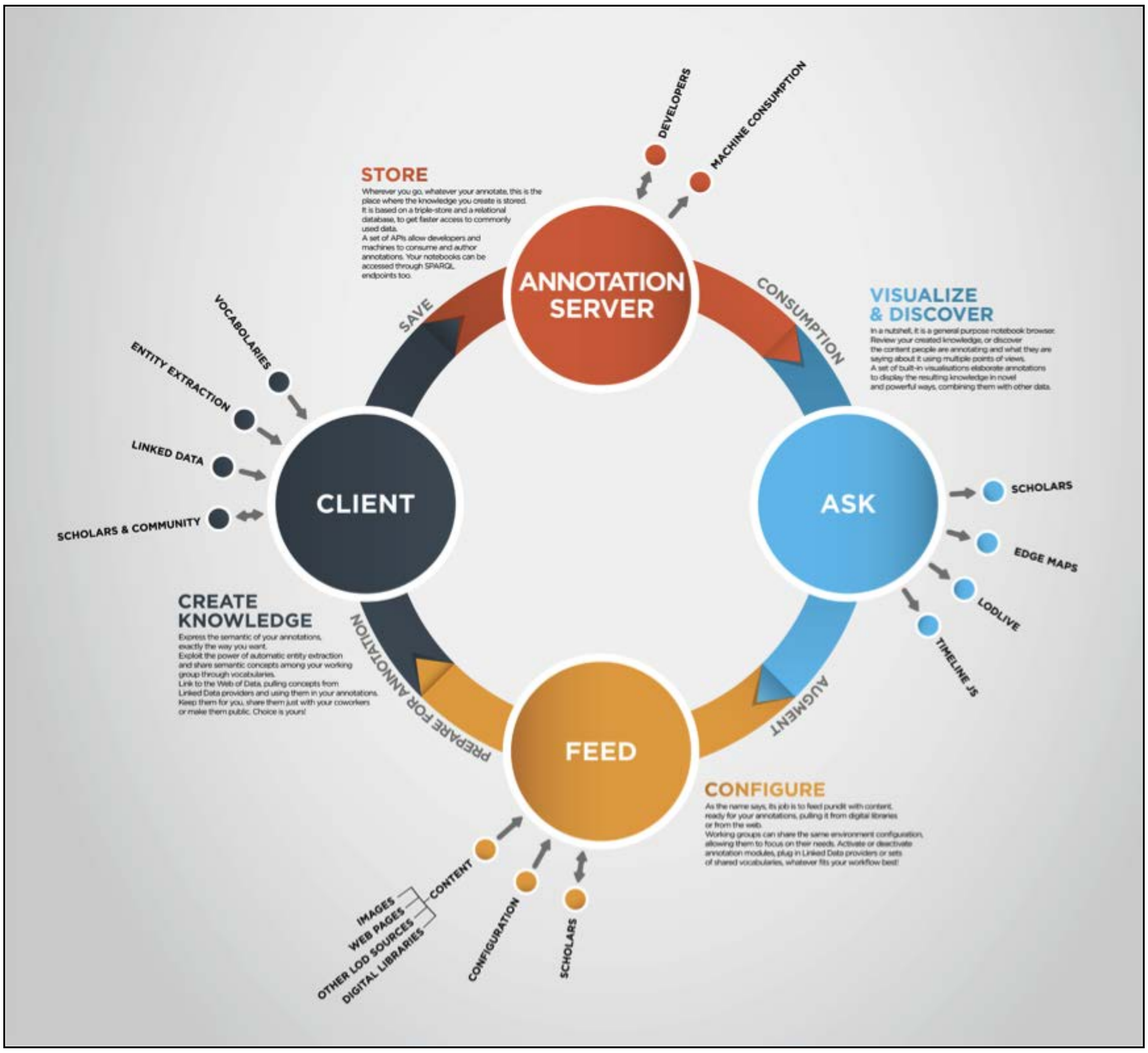
The learning materials are targeted both at scholars, showing possible use of the tools, and at developers, illustrating how the platform can be extended by means of practical demonstrative applications. All materials have a working online demo associated with it.

2 Overview of the DM2E software platform

The core components of the infrastructure that we will use as a backbone of the demonstrative applications are:

- **Pundit** as a configurable annotation tool:
  - **Pundit Server** to store annotations as RDF and provide REST API to data.
  - **Pundit Client** provides a highly configurable, web based user interface for annotating online content.
- **Ask** for collecting/managing/searching public and private annotations as well as to access vertical applications, which are, for example, visualisations of data for a specific domain;
- **Feed** to expose the *Pundit client* as a REST web service and connect it to external web applications

These four components cooperate to create end-user workflows where knowledge is created via annotations, shared and accessed via Ask, and consumed by additional applications to meet specific needs.



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 1. The software ecosystem. Different independent web applications modules interact to support annotation creation, publication and exploration workflows.

The following table summarizes the different software tools that have been put in place to develop the demonstrative applications addressed in the learning materials.

Marked in dark grey (in Table 1), we list the main components that have been developed in the framework of this project. White cells refer to existing open software that has been customised and reused to produce annotations, and examples of applications for data visualisation.

Name	**Role**	**Technology**	**Notes**	**Demo/URL**
Pundit Server	Provides storage for annotations and related RDF data	Sesame API compliant triple store, MySQL database	Provides REST APIs to consume and create annotations	Public release in summer 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	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 visualisations	http://ask.as.thepund.it/
Feed	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/
Korbo	Manages vocabularies and can be used to edit vocabularies.	PHP	This tool is currently in alpha. It provides storage for JSON vocabularies and an API to connect them to Pundit.	http://korbo.org/
Muruca DL	Legacy Net7 solution for Linked Data Digital Libraries	PHP + Sinfony	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 visualisation javascript library. Allows different contents to be dynamically displayed in the same browser tab.	Javascript + JQuery	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	Raphael JS + SVG	Used as a demonstrative vertical visualisation of	http://goo.gl/h72ku

	visualisation tool		annotations (citations among philosophers)	
TimelineJS	Open source timeline visualisation javascript library	Javascript	Used as a demonstrative vertical application	https://github.com/NUKnightLab/TimelineJS
Solr	Open source HTTP API over Lucene text indexing system	Java - HTTP API	Used to drive a demonstrative faceted browser on DM2E data	http://lucene.apache.org/solr/

Table 1. List of tools used in the DM2E platform.

In the Sections below we present an overview of the core components.

2.1 Creating rich structured annotations: Pundit

Annotation is a primary activity for scholars and professionals. It consists in enriching a content with some new information, which possibly helps in understanding or searching the content itself. While until a few decades ago annotations were drafted by hand in the margins of a book, today's web technologies have the potential to make them infinitely replicable, remotely accessible and easy to share.

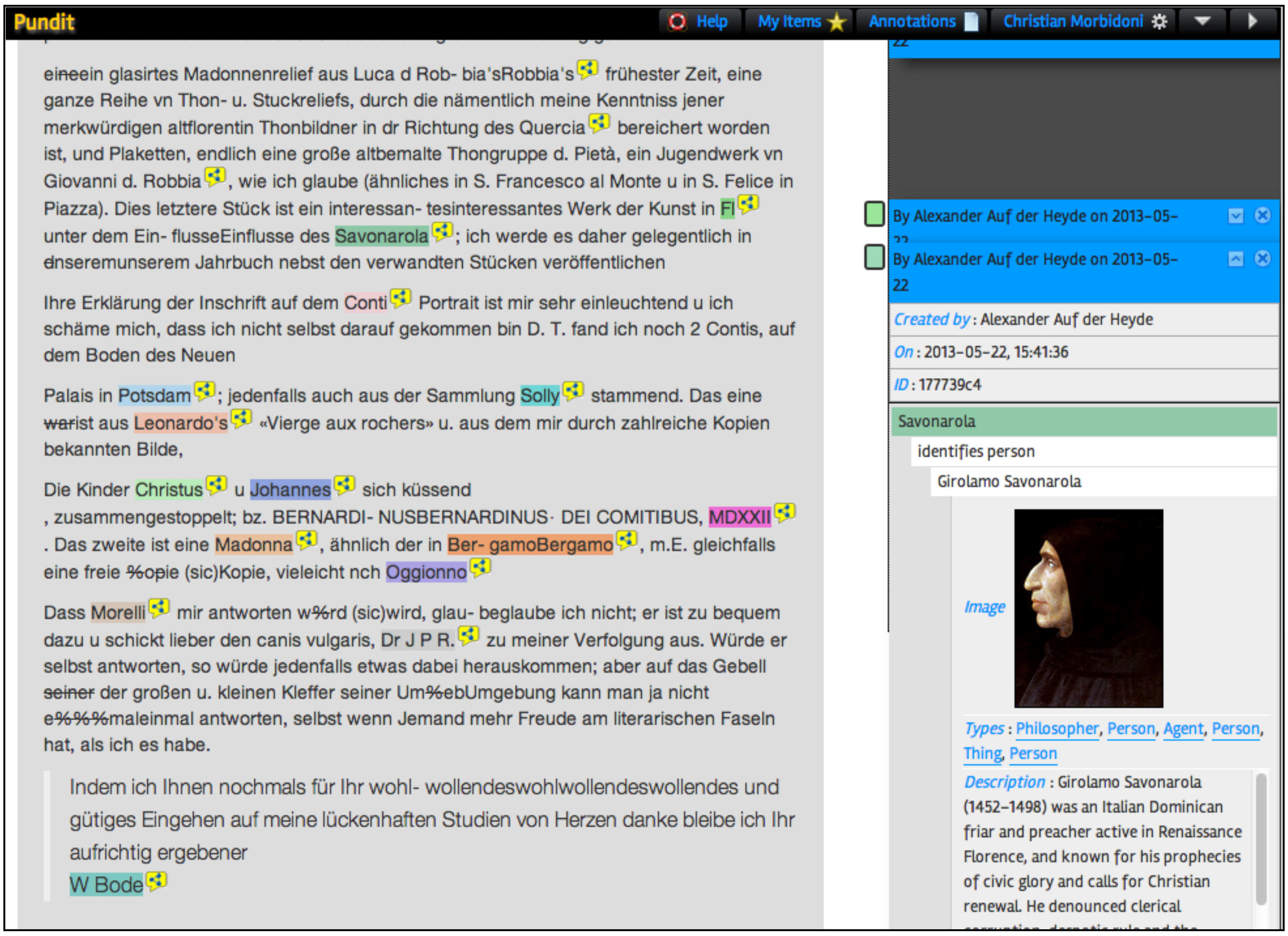
Web annotations systems and bookmarking/clipping tools are popular nowadays both among generic users (e.g. social tagging) and among scholarly communities (e.g. http://www.zotero.org, http://www.mendeley.com/).

However, existing annotation systems are generally limited to textual comments, tags or predefined metadata templates (e.g. bibliographic records). Furthermore, annotations are often isolated into closed systems and very rarely are connected to the Web of Data.

The simple idea behind Pundit is that of using annotations as a vehicle to create new semantic web data, adding links and, ultimately, knowledge to the so-called Global Data Space[1]. Once annotations become available in a standard and highly expressive form, a variety of applications can be built to visualize the resulting knowledge in specific domains.

Pundit is a novel annotation system that aims to implement this vision, by enabling annotators (e.g. scholars) to use semantically specified relations and to link to web of data entities, producing in this way accessible RDF graphs out of their work. Such RDF graphs are collections of annotations that we call “notebooks''. Notebooks can be consumed via REST APIs or standard SPARQL endpoints.

[1] http://linkeddatabook.com/editions/1.0/



ei~~ne~~ein glasirtes Madonnenrelief aus Luca d Rob- bia'sRobbia's frühester Zeit, eine ganze Reihe vn Thon- u. Stuckreliefs, durch die nämentlich meine Kenntniss jener merkwürdigen altflorentin Thonbildner in dr Richtung des Quercia bereichert worden ist, und Plaketten, endlich eine große altbemalte Thongruppe d. Pietà, ein Jugendwerk vn Giovanni d. Robbia, wie ich glaube (ähnliches in S. Francesco al Monte u in S. Felice in Piazza). Dies letztere Stück ist ein interessan- tesinteressantes Werk der Kunst in Fl unter dem Ein- flusseEinflusse des Savonarola; ich werde es daher gelegentlich in ~~d~~nseremunserem Jahrbuch nebst den verwandten Stücken veröffentlichen

Ihre Erklärung der Inschrift auf dem Conti Portrait ist mir sehr einleuchtend u ich schäme mich, dass ich nicht selbst darauf gekommen bin D. T. fand ich noch 2 Contis, auf dem Boden des Neuen

Palais in Potsdam; jedenfalls auch aus der Sammlung Solly stammend. Das eine ~~war~~ist aus Leonardo's «Vierge aux rochers» u. aus dem mir durch zahlreiche Kopien bekannten Bilde,

Die Kinder Christus u Johannes sich küssend
, zusammengestoppelt; bz. BERNARDI- NUSBERNARDINUS· DEI COMITIBUS, MDXXII
. Das zweite ist eine Madonna, ähnlich der in Ber- gamoBergamo, m.E. gleichfalls eine freie ~~%op~~ie (sic)Kopie, vieleicht nch Oggionno

Dass Morelli mir antworten w~~%~~rd (sic)wird, glau- beglaube ich nicht; er ist zu bequem dazu u schickt lieber den canis vulgaris, Dr J P R. zu meiner Verfolgung aus. Würde er selbst antworten, so würde jedenfalls etwas dabei herauskommen; aber auf das Gebell ~~seiner~~ der großen u. kleinen Kleffer seiner Um~~%eb~~Umgebung kann man ja nicht e~~%%%~~maleinmal antworten, selbst wenn Jemand mehr Freude am literarischen Faseln hat, als ich es habe.

> Indem ich Ihnen nochmals für Ihr wohl- wollendeswohlwollendeswollendes und gütiges Eingehen auf meine lückenhaften Studien von Herzen danke bleibe ich Ihr aufrichtig ergebener
> W Bode

By Alexander Auf der Heyde on 2013-05-22

By Alexander Auf der Heyde on 2013-05-22

Created by : Alexander Auf der Heyde

On : 2013-05-22, 15:41:36

ID : 177739c4

Savonarola

identifies person

Girolamo Savonarola

Image

Types : Philosopher, Person, Agent, Person, Thing, Person

Description : Girolamo Savonarola (1452–1498) was an Italian Dominican friar and preacher active in Renaissance Florence, and known for his prophecies of civic glory and calls for Christian renewal. He denounced clerical

Figure 2. A screenshot of Pundit showing different annotations on a web page. The text has been marked with connections to mentioned persons, places and works of art.

2.1.1 Understanding Pundit annotations

Annotations in Pundit are essentially triples that connect different kinds of items together. A triple has the form [subject - predicate - object], where the subject and object can be segments of text and images

```
[ text - describes - image ]
```

or entities from the web of data

```
[ text - has author - Dante(from Freebase.com) ]
```

or

```
[ image - depicts - Florence(from DBPedia.org) ]
```

The most expressive annotation interface provided by the Pundit client is the "triple composer", shown in Figure 3. It allows users to drag and drop items into triples, or select them from the web page (e.g. by selecting a text or an image), as well as searching in the available vocabularies and data sources. However, other annotation wizards support specific kinds of annotations, such as putting two segments of text in relation, or attaching tags and comments to a text segment. Image annotation of segments of images is supported by a dedicated module as shown in Figure 4.

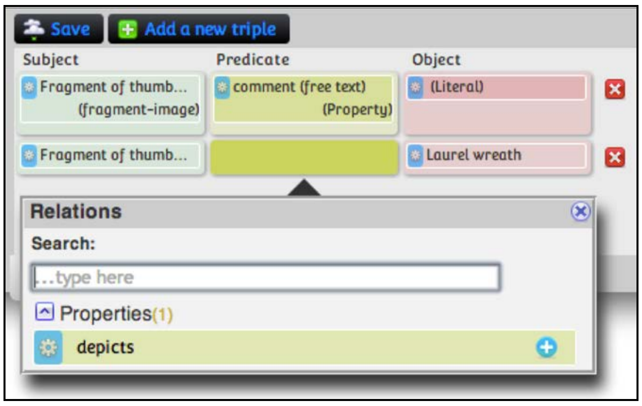


Figure 3: A detail of the Pundit triple composer. It allows the user to put in relation items from a web page (as text and images) with other page items, entities from domain vocabularies or simple textual comments.

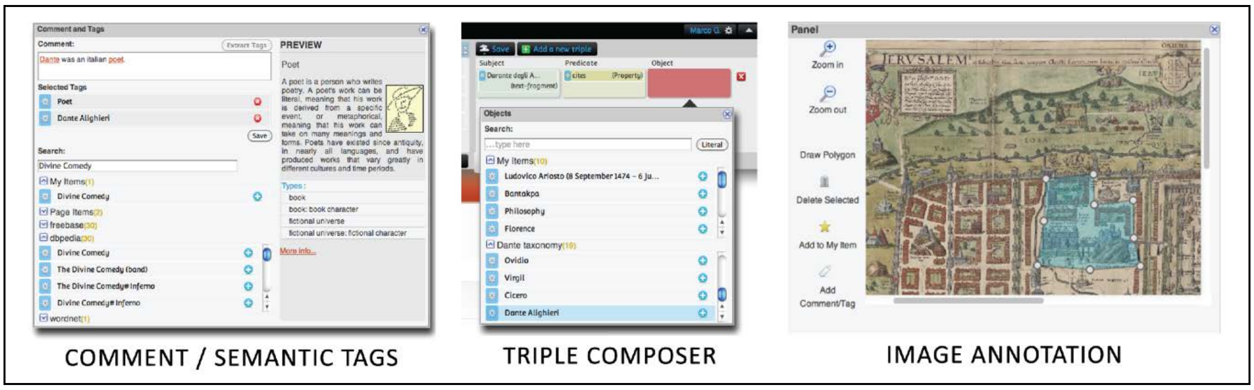


Figure 4: Examples of annotations modules in Pundit.

The Pundit client is a JavaScript application that can be deployed as a library, to be then easily included in existing web sites to make the content "annotatable''. This was achieved in wittgensteinsource.org, as well as delivered as a bookmarklet. A bookmarklet is a simple link (bookmark) that, once added to a web browser, allows Pundit to be loaded on every web page and its content annotated.

In Pundit, an annotation contains information at a two-fold level:

(1) the “annotation metadata" deals with the act of annotating, including information on the author, the time of creation and the involved web resources. Pundit is based on the Open Annotation data model (OA, http://www.openannotation.org/spec/core/) for representing this dimension.

(2) the “annotation graph'' is an RDF graph resulting from metadata and relations among web resources that a user has created by annotating. In other words, it captures the semantics of the annotation representing the user's contribution in terms of “domain knowledge''.

For example, an annotation graph could contain Wikipedia pages corresponding to Italian writers and relevant text segments from their works on wikisource.org or other open web archives, perhaps linking each text to a number of other texts from the relevant contemporary writers. We call “items'' the nodes of such a graph, which represents the annotated web resources, consisting of web pages segments or other kind of entities (places, persons, etc.).

2.1.2 Domain vocabularies

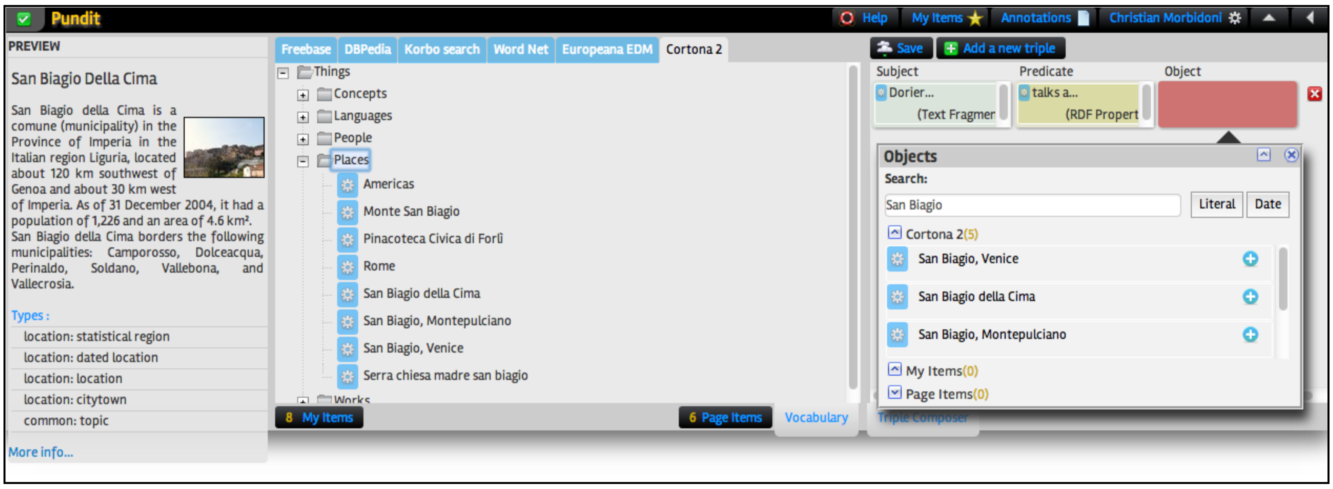
One of the most successful approaches to fostering the reuse of data on the web is to create a consensus around vocabularies and ontologies within a certain community. In Pundit we try to follow this pattern by making it possible to deploy customised annotation clients, in the form of JavaScript libraries or bookmarklets, which can be distributed to users by community leaders.

A custom client may include a precise set of a well-defined *set of relations* to be used in annotations to create typed links among items or *entities taxonomies* where relevant web entities are collected and ready to be annotated.

Both taxonomies and relations are represented in JSON and can be easily extracted from existing vocabularies (e.g. SKOS) or ontologies, as we did in the Wittgenstein's brown book pilot (see http://dm2e.eu/wittgenstein-incubator-workshop-bergen-6122013/ for details).

To understand how to create a vocabulary for Pundit, please refer to online documentation (http://www.thepund.it/documentation/use-and-install-the-client/) or to the User Tutorials number 1 and 2 (available in the DM2E wiki at: http://wiki.dm2e.eu/).

In Figure 5 we show the Pundit annotation toolbar. One can see that domain vocabularies can be explored as a taxonomy or searched directly from the triple composer to then be used in annotations.



San Biagio della Cima is a comune (municipality) in the Province of Imperia in the Italian region Liguria, located about 120 km southwest of Genoa and about 30 km west of Imperia. As of 31 December 2004, it had a population of 1,226 and an area of 4.6 km². San Biagio della Cima borders the following municipalities: Camporosso, Dolceacqua, Perinaldo, Soldano, Vallebona, and Vallecrosia.

Types :

location: statistical region
location: dated location
location: location
location: citytown
common: topic

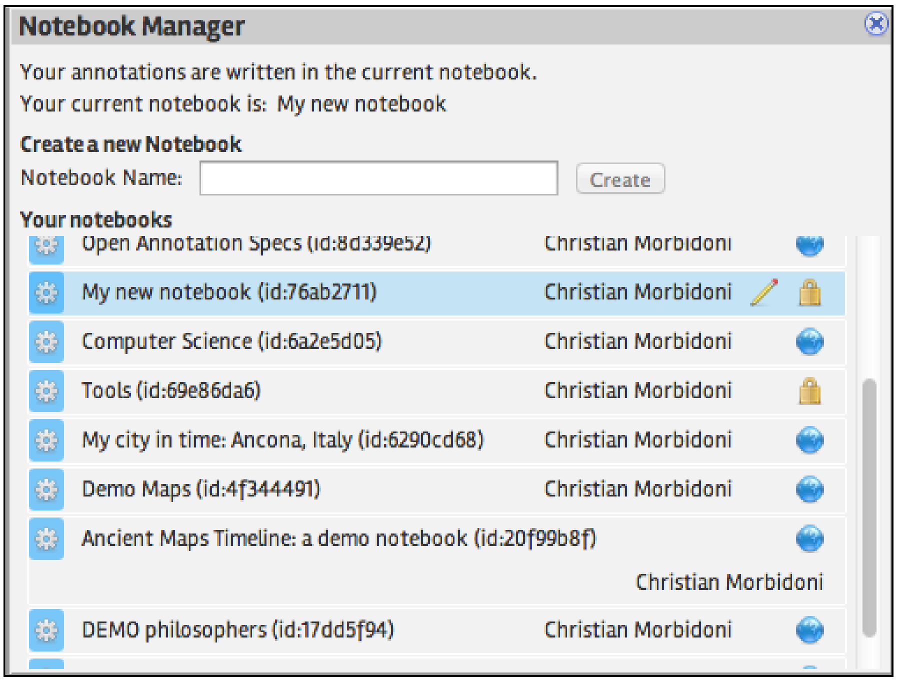
More info...

Figure 5. Domain vocabularies and the triple composer.

2.1.3 Notebooks

Aggregating items in a collection, for sharing and publishing, is a common pattern in social clipping and bookmarking tools.

In Pundit, annotations are collected in so called *notebooks* that users can optionally make publicly accessible. When a notebook is public, the annotations contained in it are not only shown in the Pundit client (e.g. when a user loads the Pundit bookmarklet on one of the annotated web pages) but, more interestingly, a notebook can be consumed by means of open REST APIs and accessed by a variety of web applications. Each notebook provides a SPARQL endpoint to query its content. In other words, a notebook is an independent RDF graph created by a given user in time and connecting a variety of web resources.


Notebook Manager

Your annotations are written in the current notebook.
Your current notebook is: My new notebook

Create a new Notebook
Notebook Name: Create

Your notebooks

Open Annotation Specs (id:8d339e52)	Christian Morbidoni	🌐
My new notebook (id:76ab2711)	Christian Morbidoni	✏ 🔒
Computer Science (id:6a2e5d05)	Christian Morbidoni	🌐
Tools (id:69e86da6)	Christian Morbidoni	🔒
My city in time: Ancona, Italy (id:6290cd68)	Christian Morbidoni	🌐
Demo Maps (id:4f344491)	Christian Morbidoni	🌐
Ancient Maps Timeline: a demo notebook (id:20f99b8f)	Christian Morbidoni	🌐
DEMO philosophers (id:17dd5f94)	Christian Morbidoni	🌐

Figure 6. The Pundit notebook manager.

As shown in Figure 6, Pundit includes a dedicated UI for supporting users in managing their notebooks. Notebooks can be private or public and their status can be changed at any time by the owner. The pencil icon indicates what notebook is currently used and can be easily changed.

2.1.4 Online documentation

The following resources have been made available to provide a starting point for "advanced" users and developers.

- Pundit quick guide:
  http://www.thepund.it/documentation/quick-start-guide-to-pundit/
- Install the Pundit client
  http://www.thepund.it/introductory-videos/use-and-install-the-client/
- Advanced: using named contents in your pages
  http://www.thepund.it/documentation/play-nice-with-pundit/
- Pundit JavaScript client API
  http://release-bot.thepund.it/latest/docs/
- Pundit Client source code
  https://github.com/net7/pundit

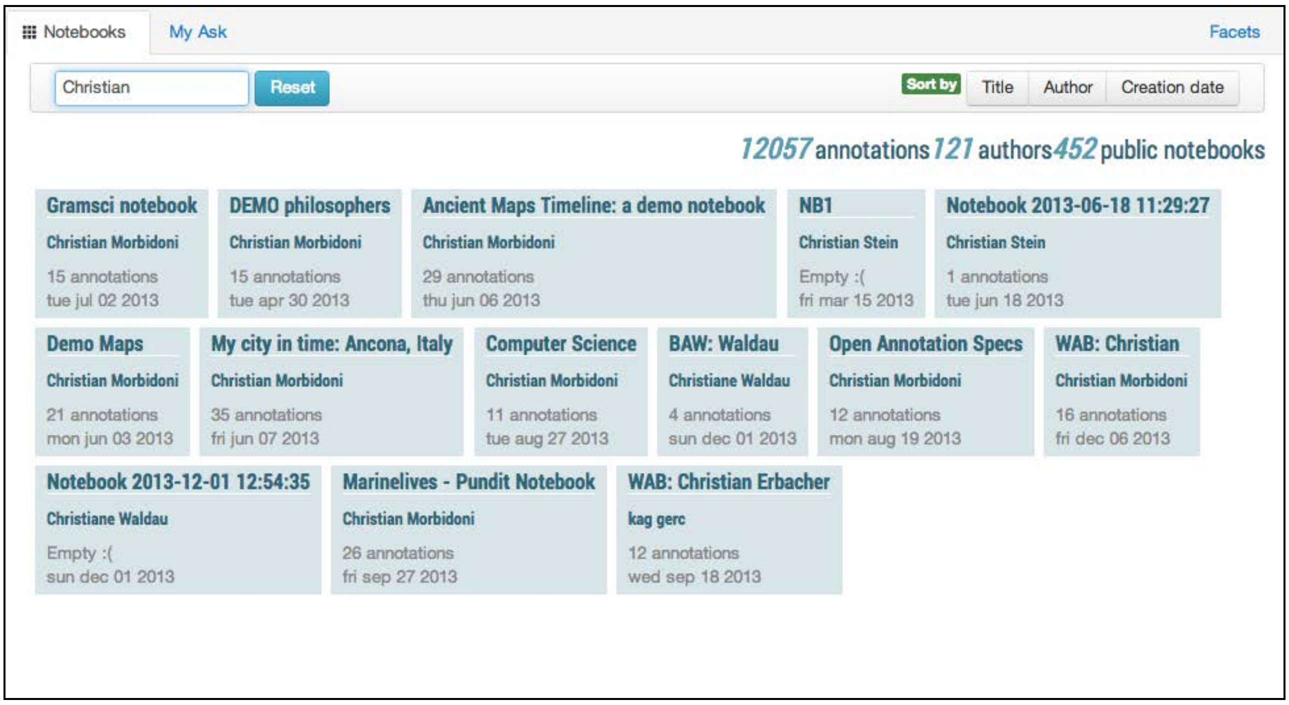
- Annotation server REST API
  http://thepund.it/documentation/pundit-server-api/

2.2 Publishing, sharing and discovering annotations: Ask

Ask (http://ask.as.thepund.it) is a web application where public notebooks stored in Pundit can be searched and explored. At the time of writing a new version of the tool is being released.

2.2.1 Searching public notebooks

The main page of Ask allows to search through public notebooks, by specifying authors' name, date or simply by searching the entire text for their titles and descriptions. This is shown in Figure 7.



12057 annotations 121 authors 452 public notebooks

Gramsci notebook
Christian Morbidoni
15 annotations
tue jul 02 2013

DEMO philosophers
Christian Morbidoni
15 annotations
tue apr 30 2013

Ancient Maps Timeline: a demo notebook
Christian Morbidoni
29 annotations
thu jun 06 2013

NB1
Christian Stein
Empty :(
fri mar 15 2013

Notebook 2013-06-18 11:29:27
Christian Stein
1 annotations
tue jun 18 2013

Demo Maps
Christian Morbidoni
21 annotations
mon jun 03 2013

My city in time: Ancona, Italy
Christian Morbidoni
35 annotations
fri jun 07 2013

Computer Science
Christian Morbidoni
11 annotations
tue aug 27 2013

BAW: Waldau
Christiane Waldau
4 annotations
sun dec 01 2013

Open Annotation Specs
Christian Morbidoni
12 annotations
mon aug 19 2013

WAB: Christian
Christian Morbidoni
16 annotations
fri dec 06 2013

Notebook 2013-12-01 12:54:35
Christiane Waldau
Empty :(
sun dec 01 2013

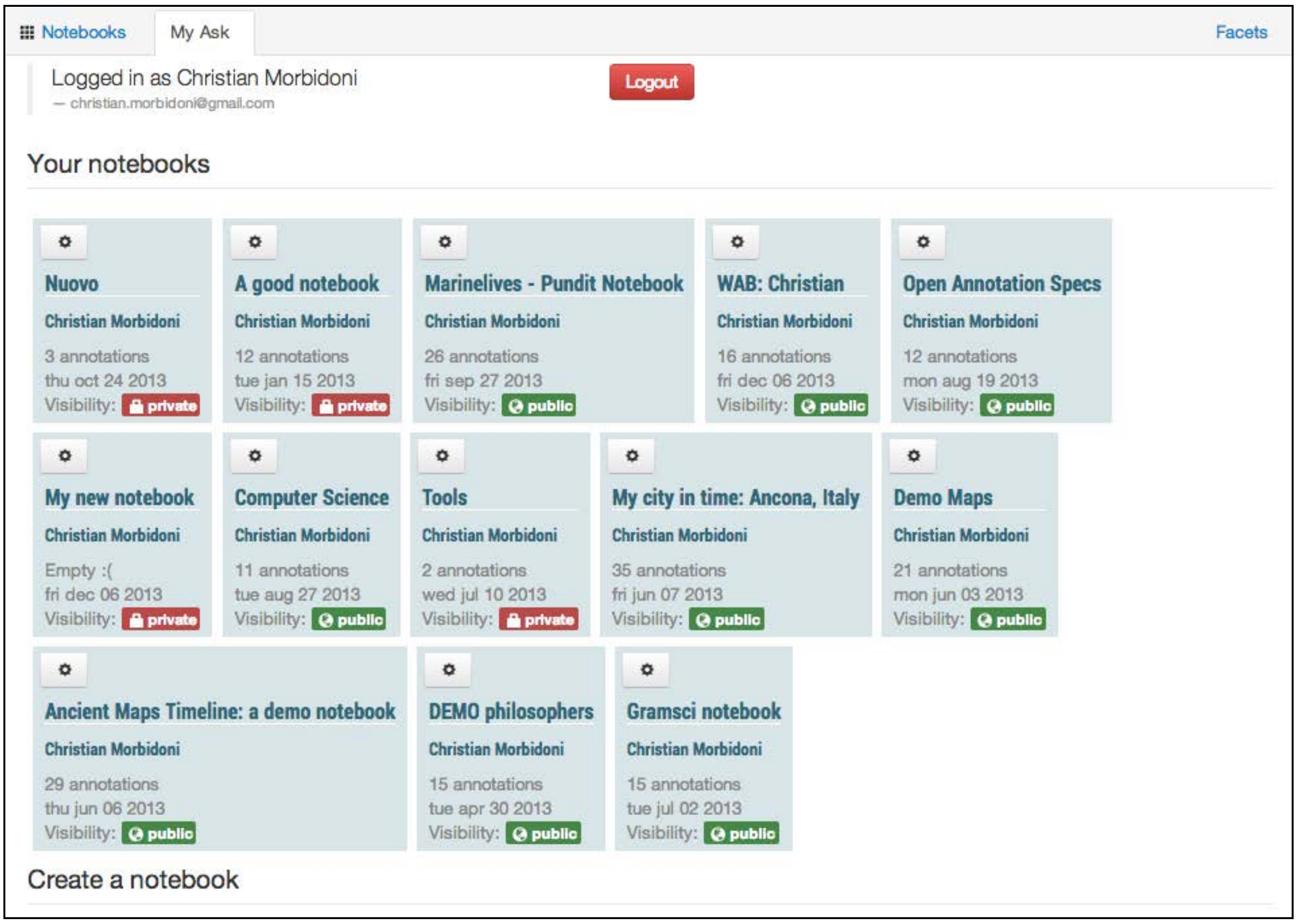
Marinelives - Pundit Notebook
Christian Morbidoni
26 annotations
fri sep 27 2013

WAB: Christian Erbacher
kag gerc
12 annotations
wed sep 18 2013

Figure 7. Searching public notebooks in Ask.

