D 7.5 - Web-enabled Public Showcase

Summary: This deliverable describes Web-enabled public showcases for public dissemination and presentation of the project. The showcase is built around publicly available, Web-based CODE prototypes. The purpose of each prototype is briefly described followed by typical use case descriptions demonstrated along a typical use case.

<table>
<thead>
<tr>
<th>Project Acronym</th>
<th>CODE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Grant Agreement number</td>
<td>296150</td>
</tr>
<tr>
<td>Project Title</td>
<td>Commercially Empowered Linked Open Data Ecosystems in Research</td>
</tr>
<tr>
<td>Date</td>
<td>2014-05-02</td>
</tr>
<tr>
<td>Nature</td>
<td>R (Report)</td>
</tr>
<tr>
<td>Dissemination level</td>
<td>PU (Public)</td>
</tr>
<tr>
<td>WP Lead Partner</td>
<td>MindMeister</td>
</tr>
<tr>
<td>Revision</td>
<td>Final revision</td>
</tr>
<tr>
<td>Authors</td>
<td>Patrick Höfler, Belgin Mutlu, Florian Stegmaier, Robert Knight, Vedran Sabol, Roman Kern, Michael Hollauf, Michael Granitzer</td>
</tr>
</tbody>
</table>

Consortium: [Icons of participating institutions]

Statement of originality: This document contains original unpublished work except where clearly indicated otherwise.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

This document reflects only the author’s views and the European Community is not liable for any use that might be made of the information contained herein. © CODE Consortium, 2012
Project Officer & Project Coordinators

<table>
<thead>
<tr>
<th>Role</th>
<th>Name</th>
<th>Organization</th>
</tr>
</thead>
<tbody>
<tr>
<td>Project Officer</td>
<td>Stefano Bertolo</td>
<td>European Commission</td>
</tr>
<tr>
<td></td>
<td></td>
<td>DG CONNECT Unit G3</td>
</tr>
<tr>
<td></td>
<td></td>
<td>EUROFORUM 01/293 - 10 rue Robert Stumper, L-2557 Luxembourg</td>
</tr>
<tr>
<td></td>
<td></td>
<td><a href="mailto:stefano.bertolo@ec.europa.eu">stefano.bertolo@ec.europa.eu</a></td>
</tr>
<tr>
<td>Project Coordinator</td>
<td>Stefanie Lindstaedt</td>
<td>Inffeldgasse 21a, 8010 Graz, Austria</td>
</tr>
<tr>
<td></td>
<td></td>
<td>+43 (316) 873-9250 (phone)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>+43 (316) 873-9254 (fax)</td>
</tr>
<tr>
<td></td>
<td></td>
<td><a href="mailto:slind@know-center.at">slind@know-center.at</a></td>
</tr>
<tr>
<td>Scientific Coordinator</td>
<td>Michael Granitzer</td>
<td>Innstrasse 33, D-94032 Passau</td>
</tr>
<tr>
<td></td>
<td></td>
<td>+49(0)851-509-3305</td>
</tr>
<tr>
<td></td>
<td></td>
<td><a href="mailto:michael.granitzer@uni-passau.de">michael.granitzer@uni-passau.de</a></td>
</tr>
</tbody>
</table>