2.2.2 Managing personal notebooks

The MyAsk tab shows, upon login, only personal notebooks and allows to set them as public or private, to delete them and to create new ones. This is shown in Figure 8.


Notebooks | My Ask | Facets

Logged in as Christian Morbidoni
— christian.morbidoni@gmail.com

Logout

Your notebooks

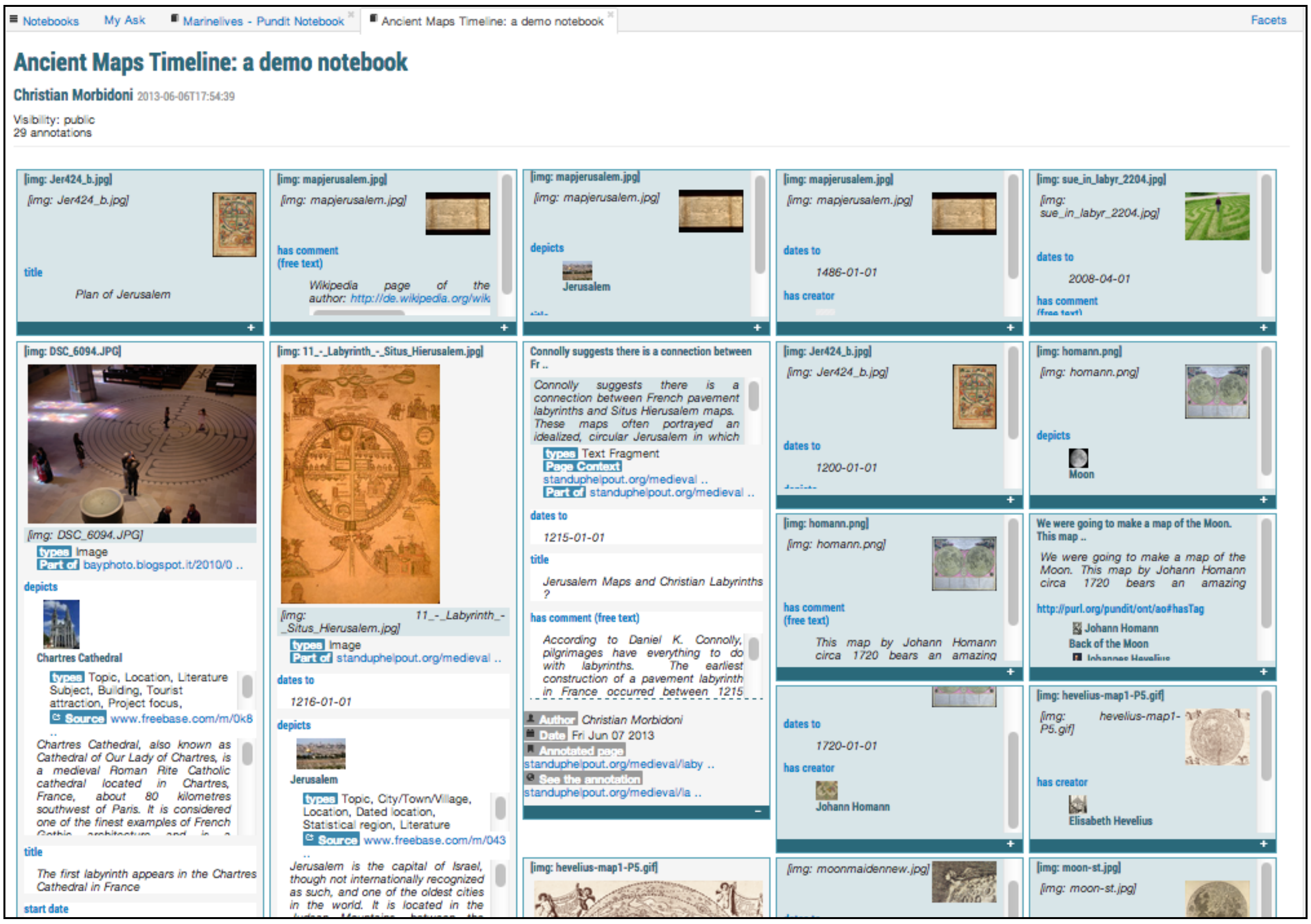
Notebook	Author	Annotations	Date	Visibility
Nuovo	Christian Morbidoni	3 annotations	thu oct 24 2013	private
A good notebook	Christian Morbidoni	12 annotations	tue jan 15 2013	private
Marinelives - Pundit Notebook	Christian Morbidoni	26 annotations	fri sep 27 2013	public
WAB: Christian	Christian Morbidoni	16 annotations	fri dec 06 2013	public
Open Annotation Specs	Christian Morbidoni	12 annotations	mon aug 19 2013	public
My new notebook	Christian Morbidoni	Empty :(	fri dec 06 2013	private
Computer Science	Christian Morbidoni	11 annotations	tue aug 27 2013	public
Tools	Christian Morbidoni	2 annotations	wed jul 10 2013	private
My city in time: Ancona, Italy	Christian Morbidoni	35 annotations	fri jun 07 2013	public
Demo Maps	Christian Morbidoni	21 annotations	mon jun 03 2013	public
Ancient Maps Timeline: a demo notebook	Christian Morbidoni	29 annotations	thu jun 06 2013	public
DEMO philosophers	Christian Morbidoni	15 annotations	tue apr 30 2013	public
Gramsci notebook	Christian Morbidoni	15 annotations	tue jul 02 2013	public

Create a notebook

Figure 8. Managing personal notebooks in Ask.

2.2.3 Viewing notebooks content

By clicking a notebook users can see their content, which consists of annotations, possibly made at different times and on different web pages. They are visualised by default as a number of “boxes”, as shown in Figure 9. If expanded, a box shows all the triples included in an annotation, providing links to external resources if mentioned. Clicking the “see annotation” link shows the annotation in its original context on a separate page, where Pundit can be used to inspect the annotation or to annotate further.



Notebooks My Ask Marinelives - Pundit Notebook Ancient Maps Timeline: a demo notebook Facets

Ancient Maps Timeline: a demo notebook

Christian Morbidoni 2013-06-06T17:54:39

Visibility: public
29 annotations

[img: Jer424_b.jpg]
[img: Jer424_b.jpg]
title
Plan of Jerusalem

[img: mapjerusalem.jpg]
[img: mapjerusalem.jpg]
has comment (free text)
Wikipedia page of the author: http://de.wikipedia.org/wik

[img: mapjerusalem.jpg]
[img: mapjerusalem.jpg]
depicts
Jerusalem

[img: mapjerusalem.jpg]
[img: mapjerusalem.jpg]
dates to
1486-01-01
has creator

[img: sue_in_labyr_2204.jpg]
[img: sue_in_labyr_2204.jpg]
dates to
2008-04-01
has comment (free text)

[img: DSC_6094.JPG]
[img: DSC_6094.JPG]
types Image
Part of bayphoto.blogspot.it/2010/0 ..
depicts
Chartres Cathedral
types Topic, Location, Literature Subject, Building, Tourist attraction, Project focus, ..
Source www.freebase.com/m/0k8
Chartres Cathedral, also known as Cathedral of Our Lady of Chartres, is a medieval Roman Rite Catholic cathedral located in Chartres, France, about 80 kilometres southwest of Paris. It is considered one of the finest examples of French
title
The first labyrinth appears in the Chartres Cathedral in France
start date

[img: 11_-_Labyrinth_-_Situs_Hierusalem.jpg]
[img: 11_-_Labyrinth_-_Situs_Hierusalem.jpg]
types Image
Part of standuphelpout.org/medieval ..
dates to
1216-01-01
depicts
Jerusalem
types Topic, City/Town/Village, Location, Dated location, Statistical region, Literature ..
Source www.freebase.com/m/043
Jerusalem is the capital of Israel, though not internationally recognized as such, and one of the oldest cities in the world. It is located in the

Connolly suggests there is a connection between Fr ..
Connolly suggests there is a connection between French pavement labyrinths and Situs Hierusalem maps. These maps often portrayed an idealized, circular Jerusalem in which
types Text Fragment
Page Context standuphelpout.org/medieval ..
Part of standuphelpout.org/medieval ..
dates to
1215-01-01
title
Jerusalem Maps and Christian Labyrinths ?
has comment (free text)
According to Daniel K. Connolly, pilgrimages have everything to do with labyrinths. The earliest construction of a pavement labyrinth in France occurred between 1215
Author Christian Morbidoni
Date Fri Jun 07 2013
Annotated page standuphelpout.org/medieval/laby ..
See the annotation standuphelpout.org/medieval/la ..

[img: hevelius-map1-P5.gif]

[img: Jer424_b.jpg]
[img: Jer424_b.jpg]
dates to
1200-01-01

[img: homann.png]
[img: homann.png]
has comment (free text)
This map by Johann Homann circa 1720 bears an amazing

dates to
1720-01-01
has creator
Johann Homann

[img: moonmaidennew.jpg]

[img: homann.png]
[img: homann.png]
depicts
Moon

We were going to make a map of the Moon. This map ..
We were going to make a map of the Moon. This map by Johann Homann circa 1720 bears an amazing
http://purl.org/pundit/ont/ao#hasTag
Johann Homann
Back of the Moon

[img: hevelius-map1-P5.gif]
[img: hevelius-map1-P5.gif]
has creator
Elisabeth Hevelius

[img: moon-st.jpg]
[img: moon-st.jpg]

Figure 9. The content of a notebook in Ask. Small boxes represent annotations on some image or text found on the web. Expanding an annotation reveals comments, metadata and relations that this object has with others. Clicking the “see annotation” link, shows the annotation in its original context on a separate page.

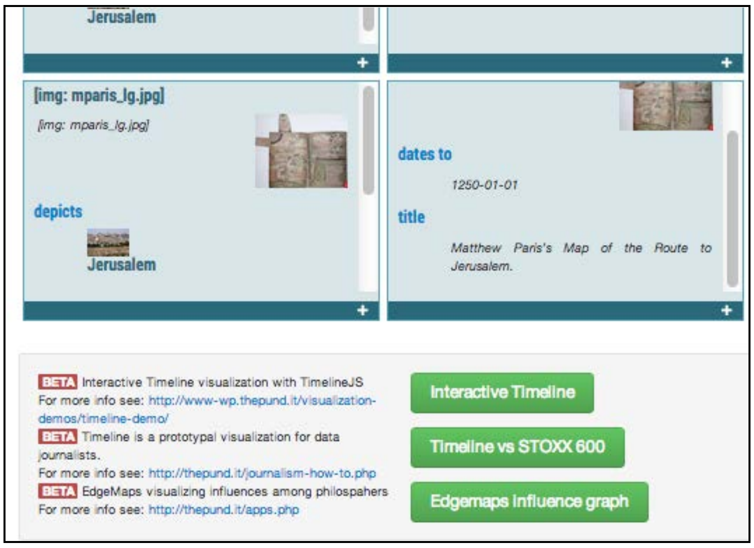
2.2.4 Vertical visualisations

As one can see, Ask provides a very generic and domain independent view on annotations.

However, the advantage of having annotations as structured (in RDF), platform independent and dereferenceable URLs, is that the very same data can be consumed by separate applications to provide e.g. specialised visualisation of annotations.

In Ask, it is relatively easy to plug-in external applications to deliver alternative data visualisation. We did this for some of the demonstrative applications presented in the learning material. The integration of such applications is very simple and the only requirement is that of providing a REST API that takes as input the ID of the notebook to be visualised.

We did this for the Timeline demo (discussed in the User Tutorial 1) and the Edgemaps demo (User Tutorial 2). As shown in Figure 10, green buttons perform external applications to visualize the notebook: if the required triples are contained in the annotations, a Timeline or an Edgemap will be populated.



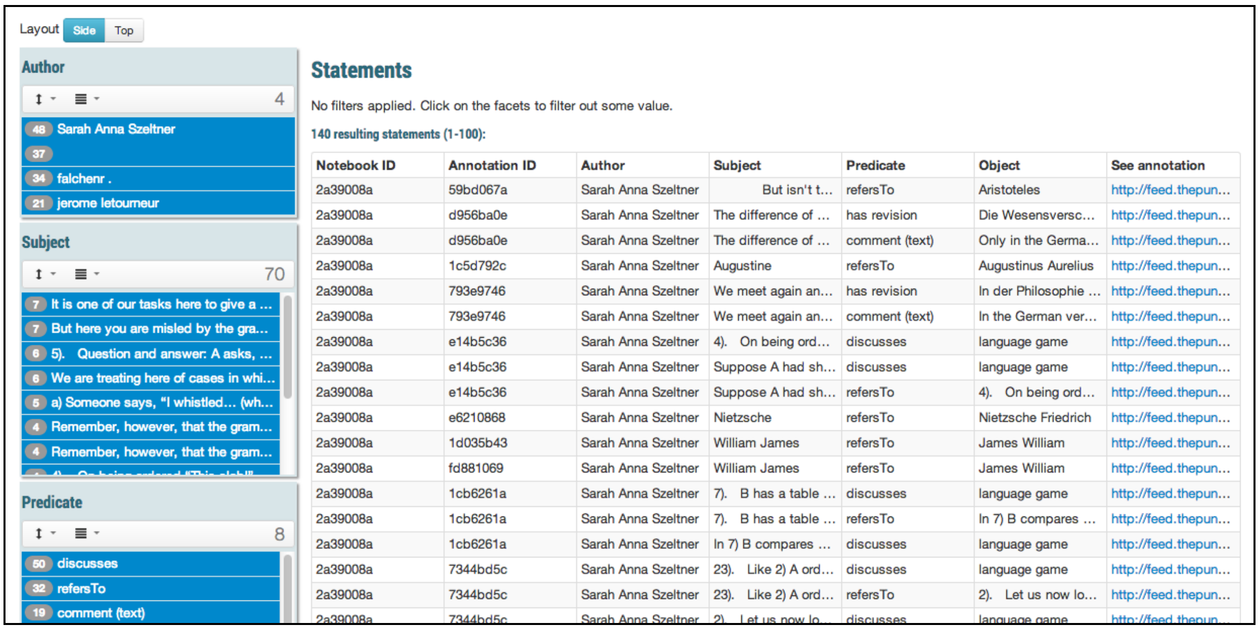
BETA Interactive Timeline visualization with TimelineJS
For more info see: http://www-wp.thepund.it/visualization-demos/timeline-demo/
BETA Timeline is a prototypal visualization for data journalists.
For more info see: http://thepund.it/journalism-how-to.php
BETA EdgeMaps visualizing influences among philospahers
For more info see: http://thepund.it/apps.php

Figure 10. Links to vertical visualisations in Ask.

2.2.5 The notebooks faceted browser

Ask is currently subject to intense development, and one of the most interesting recent features is the ***notebooks faceted browser***. This is a very powerful feature for analysing a corpus of annotations.

The notebooks faceted browser is a way of exploring the whole graph composed by all the annotations contained in a set of notebooks. A user can select (by opening them in Ask) a set of public notebooks he finds of interest, then possibly adding his own private ones (again, by simply opening them) and then, by clicking the “Facets” link at top-right, he can then explore all the opened notebooks as shown in Figure 11.


Layout
Side
Top
Author
4
48 Sarah Anna Szeltner
37
34 falchenr .
21 jerome letourneur
Subject
70
7 It is one of our tasks here to give a ...
7 But here you are misled by the gra...
6 5). Question and answer: A asks, ...
6 We are treating here of cases in whi...
5 a) Someone says, "I whistled... (wh...
4 Remember, however, that the gram...
4 Remember, however, that the gram...
Predicate
8
50 discusses
32 refersTo
19 comment (text)

Statements

No filters applied. Click on the facets to filter out some value.

140 resulting statements (1-100):

Notebook ID	Annotation ID	Author	Subject	Predicate	Object	See annotation
2a39008a	59bd067a	Sarah Anna Szeltner	But isn't t...	refersTo	Aristoteles	http://feed.thepun...
2a39008a	d956ba0e	Sarah Anna Szeltner	The difference of ...	has revision	Die Wesensversc...	http://feed.thepun...
2a39008a	d956ba0e	Sarah Anna Szeltner	The difference of ...	comment (text)	Only in the Germa...	http://feed.thepun...
2a39008a	1c5d792c	Sarah Anna Szeltner	Augustine	refersTo	Augustinus Aurelius	http://feed.thepun...
2a39008a	793e9746	Sarah Anna Szeltner	We meet again an...	has revision	In der Philosophie ...	http://feed.thepun...
2a39008a	793e9746	Sarah Anna Szeltner	We meet again an...	comment (text)	In the German ver...	http://feed.thepun...
2a39008a	e14b5c36	Sarah Anna Szeltner	4). On being ord...	discusses	language game	http://feed.thepun...
2a39008a	e14b5c36	Sarah Anna Szeltner	Suppose A had sh...	discusses	language game	http://feed.thepun...
2a39008a	e14b5c36	Sarah Anna Szeltner	Suppose A had sh...	refersTo	4). On being ord...	http://feed.thepun...
2a39008a	e6210868	Sarah Anna Szeltner	Nietzsche	refersTo	Nietzsche Friedrich	http://feed.thepun...
2a39008a	1d035b43	Sarah Anna Szeltner	William James	refersTo	James William	http://feed.thepun...
2a39008a	fd881069	Sarah Anna Szeltner	William James	refersTo	James William	http://feed.thepun...
2a39008a	1cb6261a	Sarah Anna Szeltner	7). B has a table ...	discusses	language game	http://feed.thepun...
2a39008a	1cb6261a	Sarah Anna Szeltner	7). B has a table ...	refersTo	In 7) B compares ...	http://feed.thepun...
2a39008a	1cb6261a	Sarah Anna Szeltner	In 7) B compares ...	discusses	language game	http://feed.thepun...
2a39008a	7344bd5c	Sarah Anna Szeltner	23). Like 2) A ord...	discusses	language game	http://feed.thepun...
2a39008a	7344bd5c	Sarah Anna Szeltner	23). Like 2) A ord...	refersTo	2). Let us now lo...	http://feed.thepun...
2a39008a	7344bd5c	Sarah Anna Szeltner	2). Let us now lo...	discusses	language game	http://feed.thepun...

Figure 11. Notebooks Faceted browser in Ask.

Facets include authors of annotations, annotated page, relations used in triples and objects of the triples. By filtering results users can discover, for example, all annotated texts where a person (e.g. Aristotle) is mentioned, or pictures where a place is depicted, or where an “agrees-with” relation has been used to link to the texts. It is important to notice that data is non-anonymised, which means that once an information is discovered, it is always possible to understand who said what.

2.3 Using Pundit as-a-service: Feed

Pundit can be deployed in different ways. The most straightforward way is that of including it as a Javascript library to make the content of a digital archive annotatable. However, in most cases the annotation environment is something that might be better to handle as a separate application, rather than including it directly in web sites. This includes cases where we want users to be able to annotate several web sites where we do not have control (e.g. Wikipedia pages, web journals, etc.).

In other cases it can be useful to provide the annotation environment as a service. This means that calling an HTTP API it should be possible to push some content into the annotation environment and choose a configuration (e.g. domain vocabularies) to be used in Pundit.

This approach is implemented in Feed (http://feed.thepund.it).

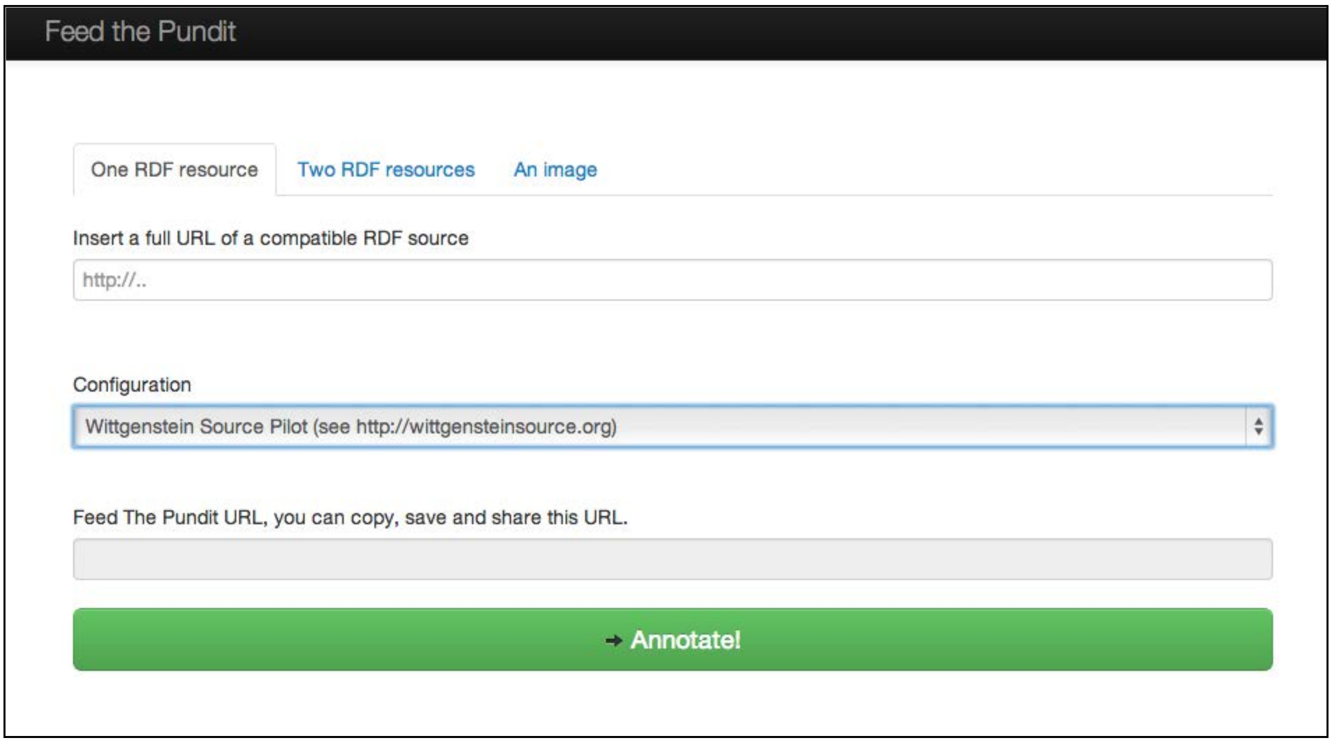
Feed API for DM2E data is available at:

*http://feed.thepund.it/?dm2e={**DM2E_Linked_Data_URL**}&conf={**Pundit_Configuration**}*

Where *DM2E_Linked_Data_URL* is the dereferenceable URL of some resource represented in RDF in conformance with the DM2E data model (http://onto.dm2e.eu/), and *Pundit_Configuration* is one of the preloaded configurations of Pundit.

When the API is called, Feed gets the RDF representation of the resource and parses it to extract relevant metadata (e.g. authorship, dates and, in general, connections with other resources that might be of interest to the user to display). Most importantly, it grabs the content to be annotated (e.g. an image, a text of simple HTML) from the actual content provider. This is done “on the fly” via a special RDF property defined in the DM2E data model (dm2e:hasAnnotatableVersionAt).

In addition, Feed has a simple web user interface to access the API, shown in Figure 12.



Feed the Pundit

Insert a full URL of a compatible RDF source

http://..

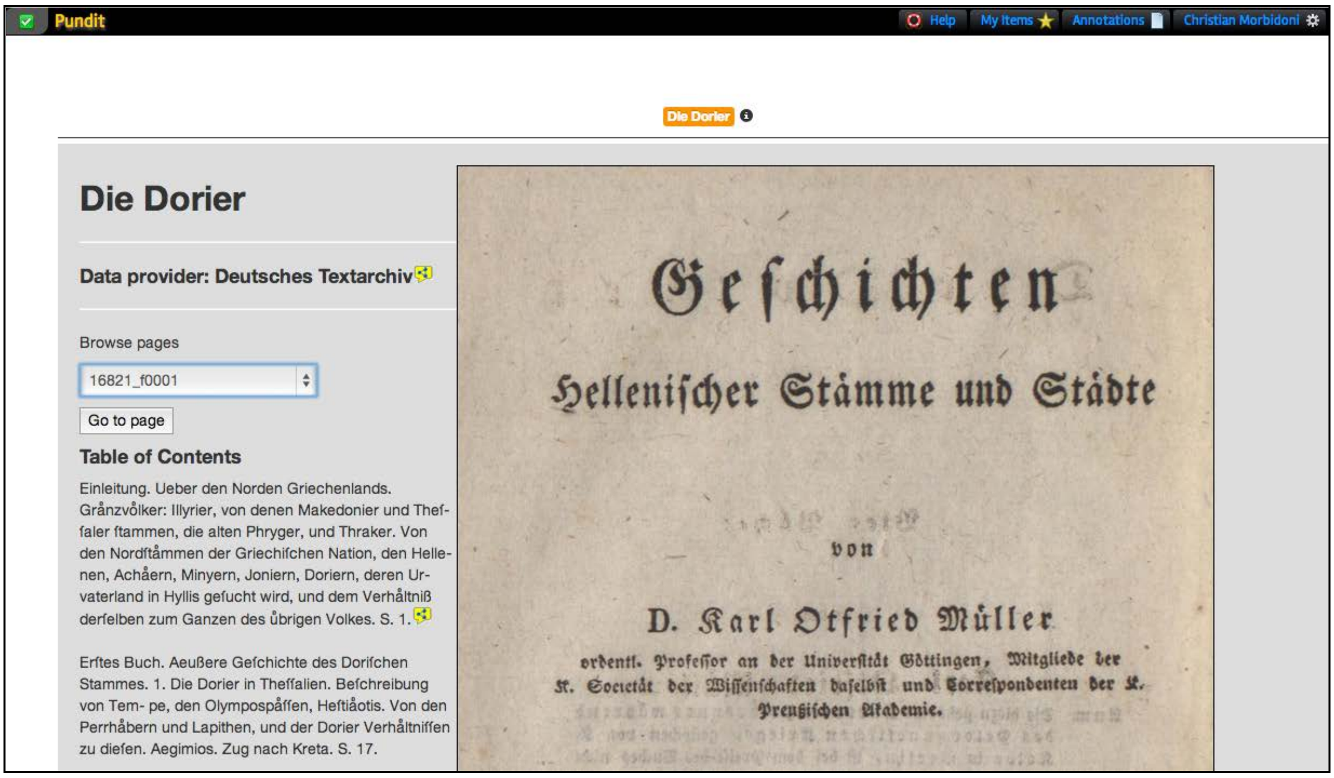
Configuration

Wittgenstein Source Pilot (see http://wittgensteinsource.org)

Feed The Pundit URL, you can copy, save and share this URL.

Figure 12. The simple web UI of Feed.

Feed can be extended to support specific RDF vocabularies, as it happens for supporting the DM2E data model. This allows the user to handle different kind of resources in different ways. For example Manuscripts in DM2E are displayed with a simple navigation box that allows users to reach specific pages in the manuscript (Figure 13).



Die Dorier

Data provider: Deutsches Textarchiv

Browse pages

16821_f0001

Table of Contents

Einleitung. Ueber den Norden Griechenlands. Grånzvőlker: Illyrier, von denen Makedonier und Theſ-ſaler ſtammen, die alten Phryger, und Thraker. Von den Nordſtåmmen der Griechiſchen Nation, den Helle-nen, Achåern, Minyern, Joniern, Doriern, deren Ur-vaterland in Hyllis geſucht wird, und dem Verhåltniß derſelben zum Ganzen des ůbrigen Volkes. S. 1.

Erſtes Buch. Aeußere Geſchichte des Doriſchen Stammes. 1. Die Dorier in Theſſalien. Beſchreibung von Tem- pe, den Olympospåſſen, Heſtiåotis. Von den Perrhåbern und Lapithen, und der Dorier Verhåltniſſen zu dieſen. Aegimios. Zug nach Kreta. S. 17.

Geſchichten

Helleniſcher Stämme und Städte

von

D. Karl Otfried Müller

ordentl. Profeſſor an der Univerſität Göttingen, Mitgliede der K. Societät der Wiſſenſchaften daſelbſt und Correſpondenten der K. Preußiſchen Akademie.

Figure 13. Feed showing a DM2E manuscript ready to be annotated with Pundit.

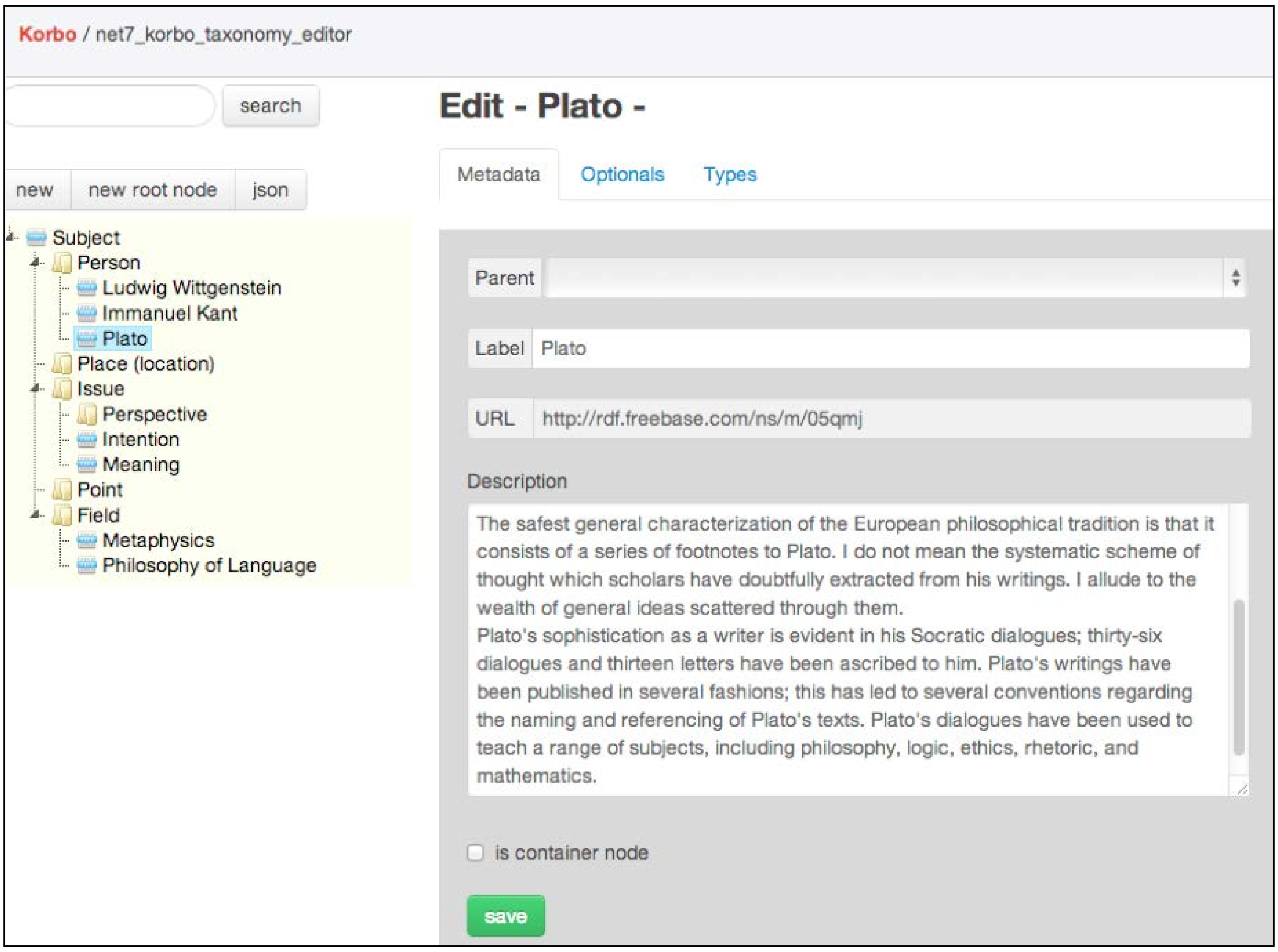
Details about feed and how DM2E data is handled can be found in Demonstrative Application 3 attached to this deliverable (*Attachment A3*).

2.4 Korbo and Taxonomies

Korbo (http://korbo.org) is a PHP component to manage vocabularies and taxonomies.

The component is in alpha, currently under a major code revision phase. Documentation and a working demo instance is available on the web site at http://manager.korbo.org/.

In Korbo users can create so-called Baskets, which can then be populated via a web interface (Figure 14). Users can create "folders" (or container nodes) just by assigning labels, and populating them with entities from Freebase.com or by creating their own new entities.



Edit - Plato -

Parent

Label Plato

URL http://rdf.freebase.com/ns/m/05qmj

Description

The safest general characterization of the European philosophical tradition is that it consists of a series of footnotes to Plato. I do not mean the systematic scheme of thought which scholars have doubtfully extracted from his writings. I allude to the wealth of general ideas scattered through them.
Plato's sophistication as a writer is evident in his Socratic dialogues; thirty-six dialogues and thirteen letters have been ascribed to him. Plato's writings have been published in several fashions; this has led to several conventions regarding the naming and referencing of Plato's texts. Plato's dialogues have been used to teach a range of subjects, including philosophy, logic, ethics, rhetoric, and mathematics.

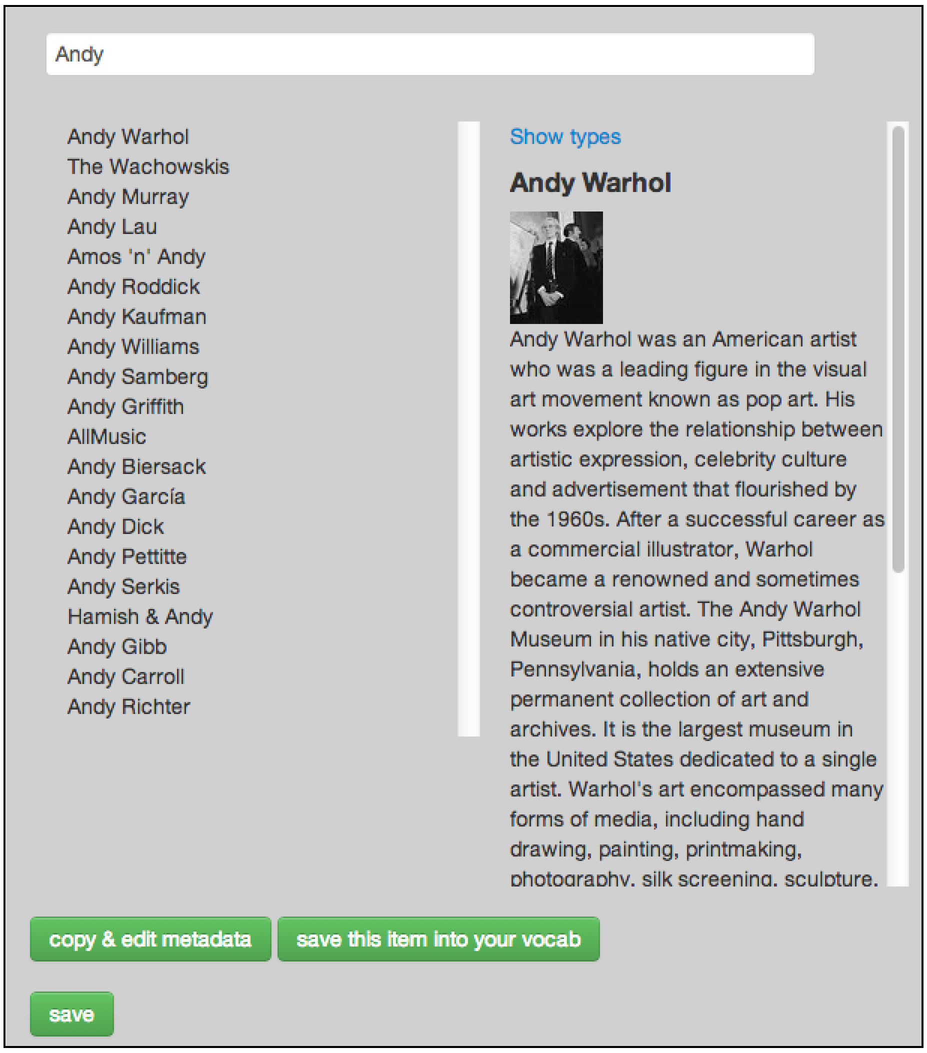
☐ is container node

Figure 14. The Korbo Taxonomy editing GUI.