Document Revision History

<table>
<thead>
<tr>
<th>Revision</th>
<th>Date</th>
<th>Author</th>
<th>Organization</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1st Draft</td>
<td>2014-04-14</td>
<td>Vedran Sabol</td>
<td>Know-Center</td>
<td>Document structure</td>
</tr>
<tr>
<td>2nd Draft</td>
<td>2014-04-27</td>
<td>Patrick Höfler, Belgin Mutlu</td>
<td>Know-Center</td>
<td>Query Wizard and Visualization Wizard</td>
</tr>
<tr>
<td>3rd Draft</td>
<td>2014-04-28</td>
<td>Florian Stegmaier</td>
<td>University of Passau</td>
<td>42-data platform</td>
</tr>
<tr>
<td>4th draft</td>
<td>2014-04-29</td>
<td>Robert Night</td>
<td>Mendeley</td>
<td>Mendeley Desktop</td>
</tr>
<tr>
<td>5th draft</td>
<td>2014-04-30</td>
<td>Roman Kern</td>
<td>Know-Center</td>
<td>Annotator Tool</td>
</tr>
<tr>
<td>7th Draft</td>
<td>2014-05-02</td>
<td>Vedran Sabol</td>
<td>Know-Center</td>
<td>Proof-reading, content integration, reformatting</td>
</tr>
<tr>
<td>8th Draft</td>
<td>2014-05-02</td>
<td>Michael Granitzer</td>
<td>University of Passau</td>
<td>Additional images</td>
</tr>
<tr>
<td>Final</td>
<td>2014-05-02</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
# Table of Contents

1. **Introduction** - 2

2. **Public Demonstrators** - 3
   2.1 **42-Data Platform** - 3
      2.1.1 **Description** - 3
      2.1.2 **Enabling Discussions of Primary Research Data** - 4
   2.2 **Annotator Tool** - 11
      2.2.1 **Use Case** - 11
   2.3 **Query Wizard** - 13
      2.3.1 **Description** - 13
      2.3.2 **Search, Filtering and Mindmap Generation** - 14
      2.3.3 **Data Set Aggregation and Visualization** - 18
   2.4 **Visualization Wizard** - 19
      2.4.1 **Description** - 19
      2.4.2 **Use Cases** - 19
      2.4.3 **Filtering** - 24
      2.4.4 **Aggregation** - 25
      2.4.5 **Mind Mapping** - 26
   2.5 **Mendeley Research Paper Mining** - 26
   2.6 **MindMeister Semantic Mind Maps** - 30
      2.6.1 **MindMeister Wunderkind** - 30
      2.6.2 **RDF / SKOS Publisher** - 35
      2.6.3 **Research Presentation Creation** - 37

3. **References** - 41
1 Introduction

Within the project a number of publicly available Web-enabled prototypes were developed. For the purpose of public dissemination and presentation of the project, this deliverable describes these prototypes from the user’s point of view. Simple use cases are given in the form of a sequence of steps accompanied by screenshots, illustrating how each prototype is used and applied. Descriptions of following prototypes are included:

- 42 data platform: a data-centric question and answering portal that builds around main CODE services
- Annotator Tool: a tool for intuitively annotating documents, creating models by applying machine learning techniques, and sharing the created models with others.
- Query Wizard: an easy to use user interface for searching, refining and transforming Linked Data
- Visualization Wizard: a tool for visual analysis of Linked Data which supports the user by automating the visualization process
- Mendeley Research Paper Mining: the prototype extracts facts in form of tables and figures from research papers, links them to Open Data and enables their utilization through the Mendeley Desktop Client
- MindMeister Semantic Mind Maps: prototype emphasizes the consumption and provision of Linked Open Data through mind maps on the MindMeister platform
2 Public Demonstrators

2.1 42-Data Platform

2.1.1 Description

42-data is a data-centric question and answering portal that builds around four main services for its users, namely: (i) Bookmark (Linked) Open Data found on the Web to your library or upload and semantically lift non Semantic Data, (ii) discover interesting insights in (Linked) Open Data through the provision of easy-to-use querying and analytical facilities, (iii) research-related discussions using (Linked) Open Data and (iv) donation-based revenue model.

On the basis of these services three main achievements are fostered that are currently not recognizable in other portals. In particular:

- **Lift non-Semantic Web (Open) Data**: The Open Data initiative is currently in a very active but early phase. There exist plenty of data silos that have not been integrated and interlinked to the Linked Open Data cloud. 42-data integrates crowd-sourced workflows to lift offline statistical data into the Linked Open Data cloud along with sophisticated disambiguation techniques of semantic concepts.

- **Community-driven insight generation**: Through discussions tailored to a specific domain, we hope to stipulate new discoveries and interesting findings based on open data. It is well known in academia that discoveries come out of intense communication processes. 42-data supports exactly those.