As shown in Figure 15, users can search for entities, a person in this case, and find relevant matching on Freebase. If a good match is found, users can choose among two options:

1. Make a copy of the Freebase entity, which means creating a new personal one with a new URI, and being allowed to edit its metadata at will;

2. Use the actual entity from Freebase (with its original URI).


Andy

Andy Warhol
The Wachowskis
Andy Murray
Andy Lau
Amos 'n' Andy
Andy Roddick
Andy Kaufman
Andy Williams
Andy Samberg
Andy Griffith
AllMusic
Andy Biersack
Andy García
Andy Dick
Andy Pettitte
Andy Serkis
Hamish & Andy
Andy Gibb
Andy Carroll
Andy Richter

Show types

Andy Warhol

Andy Warhol was an American artist who was a leading figure in the visual art movement known as pop art. His works explore the relationship between artistic expression, celebrity culture and advertisement that flourished by the 1960s. After a successful career as a commercial illustrator, Warhol became a renowned and sometimes controversial artist. The Andy Warhol Museum in his native city, Pittsburgh, Pennsylvania, holds an extensive permanent collection of art and archives. It is the largest museum in the United States dedicated to a single artist. Warhol's art encompassed many forms of media, including hand drawing, painting, printmaking, photography, silk screening, sculpture,

Figure 15. Searching Freebase and copying or including entities.

Although Korbo was designed from the beginning to work with Pundit, it evolved as a stand alone application and a generic taxonomy management tool.

However, using it in combination with Pundit is very easy. Korbo provides REST APIs to access Baskets and supports the Pundit vocabulary JSON format. This means it is possible to create Entities Taxonomies and Relations Sets in Korbo and then include them in Pundit by simply editing its configuration file, adding one line like the following:

```
vocabularies: [
        "http://korbo.netseven.it/84?jsonp=_PUNDIT.vocab.initJsonpVocab"
    ]
```

On the other hand, Pundit has a special plugin for Korbo, so that, once a Basket is available in Korbo, it can be easily connected to Pundit as a “Selector”. This means that Pundit will use Korbo search APIs to let users search a Basket and use the resulting entities in semantic annotations.

The following code shows the Pundit configuration that installs a Korbo Basket as a Selector. In the case of public Baskets, the only information needed is the Basket id.

```
'selectors': {
            'KorboBasket': {
                name: 'korbo',
                label: 'Korbo search',
                active: true,
                baskets: [82]
            }
        }
```

3 Developing on top of the platform: demonstrative applications

A number of demonstrative applications and prototypes have been developed within the DM2E project. In this section, we include 4 practical demonstrations targeted to developers, where we describe and discuss implementation and design decisions. In these software prototypes, each of which is available as a working online demo and source code, we provide guidelines to build on top of the DM2E tools (WP3) and of DM2E Linked Data (WP1-WP2).

Four demonstrative applications are provided as attachments to the present deliverable, as well as published on the project wiki.

List of demonstrative application attached:

A1 - Demonstrative application 1 - *Edgemap annotation viewer*

A2 - Demonstrative application 2 - *Timeline annotation viewer*

A3 - Demonstrative application 3 - *Consuming DM2E data in Feed*

A4 - Demonstrative application 4 - *A Solr based faceted search over DM2E LD*

4 Tutorials

Five tutorials targeted at scholars and users of the platform have been created and refined during the project. Some of these tutorials were used to conduct experiments and surveys reported in D 1.4 of the DM2E project, such as the Wittgenstein Brown Book experiment. Others was used at conferences and workshops, or in collaboration with other projects as in the case of Burckhardtsource.org, where Pundit was adopted as the annotation tool and in the MarineLive Pundit bookmaklet (http://thepund.it/bm/marinelives/), used to explore the functionalities of the tools in the Marinelives project.

The tutorials are attached to the present deliverable:

T1 - Pundit Tutorial number 1 - *Creating annotations with Pundit*

T2 - Pundit Tutorial number 2 - *Deploying the annotation environment*

T3 - Pundit Tutorial number 3- *Sharing/discovering/visualising annotations in Ask*

T4 - Pundit Tutorial number 4 - *Timeline Demo: building an interactive timeline with annotations*

T5 - Pundit Tutorial number 5 - *Edgemaps visualisation with Pundit annotations*

Translations of each tutorial are available in Italian, German and Norwegian in the DM2E Wiki http://wiki.dm2e.eu/.

4.1 Other resources

- Some Pundit bookmarklets created for different demos: http://www.thepund.it/bm/
- Philosophers influence edgemep - Draft Guide: http://www.thepund.it/visualization-demos/philosophers-demo-howto/
- Simple interactive timeline - Draft Guide: http://www.thepund.it/visualization-demos/timeline-demo/
- Journalism example - online demo: http://ask.thepund.it/?#/timeline/31951d93/20120927
- Burckhardt correspondence - online prototypes
  - Timeline navigation of annotated letters
    http://metasound.dibet.univpm.it/timelinejs/examples/bur-bode.html
  - Edgemaps based visualisation of citations and persons in the letters
    http://metasound.dibet.univpm.it/edgemaps/maps/test.html#paint;map;

5 Demonstrative videos

Three videos were created during the second year of the project.

Introduction to Pundit (http://vimeo.com/85261745). This video introduces the concept of semantic web annotation and presents Pundit at a sophisticated level.

DM2E Pundit and the Early Days of Linked Open Data (http://vimeo.com/85517504)**.** This video was produced for the LODLAM 1013 Challenge and won the first prize (http://dm2e.eu/pundit-winner-at-the-lodlam-competition/).

Pundit Screencast (http://vimeo.com/85732656). Shows Pundit in action, illustrating annotation editing and sharing with the DM2E tools.

6 Conclusion and Next Steps

Pundit, Ask and Feed are under constant development and are used outside the boundaries of the DM2E project. We expect that new features and new demonstrative prototypes will appear in the coming months. These will be documented in the Pundit web site, as well as posted on the DM2E project wiki.

With respect to DM2E, in the next stage of the project the effort of the Work Package 3 Task 3.3 will be directed toward improving integration with the DM2E Linked Data and the user experience, and to set up demonstrative applications to support Task 3.4, where the scholarly domain model will be applied and evaluated by deploying customised versions of the tools and delivering them to users.

7 A1 - Edgemap annotation viewer

7.1 Introduction

In this document we discuss a demonstrative application built on top of the Pundit server REST API. The demo is meant as an example to demonstrate how public semantic annotations made with Pundit can be consumed by developers and used to drive specialised visualisations. In this case our goal was to create dynamically an Edgemap (http://mariandoerk.de/edgemaps/) that tracks influences among philosophers, reflecting the annotations contained in one or more public notebooks.

This document is targeted at developers and discusses the prototype from a technical view point. Before going through this document we recommend to users to have a look at the tutorial titled *"Timeline Demo: building an interactive timeline with annotations"*, which presents the prototype from an end-user perspective, demonstrating its usage.

To see the resulting demonstrative application, please go to

http://thepund.it/edgemaps_demo/demo.html?nbs={6bdcd4a5}&source={pundit}#phils;time;;/en/plato;

where the content of a sample notebook is visualised as an Edgemap. By clicking on one circle (representing one philosopher) you can see all the influence relations that the philosopher has with other philosophers.

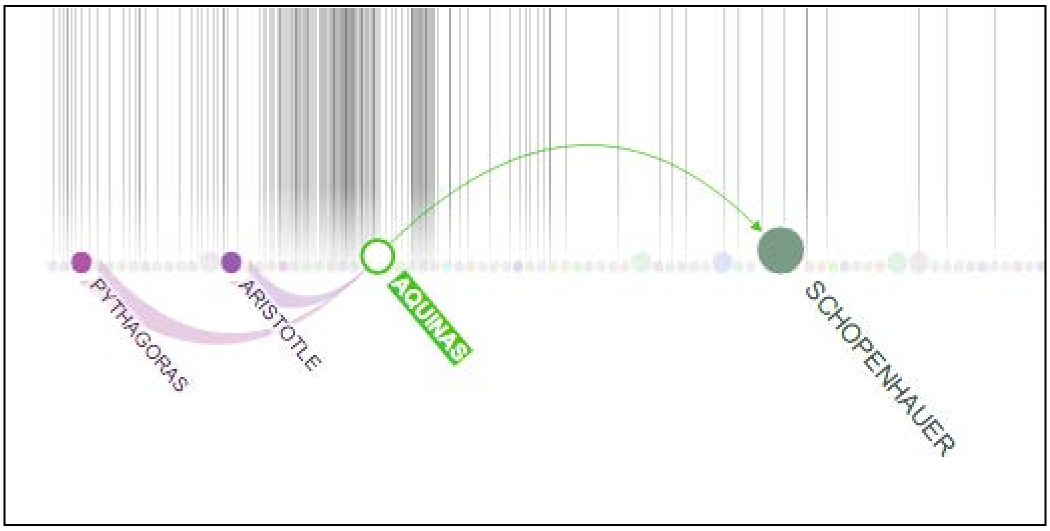


Figure 16. Two kinds of edges are used in this Edgemap. The dotted one is used e.g. to show that Aristotle influenced Aquinas, the thin one is used to state that e.g. Schopenhauer was influenced by Aquinas.

Both Pundit client and server are open-source and available at GitHub:

- Pundit client, https://github.com/net7/pundit
- Pundit server, https://github.com/net7/pundit-server

The full source code of the Pundit Edgemaps demonstrative application is available at https://github.com/chrmor/Pundit-Edgempas-demo.

7.2 Customising and deploying Pundit

The first thing we did was to create a customised instance of Pundit to enable users to produce semantic annotations with a precise structure.

We want our users to be able to mark “citations” in the text. This means being able to connect two segments of texts, from two works by different authors, specifying that one cites the other. The idea is then to transform each annotation of this kind into an edge connecting the two authors.

Furthermore, annotations will have to contain information about the authors of the two annotated texts. In a real world setting it would be possible to pre-load such information or derive it from available open dataset (such as Freebase or DBpedia). However, in this demonstration we decided for simplicity’s sake to leave to the annotator the task of creating this information.

7.3 Annotations and annotation vocabulary

We wanted to enable users to create annotations of two kinds:

1. *Annotations connecting a text to an author.*
   These annotations include triples like the following:

   text - has author - author_1
   text - cites - author_1

2. *Annotations connecting a text to another text from a different author.*
   These include the following triples:

   text_1 - has author - author_1
   text_2 - has author - author_2
   text_1 - cites - text_2

In order to make such annotations possible we created a simple Relation Set including the “cites” RDF property defined in the CiTO ontology, available at http://goo.gl/dFHJVw.

Reusing available ontologies is generally seen as good practice, but it is not always possible. In some cases users will have to create their own RDF ontology, possibly containing few properties.

The Relations Set we created in JSON file is shown in the following code.

```
_PUNDIT.vocab.initJsonpVocab({
    "error_code": "200",
    "error_message": "OK",
    "result":
    {
        "vocab_label": "DEMO Philosophers relations",
        "vocab_id": "demophrel01",
        "vocab_type": "predicates",
        "items": [
            {
                "type": ["predicate"],
                "rdftype": ["http://www.w3.org/1999/02/22-rdf-syntax-ns#Property"],
                "label": "has comment (free text)",
                "description": "Any comment related to the selected fragment of text or
image",
                "domain": ["http://purl.org/pundit/ont/ao#fragment-image",
"http://purl.org/pundit/ont/ao#fragment-text", "http://xmlns.com/foaf/0.1/Image"],
                "range": ["http://www.w3.org/2000/01/rdf-schema#Literal"],
                "value": "http://schema.org/comment"
            },
            {
                "type": ["predicate"],
                "rdftype": ["http://www.w3.org/1999/02/22-rdf-syntax-ns#Property"],
                "label": "has creator",
                "description": "The selected text fragment has been created by a specific
Person",
                "domain": ["http://purl.org/pundit/ont/ao#fragment-text",
"http://purl.org/pundit/ont/ao#fragment-image", "http://xmlns.com/foaf/0.1/Image"],
                "range": ["http://www.freebase.com/schema/people/person",
"http://xmlns.com/foaf/0.1/Person"],
                "value": "http://purl.org/dc/terms/creator"
            },
            {
                "type": ["predicate"],
                "rdftype": ["http://www.w3.org/1999/02/22-rdf-syntax-ns#Property"],
                "label": "cites",
                "description": "The selected text fragment cites another text fragment,
or a Work or a Person",
                "domain": ["http://purl.org/pundit/ont/ao#fragment-text"],
                "range": [
                    "http://purl.org/pundit/ont/ao#fragment-text",
                    "http://www.freebase.com/schema/people/person",
                    "http://xmlns.com/foaf/0.1/Person",
                    "http://www.freebase.com/schema/book/written_work",
                    "http://www.freebase.com/schema/book/book"
                ],
                "value": "http://purl.org/spar/cito/cites"
            }
        ]
    }
});
```

We then created a simple Entities Taxonomy with a single class of items (Philosophers) and we populated it automatically by parsing the data already available in the Edgemap. Edgemaps uses a JSON format to represent data. An example of the original data format can be found at http://mariandoerk.de/edgemaps/demo/philosophers.js.

The resulting Pundit JSON vocabulary contains items like the following:

```
{ "value": "http://rdf.freebase.com/en.immanuel_kant",
  "label": "Immanuel Kant",
  "rdftype": ["http://www.freebase.com/schema/people/person",
             "http://www.freebase.com/common/topic",
             "http://www.freebase.com/book/author"],
  "description": "Immanuel Kant (IPA: [ɪˈmanuɛl kant]; 22 April 1724 –12 February
       1804) was an 18th-century German philosopherfrom the Prussian city of
       Königsberg (now Kaliningrad,Russia). He is regarded as one of the most
       influentialthinkers of modern Europe and of the late Enlightenment.Kant created
       a new widespread perspective in philosophywhich influenced philosophy through
       the 21st Century. Healso published important works of epistemology, as well
       asworks relevant to religion, law, and history....",
  "nodetype": "node",
  "image":
       "https://usercontent.googleapis.com/freebase/v1/image/guid/9202a8c04000641f8000000
004a6d063",
       "children": []
}
```

As you can see each item represents a philosopher and its URI and metadata are extracted from Freebase.com. See the entire JSON file at https://dl.dropboxusercontent.com/u/769042/pundit-demo-philosophers.json.

7.4 Configuring Pundit

Once our Pundit vocabularies were online and reachable at a stable URL, we created a new Pundit configuration to include them by simply adding the following lines:

```
vocabularies: [
"https://dl.dropboxusercontent.com/u/769042/pundit-demo-philosophers.json",
"https://dl.dropboxusercontent.com/u/769042/pundit-demo-philosophers-predicates.json”]
```

In addition to configuring the vocabularies, the Pundit configuration allows to tune in other features of the annotation UI. The following is the configuration (with comments) that we used for this demo.

```
punditConfig = {
  // Set debug to true to see debugging messages in your browser javascript consolle.
  debugAllModules: false,
  // The url of the Pundit server that will be used to store and retrieve annotations
  annotationServerBaseURL : 'http://as.thepund.it:8080/annotationserver/',
  // The vocabularies (Relations Sets and Entities Taxnonomies) to be included
  vocabularies: [
    "https://dl.dropboxusercontent.com/u/769042/pundit-demo-philosophers.json",
"https://dl.dropboxusercontent.com/u/769042/pundit-demo-philosophers-predicates.json"
  ],
  // Set this one to “true” if you want to include the default Relations Set
  useBasicRelations: false,
  // The modules parameter allows you to turn on/off and configure specific UI modules
  modules: {
    // With this module you can provide to your users an introduction or help page
    'pundit.Help': { introductionFile:
                    'http://thepund.it/bm/demo-philosophers/demo-introduction.html',
                introductionWindowTitle: 'Welcome to Pundit Philosopher DEMO! :)',
                showIntroductionAtLogin: true
    },
  // The following configures the contact form, where users can enter feedback or feature
requests, etc. The “list” parameter is an identifier for a set of mail addresses to which
the user messages will be forwarded. It can be configured from the Pundit server admin
panel.
  'pundit.ContactHelper': {
   instances: [{
     title: 'Contact us!',
     comment: 'Drop us a line, give us some feedback!',
     list: 'demophilosophers'
   }]},
   // The notebook manager allows users to see and edit notebooks properties from Pundit,
as well as to create new notebooks
   'pundit.NotebookManager': {
     active: true,
     notebookSharing: false,
     askBaseURL: 'http://ask.as.thepund.it/#/myNotebooks/'},
   // Turn on/off the image fragments annotation
   'pundit.ImageFragmentHandler': { active: false },
   'pundit.ImageAnnotationPanel': { active: false },
   // Turn on/off annotation of the full page
   'pundit.PageHandler': { active: false },
   // Turn on/off the automatic entity extraction feature
   'pundit.Recognizer': { active: false },
   // Turn on/off the comment/tag annotations
   'pundit.CommentTagPanel': {
      active: true, enableEntitiesExtraction: true},
   // Selectors are open data sources where entities can be looked up (in addition to
custom vocabularies)
   'selectors': {
          'Freebase': { active: false },
          'DBPedia': { active: false },
          'Wordnet': { active: false }
}}};
```

7.5 Publishing the bookmarklet

In this experiment we wanted users to be able to annotate a variety of pages on the web, such as, for example, pages from http://wikisource.org, which contains original works from a variety of authors as open content.

To achieve this we created a specific Pundit bookmarklet, and a simple HTML page where users can install it into their browsers: http://thepund.it/bm/demo-philosophers/bookmarklet.html.

To learn how to configure and deploy your Pundit bookmarklet please refer to the tutorial titled *"Deploying the annotation environment"* or read the *README.TXT* file that can be found in the Pundit client package in the *"bookmarklet_build"* directory.

7.6 Extending the Edgemaps data model

Now that we created an appropriate Pundit instance to let the user create a specific kind of annotations, let us see how the interaction with Edgempas works.

First of all, we extended the Edgempas data model to store, in addition to Philosophers data, details of the related Pundit annotations.

A single philosopher is represented in Edgempas with the following syntax:

```
"/en/aristotle":{
    "id":"/en/aristotle",
    "guid":"9202a8c04000641f8000000000003bdf",
    "img_guid":"9202a8c04000641f8000000007443028",
    "name":"Aristotle",
    "abstract":"Aristotle was a Greek philosopher....",
    "birthyear":"-383",
    "name_short":"ARISTOTLE",
    // philosophers that was influenced by Aristotle
    "to":{  "/en/adam_smith":1,
            "\en\anselm_of_canterbury":1,
            "\en\baruch_spinoza":1,
            .....
    },
    // philosophers that influenced Aristotle
    "fr":{  "/en/anaximander":1,
            "/en/democritus":1,
            "\/en\/empedocles":1,
            .....
    },
    // the number ot philosophers that was influenced by Aristotle
    "to_count":32,
    // the number of philosophers that influenced Aristotle
    "fr_count":7,
    "d1":0.438690026767,
    "d2":0.730099304692}
```

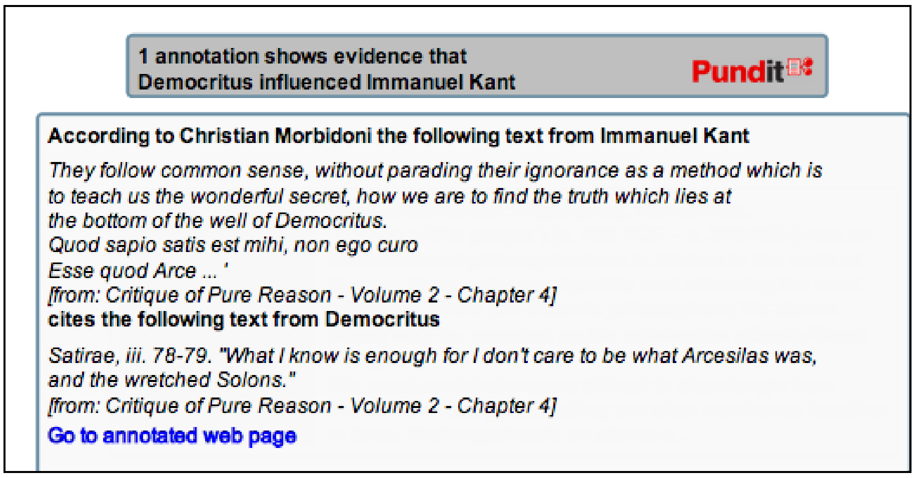
We extended this simple data model by adding the following attribute to the philosophers' nodes, resulting in the following JSON code:

```
"/en/aristotle":{
  "annotations": {
    // the following object includes annotations linking Aristotle to Aquinas
    "/en/thomas_aquinas" {
       // the following indicates a single annotation by its ID
      "c65a1b5b": {
         // The sentence that was annotated
        "annotated_sentence": "But the principle of human acts is not in man himself,
        but outside him: since man's appetite is moved to act, by the appetible object
        which is outside him, and is as a "mover unmoved" (De Anima iii, 10).
        Therefore there is nothing voluntary in human acts.",
       // the name of the Pundit user who did the annotation
       "annotator": "Romeo Zitarosa",
       // the web page where the annotation was made
       "page": "http://en.wikisource.org/wiki/Summa_Theologiae/
             First_Part_of_the_Second_Part/Question_6",
       // the predicate (RDF property) used to link the text to the cited philosopher
       "rel": "cites"}}}
  ....
 }
```

In the case of an annotation connecting two texts (the second type of annotation previously discussed), the JSON file has some more information. Here is an example:

```
"/en/democritus":{
  "annotations": {
    "/en/immanuel_kant": {
        "0aa418f1": {
            annotated_sentence: "They follow common sense, without parading their
                    ignorance as a method which is to teach us the wonderful secret,
                    how we are to find  the truth which lies at the bottom of the well
                    of Democritus. Quod sapio satis est mihi, non ego curo Esse quod
                    Arcesilas aerumnosique Solones..."
            // the target sentece is the object of the annotation
            annotated_target_sentence: "Satirae, iii. 78-79. What I know is enough
                    for I don't care to be what Arcesilas was, and the wretched
                    Solons."
            annotator: "Christian Morbidoni"
            page: "http://en.wikisource.org/wiki/
                      Critique_of_Pure_Reason/Volume_2/Chapter_4"
            rel: "cites"
           // The target page is where the target sentence belongs to
            //(in this case it is the same page as the subject sentence)
            target_page: "http://en.wikisource.org/wiki/
                      Critique_of_Pure_Reason/Volume_2/Chapter_4"
        }}}
      ....
  }
```

In order to make such additional information visible we extended the Edgemaps code (SVG based) to modify the behaviour of the preview box displayed at the bottom-right of the page: when the user moves his mouseover a circle connected to the currently selected one, such a box displays annotations-related information, including a link to see the annotation in its context (as shown in the following figure).


1 annotation shows evidence that
Democritus influenced Immanuel Kant

Pundit

According to Christian Morbidoni the following text from Immanuel Kant

They follow common sense, without parading their ignorance as a method which is to teach us the wonderful secret, how we are to find the truth which lies at the bottom of the well of Democritus.
Quod sapio satis est mihi, non ego curo
Esse quod Arce ... '
[from: Critique of Pure Reason - Volume 2 - Chapter 4]

cites the following text from Democritus

Satirae, iii. 78-79. "What I know is enough for I don't care to be what Arcesilas was, and the wretched Solons."
[from: Critique of Pure Reason - Volume 2 - Chapter 4]

Go to annotated web page

Figure 17. The preview box modified to show annotations information.

7.7 Getting annotations data from the Pundit server

Finally we included in the application an additional javascript file (named *get-pundit.js*) that queries the Pundit server to retrieve annotations in a specified notebook and to create the extended Edgemaps JSON file described so far.

In this section we comment the most relevant code fragments, while the full code can be found in the GIT repository.

The resulting application can be controlled by special parameters in the URL. The following is an example invocation:

http://thepund.it/edgemaps_demo/demo.html?nbs={17dd5f94}&source={pundit}#phils;

The parameters we added are:

- **nbs**, listing the ids of the Pundit notebooks to be retrieved
- **source**, which indicates the data source. We set it to "pundit" to specify that data have to be imported from a Pundit vocabulary, while we set it to "freebase" to simply get data from freebase (the default behaviour of the original Edgemaps demo)

To retrieve the annotations we make subsequent calls to the Pundit server.

First we query a single notebook and retrieve all the annotations in it:

```
function getNotebookAnnotations(notebookId) {
    var self = this,
    args = {
        url: annotationServerApi +
                    "open/notebooks/" + notebookId + "/annotations/metadata",
        headers : {"Accept": "application/json"},
        handleAs: "json",
        load: function(r) {
            self.handleNotebookAnnotationsMetadata(r);
        },
        error: function(error) {
        }
    };
    dojo.xhrGet(args);
}
```

Once the list of annotations with related metadata is obtained, we get each single annotation id.

```
function handleNotebookAnnotationsMetadata(annotations) {
  var cont = 0;
  for (ann in annotations) {
       cont ++;
  }
  numberOfAnnotationsInCurrentNotebook = cont;
  for (url in annotations) {
       getAnnotationGraph(annotations[url],
              annotations[url]["http://purl.org/pundit/ont/ao#id"][0].value);
       }
}
```

For each annotation we then retrieve its graph, composed by all the triples the user included in the annotation.

```
function getAnnotationGraph(annotationMetadata, annotationId) {
    var self = this,
    args = {
        url: annotationServerApi + "open/annotations/" + annotationId + "/graph",
        headers : {"Accept": "application/json"},
        handleAs: "json",
        load: function(r) {
            self.handleAnnotationGraph(annotationId, annotationMetadata, r);
        },
        error: function(error) {
        }
    };
    dojo.xhrGet(args);
}
```

There is another piece of data that we have to take into account: the annotation items.

An annotation includes two graphs: the first one contains the triples that the user created, the second (the items graph) contains metadata about each resource used in the user-created triples (e.g. labels, types, pictures, etc.). The item graph is stored in a separate context as it is semantically different from the other graph: it includes triples that describe the resources, but such triples were not created by the user, they were automatically imported from vocabularies.

To get this data we have to make an additional API call:

```
function getAnnotationItems(annotationId, annotationMetadata, graph) {
    var self = this,
    args = {
        url: annotationServerApi + "open/annotations/" + annotationId + "/items",
        headers : {"Accept": "application/json"},
        handleAs: "json",
        load: function(r) {
            self.handleAnnotationMetadataAndGraphAndItems(annotationMetadata, graph,r);
        },
        error: function(error) {
        }
    };
    dojo.xhrGet(args);
}
```

At this point we have all the data we need to generate the extended Edgemaps JSON data.

It is important to note that after this experiment was coded, a new API was added to the Pundit server, in order to avoid the large amount of subsequent calls. Such a new API allows the user to get all the data in a single API call:

GET /notebooks/{notebookid}/

The resulting data is a JSON file containing all the annotations in the notebook and for each annotation it includes its graph and its items, in the following format:

```
"metadata": {
             // Notebook's metadata
            *JSON/RDF*
        },
"annotations": [
            {
                "metadata": {
                // Annotation's metadata
                *JSON/RDF*
            },
            "graph": {
                // Annotation's triples
                *JSON/RDF*
            },
            "items": {
                // Annotation's items
                *JSON/RDF*
            }
       },
       {
           "metadata": {
               // Annotation's metadata
               *JSON/RDF*
           },
           "graph": {
               // Annotation's triples
               *JSON/RDF*
           },
           "items": {
               // Annotation's items
               *JSON/RDF*
           }
       },
       .......
       ]
    }
```

7.8 Generating the extended Edgemaps JSON

The last thing we did was generating the extended Edgemaps JSON data from the data we gathered from the Pundit server.

This is done by the *handleAnnotationMetadataAndGraphAndItems* Javascript function.

The only complexity in such a function is the handling of RDF/JSON, which is the default format used by the Pundit server to serve RDF triples. A triple in such a format is represented as follows:

{ "S" : { "P" : [O] } }

where *S* is the subject of the triple, *P* is the predicate and *O* is the object.

While subject and predicate can be represented by a URI, additional information is needed for correctly describing the object, which can be a literal.

Here is an example of an RDF triple in RDF/JSON:

```
{
  "http://example.org/about" :
    {
       "http://purl.org/dc/elements/1.1/title":
             [ { "type" : "literal" , "value" : "Anna's Homepage" } ]
    }
}
```

As one can see the *type* attribute is used to distinguish between literals and URIs, while the *value* attribute specifies the actual value of the object (a URL or a Literal).

Full documentation of the format can be found at http://docs.api.talis.com/platform-api/output-types/rdf-json.

The following portion of code parses the RDF/JSON triples to get the data we need:

```
// for all the subjects of triples in the annotation graph ...
for(subj in graph) {
  var screator;
  // scan all the predicates used in combination with
  // the given subject to find the dcterms:creator,
  // which is the author of the annotated sentence
  for (pred in graph[subj]) {
    if (pred == "http://purl.org/dc/terms/creator") {
        screator = graph[subj]["http://purl.org/dc/terms/creator"][0].value;
    }
  }

  ....

  for (pred in graph[subj]) {
    if (pred.indexOf("http://purl.org/spar/cito/") !== -1) {
      predicate = items[pred]["http://www.w3.org/2000/01/rdf-schema#label"][0].value;
    // if the subject of the triple is a sentence (a text-fragment)...
    if (items[subj]["http://www.w3.org/1999/02/22-rdf-syntax-ns#type"][0].value
              == "http://purl.org/pundit/ont/ao#text-fragment"  ||
        items[subj]["http://www.w3.org/1999/02/22-rdf-syntax-ns#type"][0].value ==
        "http://purl.org/pundit/ont/ao#fragment-text") {
     // get the ID of the author of the sentence...
     citingId =
       "/" + screator.split("/")[screator.split("/").length - 1].replace(".","/");
     // get the text of the sentence ...
     citingSentence =
        items[subj]["http://purl.org/dc/elements/1.1/description"][0].value;
    // get the page that was annotated ...
    annotatedPage =
        items[subj]["http://purl.org/pundit/ont/ao#hasPageContext"][0].value;
    }

    // Get the object of the triple ...
    obj = graph[subj][pred][0].value;

    if (items[obj]["http://www.w3.org/1999/02/22-rdf-syntax-ns#type"][0].value
        == "http://purl.org/pundit/ont/ao#text-fragment" ||
        items[obj]["http://www.w3.org/1999/02/22-rdf-syntax-ns#type"][0].value ==
        "http://purl.org/pundit/ont/ao#fragment-text") {
        // get the text of the sentence ...
        citedSentence =
        items[obj]["http://purl.org/dc/elements/1.1/description"][0].value;
        // get the page where the sentence belongs to ...
        annotatedTargetPage =
               items[obj]["http://purl.org/pundit/ont/ao#hasPageContext"][0].value;
         // get the author of the sentence ...
         var ocreators = graph[obj]["http://purl.org/dc/terms/creator"];
        // if the author is not specified, doscard the annotation ...
       if (typeof(ocreators) === 'undefined') {
               continue;
       }
        citedUrl = ocreators[0].value;
       // get the id of the author
        citedId =
               "/" + citedUrl.split("/")[citedUrl.split("/")
                                        .length - 1].replace(".","/");
       // if the object is not a sentence, it should be a philosopher ...
        } else {
               citedUrl = obj;
               citedId =
                     "/" + citedUrl.split("/")[citedUrl.split("/")
                                        .length - 1].replace(".","/");
        }
     }
   }
}
```

Now we can use the extracted data to populate the Edgemaps JSON. This is fairly straightforward and is done in the *updateEdgemapsData* function:

function updateEdgemapsData(annotationId, annotator, annotatedPage, annotatedTargetPage, citedId, citingId, citedSentence, citingSentence, predicate) {

```
function updateEdgemapsData(annotationId, annotator, annotatedPage,
                                 annotatedTargetPage, citedId, citingId,
                                 citedSentence, citingSentence, predicate) {
      //Annotations are taken into consideration only if cited and citing philosophers
      are already present in the JSON data
      if (phils.ph[citedId] == undefined || phils.ph[citingId] == undefined) {
             return;
      }
      if (phils.ph[citedId].to[citingId] == undefined) {
             phils.ph[citedId].to[citingId] = 1;
             phils.ph[citedId].to_count ++;
              phils.to_max ++;
      }
      if (phils.ph[citingId].fr[citedId] == undefined) {
             phils.ph[citingId].fr[citedId] = 1;
             phils.ph[citingId].fr_count ++;
              phils.fr_max ++;
      }
      if (phils.ph[citedId].annotations == undefined) {
             phils.ph[citedId].annotations = {};
```

```
       }
       if (phils.ph[citedId].annotations[citingId] == undefined) {
              phils.ph[citedId].annotations[citingId] = {};
       }
       phils.ph[citedId].annotations[citingId][annotationId] = {};
       phils.ph[citedId].annotations[citingId][annotationId].annotator = annotator;
       phils.ph[citedId].annotations[citingId][annotationId].annotated_sentence =
                                                                   citingSentence;
       phils.ph[citedId].annotations[citingId][annotationId].page = annotatedPage;
       phils.ph[citedId].annotations[citingId][annotationId].rel = predicate;

       if (annotatedTargetPage !== undefined && annotatedTargetPage !== "") {
              phils.ph[citedId].annotations[citingId][annotationId].target_page =
                                                            annotatedTargetPage;
       phils.ph[citedId].annotations[citingId][annotationId].annotated_target_sentence
                                                            = citedSentence;

       }

};
```

8 A2 - Timeline annotation viewer

8.1 Introduction

In this document we discuss a demonstrative application built on top of the Pundit server.

The application is made of a server side component, Java based, that retrieves annotations from the Pundit server REST API and outputs JSON data.

The client side is constituted by TimelineJS (http://timeline.knightlab.com/), which displays the JSON data as a nice timeline.

This document is targeted at developers who want to build similar applications or extend this one with additional features. Before going through this document we recommend to users to have a look at the tutorial titled *"Edgemaps visualization with Pundit annotations"*, which presents the prototype from an end-user perspective, demonstrating its usage.

To see the resulting demonstrative application, please go to

http://metasound.dibet.univpm.it/timelinejs/examples/pundit.html?notebook-ids=6290cd68

where a sample notebook made with Pundit by annotating various web pages is shown as a timeline.

This demonstration was developed in collaboration with Semedia, University of Ancona (http://semedia.dibet.univpm.it/), where the demo is hosted.

Both Pundit client and server are open-source and available at GitHub:

- Pundit client, https://github.com/net7/pundit
- Pundit server, https://github.com/net7/pundit-server

8.2 Customising and deploying Pundit

In order to be displayed in our timeline, annotations will have to contain at least one triple that associates a date to the annotated object, be it a web page, a portion of text or an image. Furthermore, we would like to show other selected information if presented as, for example, the person or city depicted in a picture or the author of a certain text.

To enable compliant annotations we configured and deployed Pundit as a bookmarklet.

8.3 Annotations and annotation vocabulary

Compliant annotations will have to include one triple of the form:

annotated object - dates to - YYY-MM-DD

This is the only requirement for an annotation to be displayed in the timeline.

A simple Relation Set was created that includes a "dates-to" property and some other properties, such as "depicts", "cites", "has author".

Such a Relation Set is available at http://metasound.dibet.univpm.it/timelinejs/pundit_conf/timeline_demo_relations.jsonp

With respect to Entities Taxonomy (hierarchical custom vocabularies in Pundit), we didn't provide any specifics for this example, rather we invited users to search for entities in Freebase or in DBpedia.

8.4 Configuring Pundit

Once the simple Relation Set was created, we included it in the Pundit configuration that we can see at http://metasound.dibet.univpm.it/timelinejs/pundit_conf/pundit-conf.js.

To read more about Pundit configuration the users shouldrefer to the "Deploying the annotation environment" tutorial.

8.5 Publishing the bookmarklet

In this demonstration we expect users to annotate generic pages they find on the web.

To achieve this we created a specific Pundit bookmarklet, and a simple HTML page where users can install it into their browsers:

http://thepund.it/bm/demo-timeline/

To learn how to configure and deploy your Pundit bookmarklet please refer to the tutorial titled *"Deploying the annotation environment"* or read the *README.TXT* file that you find in the Pundit client package in the *"bookmarklet_build"* directory.

8.6 The server side component

The server side component is a Java class exposed as a REST web service thanks to Jersey (https://jersey.java.net/) a well know Java library that makes it easy to expose class methods as RESTful API.

The Java code for this class is listed at the end of this document. Please note that, being a demonstrative application, we didn't invest time in producing professional code.

Let us illustrate the main functionalities and methods that such a class implements.

The main method has the following signature

```
@GET
@Path("/to.jsonp")
@Produces(MediaType.TEXT_PLAIN)
public String getTimelineJSON   (@QueryParam("notebook-ids") String ids,
                                  @QueryParam("api") String annotationApi,
                                  @QueryParam("namespace") String notebooksNamespace,)
```

As you can see, the method has three parameters. The first indicates the ids of the notebooks that will be queried, separated by commas. The second parameter is the root

URL of the Pundit serve API. This means the web service is completely de-coupled from the Pundit server and different Pundit server's instances can be used within the same web service.

The third parameter is the namespace used by the Pundit server to represent notebooks as RDF resources. In the case of the Net7 public server such a namespace is set to *http://swickynotes.org/notebook/resource/.* This means that a notebook with id *12345* is represented by resources with URL equal to *http://swickynotes.org/notebook/resource/12345.*

Please note that Java annotations are used by Jersey to automatically expose the method as a REST API whose location is "/to.jsonp" and which accepts three parameters via HTTP GET:
notebook-ids, *api* and *namespace*.

Example API call:

http://metasound.dibet.univpm.it:8080/PunditTimeliner/to.jsonp?notebook-ids=6290cd68&namespace=http://swickynotes.org/notebook/resource/&api=http://as.thepund.it:8080/annotationserver/api/open/

8.7 Getting annotations and parsing their RDF graphs

The method that connects to the Pundit server is *getNotebookJSON*, which simply calls the Pundit server API to get one notebook in JSON format.

```
	private JSONObject getNotebookJSON(String id) throws IOException {

		String parameters = "notebooks/" + id + "/";
		URL call = new URL(annotationApi + parameters);
		BufferedReader reader =
			new BufferedReader(new InputStreamReader(call.openStream()));
		String json = "";
		String line = null;
		while ((line = reader.readLine()) != null) {
			json += line;
		}

		return JSONObject.fromObject(json);

	}
```

This methods simply invokes the following API

GET /notebooks/{notebook-id}/

which returns the metadata about the notebook (its creation time, author, etc) as well as all the annotations contained in it, in the following format:

```
"metadata": {
            // Notebook's metadata
           *JSON/RDF*
        },
"annotations": [
           {
               "metadata": {
               // Annotation's metadata
               *JSON/RDF*
           },
           "graph": {
               // Annotation's triples
               *JSON/RDF*
           },
           "items": {
               // Annotation's items
               *JSON/RDF*
           }
       },
       {
           "metadata": {
               // Annotation's metadata
               *JSON/RDF*
           },
           "graph": {
               // Annotation's triples
               *JSON/RDF*
           },
           "items": {
               // Annotation's items
               *JSON/RDF*
           }
       },
       .......
       ]
    }
```

To parse JSON we used the JsonLib library (http://json-lib.sourceforge.net/). The following code gets one notebook in JSON format and separates its metadata from its annotations.

```
JSONObject obj = getNotebookJSON(notebookIds[i]);
JSONObject metadata = (JSONObject)obj.get("metadata");
JSONArray annotations = (JSONArray)obj.get("annotations");
```

In the following method you can see how the JsonLib primitives can be used to query the RDF graph in RDF/JSON format. As this format arranges subject, predicates and objects of the triples in the graph hierarchically, not all the queries are easy to perform in this way.

In our case we need to match triples with given subjects and predicates. This is straightforward and exemplified in the following code.

```
	/**
	 * Retrieves from the RDF graph, represented in RDF/JSON, the values of the
triples with the given subject and predicate.
	 * @param rdfSubject. The subject of the triples to be matched
	 * @param rdfProperty. The predicate of the triples to be matched
	 * @param graph The RDF graph represented in RDF/JSON
	 * @return
	 */
	private ArrayList<String> getRDFPropertyValues(String rdfSubject, String
rdfProperty, JSONObject graph) {

		ArrayList<String> values = new ArrayList<String>();
		// Get the JSOB object representing all the triples
		// with the given subject ...
		JSONObject triples = (JSONObject)graph.get(rdfSubject);
		if (triples == null) return values;
		// Get the JSON array with all the values of the matched triples ...
		JSONArray objectValues = (JSONArray)triples.get(rdfProperty);
		if (objectValues == null) return values;
		// For each object, get his string value and add it to the results ...
		for (Object objv : objectValues) {
			JSONObject obj = (JSONObject)objv;
			String objvString = (String)obj.get("value");
			if (objvString != null) {
				values.add(objvString);
			}

		}

		return values;
	}
```

Other RDF-to-JSON serialization formats exist, in particular, the relatively recent JSON-LD which has been recently updated to provide easier interaction with the graph.

However, if the users want to fully exploit the potential of the RDF data model, they should consider using one of the available RDF API implementations, such as OpenRDF Sesame (http://www.openrdf.org/), which provides a standard and powerful query language such as SPARQL (http://www.w3.org/TR/sparql11-query/).

8.8 Generating TimelineJS data

TimelineJS loads data from a JSONP (http://en.wikipedia.org/wiki/JSONP) data source, which has to output data in a simple format. The example we show below is the output of our web service where a timeline of two slides is represented:

```
storyjs_jsonp_data={
    "timeline": {
        "headline": "My city in time: Ancona, Italy",
        "type": "default",
        "text": "A timeline view of this public notebook.",
        "startDate": "1000,1,1",
        "endDate": "2012,1,1"
        "date": [
            {
                // Each slide has a URL that identified it
                "uri": "http://www.antiwarsongs.org/canzone.php",
                // the startDate attribute is needed to position the slide in time
                "startDate": "1920,01,01",
                // The title of the slide
                "headline": "Canto della rivolta dei bersaglieri di Ancona&nbsp;",
                // We embed HTML in the text as a simple way of including
                 // metadata and link to related annotated resources
                "text": "
                    <p><a
                    href='http://www.antiwarsongs.org/canzone.php?id=28493&lang=it'
                    target='_blank'>Go to annotated page >><\/a><\/p><p>
                    <br/>Soldato proletario che parti per Valona\nnon ti scordar del
                    popolo d'Ancona\nche volle col suo sangue la tua liberazione\nsol
                    con la ribellione sorge radiosa la libertà.\n\nAndiamo via senza
                    ....
                    ....
                    <\/i><\/p>"
            },
            {
                "uri":
                    "http://purl.org/pundit/fragment/image/
                                  407231e5a1366cfee1502464dcbef266",
                "asset": {
                    "media":
                           "http://www.amicidellemarche.it/galeazzi/
                                         stamira_grande1.JPG"
                },
                "text": "<p><a
                    href='http://feed.thepund.it/?img=http://www.amicidellemarche.it/g
                    aleazzi/stamira_grande1.JPG&conf=timeline-demo.js'
                    target='_blank'>Go to annotated page
                    >><\/a><\/p><p><i>Comments:<br/>This is probably the old church
                    that was replaced by the Duomo we have today<\/i><\/p>"
            }
        ]
    }
}
```

As you can see each slide contains some basic metadata that TimelineJS parses to create the UI. However we can include custom metadata, texts, images and links by using the “text” attribute. Its value allows for arbitrary HTML, thus we can query the RDF graph to extract relevant triples and creating custom HTML to show their content.

The following method is an example of how it can be done.

```
/**
* Returns an HTML representation of the given RDF triple
* @param subject. the subject of the triple
* @param property. the predicate
* @param propertyLabel. the label of the predicate
* @param graph. the RDf annotation graph in RDF/JSON format
* @param itemsGraph. the Items of the annotation as an RDF/JSON serialized graph
* @return
* @throws UnsupportedEncodingException
*/
public String printObjectPropertyValue(String subject, String property,
              String propertyLabel, JSONObject graph, JSONObject itemsGraph)
                            throws UnsupportedEncodingException {
       String html = "";

       // get the values of the matched triples
       ArrayList<String> values = getRDFPropertyValues(subject, property, graph);
       if (values != null && !values.isEmpty()) {
              String content ="";
              // For each value ...
              for (int f = 0; f < values.size(); f++) {
                     // Get the rdf:label of the object value. NOTE: this information
                     // is stored in the annotation Items graph.
                     ArrayList<String> label = getRDFPropertyValues(values.get(f),
                                          Namespaces.RDFS_LABEL, itemsGraph);
                     content += "<br/>";
                     // Get the image associated to the value, if any …
                     ArrayList<String> thumbs = getRDFPropertyValues(values.get(f),
                                   Namespaces.PROPERTY_IMAGE, itemsGraph);
                     // if there is a n image, create a simple HTML enclosing it ...
                     if (thumbs != null && !thumbs.isEmpty()) {
                            content += "<img src='" + thumbs.get(0)
                                          + "' height='40'></img><br/>";
                     }
                     String url = values.get(f).split("#")[0];
                     // Create an hyperlink to the resource...
                     content += "<a href=\"" + url + "\">" + label + "</a>";
              }
```

```
                html += "<p>" + propertyLabel + ":" + content + "</p>";
        }
        return html;
}
```

Time information can be added by querying the RDF for the appropriate triple, those that have a specific predicate. In our case this predicate is *http://purl.org/pundit/ont/oa#periodStartDate.*

The following code, queries the RDF graph using one of the methods previously defined and puts the corresponding date into the resulting TimelineJS JSON.

```
public static final String PROPERTY_STARTDATE =
        "http://purl.org/pundit/ont/oa#periodStartDate";
ArrayList<String> startDates = getRDFPropertyValues(subject,
                Namespaces.PROPERTY_STARTDATE, graph);
if (startDates != null && !startDates.isEmpty()) {
        if (node.get("startDate") == null) {
                // A string transformation is needed to go from the standard XMLSchema
                // date format used in RDF to the comma separated format
                // used in TimelineJS
                String date = transformDateformat(startDates.get(0));
                // Update the JSON object to include the startDate attribute ...
                node.put("startDate", date);
                // Update the min and max dates attributed in the JSON,
                // which sets the tomporat axis
                updateMinMaxDates(date);

        };
}
```

Where the updateMinMaxDates can be implemented as the follows:

```
private void updateMinMaxDates(String date) {
        Integer dateInt = Integer.parseInt(date.split(",")[0]);
        if (dateInt.intValue() < minDate) {
                minDate = dateInt.intValue();
        }
        if (dateInt.intValue() > maxDate) {
                maxDate = dateInt.intValue();
        }
}
```

The full Java class described in this section can be found on the Git repository: https://github.com/chrmor/Pundit-Timeline-Demo/.

8.9 The client component: TImelineJS

The client side component is a simple HTML page including the TimelineJS javascript library and specifying where that data will be loaded from the web service described so far.

The full HTML, created by customising the default TimelineJS template page, is the following:

```
<script type="text/javascript"
src="http://ajax.googleapis.com/ajax/libs/jquery/1.7.2/jquery.min.js"></script>
<script type="text/javascript"
src="http://metasound.dibet.univpm.it/timelinejs/compiled/js/storyjs-embed.js">
</script>
<script>
      $(document).ready(function() {
      var decodedSearch = decodeURI(window.location.search);
       if ( (decodedSearch.indexOf("namespace=") === -1) &&
                             (decodedSearch.indexOf("api=") === -1)) {
                decodedSearch +=
                    "&namespace=http://swickynotes.org/notebook/resource/
                          &api=http://as.thepund.it:8080/annotationserver/api/open/";
       }
        var s =
             "http://metasound.dibet.univpm.it:8080/PunditTimeliner/
                    to.jsonp" + decodedSearch;
        createStoryJS({
                    type:        'timeline',
                    width:       '100%',
                    height:      '100%',
                    source:      s,
                    embed_id:    'my-timeline'
                });
            });
    </script>
    <!-- Style-->
    <style>
      html, body {
       height:100%;
       padding: 0px;
       margin: 0px;
      }
    </style>
    <!-- HTML5 shim, for IE6-8 support of HTML elements--><!--[if lt IE 9]>
    <script
       src="http://html5shim.googlecode.com/svn/trunk/html5.js"></script><![endif]-->
  </head>
</html>
<body>
  <!-- BEGIN Timeline Embed -->
  <div id="my-timeline"></div>
  <!-- END Timeline Embed-->

  <script>
    (function(i,s,o,g,r,a,m){i['GoogleAnalyticsObject']=r;i[r]=i[r]||function(){
    (i[r].q=i[r].q||[]).push(arguments)},i[r].l=1*new Date();a=s.createElement(o),
    m=s.getElementsByTagName(o)[0];a.async=1;a.src=g;m.parentNode.insertBefore(a,m)
    })(window,document,'script','//www.google-analytics.com/analytics.js','ga');

    ga('create', 'UA-41569925-1', 'univpm.it');
    ga('send', 'pageview');

  </script>

</body>
```

9 A3 - Consuming DM2E data in Feed

9.1 Introduction

In this document we demonstrate the consumption of RDF published as Linked Data within the DM2E project. Such data is structured according to the DM2E data model, published at http://onto.dm2e.eu/schemas/dm2e/1.0/.

We discuss how this is done in Feed (http://feed.thepund.it), one of the components of the DM2E software platform.

9.2 Feed overview

Feed is a PHP based component developed in WP3 of DM2E in order to provide an as-a-service access to Pundit.

Feed provides a simple REST API to specifically interact with DM2E data, based on the DM2E data model and on Linked Data.

The API is the following:

*http://feed.thepund.it/?dm2e=**{DM2E_Linked_Data_URL}**&conf=**{Pundit_Configuration}***

Where *DM2E_Linked_Data_URL* is the dereferenceable URL of some resource represented in RDF in conformance with the DM2E data model, and *Pundit_Configuration* is one of the preloaded configuration of Pundit.

When the API is called, Feed gets the RDF representation of the resource and parses it to extract relevant metadata (e.g. authorship, dates and, in general, connections with other resources that might be of interest to display to the user) and, most importantly, it grabs the content to be annotated (e.g. an image, a text of simple HTML) from the actual content provider. This is done "on the fly" via a special RDF property defined in the DM2E data model (dm2e:hasAnnotatableVersionAt).

An HTML representation of the resource is then created and loaded into Pundit, using the provided configuration.

Go to http://demo.feed.thepund.it/?dm2e=http://data.dm2e.eu/data/item/bbaw/dta/16821 to see an example of a DM2E digital object in Feed.

9.3 Browsing the DM2E Linked Data in Feed

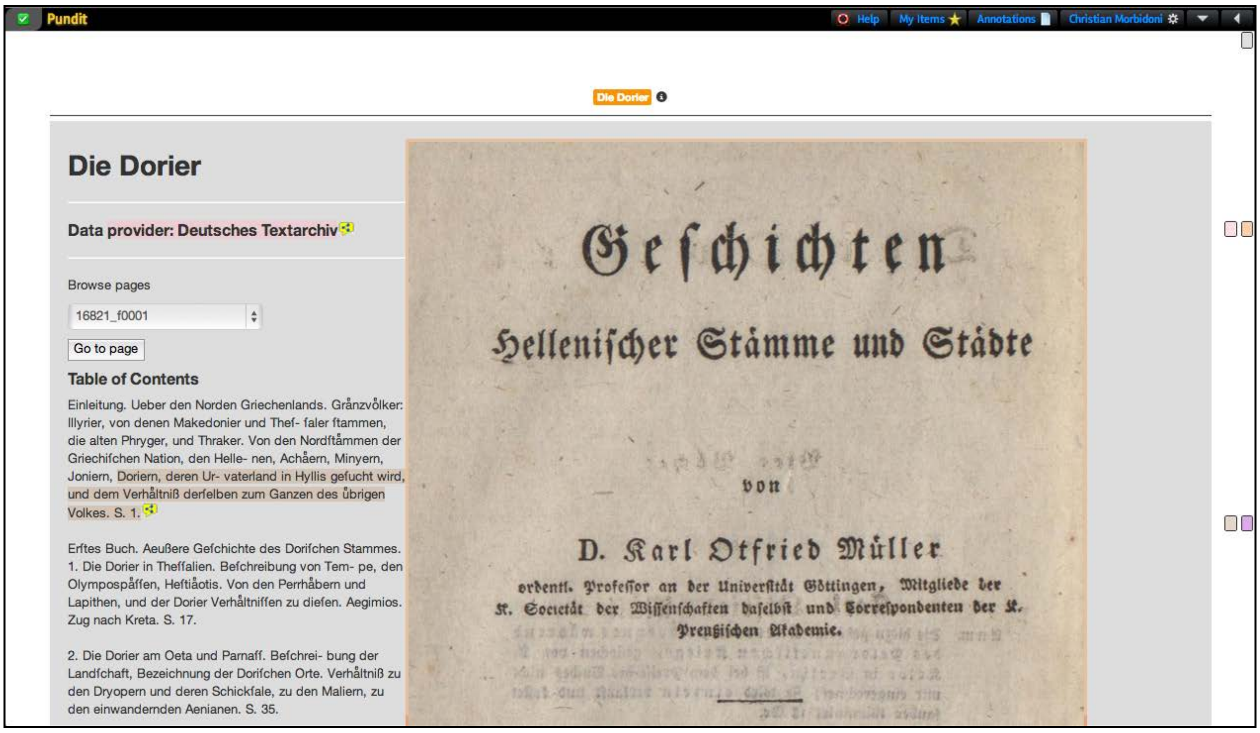
Two kinds of resources are supported in Feed at the current stage:

- ***Manuscripts (or Books):*** in this case a navigation bar is shown that allows users to go to a specific page;
- ***Pages*:** in this case the content is shown, ready to be annotated, and a link to the entire manuscript is provided.

Manuscripts are a particular kind of Cultural Heritage Objects (CHO) that contain other CHOs that usually represent single Pages. Both these types of CHOs are represented as Linked Data in the form of RDF, containing metadata as well as links among Manuscripts and Pages.

The following screenshot shows an example of a landing page in the annotation environment, where a Manuscript is represented. RDF data is parsed to visualize basic metadata, a table of contents (if present) and a preview of the Manuscript pages.

Single pages can be reached by using the “Browse page” dropdown menu.



Die Dorier

Data provider: Deutsches Textarchiv

Browse pages

16821_f0001

Go to page

Table of Contents

Einleitung. Ueber den Norden Griechenlands. Grȧnzvȯlker: Illyrier, von denen Makedonier und Theſ- ſaler ſtammen, die alten Phryger, und Thraker. Von den Nordſtȧmmen der Griechiſchen Nation, den Helle- nen, Achȧern, Minyern, Joniern, Doriern, deren Ur- vaterland in Hyllis geſucht wird, und dem Verhȧltniß derſelben zum Ganzen des ůbrigen Volkes. S. 1.

Erſtes Buch. Aeußere Geſchichte des Doriſchen Stammes. 1. Die Dorier in Theſſalien. Beſchreibung von Tem- pe, den Olympospȧſſen, Heſtiȧotis. Von den Perrhȧbern und Lapithen, und der Dorier Verhȧltniſſen zu dieſen. Aegimios. Zug nach Kreta. S. 17.

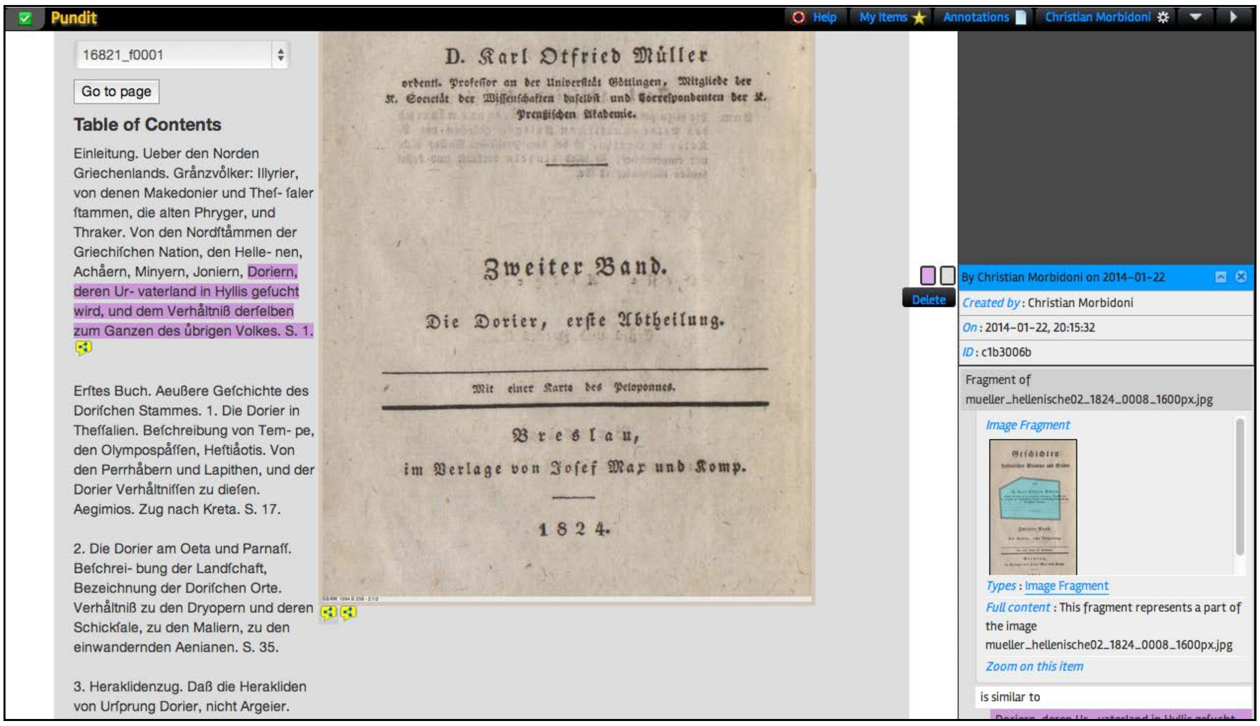
2. Die Dorier am Oeta und Parnaſſ. Beſchrei- bung der Landſchaft, Bezeichnung der Doriſchen Orte. Verhȧltniß zu den Dryopern und deren Schickſale, zu den Maliern, zu den einwandernden Aenianen. S. 35.

Geſchichten
Helleniſcher Stȧmme und Stȧdte
von
D. Karl Otfried Můller
ordentl. Profeſſor an der Univerſitȧt Gȯttingen, Mitgliede der K. Societȧt der Wiſſenſchaften daſelbſt und Correſpondenten der K. Preußiſchen Akademie.

Figure 18. A manuscript ready to be annotated in Feed.

As you can see, some example annotations are present in the page. This is shown by different colour highlighting on the text and small yellow icons to expand the annotations.

In the following screenshot, some expanded annotations are shown, demonstrating how a portion of text can be commented on or even connected with precise areas of an image.



Pundit

Help | My Items | Annotations | Christian Morbidoni

16821_f0001

Go to page

Table of Contents

Einleitung. Ueber den Norden Griechenlands. Grånzvölker: Illyrier, von denen Makedonier und Theſ- ſaler ſtammen, die alten Phryger, und Thraker. Von den Nordſtåmmen der Griechiſchen Nation, den Helle- nen, Achåern, Minyern, Joniern, Doriern, deren Ur- vaterland in Hyllis geſucht wird, und dem Verhåltniß derſelben zum Ganzen des übrigen Volkes. S. 1.

Erſtes Buch. Aeußere Geſchichte des Doriſchen Stammes. 1. Die Dorier in Theſſalien. Beſchreibung von Tem- pe, den Olympospåſſen, Heſtiåotis. Von den Perrhåbern und Lapithen, und der Dorier Verhåltniſſen zu dieſen. Aegimios. Zug nach Kreta. S. 17.

2. Die Dorier am Oeta und Parnaſſ. Beſchrei- bung der Landſchaft, Bezeichnung der Doriſchen Orte. Verhåltniß zu den Dryopern und deren Schickſale, zu den Maliern, zu den einwandernden Aenianen. S. 35.

3. Heraklidenzug. Daß die Herakliden von Urſprung Dorier, nicht Argeier.

D. Karl Otfried Müller

ordentl. Profeſſor an der Univerſität Göttingen, Mitgliede der K. Societät der Wiſſenſchaften daſelbſt und Correſpondenten der K. Preußiſchen Akademie.

Zweiter Band.

Die Dorier, erſte Abtheilung.

Mit einer Karte des Peloponnes.

Breslau,

im Verlage von Joſef Max und Komp.

1824.

Delete

By Christian Morbidoni on 2014-01-22

Created by: Christian Morbidoni

On: 2014-01-22, 20:15:32

ID: c1b3006b

Fragment of mueller_hellenische02_1824_0008_1600px.jpg

Image Fragment

Types: Image Fragment

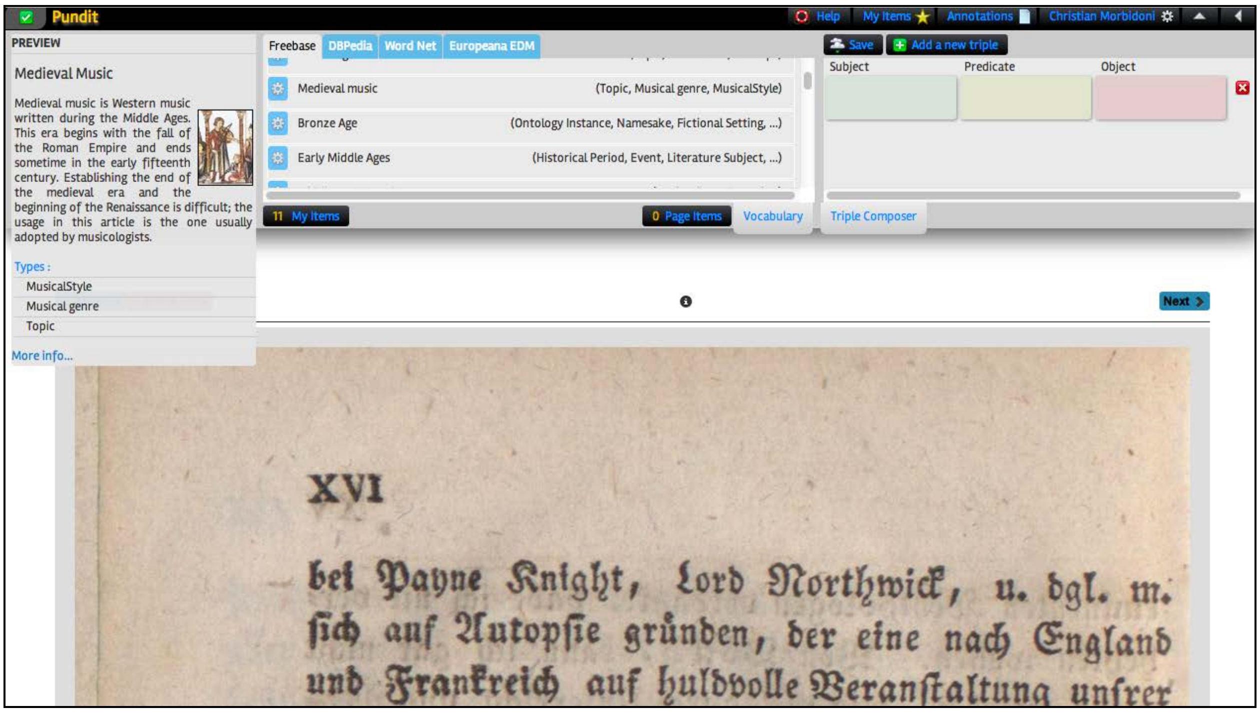
Full content: This fragment represents a part of the image mueller_hellenische02_1824_0008_1600px.jpg

Zoom on this item

is similar to

Figure 19. An annotation made on the table of contents of a book.

The following screenshot shows how a single Page is visualised. A link to the enclosed manuscript as well as to previous and next pages is provided. In this screenshot you can also see the Pundit toolbar, that can be used to search for relevant Linked Data entities and create semantic annotations.



Pundit

Help | My Items | Annotations | Christian Morbidoni

PREVIEW

Medieval Music

Medieval music is Western music written during the Middle Ages. This era begins with the fall of the Roman Empire and ends sometime in the early fifteenth century. Establishing the end of the medieval era and the beginning of the Renaissance is difficult; the usage in this article is the one usually adopted by musicologists.

Types:
- MusicalStyle
- Musical genre
- Topic

More info...

Freebase | DBPedia | Word Net | Europeana EDM

Medieval music	(Topic, Musical genre, MusicalStyle)
Bronze Age	(Ontology Instance, Namesake, Fictional Setting, ...)
Early Middle Ages	(Historical Period, Event, Literature Subject, ...)

11 My Items | 0 Page Items | Vocabulary

Save | Add a new triple

Subject	Predicate	Object

Triple Composer

Next

XVI

bei Payne Knight, Lord Northwick, u. dgl. m. ſich auf Autopſie gründen, der eine nach England und Frankreich auf huldvolle Veranſtaltung unſrer

Figure 20. The Pundit toolbar showing results from search on Freebase. Such entities can be used in the triple composer (right box) to compose triples.

9.4 Parsing RDF in PHP

Parsing RDF/XML, which is the standard format in Linked Data, in PHP can be very cumbersome without an appropriate library. The very same RDF graph can be serialised in many different ways, all valid and XML compliant. This means that relying on XML/DOM parsing cannot easily result in a generic code that supports all variants.

A similar issue applies to other serialization formats.

We used the EasyRDF PHP library[2] as an abstraction layer. This makes RDF parsing and querying very easy, even if the queries expressiveness is very limited if compared to SPARQL.

In Feed, the PHP function that parses DM2E Linked Data is *retrievePunditContentDm2e.* Let us shortly comment on its implementation.

Loading RDF from a Linked Data source is as easy as calling a library function with a dereferenceable URL as parameter.

```
        $url = $this->url;
        $this->dm2eGraph = EasyRdf_Graph::newAndLoad($url);
```

One limitation we found is the impossibility in the current release of EasyRDF, to use extended URLs in queries, allowing only URIrefs a parameters. Therefore, we have to pay particular attention to defining all the namespace we need.

```
        EasyRdf_Namespace::set('edm','http://www.europeana.eu/schemas/edm/');
        EasyRdf_Namespace::set('dm2e','http://onto.dm2e.eu/schemas/dm2e/1.1/');
        $this->nsDc =
              EasyRdf_Namespace::prefixOfUri('http://purl.org/dc/elements/1.1/');
        $this->nsDct = EasyRdf_Namespace::prefixOfUri('http://purl.org/dc/terms/');
        $this->nsSpar = EasyRdf_Namespace::prefixOfUri('http://purl.org/spar/pro/');
```

At this point we can extract relevant values from RDF triples, for example the following code looks up the dm2e:hasAnnotatableVersionAt property, which is used to link to a representation annotatable with Pundit.

```
private function extractDm2eAnnotableFormatByDom() {
       // get all resources in the RDF that appears in a triples
       //where the predicate is edm:aggregatedCHO
        $aggs = $this->dm2eGraph->resourcesMatching('edm:aggregatedCHO');
       // return the value of the triple where the predicate is
       //dm2e:hasAnnotatableVersionAt, and the subject is the aggregation
        return  $aggs[0]->get('dm2e:hasAnnotatableVersionAt');
}
```

[2] http://www.easyrdf.org/

10 A4 - A Solr based faceted search over DM2E LD

10.1 Introduction

The DM2E consortium has published a considerable amount of digital cultural heritage objects and data about them in the form of Linked Data.

The data modeling and schema mapping work carried on in the project lead to a uniform ontology that is being used as a semantic backbone to make data homogeneous among the different data providers and collections.

While access to data is perfectly supported by the Linked Data API and the availability of a SPARQL endpoint, end-users benefit from more friendly search and browsing interfaces.

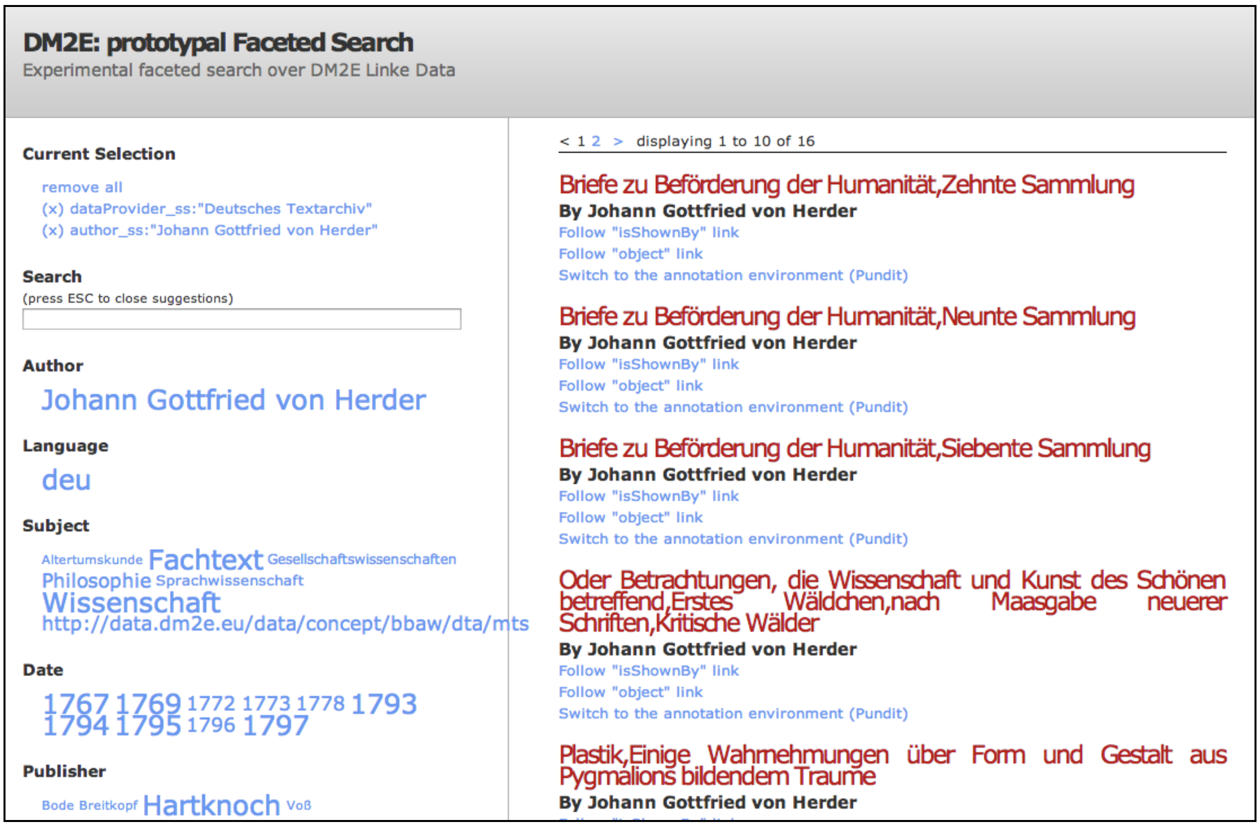
Primo (http://www.exlibrisgroup.com/category/PrimoOverview) is an established solution for indexing and searching in a digital library. EXLIBRIS, partner of the DM2E project, provided a Primo instance for full text search and faceted navigation over DM2E metadata, harvested via the OAI-PMH protocol (http://bit.ly/1dUOnly).

Similar functionalities can be reached by developing on top of Linked Data and leveraging popular open-source tools. The prototype described in this document aims at demonstrating how standards like SPARQL and RDF can be easily combined with well established open-source technologies like Solr (http://lucene.apache.org/solr/) to provide full-text and faceted search/browsing. A simple approach has been applied, which consists of using ad-hoc SPARQL queries to index RDF resources as Lucene documents.

This demo has been developed by Net7 SRL, Pisa, in collaboration with Semedia, Università Politecnica delle Marche, Ancona, where the prototype is hosted:

http://metasound.dibet.univpm.it/dm2e/ajax-solr-master/examples/dm2e/

The code of this demonstrative application is available on GitHub at https://github.com/chrmor/RDF2Solr.


DM2E: prototypal Faceted Search
Experimental faceted search over DM2E Linke Data

Current Selection

remove all
(x) dataProvider_ss:"Deutsches Textarchiv"
(x) author_ss:"Johann Gottfried von Herder"

Search
(press ESC to close suggestions)

Author
Johann Gottfried von Herder

Language
deu

Subject
Altertumskunde Fachtext Gesellschaftswissenschaften Philosophie Sprachwissenschaft Wissenschaft http://data.dm2e.eu/data/concept/bbaw/dta/mts

Date
1767 1769 1772 1773 1778 1793 1794 1795 1796 1797

Publisher
Bode Breitkopf Hartknoch Voß

< 1 2 > displaying 1 to 10 of 16

Briefe zu Beförderung der Humanität,Zehnte Sammlung
By Johann Gottfried von Herder
Follow "isShownBy" link
Follow "object" link
Switch to the annotation environment (Pundit)

Briefe zu Beförderung der Humanität,Neunte Sammlung
By Johann Gottfried von Herder
Follow "isShownBy" link
Follow "object" link
Switch to the annotation environment (Pundit)

Briefe zu Beförderung der Humanität,Siebente Sammlung
By Johann Gottfried von Herder
Follow "isShownBy" link
Follow "object" link
Switch to the annotation environment (Pundit)

Oder Betrachtungen, die Wissenschaft und Kunst des Schönen betreffend,Erstes Wäldchen,nach Maasgabe neuerer Schriften,Kritische Wälder
By Johann Gottfried von Herder
Follow "isShownBy" link
Follow "object" link
Switch to the annotation environment (Pundit)

Plastik,Einige Wahrnehmungen über Form und Gestalt aus Pygmalions bildendem Traume
By Johann Gottfried von Herder

Figure 21. A screenshot of the prototypal faceted browser over DM2E content.

10.2 Ingredients

10.2.1 Solr

Solr is a de-facto standard tool for full text indexing and is based on the Lucene engine.

Solr API is completely REST based and indexing and retrieving documents is easy in all programming languages. Java developers might find it useful to build on top of the SolrJ library (http://wiki.apache.org/solr/Solrj), that we used in our demonstration.

An interesting feature of the Sol API is that of supporting facets. Facets are particular fields that can be used to filter results of a query and to generate faceted browsers. This technical solution, even if it requires a previous indexing of original RDF data, provides very high performances.

10.2.2 Ajax-Solr

Ajax-solr (https://github.com/evolvingweb/ajax-solr) is a basic and simple to configure Javascript faceted browser for Solr.

With the knowledge of the Solr index schema, it is easy to customize facets by including HTML code. For example, once the "author_ss" field is available, adding a *tagcloud* based on the corresponding facet, consists of defining HTML elements likefollows:

```
<h2>Author</h2>
<div class="tagcloud" id="author_ss"></div>
```

And writing Javascript code to attach behaviors to elements. As show in the following code:

```
var fields = ['subject_ss', 'publisher_ss', 'author_ss', 'issued_ss',
'language_ss', 'dataProvider_ss'];
    for (var i = 0, l = fields.length; i < l; i++) {
      Manager.addWidget(new AjaxSolr.TagcloudWidget({
        id: fields[i],
        target: '#' + fields[i],
        field: fields[i]
      }));
    }
```

10.2.3 SPARQL

SPARQL is the standard RDF query language and protocol. As a language it is based on graph paths and is very flexible in matching even complex paths against RDF datasets. As a protocol it is HTTP based and easy to use from any programming languages.

Java developers are generally facilitated in interacting with RDF data, as different APIs and open implementations are available.

10.2.4 OpenRDF Sesame

In our application we used OpenRDF Sesame (http://openrdf.org), as it implements a de-facto standard Java API for triple-stores, which is implemented in a variety of commercial and open-source products.

10.3 Indexing RDF in Solr via SPARQL

Lucene indexes documents via a number of textual *fields,* that have to be defined in the indexing schema. Our goal was to find an easy way of creating and populating fields that reflect the information stored in some selected triples from the DM2E RDF dataset.

As RDF is designed with a flexible and “fluid” data schema, it is subject to changes. This happens not only because ontologies can change over time, but, most importantly, because new, different in structure, and possibly unexpected data could be added. This would, for example, enable the building of new indexes based not only on DM2E data, but on a data mashup from different sources.

With this scenario in mind, we wanted to obtain a certain freedom in rebuilding the Lucene index without having to go through complex schema editing.

Solr is not as flexible as RDF with respect to “on-the-fly” schema editing. However, it provides the so-called automatic fields that can be extended on the fly. Automatic fields are mandatory and identified by an “ss” suffix in their name.

10.3.1 Example queries

We used SPARQL queries with the following structure:

```
select distinct ?uri ?field ?value where { CUSTOM SPARQL QUERY }
```

Where CUSTOM SPARQL QUERY has been replaced with a number of ad-hoc queries to match relevant informations in the RDF graph, and where the matches for the variables will be used respectively as identifier of the indexed document in Solr (*?uri*), field to be indexed (*?field*), and corresponding values (*?value*).

For example, the following query has been used to index all the documents that are CHOs (Cultural Provided Objects), but that are not single *Pages.* Note that in this simple case a single SPARQL query is enough to index all the attributes of the resources that have some other resource with a valid rdf:label.

```
select distinct ?uri ?field ?value where {
                graph ?g {
                        ?uri <http://www.w3.org/1999/02/22-rdf-syntax-ns#type>
                                <http://www.europeana.eu/schemas/edm/ProvidedCHO>.
                        FILTER NOT EXISTS {?uri <http://purl.org/dc/elements/1.1/type>
                                <http://onto.dm2e.eu/schemas/dm2e/1.1/Page>}
                        FILTER NOT EXISTS {?uri <http://purl.org/dc/elements/1.1/type>
                                <http://purl.org/spar/fabio/#Page>}
                        ?uri ?field ?v.
                        ?v <http://www.w3.org/2004/02/skos/core#prefLabel> ?value.
                }
        }
```

This for example would match triples like the following:

dm2e:Example_CHO dc:creator dm2e:Leonardo_Da_vinci.
dm2e:Leonardo_Da_vinci rdf:label "Leonardo".
dm2e:Leonardo_Da_vinci rdf:label "Leonardo Da Vinci".

by creating an automatic field named ***creator_ss*** with value equals to "Leonardo, Leonardo Da Vinci".

10.3.2 Advanced: Indexing Pundit annotations

In other cases, it is difficult to capture the target path with a query like the previous one.

For example, let us illustrate a more complex case. Let's suppose we want to index not only DM2E data, but also RDF data produced as a result of annotations with Pundit. In Pundit users have notebooks. And such notebooks include triples that describe in some way the annotated resource.

Let us now suppose we want to create a Solr field (e.g. *related_notebooks_ss*) with the names of all the public Pundit notebooks that contain an annotation involving the resource.

The following SPARQL query would select such notebooks:

```
select distinct ?uri ?value where {
        ?n <http://purl.org/pundit/ont/ao#includes> ?a.
        ?n <http://purl.org/dc/terms/creator> ?auth.
        ?auth <http://xmlns.com/foaf/0.1/name> ?value.
        ?a <http://www.openannotation.org/ns/hasTarget> ?f.
        ?f <http://purl.org/dc/terms/isPartOf> ?uri.
        FILTER regex(str(?uri), "data.dm2e.eu","i") }
```

In this case, we do not have a single RDF property which corresponds to the field (as was happening in the *dc:creator* example). The "related_notebook_ss" field represents in fact a name for a given path in the graph.

10.4 Example Java code

This is the main function we used to create a document to be indexed in Solr by querying the DM2E SPARQL endpoint provided by WP2.

```
    /**
     * Given a SPARQL query with at least a ?uri and a ?value variables are binded,
     * indexes the Documents identified by each match of the ?uri variable,
     * adding fields equals to each match of the ?field variable and corresponding values,
     * matched by the ?value variable.
     * If no matches are found for the ?field variable, the default field is used, if provided.
     * @param query The SPARQL query
     * @param conn a connection to a Sesame Repository
     * @param solrDocs A set of Solr Input Documents to be updated with the new fields
     * @param defaultField an optional default field
     * @throws RepositoryException
     * @throws MalformedQueryException
     * @throws QueryEvaluationException
     * @throws MalformedURLException
     * @throws IOException
     */
private void indexByQuery(String query, RepositoryConnection conn, HashMap<String,
SolrInputDocument> solrDocs, String defaultField) throws RepositoryException,
MalformedQueryException, QueryEvaluationException, MalformedURLException, IOException {

        // Execute the query ....
        TupleQuery tquery = conn.prepareTupleQuery(QueryLanguage.SPARQL, query);
        TupleQueryResult res = tquery.evaluate();

        int id = 0;

        // foreach result
        while (res.hasNext()) {

                BindingSet bs = res.next();
                String uri = bs.getBinding("uri").getValue().stringValue();
                String field;

                // if no value has been matched skip ...
                if (bs.getBinding("value") == null) continue;

                // if the ?field variable has a match ...
                if (bs.getBinding("field")!=null) {
                Value fv = bs.getBinding("field").getValue();
                // if the value is a RDF resource we treat it accordingly
                // by checking for rdf:type and adding a special field
```

```
            if (fv instanceof Resource) {
                    if (fv.stringValue().equals("http://purl.org/dc/elements/1.1/type")) {
                            field = "topic_ss";
                    } else {
                            field = truncate(fv.stringValue()) + "_ss";
                    }

            } else {
                    field = fv.stringValue() + "_ss";
            }

            // Else if the default filed is provided use it instead ...
            } else if (defaultField != null){
            field = defaultField;
            } else {
            return;
            }

            Value valval = bs.getBinding("value").getValue();
            String val;
            // If the value is not a RDF resource we skip some unwanted chars
            if (valval instanceof Resource) {
            val = valval.stringValue();
            } else {
            val = valval.stringValue().replaceAll("\\[", "").replaceAll("\\]", "");
            }

            System.out.println(uri + " " + field + " " + val);

            SolrInputDocument doc;
            // Check if the document already exists. In this case add fields to the existinf
document...
            if (solrDocs.containsKey(normalizeWWWUri(uri))) {
            doc = solrDocs.get(normalizeWWWUri(uri));
            } else {
            // Create a new document
            doc = new SolrInputDocument();
            // Assign an incremental ID ...
            doc.addField("id", id++);
            // Add the URI field (mandatory) ...
            doc.addField("uri_ss", normalizeWWWUri(uri));
            // store the doc in the buffer ...
            solrDocs.put(normalizeWWWUri(uri), doc);
            }
            // The text field ha a special meaning: it contains text for the full text index.
            if (field.equals("text")) {
            doc.setField(field, val, 1.0f);
            } else if (!configuration.getTags_black_list().contains(val)) {
            doc.addField(field, val, 1.0f);
            }


        }
        }
```

11 T1 - Creating annotations with Pundit

11.1 Introduction

The present tutorial proposes three examples of annotation assignments, with a two-fold objective:

- to let users experiment with and understand the different ways of annotating a document in Pundit;
- to help users to create a critical mass of annotations to begin experimenting with Ask, the next Pundit component that will be introduced in Tutorial 2.

The annotation tutorials have been developed in the interaction with the Wittgenstein Archive Bergen (WAB) research group and subsequently tailored on real examples of research activity.

11.2 Audience

The tutorial is addressed to potential users as well as to developers interested in understanding how the process of annotation works in practice.

11.3 Objective

The objective of this tutorial is to show how to perform annotations with Pundit. The activity of annotating aims at solving specific research questions such as the evolution of an author’s idea on a specific question or the reconstruction of the author’s thought in different works. Pundit allows one to answer these questions thanks to the different functionalities it offers, which will be described in practice in the assignments. In particular, we will show how to perform annotations using the **triple composer**:

- Connecting different text fragments through appropriate properties in the same page and in different pages;
- Connecting a text fragment to an entity (ex. “grammar”);
- Adding free comments to text.

11.4 Description

Before creating annotations and experimenting with the proposed topics, first one must create a new notebook. This will make it easier to later analyse the results of the exercises.

To create a new notebook in Pundit and to give it a meaningful name:

- Go to Pundit (e.g. click on the following link: http://feed.thepund.it/?url=http://wittgensteinsource.org/Ts-310,1[2]et2[1]_n.rdf&conf=wab.js)
- In the Pundit top bar, click on “Your name” and then on “Manage Notebooks”
- In the “Create a new notebook” field, type the name of your new notebook.

- Set the new notebook as the "current notebook", click on the gear button next to it and choose "Set as Current Notebook", where the annotations you create from now on will be saved.

Assignments are then structured with a **goal** (which includes both a research objective and the description of what to obtain through Pundit) and a **hands-on** experience (in which all the steps required to reach the goal are described).

11.4.1 Assignment 1 - Grammar in the Brown Book

Goal

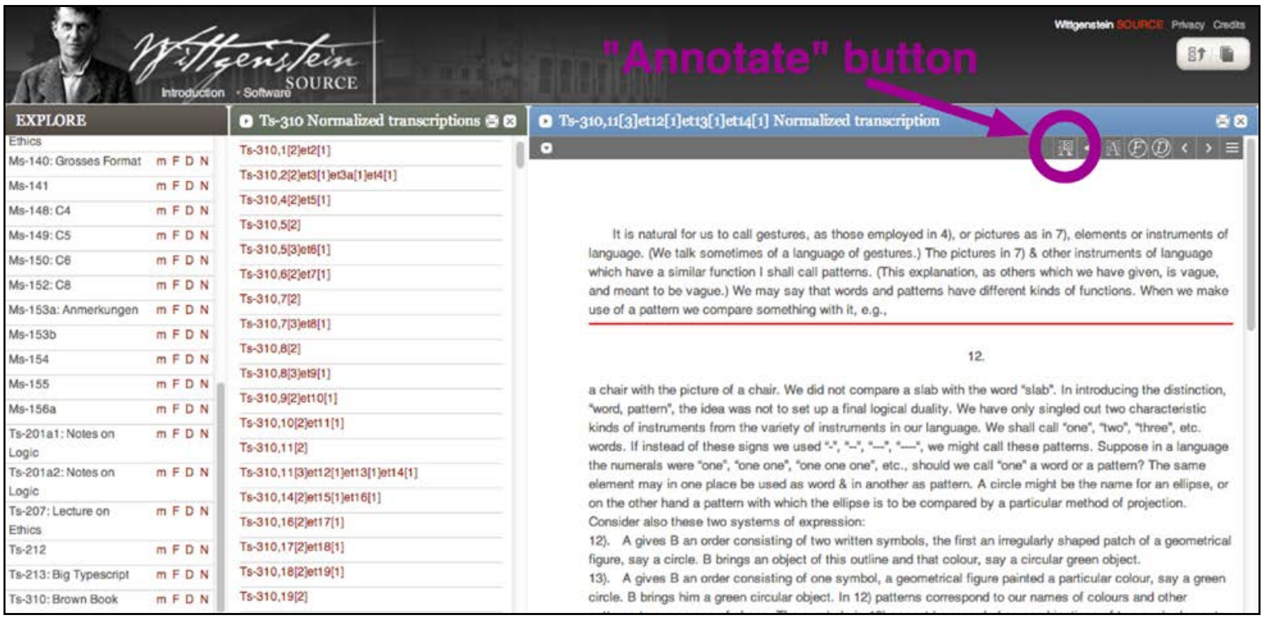
Annotate some Ts-310 (Brown book) passages which are relevant for the discussion of what **grammar** is.

To do so, you are going to establish a link between a portion of text and a concept taken from a controlled vocabulary. The final shape of the annotation will be:

- "Portion of text" : discusses : grammar

Hands-on

- Go to http://wittgensteinsource.org and browse Ts-310 to find a relevant text section.
- Once you have found the transcription you want to annotate, click the "Annotate" button (as shown in the figure below).



Ethics
Ms-140: Grosses Format m F D N
Ms-141 m F D N
Ms-148: C4 m F D N
Ms-149: C5 m F D N
Ms-150: C6 m F D N
Ms-152: C8 m F D N
Ms-153a: Anmerkungen m F D N
Ms-153b m F D N
Ms-154 m F D N
Ms-155 m F D N
Ms-156a m F D N
Ts-201a1: Notes on Logic m F D N
Ts-201a2: Notes on Logic m F D N
Ts-207: Lecture on Ethics m F D N
Ts-212 m F D N
Ts-213: Big Typescript m F D N
Ts-310: Brown Book m F D N

Ts-310,1[2]et2[1]
Ts-310,2[2]et3[1]et3a[1]et4[1]
Ts-310,4[2]et5[1]
Ts-310,5[2]
Ts-310,5[3]et6[1]
Ts-310,6[2]et7[1]
Ts-310,7[2]
Ts-310,7[3]et8[1]
Ts-310,8[2]
Ts-310,8[3]et9[1]
Ts-310,9[2]et10[1]
Ts-310,10[2]et11[1]
Ts-310,11[2]
Ts-310,11[3]et12[1]et13[1]et14[1]
Ts-310,14[2]et15[1]et16[1]
Ts-310,16[2]et17[1]
Ts-310,17[2]et18[1]
Ts-310,18[2]et19[1]
Ts-310,19[2]

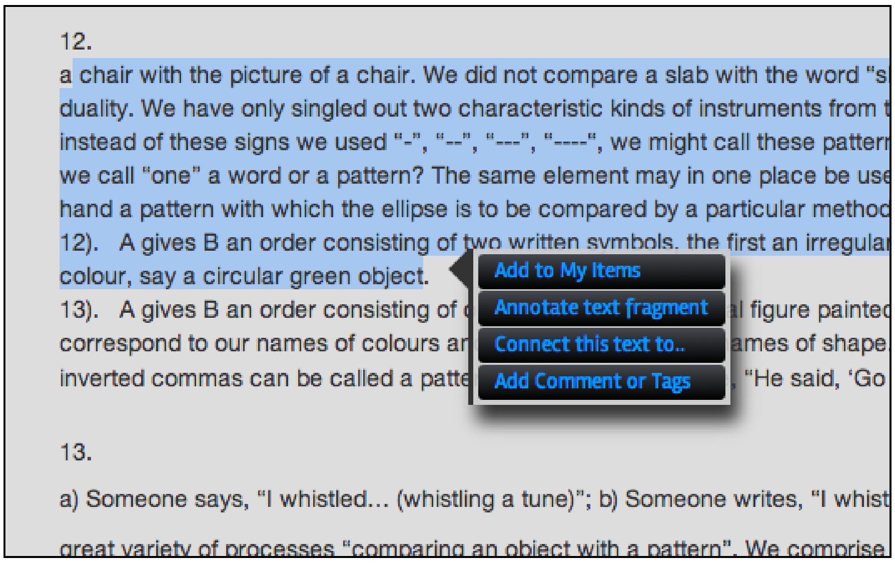
It is natural for us to call gestures, as those employed in 4), or pictures as in 7), elements or instruments of language. (We talk sometimes of a language of gestures.) The pictures in 7) & other instruments of language which have a similar function I shall call patterns. (This explanation, as others which we have given, is vague, and meant to be vague.) We may say that words and patterns have different kinds of functions. When we make use of a pattern we compare something with it, e.g.,

12.

a chair with the picture of a chair. We did not compare a slab with the word "slab". In introducing the distinction, "word, pattern", the idea was not to set up a final logical duality. We have only singled out two characteristic kinds of instruments from the variety of instruments in our language. We shall call "one", "two", "three", etc. words. If instead of these signs we used "-", "--", "---", "----", we might call these patterns. Suppose in a language the numerals were "one", "one one", "one one one", etc., should we call "one" a word or a pattern? The same element may in one place be used as word & in another as pattern. A circle might be the name for an ellipse, or on the other hand a pattern with which the ellipse is to be compared by a particular method of projection. Consider also these two systems of expression:
12). A gives B an order consisting of two written symbols, the first an irregularly shaped patch of a geometrical figure, say a circle. B brings an object of this outline and that colour, say a circular green object.
13). A gives B an order consisting of one symbol, a geometrical figure painted a particular colour, say a green circle. B brings him a green circular object. In 12) patterns correspond to our names of colours and other

Figure 22 The annotate button in WittgensteinSource.org brings users into annotation mode using Feed and Pundit.

- Feed, the annotation environment, will open in a new tab.
- Use the mouse to **select the text that you want to annotate**, then click "Annotate text fragment" from the contextual menu (as shown in the figure below).



12.

a chair with the picture of a chair. We did not compare a slab with the word "sl
duality. We have only singled out two characteristic kinds of instruments from t
instead of these signs we used "-", "--", "---", "----", we might call these patterr
we call "one" a word or a pattern? The same element may in one place be use
hand a pattern with which the ellipse is to be compared by a particular methoc
12). A gives B an order consisting of two written symbols, the first an irregular
colour, say a circular green object.
13). A gives B an order consisting of c l figure painted
correspond to our names of colours ar ames of shape
inverted commas can be called a patte , "He said, 'Go

13.

a) Someone says, "I whistled… (whistling a tune)"; b) Someone writes, "I whist
great variety of processes "comparing an object with a pattern". We comprise

Figure 23. Selection of text to be annotated n Pundit.

- The text fragment will be set as subject of your "triple" in the "triple composer" in the top-right part of the Pundit bar.
- **Choose the "discusses" relation** to use as predicate of your triple: click on the "predicate" yellow box, find it and click on the "+" button next to it (as shown in the figure below).

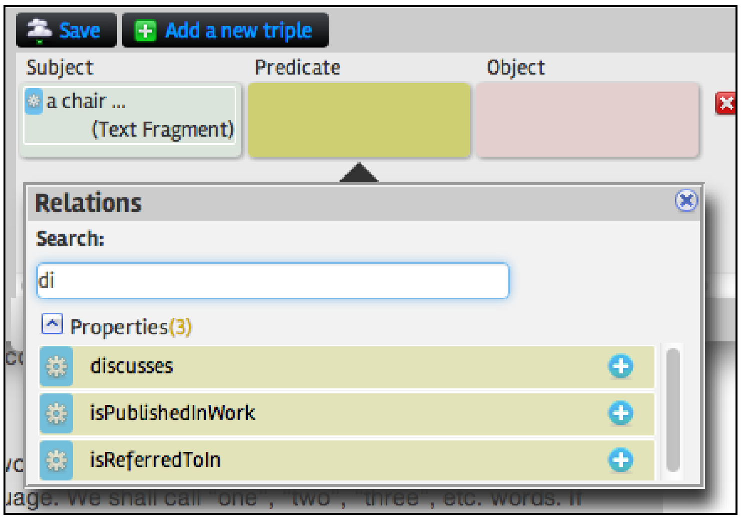


Figure 24. Composing triples with the triple composer: choosing a relation as predicate of the triple.

- **Choose an object**: click on the "object" red box and search for "grammar". Results will be returned from a variety of sources, find the one from the WAB:Subjects vocabulary and click on the "+" button next to it (as shown in the figure below).

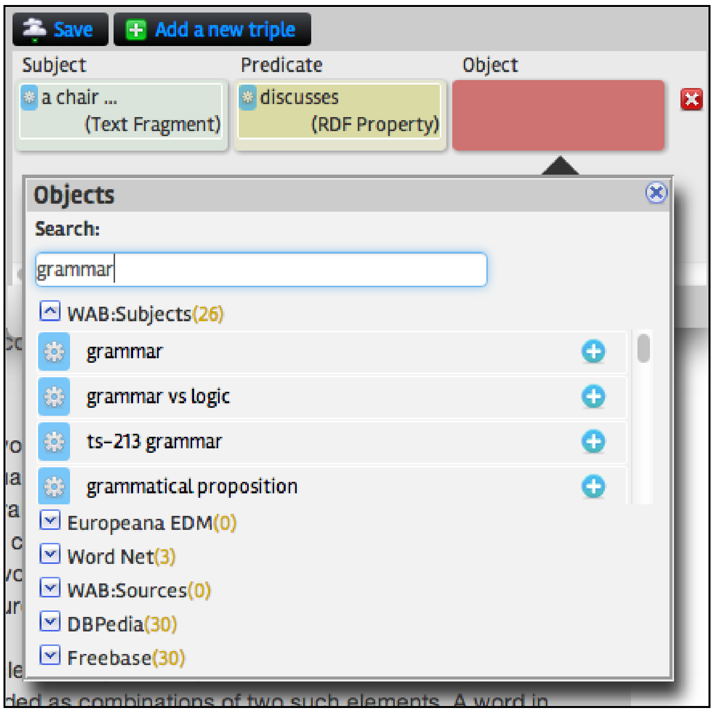


Figure 25. Searching domain vocabularies and external data sources to be used as object of the triples.

- An annotation can contain multiple triples. To add another, click on “Add new triple” and repeat the steps.
- You can remove items from the subject, predicate and object boxes by clicking on the little wheel button to the left of each item, and then choosing “remove item” from the contextual menu.
- When satisfied **click the “Save” button**: the annotation should load in a few seconds and be displayed in the page (as shown in the figure below).

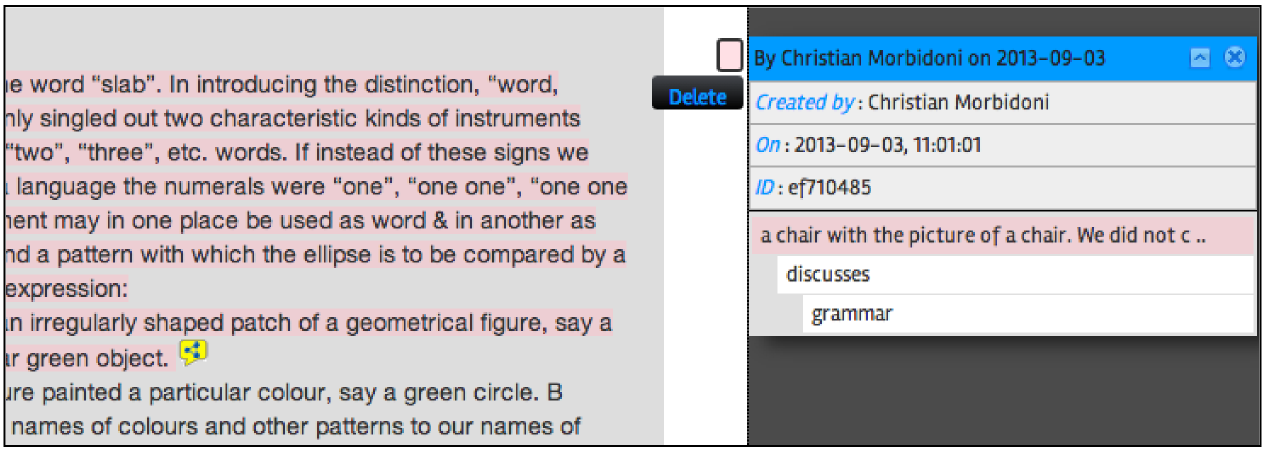

Delete
By Christian Morbidoni on 2013-09-03
Created by : Christian Morbidoni
On : 2013-09-03, 11:01:01
ID : ef710485
a chair with the picture of a chair. We did not c ..
discusses
grammar

Figure 26. An annotation on a text.

11.4.2 Assignment 2 - Different versions of the same text

Goal

There are a great number of "hasOtherVersion" Bemerkungen pairs for Ms-115 and Ts-310. Investigate such Bemerkungen pairs, finding pieces of revised text[3]. Connect them and explain the meaning of the link with a free comment.

To do so, you are going to establish one link between two portions of text and another link between one of these two texts and a free comment. The final annotation will include two triples, of the following shape:

- "Portion of text" : has revision : "Another portion of text"
- "Portion of text" : has comment : "Originally it said … and got changed because …"

Hands on

- Click one of the following links, which will open the two versions of a Bemerkung side by side

➔ http://feed.thepund.it/?lurl=http://www.wittgensteinsource.org/Ts-310,110[3]_n.rdf&rurl=http://www.wittgensteinsource.org/Ms-115,282[2]et283[1]_n.rdf&conf=wab.js

➔ http://feed.thepund.it/?lurl=http://www.wittgensteinsource.org/Ts-310,6[2]et7[1]_n.rdf&rurl=http://www.wittgensteinsource.org/Ms-115,124[4]et125[1]_n.rdf&conf=wab.js

➔ http://feed.thepund.it/?lurl=http://www.wittgensteinsource.org/Ts-310,39[2]et40[1]et41[1]_n.rdf&rurl=http://www.wittgensteinsource.org/Ms-115,161[2]et162[1]et163[1]_n.rdf&conf=wab.js

➔ http://feed.thepund.it/?lurl=http://www.wittgensteinsource.org/Ts-310,102[3]_n.rdf&rurl=http://www.wittgensteinsource.org/Ms-115,259[3]et260[1]_n.rdf&conf=wab.js

➔ http://feed.thepund.it/?lurl=http://www.wittgensteinsource.org/Ts-310,74[2]et75[1]et76[1]_n.rdf&rurl=http://www.wittgensteinsource.org/Ms-115,202[2]et203[1]_n.rdf&conf=wab.js

➔ http://feed.thepund.it/?lurl=http://www.wittgensteinsource.org/Ts-310,81[1]_n.rdf&rurl=http://www.wittgensteinsource.org/Ms-115,228[3]et229[1]_n.rdf&conf=wab.js

➔ http://feed.thepund.it/?lurl=http://www.wittgensteinsource.org/Ts-310,1[1]_n.rdf&rurl=http://www.wittgensteinsource.org/Ms-115,118[2]_n.rdf&conf=wab.js

➔ http://feed.thepund.it/?lurl=http://www.wittgensteinsource.org/Ts-310,1[1]_n.rdf&rurl=http://www.wittgensteinsource.org/Ms-115,118[3]_n.rdf&conf=wab.js

➔ http://feed.thepund.it/?lurl=http://www.wittgensteinsource.org/Ts-310,1[2]et2[1]_n.rdf&rurl=http://www.wittgensteinsource.org/Ms-

[3] For a complete list of such pairs see http://wab.uib.no/cost-a32_philospace/wittgenstein.owl or A. Pichler & D.C.P. Smith 2013: A list of correspondences between Wittgenstein Ts-310 and Ms-115ii. In: Mind, Language and Action. Contributions of the Austrian Ludwig Wittgenstein Society. Edited by Danièle Moyal-Sharrock, Annalisa Coliva and Volker Munz. pp. 311-318. (A) Kirchberg am Wechsel: ALWS.

115,118[4]_n.rdf&conf=wab.js

➔ http://feed.thepund.it/?lurl=http://www.wittgensteinsource.org/Ts-310,1[2]et2[1]_n.rdf&rurl=http://www.wittgensteinsource.org/Ms-115,118[5]et119[1]et119[2]_n.rdf&conf=wab.js

➔ http://feed.thepund.it/?lurl=http://www.wittgensteinsource.org/Ts-310,2[2]et3[1]et3a[1]et4[1]_n.rdf&rurl=http://www.wittgensteinsource.org/Ms-115,119[3]et120[1]_n.rdf&conf=wab.js

➔ http://feed.thepund.it/?lurl=http://www.wittgensteinsource.org/Ts-310,2[2]et3[1]et3a[1]et4[1]_n.rdf&rurl=http://www.wittgensteinsource.org/Ms-115,120[2]_n.rdf&conf=wab.js

➔ http://feed.thepund.it/?lurl=http://www.wittgensteinsource.org/Ts-310,2[2]et3[1]et3a[1]et4[1]_n.rdf&rurl=http://www.wittgensteinsource.org/Ms-115,120[3]et121[1]_n.rdf&conf=wab.js

➔ http://feed.thepund.it/?lurl=http://www.wittgensteinsource.org/Ts-310,4[2]et5[1]_n.rdf&rurl=http://www.wittgensteinsource.org/Ms-115,121[2]_n.rdf&conf=wab.js

➔ http://feed.thepund.it/?lurl=http://www.wittgensteinsource.org/Ts-310,5[2]_n.rdf&rurl=http://www.wittgensteinsource.org/Ms-115,121[4]et122[1]_n.rdf&conf=wab.js

➔ http://feed.thepund.it/?lurl=http://www.wittgensteinsource.org/Ts-310,5[2]_n.rdf&rurl=http://www.wittgensteinsource.org/Ms-115,122[2]_n.rdf&conf=wab.js

➔ http://feed.thepund.it/?lurl=http://www.wittgensteinsource.org/Ts-310,5[3]et6[1]_n.rdf&rurl=http://www.wittgensteinsource.org/Ms-115,122[3]et124[1]_n.rdf&conf=wab.js

➔ http://feed.thepund.it/?lurl=http://www.wittgensteinsource.org/Ts-310,5[3]et6[1]_n.rdf&rurl=http://www.wittgensteinsource.org/Ms-115,124[2]_n.rdf&conf=wab.js

➔ http://feed.thepund.it/?lurl=http://www.wittgensteinsource.org/Ts-310,5[3]et6[1]_n.rdf&rurl=http://www.wittgensteinsource.org/Ms-115,124[3]_n.rdf&conf=wab.js

➔ http://feed.thepund.it/?lurl=http://www.wittgensteinsource.org/Ts-310,7[2]_n.rdf&rurl=http://www.wittgensteinsource.org/Ms-115,125[2]_n.rdf&conf=wab.js

➔ http://feed.thepund.it/?lurl=http://www.wittgensteinsource.org/Ts-310,7[3]et8[1]_n.rdf&rurl=http://www.wittgensteinsource.org/Ms-115,125[3]et126[1]_n.rdf&conf=wab.js

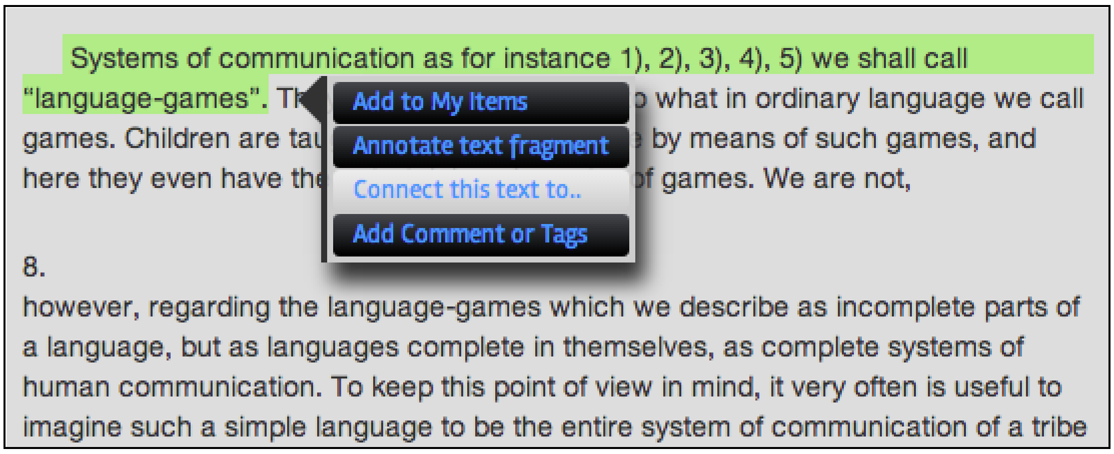
➔ http://feed.thepund.it/?lurl=http://www.wittgensteinsource.org/Ts-310,8[2]_n.rdf&rurl=http://www.wittgensteinsource.org/Ms-115,126[2]_n.rdf&conf=wab.js

➔ http://feed.thepund.it/?lurl=http://www.wittgensteinsource.org/Ts-310,8[2]_n.rdf&rurl=http://www.wittgensteinsource.org/Ms-115,126[3]_n.rdf&conf=wab.js

➔ http://feed.thepund.it/?lurl=http://www.wittgensteinsource.org/Ts-310,8[2]_n.rdf&rurl=http://www.wittgensteinsource.org/Ms-115,126[4]_n.rdf&conf=wab.js

➔ [http://feed.thepund.it/?lurl=http://www.wittgensteinsource.org/Ts-310,8[2]_n.rdf&rurl=http://www.wittgensteinsource.org/Ms-115,126[5]et127[1]_n.rdf&conf=wab.js](http://feed.thepund.it/?lurl=http://www.wittgensteinsource.org/Ts-310,8[2]_n.rdf&rurl=http://www.wittgensteinsource.org/Ms-115,126[5]et127[1]_n.rdf&conf=wab.js)

- Once you have chosen a text in the first version, select it and click "Connect this text to…" (as shown in the figure below).

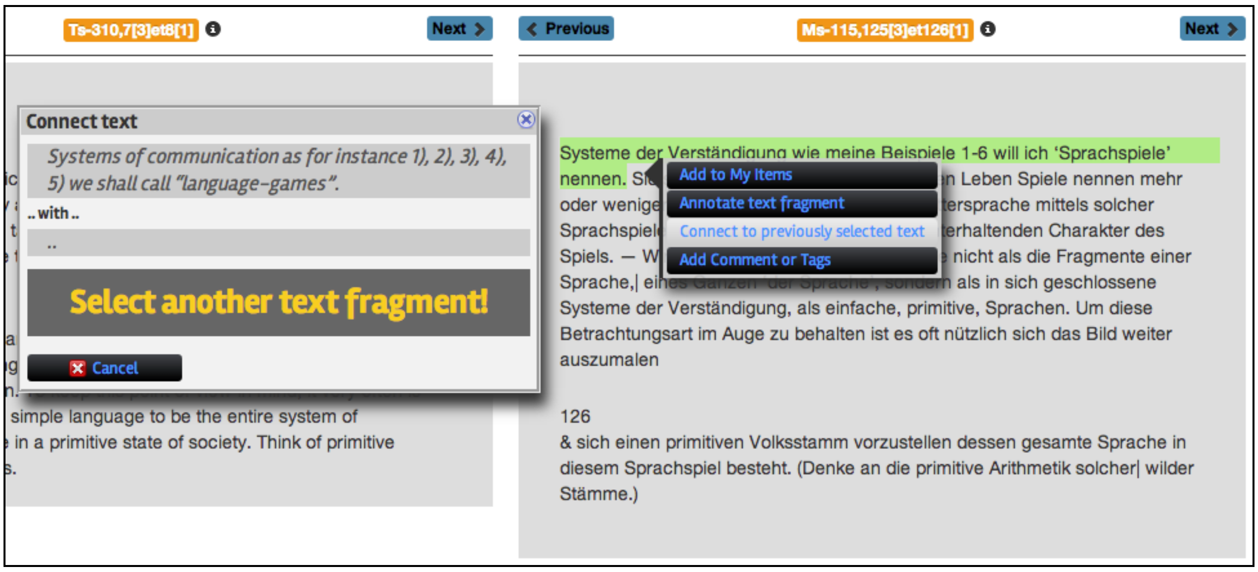


Systems of communication as for instance 1), 2), 3), 4), 5) we shall call "language-games". Th… what in ordinary language we call games. Children are tau… by means of such games, and here they even have the… of games. We are not,

8.
however, regarding the language-games which we describe as incomplete parts of a language, but as languages complete in themselves, as complete systems of human communication. To keep this point of view in mind, it very often is useful to imagine such a simple language to be the entire system of communication of a tribe

Figure 27. Selecting a text to be connected to some other text.

- The "Connect text" window can be safely closed clicking the "x" icon on the top right, without losing the first bit of selected text.
- To change the first bit of text, click the "cancel" button, and start over.
- Once you have chosen a text in the second version, select it and click "Connect to previously selected text" (as shown in the figure below).



Connect text

Systems of communication as for instance 1), 2), 3), 4), 5) we shall call "language-games".

.. with ..

..

Select another text fragment!

simple language to be the entire system of
in a primitive state of society. Think of primitive

Systeme der Verständigung wie meine Beispiele 1-6 will ich 'Sprachspiele' nennen. Si… n Leben Spiele nennen mehr oder wenige… tersprache mittels solcher Sprachspiel… erhaltenden Charakter des Spiels. — W… nicht als die Fragmente einer Sprache,| ei… als in sich geschlossene Systeme der Verständigung, als einfache, primitive, Sprachen. Um diese Betrachtungsart im Auge zu behalten ist es oft nützlich sich das Bild weiter auszumalen

126
& sich einen primitiven Volksstamm vorzustellen dessen gesamte Sprache in diesem Sprachspiel besteht. (Denke an die primitive Arithmetik solcher| wilder Stämme.)

Figure 27. Choosing the target text.

- If not satisfied with the selected texts, click the "cancel" button, and start again.

- Click the “Go to save” button. The triple composer will show the two texts in the “subject” and “object” boxes, asking for a predicate to connect them (as shown in the figure below).

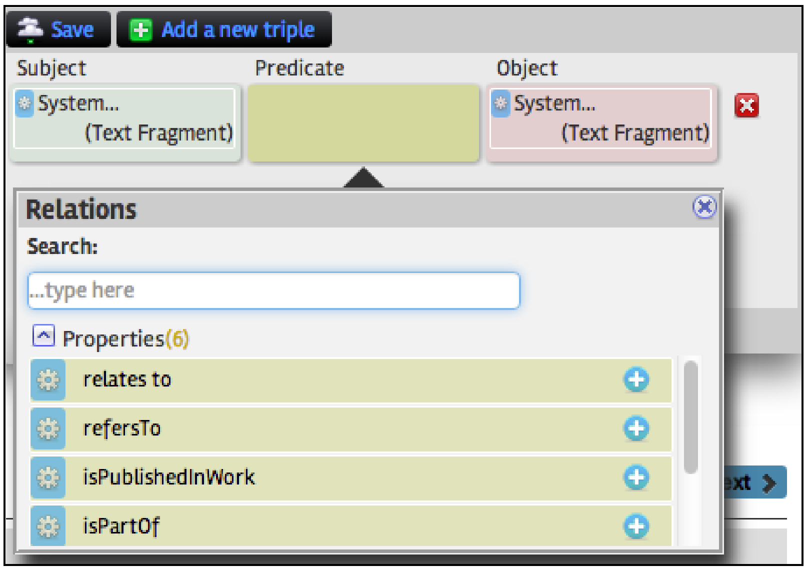


Figure 28. Choosing a relation among the allowed ones given the object and the subject already appointed.

- Find the “has revision” predicate, and click the “+” button next to it or its label to add it to the predicate box.
- Add a new triple by clicking on the “Add a new triple” button.
- Drag one of the two texts from the lower triple to the upper “subject” box. This will copy the text into the new triple (as shown in the figure below).

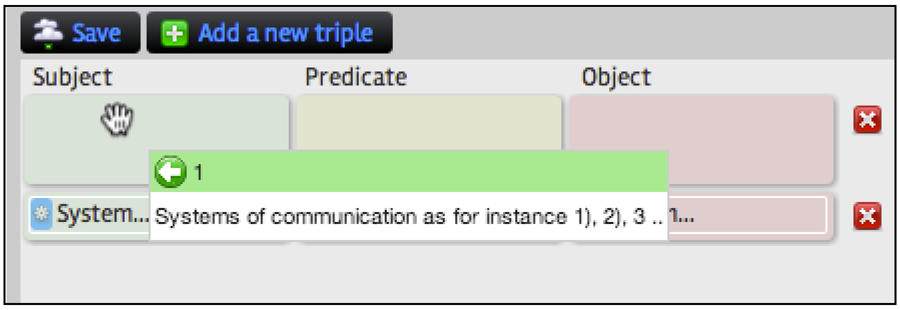
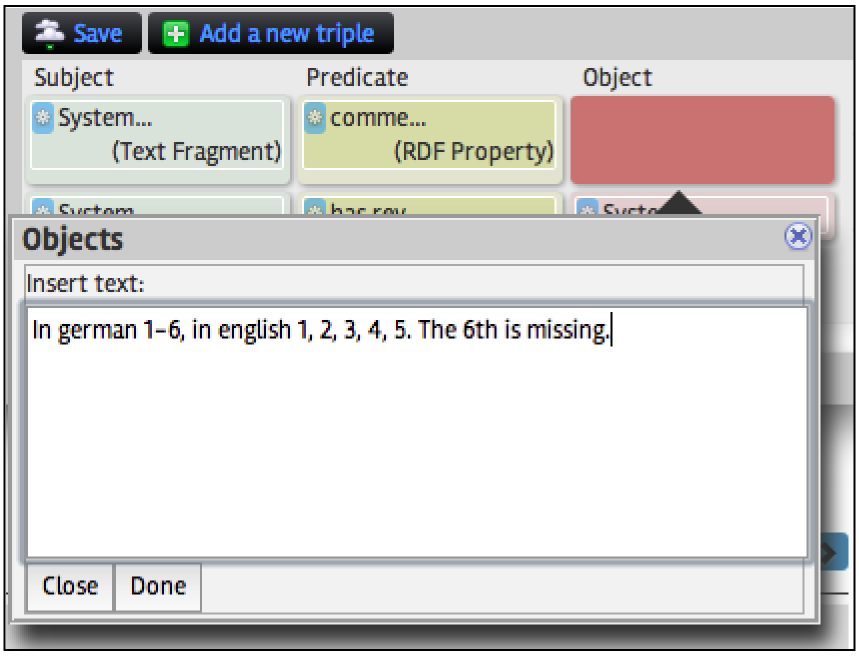


Figure 30. Drag & Drop to copy items across different triples.

- Click on the upper “predicate” box, and add the “comment (text)” relation by clicking on the “+” icon next to it or on its label.
- Click on the upper “object” box. Insert the comment in the text area that pops up (as shown in the figure below).



Subject | Predicate | Object
System... (Text Fragment) | comme... (RDF Property) |

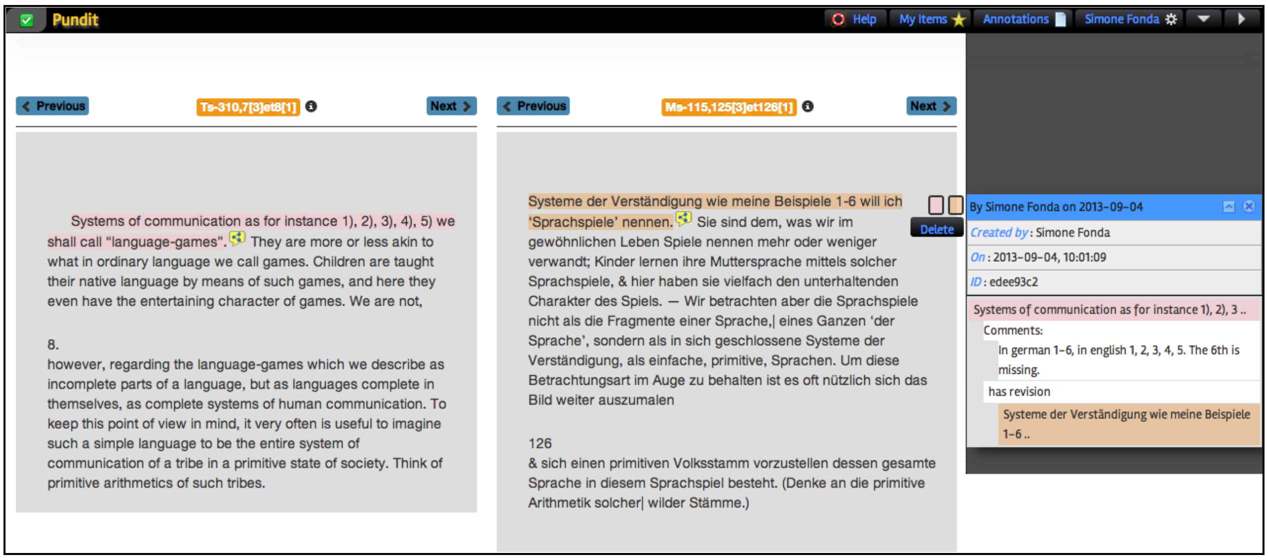
Objects

Insert text:

In german 1-6, in english 1, 2, 3, 4, 5. The 6th is missing.

Figure 31. Once the comment relation is appointed, Pundit requires a text as object. You can type and press done.

- Click on “done”. The comment will be inserted as “object” for the upper triple.
- **Click the “Save” button**: the annotation should load in a few seconds and be displayed in the page (as shown in the figure below).



Ts-310,7[3]et8[1]

Systems of communication as for instance 1), 2), 3), 4), 5) we shall call “language-games”. They are more or less akin to what in ordinary language we call games. Children are taught their native language by means of such games, and here they even have the entertaining character of games. We are not,

8.
however, regarding the language-games which we describe as incomplete parts of a language, but as languages complete in themselves, as complete systems of human communication. To keep this point of view in mind, it very often is useful to imagine such a simple language to be the entire system of communication of a tribe in a primitive state of society. Think of primitive arithmetics of such tribes.

Ms-115,125[3]et126[1]

Systeme der Verständigung wie meine Beispiele 1-6 will ich ‘Sprachspiele’ nennen. Sie sind dem, was wir im gewöhnlichen Leben Spiele nennen mehr oder weniger verwandt; Kinder lernen ihre Muttersprache mittels solcher Sprachspiele, & hier haben sie vielfach den unterhaltenden Charakter des Spiels. — Wir betrachten aber die Sprachspiele nicht als die Fragmente einer Sprache,| eines Ganzen ‘der Sprache’, sondern als in sich geschlossene Systeme der Verständigung, als einfache, primitive, Sprachen. Um diese Betrachtungsart im Auge zu behalten ist es oft nützlich sich das Bild weiter auszumalen

126
& sich einen primitiven Volksstamm vorzustellen dessen gesamte Sprache in diesem Sprachspiel besteht. (Denke an die primitive Arithmetik solcher| wilder Stämme.)

By Simone Fonda on 2013-09-04
Created by: Simone Fonda
On: 2013-09-04, 10:01:09
ID: edee93c2
Systems of communication as for instance 1), 2), 3 ..
Comments:
In german 1-6, in english 1, 2, 3, 4, 5. The 6th is missing.
has revision
Systeme der Verständigung wie meine Beispiele 1-6 ..

Figure 32. An annotations connecting two different web pages. Both are aggregated and visualised at the same time in Feed.

11.4.3 Assignment 3 - Language game developments

Goal

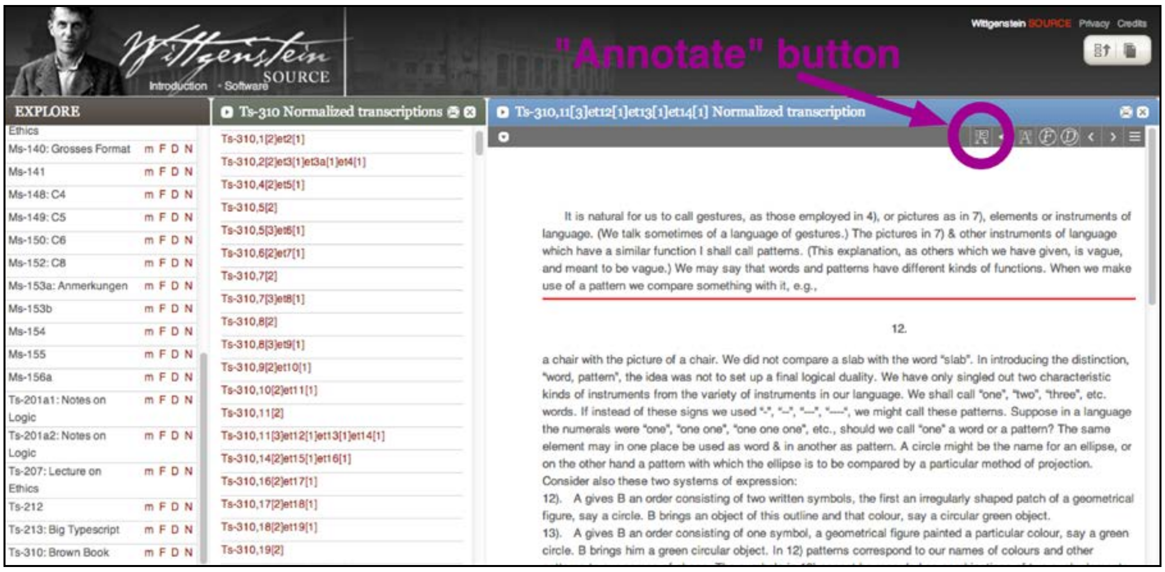
In Ts-310 (the Brown Book), connect one language game to another Bemerkung’s language game, which is a further development of the first one.

To do so, you are going to establish two links between two portions of text coming from different pages and the concept "Language game" taken from a controlled vocabulary. This will state that the identified portions are both language games. A third link will then connect the two texts together. The final annotation will include three triples, of the following shape:

- "Portion of text" : discusses : Language game
- "Another portion of text" : discusses : Language game
- "Portion of text" : refers to : "Another portion of text"

Hands-on

- Go to http://wittgensteinsource.org and browse Ts-310 to find a relevant text section.
- Once you have found the text you want to annotate, click the "annotate" button (as shown in the figure below).



Wittgenstein SOURCE
Introduction · Software
Wittgenstein SOURCE Privacy Credits
"Annotate" button
EXPLORE
Ts-310 Normalized transcriptions
Ts-310,11[3]et12[1]et13[1]et14[1] Normalized transcription

Ethics				
Ms-140: Grosses Format	m	F	D	N
Ms-141	m	F	D	N
Ms-148: C4	m	F	D	N
Ms-149: C5	m	F	D	N
Ms-150: C6	m	F	D	N
Ms-152: C8	m	F	D	N
Ms-153a: Anmerkungen	m	F	D	N
Ms-153b	m	F	D	N
Ms-154	m	F	D	N
Ms-155	m	F	D	N
Ms-156a	m	F	D	N
Ts-201a1: Notes on Logic	m	F	D	N
Ts-201a2: Notes on Logic	m	F	D	N
Ts-207: Lecture on Ethics	m	F	D	N
Ts-212	m	F	D	N
Ts-213: Big Typescript	m	F	D	N
Ts-310: Brown Book	m	F	D	N

Ts-310,1[2]et2[1]
Ts-310,2[2]et3[1]et3a[1]et4[1]
Ts-310,4[2]et5[1]
Ts-310,5[2]
Ts-310,5[3]et6[1]
Ts-310,6[2]et7[1]
Ts-310,7[2]
Ts-310,7[3]et8[1]
Ts-310,8[2]
Ts-310,8[3]et9[1]
Ts-310,9[2]et10[1]
Ts-310,10[2]et11[1]
Ts-310,11[2]
Ts-310,11[3]et12[1]et13[1]et14[1]
Ts-310,14[2]et15[1]et16[1]
Ts-310,16[2]et17[1]
Ts-310,17[2]et18[1]
Ts-310,18[2]et19[1]
Ts-310,19[2]

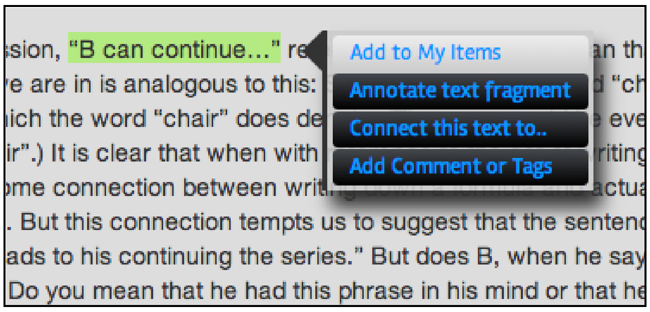
It is natural for us to call gestures, as those employed in 4), or pictures as in 7), elements or instruments of language. (We talk sometimes of a language of gestures.) The pictures in 7) & other instruments of language which have a similar function I shall call patterns. (This explanation, as others which we have given, is vague, and meant to be vague.) We may say that words and patterns have different kinds of functions. When we make use of a pattern we compare something with it, e.g.,

12.

a chair with the picture of a chair. We did not compare a slab with the word "slab". In introducing the distinction, "word, pattern", the idea was not to set up a final logical duality. We have only singled out two characteristic kinds of instruments from the variety of instruments in our language. We shall call "one", "two", "three", etc. words. If instead of these signs we used "-", "--", "---", "----", we might call these patterns. Suppose in a language the numerals were "one", "one one", "one one one", etc., should we call "one" a word or a pattern? The same element may in one place be used as word & in another as pattern. A circle might be the name for an ellipse, or on the other hand a pattern with which the ellipse is to be compared by a particular method of projection. Consider also these two systems of expression:
12). A gives B an order consisting of two written symbols, the first an irregularly shaped patch of a geometrical figure, say a circle. B brings an object of this outline and that colour, say a circular green object.
13). A gives B an order consisting of one symbol, a geometrical figure painted a particular colour, say a green circle. B brings him a green circular object. In 12) patterns correspond to our names of colours and other

Figure 33. The annotate button in WittgensteinSource.org.

- Feed, the annotation environment, will open.
- Use the mouse to select the text that you want to annotate and then **click "add to my items"** from the contextual menu (as shown in the figure below).



sion, "B can continue..." re
e are in is analogous to this:
ich the word "chair" does de
ir".) It is clear that when with
me connection between writ
. But this connection tempts us to suggest that the senten
ads to his continuing the series." But does B, when he say
Do you mean that he had this phrase in his mind or that he

Figure 35. Selecting a text.

- The text fragment will be saved into your items, which you can see by clicking the “my items” shortcut button in the Pundit top bar. Moreover, a star icon will be added to the page, next to the selected text fragment (as shown in the figure below).

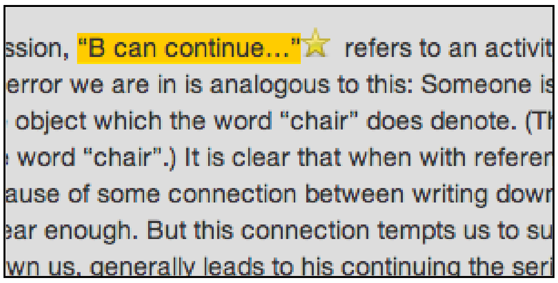

ssion, “B can continue...” refers to an activit
error we are in is analogous to this: Someone is
object which the word “chair” does denote. (Th
word “chair”.) It is clear that when with referen
ause of some connection between writing down
ear enough. But this connection tempts us to su
wn us, generally leads to his continuing the seri

Figure 36. Text already added to MyItmes is marked with a star.

- Go back to http://wittgensteinsource.org and browse Ts-310 to find an evolution of the saved language game.
- Once located, click the “annotate” button to open the Feed annotation environment on the new source.
- Select the text that you want to link to the previous language game and click “annotate text fragment”.
- The text fragment will be set as subject of your “triple” in the “triple composer” in the top-right part of the Pundit bar.
- **Choose the “discusses” relation** to use as predicate of your triple.
- Click on the “object” box and search for “language game” from the WAB:Subjects vocabulary, and add it to the “object” box.
- Add a new triple clicking on the “Add a new triple” button.
- Drag the previously saved text fragment from my items to the new “subject” box (as shown in the figure below).

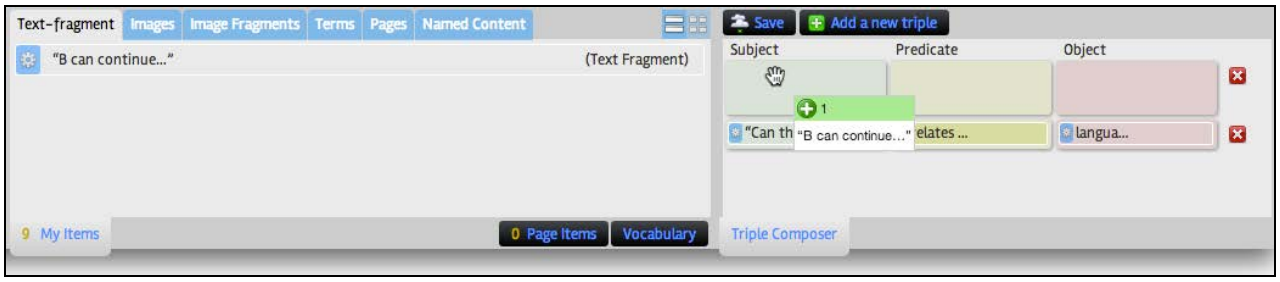


Figure 37. Drag & draop to populate triples.

- Add the “discusses” relation in the upper triple “predicate” box.
- Click on the “object”, search for “language game” and add it as object, or, as a shortcut, you can drag it from the lower triple’s “object” box to the upper one.
- The annotation should now look like the figure below.

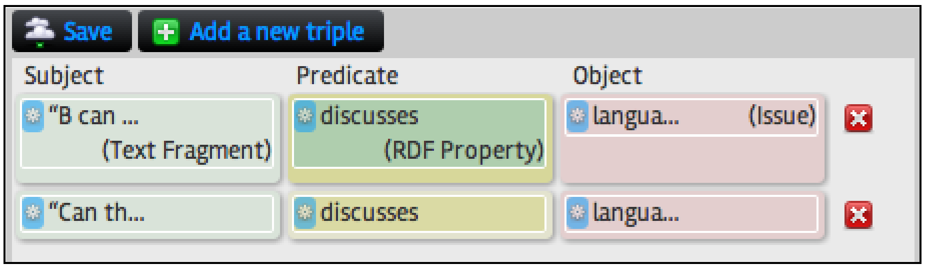
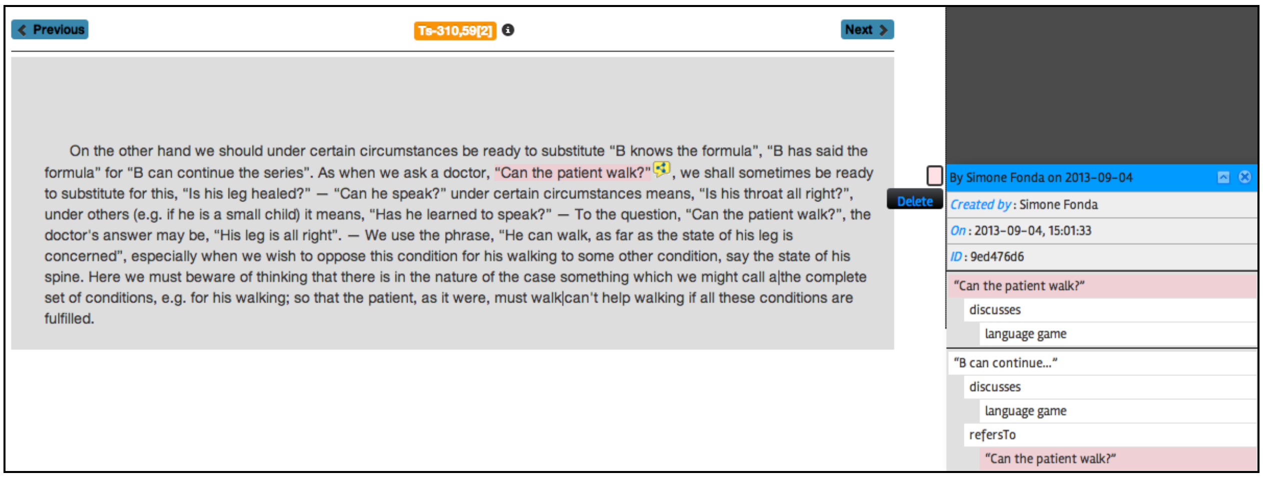


Figure 38. Multiple triples in the same annotation.

- Add a new triple (the third) clicking on the "Add a new triple" button.
- Drag the first triple's subject into the third triple's "subject" box and the second triple's subject into the third triple's "object" box. The annotation should now look as the figure below.
- Click on the third triple's "predicate" box and add the "refers to" predicate
- **Click the "Save" button**: the annotation should load in a few seconds and be displayed in the page (as shown in the figure below).



On the other hand we should under certain circumstances be ready to substitute "B knows the formula", "B has said the formula" for "B can continue the series". As when we ask a doctor, "Can the patient walk?", we shall sometimes be ready to substitute for this, "Is his leg healed?" — "Can he speak?" under certain circumstances means, "Is his throat all right?", under others (e.g. if he is a small child) it means, "Has he learned to speak?" — To the question, "Can the patient walk?", the doctor's answer may be, "His leg is all right". — We use the phrase, "He can walk, as far as the state of his leg is concerned", especially when we wish to oppose this condition for his walking to some other condition, say the state of his spine. Here we must beware of thinking that there is in the nature of the case something which we might call a|the complete set of conditions, e.g. for his walking; so that the patient, as it were, must walk|can't help walking if all these conditions are fulfilled.

By Simone Fonda on 2013-09-04
Created by : Simone Fonda
On : 2013-09-04, 15:01:33
ID : 9ed476d6

"Can the patient walk?"
discusses
language game

"B can continue..."
discusses
language game
refersTo
"Can the patient walk?"

Figure 39. Open the annotation in the Pundit side-bar.

12 T2 - Deploying the annotation environment

12.1 Introduction

Pundit is a configurable and flexible annotation environment that can be adapted to different conceptual and technical contexts.

As already seen in tutorial number 1 (Creating annotations with Pundit), a number of facilities, such as the possibility to include custom annotation vocabularies or the ability to search existing Linked Data repositories (as Freebase[4], DBpedia[5] and Wordnet), allows the user to tailor the tool to be used in a specific community.

Once the tool is properly configured, there are a number of possible technical solutions to deliver the annotation environment to final users.

This tutorial demonstrates with practical examples how Pundit can be integrated into existing Digital Libraries or directly distributed to the end-users in the form of a bookmarklet.

12.2 Audience

This tutorial is specifically targeted at **developers** or **software integrators** who want to test Pundit in a real world scenario. In general, it can be useful even to non-technicians in order to understand how the system works behind the scenes.

12.3 Objective

In this tutorial you will learn:

- How to create a vocabulary and edit the Pundit configuration accordingly
- How to include the Pundit javascript library in your web site
- How to link your web site to Feed, and use Pundit as a service
- How to deploy a boomarklet with your own configuration and publish it on the web.

12.4 Description

12.4.1 Creating a vocabulary

Annotation is a highly domain-dependent task, as different scholarly communities have different needs in terms of what kind of knowledge they wish to express via annotations.

To address this, Pundit can be configured to include a number of vocabularies, tailored for a specific annotation scenario.

[4] http://www.freebase.com/
[5] http://dbpedia.org

Vocabularies are of two kinds:

- **Entities taxonomies**, are hierarchical vocabularies where a set of relevant entities are collected to be used in annotations;
- **Relation sets**, list all the possible relations between annotated digital items and entities or between two distinct annotated digital items. For example, a "cites" relations could be used to connect two distinct texts from different web pages, or a "depicts" relation could be used to specify that a given image depicts an entity (e.g. a person or a city).

In Pundit, vocabularies are expressed in a specific JSON format, where each entity or relation is identified by a URI (in line with the Linked Data paradigm).

Such JSON files are imported at run time when Pundit is loaded on a web page, so they need to be available at a stable URL.

12.4.2 Entities taxonomies

A possible solution for building an entity taxonomy is that of selecting relevant Linked Data resources from one of the open datasets available on the Web of Data, as, for example, Freebase.com.

This approach was used in the Timeline Demo (described at http://www.thepund.it/visualization-demos/timeline-demo/), and resulted in the JSON vocabulary you can download at http://metasound.dibet.univpm.it/timelinejs/pundit_conf/timeline_demo_taxonomy.jsonp.

Let us use this example to illustrate the main characteristics of a vocabulary.

A JSON vocabulary has the following structure:

```
_PUNDIT.vocab.initJsonpVocab({
    "error_code": "200",
    "error_message": "OK",
    "result": {
        "vocab_id": "101",
        "vocab_label": "Timeline demo taxonomy",
        "vocab_type": "subjects",
        "items": [
            ...
        ]
 }})
```

First we notice that, as Pundit loads vocabularies on the fly via cross-domain ajax calls, the file has to be served in using the JSONP mechanism, by enclosing the content in a javascript callback. The name of the callback is by convention *_PUNDIT.vocab.initJsonpVocab*.

The error_code and error_message parameters can be used to signal malfunctioning to the client. This is useful when vocabularies are generated at request time by a server side application.

As you can see, a vocabulary must have a unique *vocab_id* and a *vocab_label*. The vocab type must be set to "subjects" in the case of an entity taxonomy. The *items* array contains the actual entities enclosed in the vocabulary. Let us see how single items are represented.

```
{
   "value": "http://purl.org/net7/korbo/item/33059",
   "children": [
      { "_reference": "http://rdf.freebase.com/ns/m/02h40lc"},
      { "_reference": "http://rdf.freebase.com/ns/m/04306rv"},
      { "_reference": "http://rdf.freebase.com/ns/m/0349s"}
   ],
   "label": "Languages",
   "description": "This is the root item of my example vocabulary",
   "is_root_node": true,
   "nodetype": "container"
}
```

Each item must have a unique URI to identify it within the vocabulary (*value* attribute), and can have a number of *children* items, specified by means of a *reference* attribute that points to the URI of another item in the same vocabulary. A *label* and a *description* should be present to provide information to end-users. In the case where the item is at root level in the taxonomy the *is_root_node* attribute must be set to true.

Furthermore, we can distinguish between two types of items:

- Those that represent "categories" of items, and that will be shown as folders containing children items. In this case the *nodetype* attribute must be set to *container*.
- Those that represent actual entities to be used in annotations (*nodetype*: *node*), as in the following example:

```
{
   "value": "http://rdf.freebase.com/ns/m/05qmj"
   "label": "Plato",
   "description": "Plato was a Classical Greek philosopher, mathematician, student of
Socrates, writer of philosophical dialogues, and founder of the Academy in Athens, the
first institution of higher learning in the Western world. Along with his mentor,
Socrates, and his student, Aristotle…",
   "image": "https://www.googleapis.com/freebase/v1/image/m/05qmj",
   "rdftype": [
      "http://www.freebase.com/schema/common/topic",
      "http://www.freebase.com/schema/people/person",
      "http://example.org/ontology/AncientPhilosopher"
   ],
   "nodetype": "node",
}
```

In addition to previously described attributes, a *node* item must have a *rdftype* attribute, an array of URIs identifying a number of RDF classes that the item belongs to.

Notice that, as we are re-using entities from Freebase, a good practice is to adopt the original Freebase URI as item *value*, as well as to select a number of RDF classes from the Freebase schema. However, it is always possible to specify "home-made" RDF classes (e.g. *http://example.org/ontology/AncientPhilosopher*). Optionally, an image can be associated to an item.

12.4.3 Relations sets

Relations sets are represented in JSON with the same syntax as Entities Taxonomies. Here is the basic structure (see the full example at http://metasound.dibet.univpm.it/timelinejs/pundit_conf/timeline_demo_relations.jsonp):

```
_PUNDIT.vocab.initJsonpVocab({
    "error_code": "200",
    "error_message": "OK",
    "result": {
        "vocab_label": "Timeline Demo relations",
        "vocab_id": "101",
        "vocab_type": "predicates",
        "items": [

...
        ]
    }})
```

Note that the *vocab_type* attribute must be set to “predicates” in the case of a Relations Set vocabulary. The following is an example of a relation item:

```
{
      "nodetype": "node",
       "rdftype": ["http://www.w3.org/1999/02/22-rdf-syntax-ns#Property"],
       "label": "depicts",
       "description": "An image or part of an image depicts something",
       "domain": ["http://xmlns.com/foaf/0.1/Image",
                    "http://purl.org/pundit/ont/ao#fragment-image"],
      "range": [],
       "value": "http://xmlns.com/foaf/0.1/depicts"
}
```

The *domain* and *range* attributes can be used to assist users in the usage of a relation, avoiding inconsistent or meaningless annotations. In this example, the relation *depicts* can be used to put in relation an item of type image (e.g. to annotate an image included in a web page or a fragment of an image) to a generic item. Declaring *range* or *domain* values of the two attributes can be set to a list of RDF classes, including those declared in a Entities Taxonomy vocabulary.

In this case, we used two Pundit built-in classes as *domain* and a void *range.* Note that void ranges or domains behave as wildcard, meaning that all types of items can be used respectively as object or subject of a triple when the predicate is set to the corresponding relation.

Here is the list of built-in RDF classes that you can use in your Relations Sets:

- *http://purl.org/pundit/ont/ao#fragment-text*
  Use it as domain or range when you want a relation to be usable to annotate a portion of text selected by the user within a web page
- *http://xmlns.com/foaf/0.1/Image*
  Use it as domain or range when you want a relation to be usable to annotate an image within a web page
- *http://purl.org/pundit/ont/ao#fragment-image*
  Use it as domain or range when you want a relation to be usable to annotate an portion of an image within a web page
- *http://purl.org/pundit/ont/ao#WebPage*
  Use it as domain or range when you want a relation to be usable to annotate a whole web page
- *http://www.w3.org/2000/01/rdf-schema#Literal*
  Use it as range when you want to use a relation to be used to annotate an item with some text (e.g. to add a free-text comment)

12.4.4 Configuring Pundit

Once you created a set of vocabularies, they have to be put online and resolvable via a stable URL. Then, to include them in a specific Pundit deployment you just have to edit the Pundit configuration accordingly.

The configuration is a Javascript file like the following:

```
var punditConfig = {
  debugAllModules: false,
  annotationServerBaseURL : 'http://as.thepund.it:8080/annotationserver/',
  vocabularies: [
     'http://example.org/my-entities-taxonomy.jsonp',
     'http://example.org/my-relations-set.jsonp'
    ],
    useBasicRelations: false,
};
```

Where the *vocabulary* attribute contains a number of URLs pointing to some vocabularies.

By setting the *useBasicRelations* attribute to *true,* you can include the default relations set built-in in Pundit.

The configuration includes a number of other settings, the most important being the *annotationServerBaseURL,* which tells Pundit which instance of the Pundit Server annotations will be read from and written to.

At *http://as.thepund.it:8080/annotationserver/* you can find a public Pundit Server installation, feel free to use it for testing or prototyping, but keep in mind that persistence of annotations is not guaranteed!

If you want to set up a production environment, you can download the Pundit Server from GitHub (http://github.com/net7/pundit-server) and install it into your own server, following the instructions that you can find at http://www.thepund.it/documentation/deploy-and-configure-the-pundit-server/.

A complete set of configuration parameters can be found in the client documentation at http://docs.thepund.it/classes/pundit.Configuration.html.

12.4.5 Use Pundit as a JavaScript library

Now that you have learned how to customize your Pundit installation, let us see what possibilities you have to deliver the Pundit annotation environment to your users.

A common way to proceed is to include Pundit as a JavaScript library into your web site. To do so, you just need to download the Pundit client from GitHub (http://github.com/net7/pundit), place it somewhere into your web server and add the following to each web page you want to become "annotatable":

```
<link rel="stylesheet" href="$BUILD/css/pundit.css" type="text/css">
<script src="$BUILD/dojo/dojo/dojo.js.uncompressed.js" type="text/javascript"></script>
<script src="$PUNDIT_CONF" type="text/javascript"></script>
<script>
    dojo.registerModulePath("pundit", "../../src");
    dojo.require('pundit.Init');
</script>
```

where *$BUILD* indicates the path to Pundit on your server, and $PUNDIT_CONF indicates the path to a Pundit configuration file. You can find the default configuration file, named pundit_conf.js, in the home directory.

As Pundit loads the configuration on the fly a web page is loaded, you can also use different configurations for different pages.

12.4.6 Use Pundit as a bookmarklet

Sometimes it could be useful to allow your users to annotate pages outside the boundaries of your web site and, in general, pages that are not under your control.

One possible solution is to deploy Pundit as a bookmarklet (http://en.wikipedia.org/wiki/Bookmarklet).

12.4.6.1 How to build your own Pundit bookmarklet

Requirements

- A bookmarklet, to be correctly loaded, requires an absolute URL to be loaded from. Be sure to have some publicly available space on the web.
- You will need dojo's SDK package, version 1.6.x (tested with 1.6.1, available at http://download.dojotoolkit.org/release-1.6.1/), the file name is dojo-release-1.6.1-src.tar.gz . By default the script looks for them in a directory called dojo_sdk_1.6.1, at the same level of Pundit's /src directory. This name is configurable, as will be explained below.
- You will need to patch these sources to enable the "withCredentials" xhr header. Let's say you extracted the .tar.gz inside the pundit tree, to patch it enter the directory and use patch: # cd dojo-release-1.6.1-src # patch dojo/_base/xhr.js < ../bookmarklet_build/dojo_161.patch
- The bookmarklet will need a bootstrap file, we usually call it InitBookmarklet.js. It will also need an .html file which has the required Javascript to drag the bookmarklet to your bookmark bar.
- Both these files will be automatically created by default in /src/InitBookmarklet.js and /examples/bookmarklet.html (from Pundit sources base directory).

Configure the bookmarklet

- In order to build the bookmarklet, you will need to configure the script build_bookmarklet.sh, customising the variables sdk (path to the afore mentioned SDK package), ver (the name you want to give to the bookmarklet) and bmpath (the absolute URL to load the bookmarklet from)
- To configure the bookmarklet run-time, you must modify and customize the punditConfig variable you can find in InitBookmarklet.js-template.

Build the bookmarklet

Just launch $BUILD/bookmarklet_build/build_bookmarklet.sh

Put your Pundit bookmarklet online

For your users to download and install your Pundit bookmarklet, we might want to create a simple page like the following:

http://thepund.it/bm/demo-timeline/

To do so copy the folder *$BUILD/bookmarklet_build/dojo* to your web server and add the file *InitBookmarklet.js, your configuration,* at the same level of the dojo folder.

You can start from the file *$BUILD/bookmarklet_build/InitBookmarklet.js-template* to obtain a configuration file similar to the one you find at http://thepund.it/bm/demo-timeline/InitBookmarklet.js.

Your users will now only have to drag the bookmarklet and put it in their browser toolbar to start using your annotation vocabularies and configurations.

12.4.7 Your feedback is precious

If you reached the end of this tutorial and you have questions or feedbacks, please send a mail to pundit@netseven.it.

Thank you for your interest in Pundit!

13 T3 - Sharing / discovering / visualising annotations in Ask.ThePund.it

13.1 Introducing Ask

Ask is a web platform for sharing annotations and notebooks created with Pundit.

While Pundit's role is that of enabling people to annotate web pages of interest, Ask is where such annotations, once made public, can then be searched and explored.

Additionally, in Ask you find a MyAsk tab: it offers single users a view of their personal (public or private) notebooks.

Reusing, visualising and making intelligent use of the knowledge created via annotations is probably the most important task for scholars and it poses technical and design challenges on the developers side. As a first step in this direction, we incorporated into Ask a few specialised visualisation tools to graph annotations in a timeline or in an Edgemap graph.

The following exercises will guide you through Ask and will help us to collect feedback and new ideas from the WAB community.

13.2 Audience

This tutorial is targeted at end-users, mainly scholars who want to gain an impression of the collaborative features (sharing, discovering annotations) that we implemented in DM2E tools and that exemplify the possibilities given by Semantic Web technologies to work in a digital online environment.

Although this is not a technical tutorial, it is also meant for persons from institutions or private companies that might want to aggregate communities of users and leverage data produced by annotations.

13.3 Objective

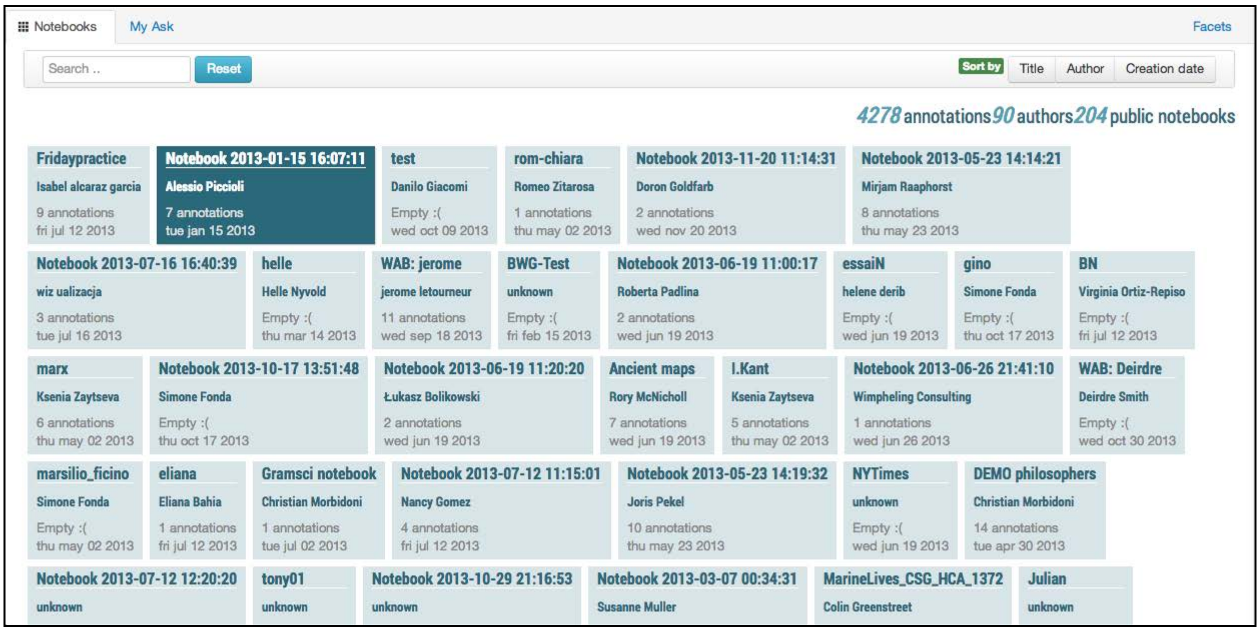
The proposed assignments which are short tasks to be performed with Ask (http://ask.thepund.it) will teach the user how to explore annotations made with Pundit in a social environment. Topics covered are: notebooks search, annotations browsing, personal notebooks management, aggregation of multiple notebooks as a semantic graph, facets driven analysis over annotations.

Assignment 4 deals with extending the platform with domain specific, vertical applications, mostly to visualize the outcome of the annotation process. The assignment provides a practical demonstration of possible interactive visualisations, hopefully encouraging novel extensions to the open tools we address here.

13.4 Description

13.4.1 Assignment 1 - Search for public notebooks

Go to http://ask.thepund.it and wait for the page to load. Public notebooks in the platform will be progressively shown along with some simple statistics as the number of annotations, authors and notebooks.


Notebooks | My Ask | Facets

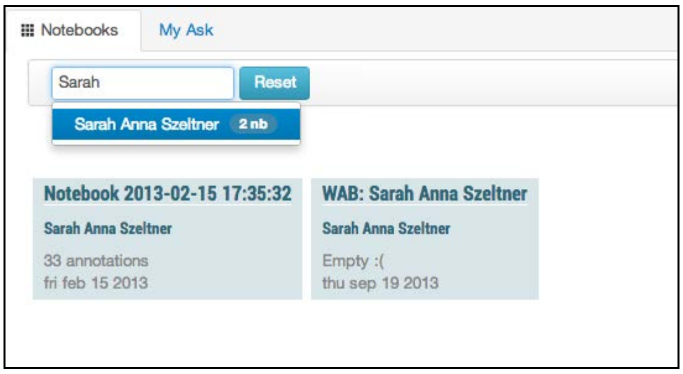
Search .. | Reset | Sort by: Title | Author | Creation date

4278 annotations 90 authors 204 public notebooks

Fridaypractice — Isabel alcaraz garcia — 9 annotations — fri jul 12 2013
Notebook 2013-01-15 16:07:11 — Alessio Piccioli — 7 annotations — tue jan 15 2013
test — Danilo Giacomi — Empty :(— wed oct 09 2013
rom-chiara — Romeo Zitarosa — 1 annotations — thu may 02 2013
Notebook 2013-11-20 11:14:31 — Doron Goldfarb — 2 annotations — wed nov 20 2013
Notebook 2013-05-23 14:14:21 — Mirjam Raaphorst — 8 annotations — thu may 23 2013
Notebook 2013-07-16 16:40:39 — wiz ualizacja — 3 annotations — tue jul 16 2013
helle — Helle Nyvold — Empty :(— thu mar 14 2013
WAB: jerome — jerome letourneur — 11 annotations — wed sep 18 2013
BWG-Test — unknown — Empty :(— fri feb 15 2013
Notebook 2013-06-19 11:00:17 — Roberta Padlina — 2 annotations — wed jun 19 2013
essaiN — helene derib — Empty :(— wed jun 19 2013
gino — Simone Fonda — Empty :(— thu oct 17 2013
BN — Virginia Ortiz-Repiso — Empty :(— fri jul 12 2013
marx — Ksenia Zaytseva — 6 annotations — thu may 02 2013
Notebook 2013-10-17 13:51:48 — Simone Fonda — Empty :(— thu oct 17 2013
Notebook 2013-06-19 11:20:20 — Łukasz Bolikowski — 2 annotations — wed jun 19 2013
Ancient maps — Rory McNicholl — 7 annotations — wed jun 19 2013
I.Kant — Ksenia Zaytseva — 5 annotations — thu may 02 2013
Notebook 2013-06-26 21:41:10 — Wimpheling Consulting — 1 annotations — wed jun 26 2013
WAB: Deirdre — Deirdre Smith — Empty :(— wed oct 30 2013
marsilio_ficino — Simone Fonda — Empty :(— thu may 02 2013
eliana — Eliana Bahia — 1 annotations — fri jul 12 2013
Gramsci notebook — Christian Morbidoni — 1 annotations — tue jul 02 2013
Notebook 2013-07-12 11:15:01 — Nancy Gomez — 4 annotations — fri jul 12 2013
Notebook 2013-05-23 14:19:32 — Joris Pekel — 10 annotations — thu may 23 2013
NYTimes — unknown — Empty :(— wed jun 19 2013
DEMO philosophers — Christian Morbidoni — 14 annotations — tue apr 30 2013
Notebook 2013-07-12 12:20:20 — unknown
tony01 — unknown
Notebook 2013-10-29 21:16:53 — unknown
Notebook 2013-03-07 00:34:31 — Susanne Muller
MarineLives_CSG_HCA_1372 — Colin Greenstreet
Julian — unknown

Figure 40. Public notebooks in Ask.

1. You can **search and sort notebooks by date, author name and title**. Try searching for the name of one of your colleagues participating in the experiment. Click on the notebook to open it in a new tab. Afterwards, go back to the "Notebooks" tab to start a new search and look for all the notebooks made within the WAB experiment, by simply typing "WAB" in the search box. Open 2 or 3 notebooks of your choice by clicking on them.

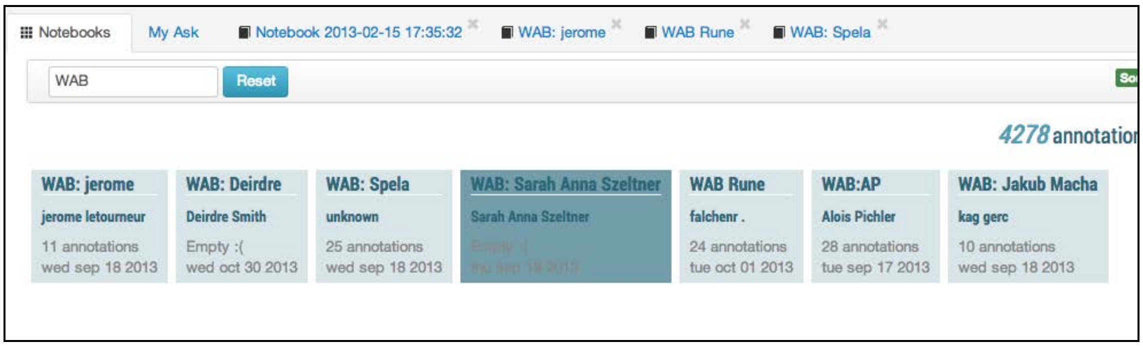

Notebooks | My Ask

Sarah | Reset

Sarah Anna Szeltner 2 nb

Notebook 2013-02-15 17:35:32 — Sarah Anna Szeltner — 33 annotations — fri feb 15 2013
WAB: Sarah Anna Szeltner — Sarah Anna Szeltner — Empty :(— thu sep 19 2013

Figure 41. Search notebooks by author.


Notebooks My Ask Notebook 2013-02-15 17:35:32 WAB: jerome WAB Rune WAB: Spela

WAB Reset

4278 annotatior

WAB: jerome	WAB: Deirdre	WAB: Spela	WAB: Sarah Anna Szeltner	WAB Rune	WAB:AP	WAB: Jakub Macha
jerome letourneur	Deirdre Smith	unknown	Sarah Anna Szeltner	falchenr .	Alois Pichler	kag gerc
11 annotations	Empty :(	25 annotations	Empty :(	24 annotations	28 annotations	10 annotations
wed sep 18 2013	wed oct 30 2013	wed sep 18 2013	thu sep 19 2013	tue oct 01 2013	tue sep 17 2013	wed sep 18 2013

Figure 42. Search for “WAB” in notebooks titles.

2. Explore a single notebook.

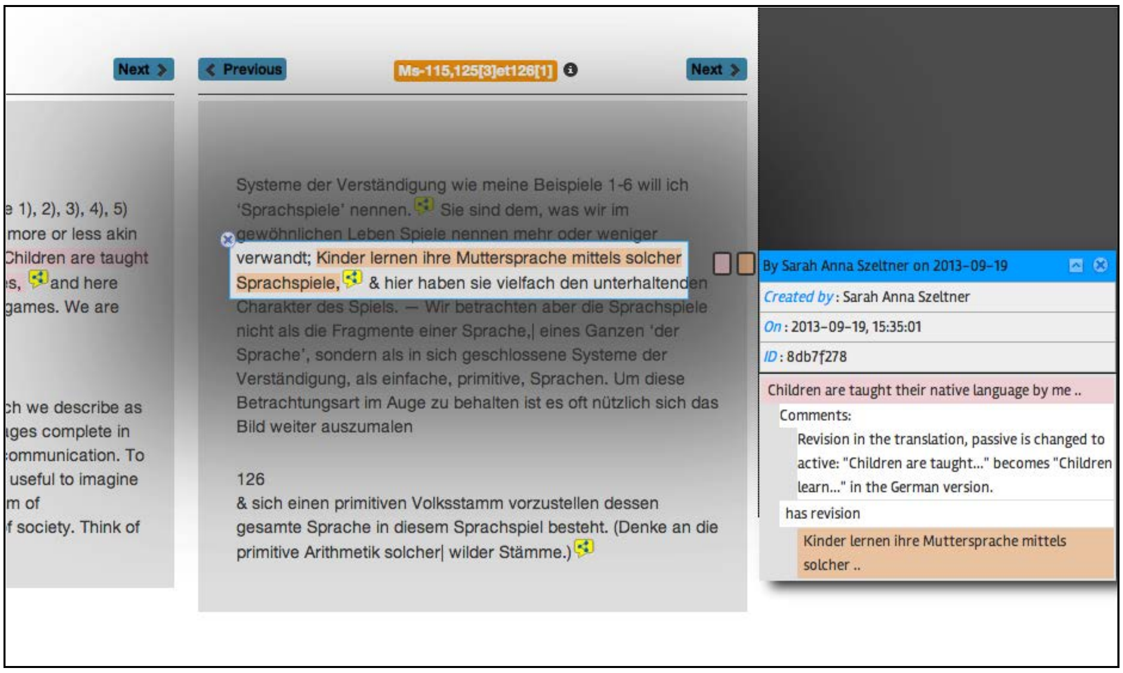
Choose one of the notebooks you have just opened and click on the respective tab to see all the annotations in the notebook.

Try to perform the following actions:

a. Expand one annotation to see its details.
b. Figure out what triple is in the annotation and its meaning
c. Figure out which predicate was used in the annotation
d. Figure out the object of the triples in one annotation
e. Find an annotation (in one of the notebooks) that has as object an element from the WAB:Subjects vocabulary
f. Take a look at the annotation in context (in the web page where it was created)

3. Respond to an annotation

When you click “See the annotation” in Ask, a new browser tab will open and the annotation will be opened in Pundit, as shown in the following picture.


Next Previous Ms-115,125[3]et126[1] Next

e 1), 2), 3), 4), 5)
more or less akin
Children are taught
s, and here
games. We are

ch we describe as
ges complete in
ommunication. To
useful to imagine
m of
f society. Think of

Systeme der Verständigung wie meine Beispiele 1-6 will ich ‘Sprachspiele’ nennen. Sie sind dem, was wir im gewöhnlichen Leben Spiele nennen mehr oder weniger verwandt; Kinder lernen ihre Muttersprache mittels solcher Sprachspiele, & hier haben sie vielfach den unterhaltenden Charakter des Spiels. — Wir betrachten aber die Sprachspiele nicht als die Fragmente einer Sprache,| eines Ganzen ‘der Sprache’, sondern als in sich geschlossene Systeme der Verständigung, als einfache, primitive, Sprachen. Um diese Betrachtungsart im Auge zu behalten ist es oft nützlich sich das Bild weiter auszumalen

126
& sich einen primitiven Volksstamm vorzustellen dessen gesamte Sprache in diesem Sprachspiel besteht. (Denke an die primitive Arithmetik solcher| wilder Stämme.)

By Sarah Anna Szeltner on 2013-09-19
Created by: Sarah Anna Szeltner
On: 2013-09-19, 15:35:01
ID: 8db7f278
Children are taught their native language by me ..
Comments:
Revision in the translation, passive is changed to active: "Children are taught..." becomes "Children learn..." in the German version.
has revision
Kinder lernen ihre Muttersprache mittels solcher ..

Figure 43 An annotation in Pundit.

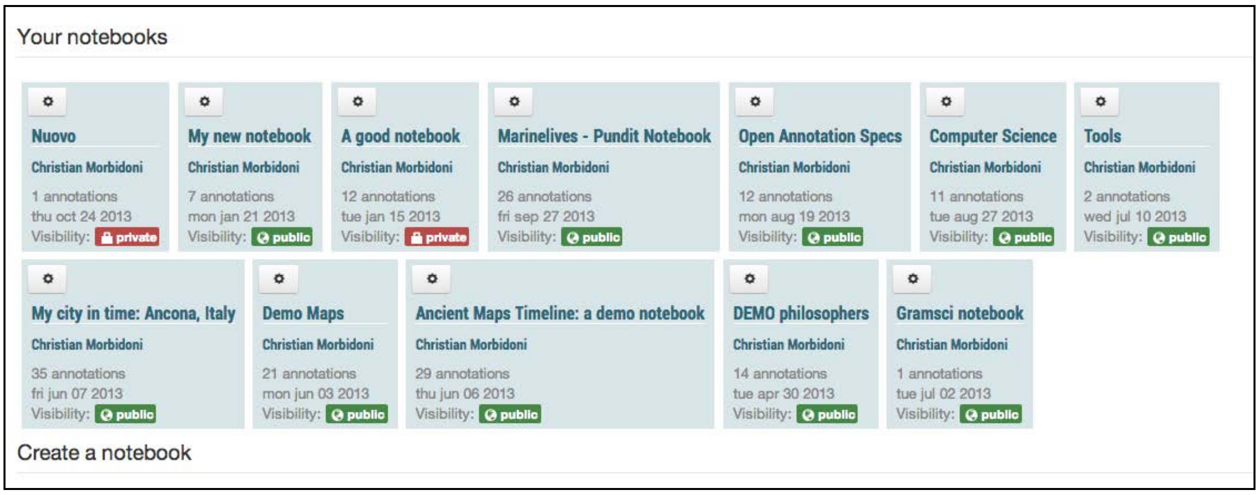
By clicking the small yellow button close to the text (e.g. the orange text in the figure above), you can use the functionalities you know from the first round of assignments (comments, tags, triple composer) to add your contribution. For example, you could attach a comment to the orange text or add a triple to link to a related entry in the WAB:Subjects vocabulary.

Try to follow these steps to make a new contribution:

a. Use Ask to find an annotation of interest
b. Go to the annotation and open it in Pundit
c. Create a new annotation on the same text fragment bringing some new explicit knowledge.

13.4.2 Assignment 2 - My Ask

Go to the "My Ask" tab. To login you can use the same credentials you use in Pundit (e.g. Google account).



Your notebooks

Nuovo
Christian Morbidoni
1 annotations
thu oct 24 2013
Visibility: private

My new notebook
Christian Morbidoni
7 annotations
mon jan 21 2013
Visibility: public

A good notebook
Christian Morbidoni
12 annotations
tue jan 15 2013
Visibility: private

Marinelives - Pundit Notebook
Christian Morbidoni
26 annotations
fri sep 27 2013
Visibility: public

Open Annotation Specs
Christian Morbidoni
12 annotations
mon aug 19 2013
Visibility: public

Computer Science
Christian Morbidoni
11 annotations
tue aug 27 2013
Visibility: public

Tools
Christian Morbidoni
2 annotations
wed jul 10 2013
Visibility: public

My city in time: Ancona, Italy
Christian Morbidoni
35 annotations
fri jun 07 2013
Visibility: public

Demo Maps
Christian Morbidoni
21 annotations
mon jun 03 2013
Visibility: public

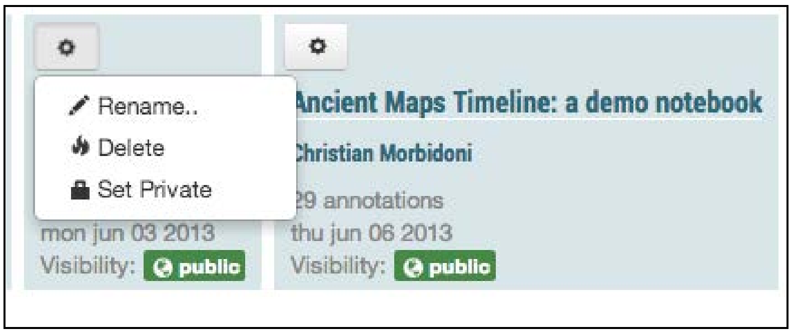
Ancient Maps Timeline: a demo notebook
Christian Morbidoni
29 annotations
thu jun 06 2013
Visibility: public

DEMO philosophers
Christian Morbidoni
14 annotations
tue apr 30 2013
Visibility: public

Gramsci notebook
Christian Morbidoni
1 annotations
tue jul 02 2013
Visibility: public

Create a notebook

Figure 44. The MyAsk tab shows personal public and private notebooks.



Rename..
Delete
Set Private
mon jun 03 2013
Visibility: public

Ancient Maps Timeline: a demo notebook
Christian Morbidoni
29 annotations
thu jun 06 2013
Visibility: public

Figure 45. Notebooks settings.

In this tab you should see all your notebooks. Try to perform the following actions:

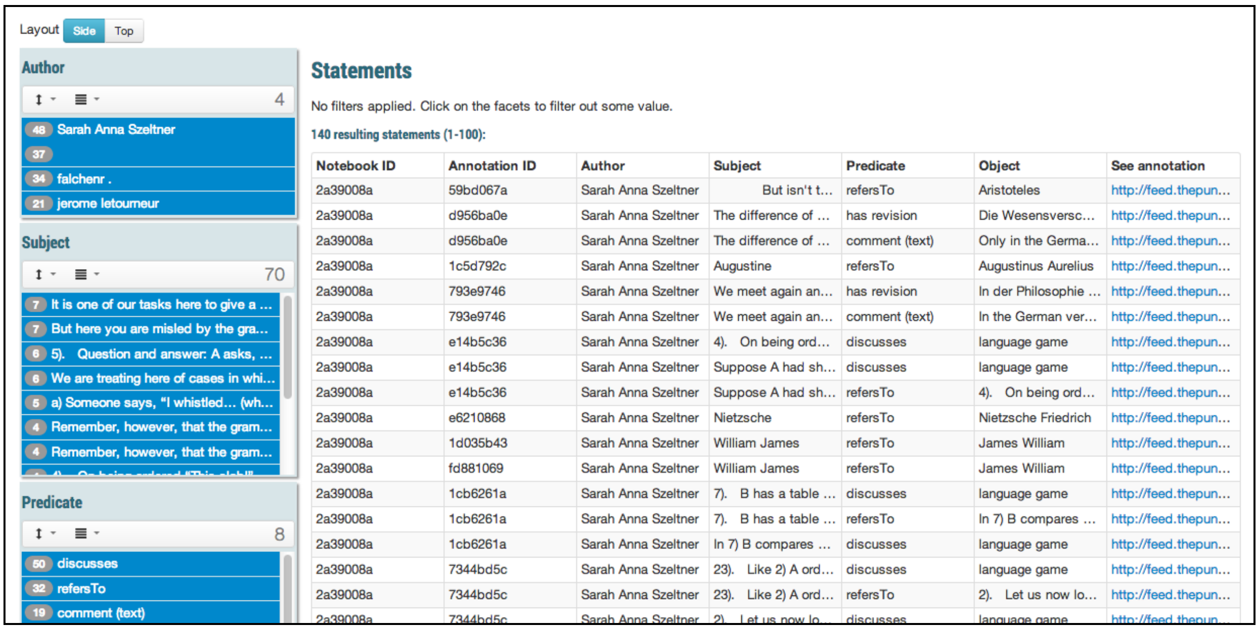
1. Rename a notebook (please do not remove the “WAB” tag from your notebook!).
2. Create a new notebook and give it a name that starts with “WAB 2”.
3. Using the “wheel” button, change visibility to the “WAB 2...” notebook. Then switch back to public.
4. Go to a page of the Brown Book and open it in Pundit. Now use the “notebook manager” to set the newly created notebook as the “current” one.
5. Create a couple of sample annotations on the page.
6. Go back to Ask and find your new annotations.
7. Delete the notebook named “WAB 2…”.
   Note: the current notebook cannot be deleted. In order to perform this action you first have to reset your current notebook to the one named “WAB...” and then delete the notebook named “WAB 2”.

13.4.3 Assignment 3 - Facets

Faceted search (or browsing) is a common paradigm to explore data. In Ask we are experimenting with it to enable a user to query the system and deeply explore the structured data (triples) that come from a number of notebooks.

Before performing this assignment, ensure you have a number of “WAB” notebooks opened (at least 3), with a certain number of annotations in them.

In Ask, click on the “Facet” button on the top right: the faceted browser will open in a new tab in Ask (see the following picture). Available facets are shown in the left bar.



Author 4
- 48 Sarah Anna Szeltner
- 37
- 34 falchenr .
- 21 jerome letourneur

Subject 70
- 7 It is one of our tasks here to give a ...
- 7 But here you are misled by the gra...
- 6 5). Question and answer: A asks, ...
- 6 We are treating here of cases in whi...
- 5 a) Someone says, “I whistled... (wh...
- 4 Remember, however, that the gram...
- 4 Remember, however, that the gram...

Predicate 8
- 50 discusses
- 32 refersTo
- 19 comment (text)

Statements

No filters applied. Click on the facets to filter out some value.

140 resulting statements (1-100):

Notebook ID	Annotation ID	Author	Subject	Predicate	Object	See annotation
2a39008a	59bd067a	Sarah Anna Szeltner	But isn't t...	refersTo	Aristoteles	http://feed.thepun...
2a39008a	d956ba0e	Sarah Anna Szeltner	The difference of ...	has revision	Die Wesensversc...	http://feed.thepun...
2a39008a	d956ba0e	Sarah Anna Szeltner	The difference of ...	comment (text)	Only in the Germa...	http://feed.thepun...
2a39008a	1c5d792c	Sarah Anna Szeltner	Augustine	refersTo	Augustinus Aurelius	http://feed.thepun...
2a39008a	793e9746	Sarah Anna Szeltner	We meet again an...	has revision	In der Philosophie ...	http://feed.thepun...
2a39008a	793e9746	Sarah Anna Szeltner	We meet again an...	comment (text)	In the German ver...	http://feed.thepun...
2a39008a	e14b5c36	Sarah Anna Szeltner	4). On being ord...	discusses	language game	http://feed.thepun...
2a39008a	e14b5c36	Sarah Anna Szeltner	Suppose A had sh...	discusses	language game	http://feed.thepun...
2a39008a	e14b5c36	Sarah Anna Szeltner	Suppose A had sh...	refersTo	4). On being ord...	http://feed.thepun...
2a39008a	e6210868	Sarah Anna Szeltner	Nietzsche	refersTo	Nietzsche Friedrich	http://feed.thepun...
2a39008a	1d035b43	Sarah Anna Szeltner	William James	refersTo	James William	http://feed.thepun...
2a39008a	fd881069	Sarah Anna Szeltner	William James	refersTo	James William	http://feed.thepun...
2a39008a	1cb6261a	Sarah Anna Szeltner	7). B has a table ...	discusses	language game	http://feed.thepun...
2a39008a	1cb6261a	Sarah Anna Szeltner	7). B has a table ...	refersTo	In 7) B compares ...	http://feed.thepun...
2a39008a	1cb6261a	Sarah Anna Szeltner	In 7) B compares ...	discusses	language game	http://feed.thepun...
2a39008a	7344bd5c	Sarah Anna Szeltner	23). Like 2) A ord...	discusses	language game	http://feed.thepun...
2a39008a	7344bd5c	Sarah Anna Szeltner	23). Like 2) A ord...	refersTo	2). Let us now lo...	http://feed.thepun...
2a39008a	7344bd5c	Sarah Anna Szeltner	2). Let us now lo...	discusses	language game	http://feed.thepun...

Figure 46. The Notebooks Faceted Browser.

Example usage: Setting the “Author” facet by clicking on an author’s name removes all the rows from the result table except those where the author is the chosen one.

Setting the “Predicate” facet to “discusses”, shows annotations where such a predicate has been used in a triple.

Try to perform the following actions:

1. Find all the annotations that use a specified predicate and are authored by a specified person.
2. Find all the annotations that have a specified text passage as the subject and use a specified predicate.
3. Open the annotation in Pundit by clicking the “See Annotation” link in the corresponding row of the results table.
4. Add a new contribution to “respond” to the annotation (as done in the previous assignment).

13.4.4 Assignment 4 - Vertical visualisations

This last assignment has the goal of guiding you through two simple examples of specialised visualisations of notebooks. Please take note of any idea that might come into your mind to improve the examples or possibly adapt them to your domain/field of interest.

Note: the demonstrative visualisation applications were build using open-source tools and are nott yet in a stable state. Things can break.

Simple timeline

1. Go to http://ask.thepund.it and search for a notebook named “Ancona”.
2. Open the notebook and have a look at the annotations and the information they encompass.
3. Scroll down the page and click on the green “Interactive Timeline” button. A new page will open where the very same annotations are shown in a timeline.
4. Browse the timeline annotation by annotation and click the “Go to annotated page” link to show an annotation in its context.

Note: this is possible thanks to the “date” triples that are present in the annotations (you can view them in Ask).

Edge Map

1. Go to http://ask.thepund.it and search for the notebook named “romeo”.
2. Open the notebook and have a look at the annotations.
3. Scroll down the page and click on the green “Edgemap influence graph” button. A new page will open showing a graph where nodes are authors of the annotated texts. The arrows connecting them have been created on the basis of citations annotated with Pundit.
4. Click on one of the bubbles to focus on a philosopher. A number of bubbles connected to the selected philosopher will be highlighted.
5. Drag the mouse pointer over one of those bubbles: a grey box should appear at the bottom right of the page. It shows the annotations that generated this connection.

6. Click on the “Go to annotated web page” to show the annotations in context with Pundit.

Note: this is possible thanks to the “cites” and “has author” predicates that were used in the annotations (see them in Ask).

14 T4 - Timeline Demo: building an interactive timeline with annotations

14.1 Introduction

In this tutorial we will guide you in experimenting with Pundit to create your own interactive Timeline by annotating web pages and pictures.

Let us show you some examples of timelines we built by annotating some pictures and historical maps on the web.

14.1.1 Sample Maps Timeline

Link: http://metasound.dibet.univpm.it/timelinejs/examples/pundit.html?notebook-ids=20f99b8f

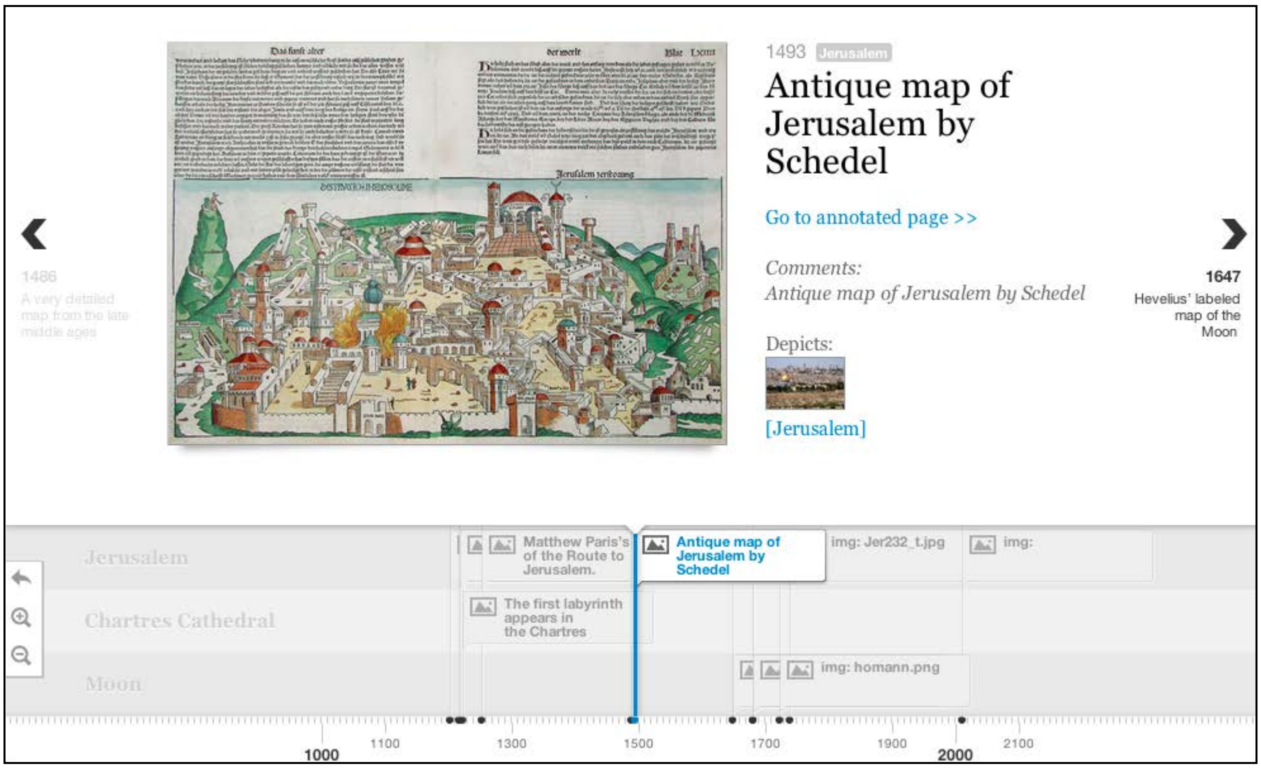
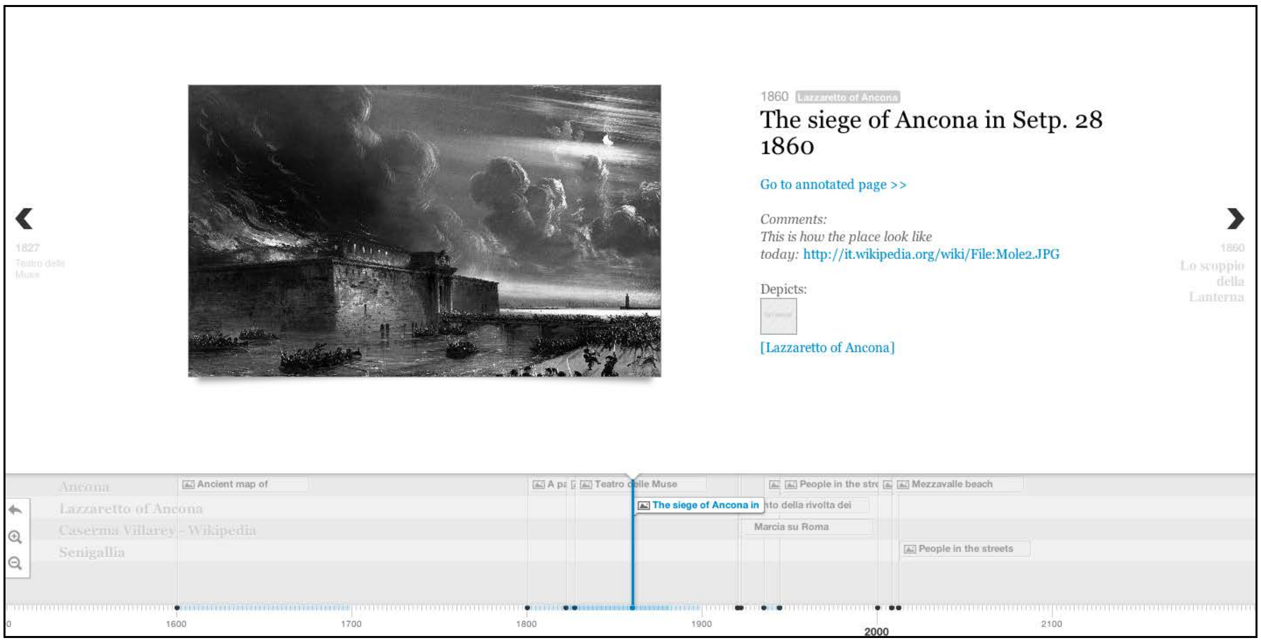

1493 Jerusalem
Antique map of Jerusalem by Schedel
Go to annotated page >>
Comments:
Antique map of Jerusalem by Schedel
Depicts:
[Jerusalem]
1486
A very detailed map from the late middle ages
1647
Hevelius' labeled map of the Moon
Jerusalem
Chartres Cathedral
Moon
Matthew Paris's of the Route to Jerusalem.
Antique map of Jerusalem by Schedel
img: Jer232_t.jpg
img:
The first labyrinth appears in the Chartres
img: homann.png
1000
1100
1300
1500
1700
1900
2000
2100

Figure 47. The Historical Maps Timeline demo.

14.1.2 Sample history of Ancona Timeline

Link: http://metasound.dibet.univpm.it/timelinejs/examples/pundit.html?notebook-ids=6290cd68


1860 Lazzaretto of Ancona

The siege of Ancona in Setp. 28 1860

Go to annotated page >>

Comments:
This is how the place look like today: http://it.wikipedia.org/wiki/File:Mole2.JPG

Depicts:

[Lazzaretto of Ancona]

Figure 48. The City of Ancona Timeline demo.

14.1.3 What is Pundit

Pundit is a novel web annotation tool that, in the DM2E project, we are continuously developing and experimenting with as a collaborative tool for scholars. Annotations in Pundit are more than side comments on a document, but rather pieces of structured data that scholars create and share, and that web applications can consume to produce interactive visualisations and navigations.

In Pundit users collect annotations in "notebooks": collections of data ready to be shared and reused by other tools.

TimeLineJS (http://timeline.knightlab.com/), a webapp to build interactive timelines, is one of such tools.

14.1.4 About this demonstration

This tutorial guides you through an exercise that shows you how to create annotations to be displayed in a Timeline.

You can build a timeline about "ancient maps", by collecting, commenting and tagging them, as in the Ancient Maps example: http://metasound.dibet.univpm.it/timelinejs/examples/pundit.html?notebook-ids=20f99b8f

... or you can build a timeline about your city, as in the City of Ancona Timeline example (http://metasound.dibet.univpm.it/timelinejs/examples/pundit.html?notebook-ids=6290cd68) or choose your favourite topic.

Warning!
To make this demo work, you will need a Google Chrome, Firefox or Safari browser.

14.2 Description

14.2.1 Get the Pundit bookmarklet

The Pundit Timeline demo bookmarklet is available here:

http://thepund.it/bm/demo-timeline

You just have to drag & drop the “Pundit-Timeline-Demo” link to your browser’s bookmark-toolbar.

14.2.2 Prepare for annotating

You can annotate both fragments of texts or images you find on the Web. However, be warned: the bookmarklet is not guaranteed to work exactly on every web page! In particular, Pundit does not work on image viewers with zooming functions etc. (it would be impossible to integrate with all possible systems), so please choose simple good old HTML pages if possible.

Some examples of annotatable resources

- Search maps in the British National Library: Example page to annotate: http://www.bl.uk/onlinegallery/sacredtexts/mparis.html
- Antique Maps and Atlases: Example page: http://www.helmink.com/Antique_Map_Ortelius_East_Indies_1/
- The British Museum: Example page: http://www.britishmuseum.org/research/collection_online/collection_object_details/collection_image_gallery.aspx?assetId=332551&objectId=1349928&partId=1
- Search Europeana for content to annotate. Note that not all the resources you find are directly annotatable with the current Pundit implementation.
- Once you are on the web page you want to annotate, click the
- “Pundit-Timeline-Demo” button you find in your browser bookmarks-toolbar, to load
- Pundit and start annotating.

14.2.3 Create a new notebook and set it as current notebook in Pundit

In the Pundit top bar, go to “Your User Name” > “Manage Notebooks”.

Under “Create a new notebook” type the name of your new notebook.

Set the new notebook as “current notebook”. To do so click on the gear button corresponding to the notebook and choose “Set as Current Notebook”.

14.2.4 Create “triples” with Pundit

An annotation in Pundit is a triple; a very simple “sentence” that says something about the annotated content. For example, “this text is citing this philosopher” or “this picture depicts this city” and so on.

A triple is in the following form:

subject - predicate - object

For example:

“some text” - cites - Plato

To learn to create your triples refer to the following steps.

Create triples with the triple composer

Select a text fragment you want to annotate and click “annotate text fragment”. The triple composer will open.

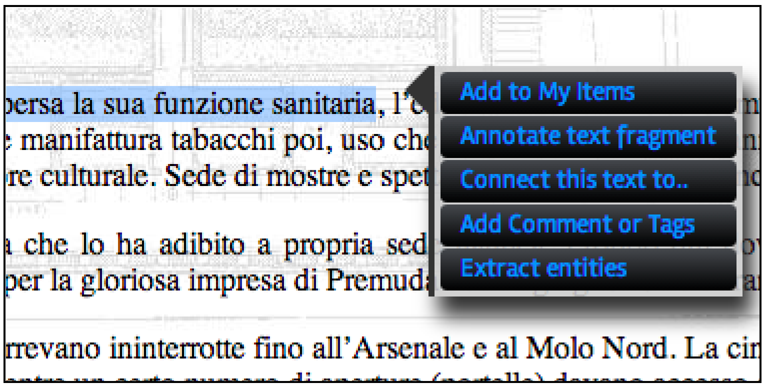


Figure 49. Selecting a text.

For images, point your mouse over the image, click on the icon that appears at the top-left and choose "annotate image"

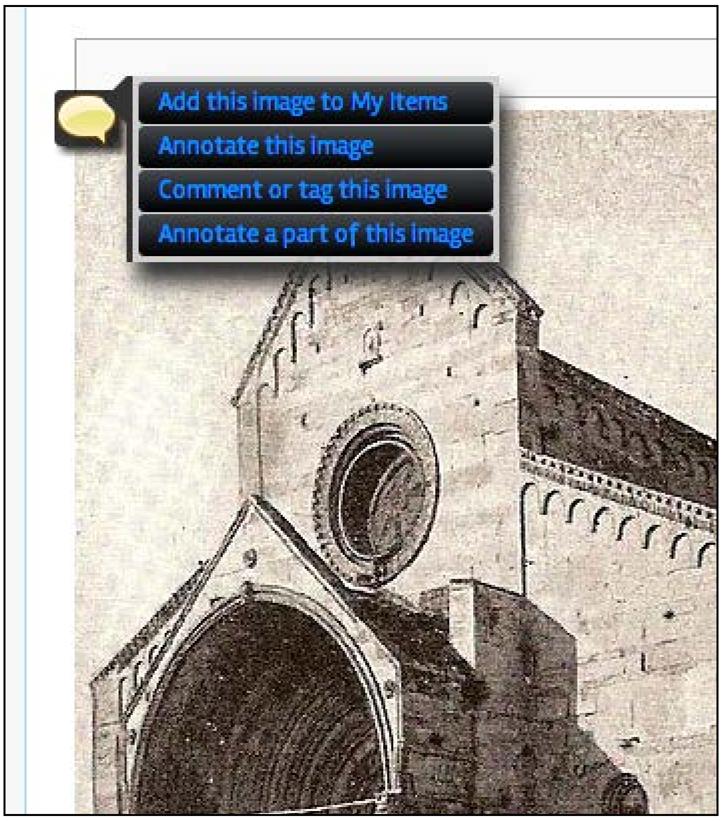

Add this image to My Items
Annotate this image
Comment or tag this image
Annotate a part of this image

Figure 50. Annotating an image.

Select the predicate by clicking on the yellow "predicate" box "date".

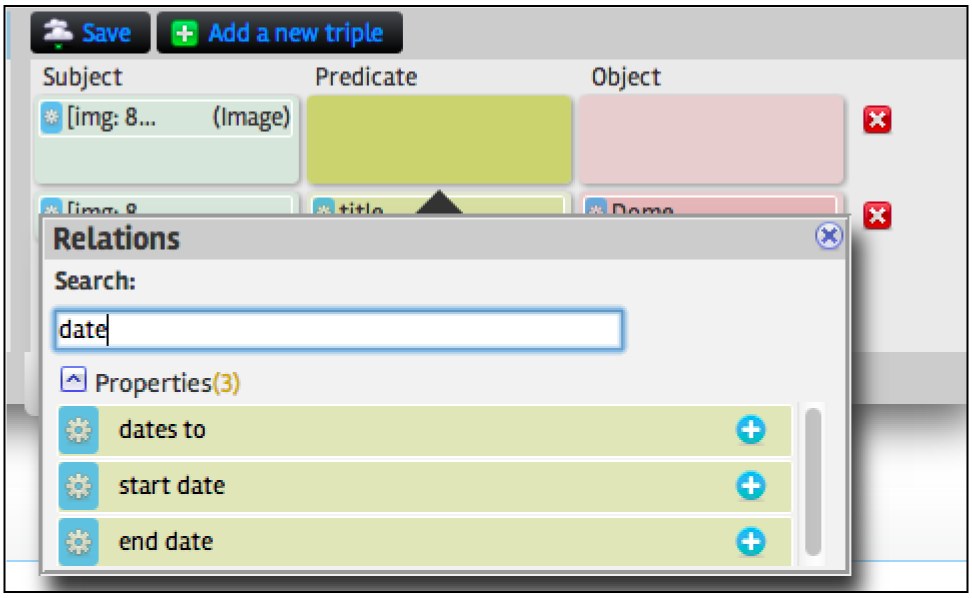
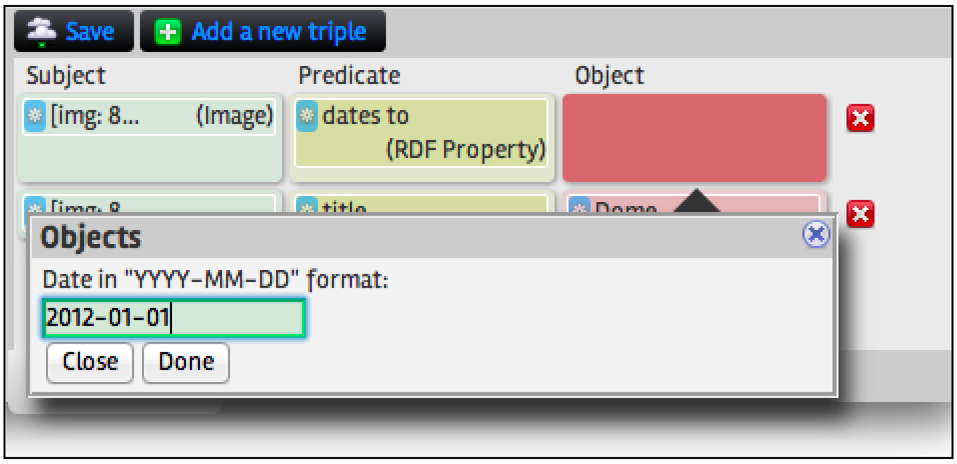


Figure 51. Choosing "date" ad predicate.

Select the object by clicking on the red "object" box. Search the available vocabularies (Freebase gives good results usually) and pick from results or, if you are using one of the textual relations (comment, title) type in a text, or a valid date when required.



Date in "YYYY-MM-DD" format:
2012-01-01

Figure 52. Dates format is YYY-MM-DD.

To add a new triple click the “Add a new triple” button.

Drag the text-fragment from the “subject” box to the empty “subject” box.

Select the predicate by clicking on the empty yellow “predicate” box.

Select the object by clicking on the empty red “object” box.

Click “Save”.

Note: if you forgot some triple you can add it later by selecting the very same image or text fragment.

14.2.5 Create timeline compliant annotations

To make your annotation visible in the TimelineJS demo you only need to add date information, by creating the following triple:

text-fragment - dates to - "YYYT-MM-DD"
or, for an image:

image - dates to - "YYYY-MM-DD"
Alternatively you can specify a period in time by using the following two triples:

image - start date - "YYYY-MM-DD"
image - end date - "YYYY-MM-DD"

Add more information to your annotation

To make your annotation more interesting you can use all the relations you find in Pundit to create additional triples.

For example:

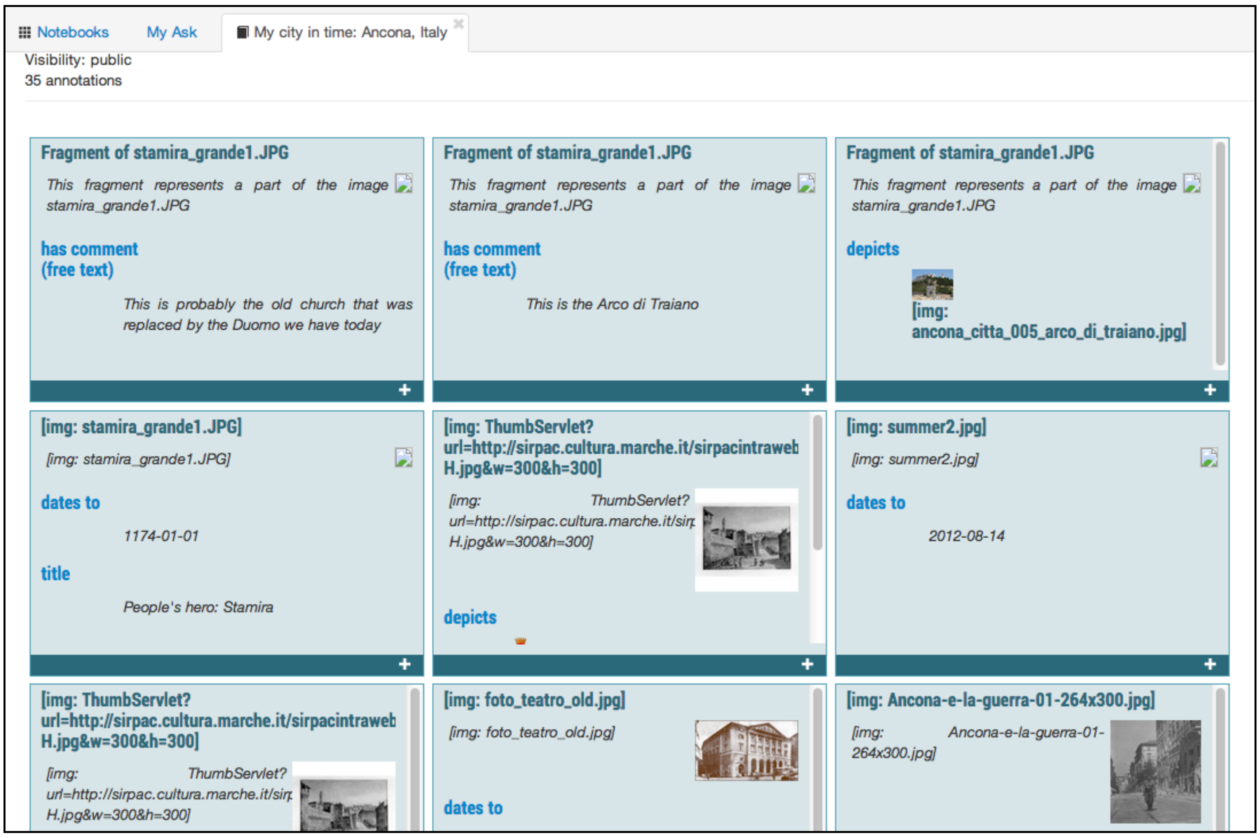
add a nice title to be displayed: image - title - "The Dome of Ancona"
add a comment: image - comment - "I used to work close to the Dome"
specify the author: image - has creator - SOME-FREEBASE_PERSON

Need help with the Triple Composer?

To learn more about Pundit annotation read the Quick Start Guide at http://www.thepund.it/documentation/quick-start-guide-to-pundit/ or watch the Introductory Videos at http://www.thepund.it/introductory-videos/.

14.2.6 Visualise your notebook in Ask the Pundit

Go to http://ask.as.thepund.it and find your public notebook, or click on "My Ask" to login and find your personal notebooks. Click the "browse" button and your annotations will be shown.



Visibility: public
35 annotations

Fragment of stamira_grande1.JPG
This fragment represents a part of the image stamira_grande1.JPG
has comment (free text)
This is probably the old church that was replaced by the Duomo we have today

Fragment of stamira_grande1.JPG
This fragment represents a part of the image stamira_grande1.JPG
has comment (free text)
This is the Arco di Traiano

Fragment of stamira_grande1.JPG
This fragment represents a part of the image stamira_grande1.JPG
depicts
[img: ancona_citta_005_arco_di_traiano.jpg]

[img: stamira_grande1.JPG]
[img: stamira_grande1.JPG]
dates to
1174-01-01
title
People's hero: Stamira

[img: ThumbServlet?url=http://sirpac.cultura.marche.it/sirpacintraweb H.jpg&w=300&h=300]
[img: ThumbServlet? url=http://sirpac.cultura.marche.it/sirp H.jpg&w=300&h=300]
depicts

[img: summer2.jpg]
[img: summer2.jpg]
dates to
2012-08-14

[img: ThumbServlet? url=http://sirpac.cultura.marche.it/sirpacintraweb H.jpg&w=300&h=300]
[img: ThumbServlet? url=http://sirpac.cultura.marche.it/sirp H.jpg&w=300&h=300]

[img: foto_teatro_old.jpg]
[img: foto_teatro_old.jpg]
dates to

[img: Ancona-e-la-guerra-01-264x300.jpg]
[img: Ancona-e-la-guerra-01-264x300.jpg]

Figure 53. Annotations in a notebook.

You can find the notebook we used to create this demo at http://ask.as.thepund.it/#/notebooks/6290cd68.

14.2.7 Visualise in the TimelineJS Demo

Once you opened your notebook in Ask the Pundit, click on the "Timeline demo" green button at the bottom of the page.

Iterate the procedure to create more annotations; refresh the visualisation page to update it.

Please give us your feedback!

Please spend 5 minutes to fill in this form: https://docs.google.com/forms/d/11-bGn_w8aPY0-0Igui4JUoMZe0x7taQsDXowVEaYv5c/viewform?usp=sharing&edit_requested=true

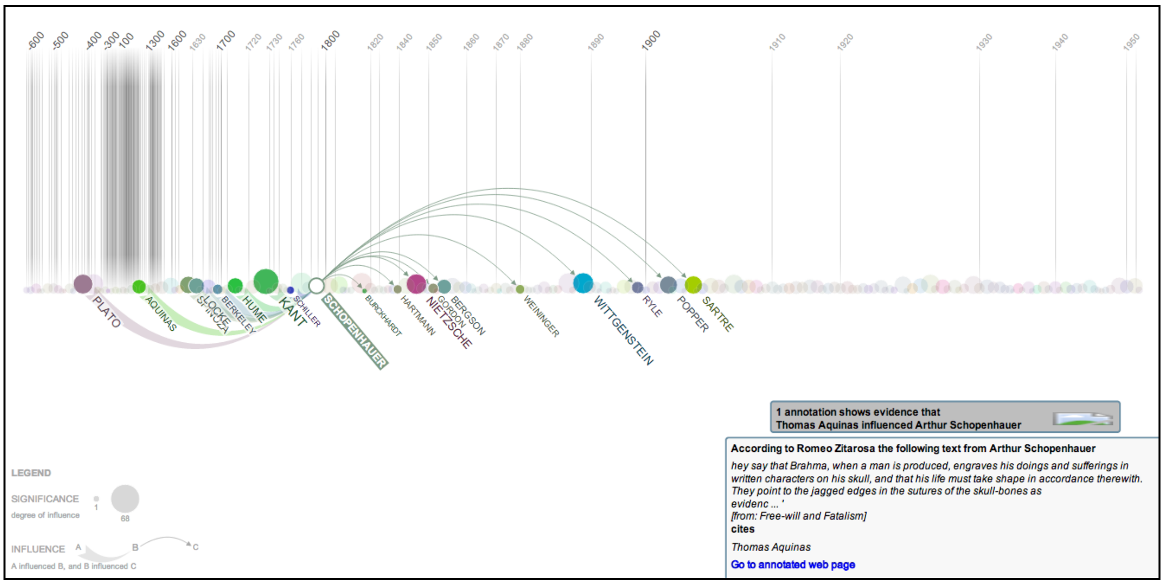
Would you like to see more cases and visualisations? Learn about the **Philosophers Demo** at http://www.thepund.it/visualization-demos/philosophers-demo-howto/.

15 T5 - Edgemaps visualisation with Pundit annotations

15.1 Introduction

This tutorial demonstrates how particular types of annotations, created by scholars on Web documents, can generate an interactive visualisation.

As a demonstrative example domain we focus on philosophy. We will guide users in the process of annotating citations among philosophical texts. Such annotations, made with Pundit, will be used to state that a text from an author “cites” another author. The annotations will be then used to create an interactive Edgemap graph (http://mariandoerk.de/edgemaps/) showing the influences among the annotated philosophers.



1 annotation shows evidence that
Thomas Aquinas influenced Arthur Schopenhauer

According to Romeo Zitarosa the following text from Arthur Schopenhauer

hey say that Brahma, when a man is produced, engraves his doings and sufferings in written characters on his skull, and that his life must take shape in accordance therewith. They point to the jagged edges in the sutures of the skull-bones as evidenc ... '
[from: Free-will and Fatalism]
cites

Thomas Aquinas

Go to annotated web page

Figure 54. The picture above shows how the Edgemap we are going to create in this tutorial will look like.

To make this demo work, you will need a Google Chrome, Firefox or Safari browser.

15.2 Audience

This tutorial is targeted at end-users, mainly scholars that wish to test Pundit and the related technologies in a simple case study.

15.3 Description

15.3.1 Get the Pundit bookmarklet

The Pundit Philosphers demo bookmarklet is available here: http://thepund.it/bm/demo-philosophers/bookmarklet.html.

You just have to drag & drop the "Pundit Philosophers DEMO" link to your browser's bookmark-toolbar.

15.3.2 Prepare for annotating

You can annotate a generic Web page using the Pundit bookmarklet. We suggest you go to http://wikisource.org and search for an interesting text. Once you are in the web page you want to annotate, click the "Pundit latest" button you find in your browser bookmarks-toolbar, to load Pundit and start annotating.

15.3.3 Create a new notebook

Create a new notebook and set it as current notebook in Pundit.

In the Pundit top bar, go to "Your User Name" > "Manage Notebooks" Under "Create a new notebook" type the name of your new notebook.
Set the new notebook as "current notebook". To do so, click on the gear button corresponding to the notebook and choose "Set as Current Notebook"

15.3.4 Create edgemaps compliant annotations

To assert that author-1 cites author-2 in his text-fragment, you must create two triples (the order is not important):

text-fragment – has creator – author-1

text-fragment – cites – author-2

To do so, you will use the "Triple composer" component like this:

Select the text fragment you want to annotate and click "annotate text fragment".

- **a.** The triple composer will open.
- **b.** Select the predicate by clicking on the yellow "predicate" box.
- **c.** Select the object by clicking on the red "object" box.
- **d.** To add a new triple click the "Add a new triple" button.
- **e.** Drag the text-fragment from the "subject" box to the empty "subject" box.
- **f.** Select the predicate by clicking on the empty yellow "predicate" box.
- **g.** Select the object by clicking on the empty red "object" box.
- **h.** Important: to make the visualisation work, you must choose a philosopher from the DEMO-PHILOSOPHERS vocabulary.
- **i.** Click "Save".

15.3.5 Visualize your notebook in Ask the Pundit

Go to http://ask.as.thepund.it and find your public notebook, or click on "My Ask" to login and find your personal notebooks. Click the "browse" button and your annotations will be shown.

15.3.6 Visualize and explore your notebook as an Edgemap

Once you opened your notebook in Ask the Pundit, click on the "Edgemaps influence graph" green button at the bottom of the page: an Edgemap graph will show the relations you established among authors.

The timeline shows each philosopher as a circle. Clicking on a circle shows all the links between this and other philosophers. Putting the mouse on the linked philosopher will show the annotations that generated the link. The annotation box shows the author and the annotated page which can be reached by clicking on "go to annotated webpage".

15.3.7 Iterate

Create more annotations repeating step 3. Refresh the visualisation page to update it.

15.3.8 Mashup notebooks

This demonstration was performed by a group of users in Pisa. We created an Edgemap where all the annotations from the participants were put together.

You can see the result following this URL:
http://thepund.it/edgemaps_demo/demo.html?nbs={0542cd8d,0f9e15a8,12377d49,144c4b17,2fcf4383,628d0485,633e5e63,6bdcd4a5,8482b3de,86cffd13,9ec66f55,b2061a6b}&source={pundit}#phils;time;

Note: If you want to include your own notebook in the collective Edgemap you just have to modify the URL by including your notebook id among the others.

To do so:

1. Get your notebook id from http://ask.tehpund.it by opening your notebook and copying the last part of the URL, which should look like the following:
*http://ask.as.thepund.it/#/notebooks/**6290cd68*** *(id = 6290cd68)*
Modify the URL by adding the notebook id. In the case of notebook ***6290cd68***:
http://thepund.it/edgemaps_demo/demo.html?nbs={6290cd68,0542cd8d,0f9e15a8,12377d49,144c4b17,2fcf4383,628d0485,633e5e63,6bdcd4a5,8482b3de,86cffd13,9ec66f55,b2061a6b}&source={pundit}#phils;time;

15.3.9 Feedback

When done, you could spend 5 minutes on completing this survey: http://goo.gl/uOe1t

Still curious? Try out these alternative paths!

ALTERNATIVE Step 2
You can annotate one of the following Muruca Digital Libraries:

http://burckhardtsource.org

In this case you should use the "annotate" button that you find in the Digital Libraries pages (no bookmarklet needed).

ALTERNATIVE Step 4

You can create more precise annotations, by linking two text fragments from different authors. In this case the annotation must have the form:

```
text-fragment-1 has creator author-1
text-fragment-2 has creator author-2
text-fragment-1 cites text-fragment-2
```