- **Exploring (Linked) Open Data**: 42-data offers services for discovering relevant data sets in the Linked Open Data cloud. The balloon service gathers necessary structural information of the Web of Data while our Query Wizard allows for easily searching the Linked Open Data cloud. Through exploring Linked Open Data, our users should be empowered to aggregate and integrate interesting data. Simple examples range from a list of Gold Medal Winners in Boxing to detailed publication statistics of journal articles on a particular topic.

While 42-Data is open to everybody, we especially target two user groups: First, researchers interested in sharing, discussing and analysing aggregated, empirical research data and second, citizens interested in discussing public data like Eurostats. The portal can be reached under:

http://www.42-data.org/

Within the portal, all services developed inside the CODE project have been integrated. In particular, the portal is built around the following prototypes to enable the prior mentioned workflows:

- **CODE Balloon** for integration of an index over the Linked Open Data cloud
- **CODE Data Extractor** for integration of unstructured primary research data (utilizing CODE Disambiguation and PDF annotation service)
- **CODE Bacon** to merge and manage data cubes
- **CODE Query Wizard** to enable the user to perform a discovery in the Linked Data cloud
- **CODE VizWizard** to create visualizations from data cubes
2.1.2 Enabling Discussions of Primary Research Data

In this section, it will be presented, how primary research data can be extracted from a scientific research paper stored as PDF document. The main aim is to discuss the findings of the paper in a broader context of a research community of the 42-data portal.

![Figure 1: Active question from a user in the 42-data portal](image)

In this example, the scenario is as following: One wants to know\(^1\), whether there are experimental results on the RCV1 benchmark available as illustrated in Figure 1. To answer we will integrate and discuss Table 1 from “Combined Regression and Ranking”\(^2\) written by D. Sculley showing specific evaluation results for the RCV1 benchmark corpus for text mining.

### 2.1.2.1 Lifting of unstructured data

The first step of this workflow undertakes the task to integrate the offline data into a semantically enriched data cube that is stored in the 42-data endpoint. The workflow is shown in the following steps.

---

2. [http://dl.acm.org/citation.cfm?id=1835804.1835928](http://dl.acm.org/citation.cfm?id=1835804.1835928)
Figure 2: Discover - entry point to integrate offline data in 42-data

Figure 3: Upload PDF document to the CODE Data Extractor prototype

It has to be mentioned that the actual processing of the PDF document is performed by the services provided by WP2.
After the upload step and the table extraction, an automatic guessing algorithm performs a pre-selection of dimensions and observations on the tables. In this step, data types are inferred if possible.

**Figure 4: Extracted tabular data**

**Figure 5: Disambiguation of table headers and insertion of provenance information**
In the last step of the triplification process, each column header is being disambiguated. Each header can be edited separately in terms of data type, linked data URL and so on. The insertion of metadata as well as provenance information is also part of this step, as illustrated in Figure 5.

After a data cube has been successfully created, the user is forwarded to the user profile page in the 42-data portal. The newly created data cube is accessible in the “Your data cubes” tab.

2.1.2.2 Discussing research data

The second workflow to discuss research data integrates the desired resource in a corresponding discussion thread. In this case, we will create a question, which is centred on this data cube displayed in the next figures. The data is taken from the data cube and added via the Visualisation Wizard. Other possible resources include (see subsequent figures):

- SPARQL Queries
- Google Spreadsheets
- Links to resources like Eurostats or the EU Open Data Portal
Figure 7: Creation of structured answers in 42-data portal

Figure 8: Finished answer for discussion thread – here, the answer consists of a free text and a visualization.
Citizens seem to be willing to use online services (see SlideShare resource), but what is the actual percentage of public services for citizens which are available online in EU countries? What does the trend look like (stagnating or rising)?

**Discussed Resource: Quality of Online Public Service Provision**

*Quality of Online Public Service Provision*

Slideshare presentation on Quality of Online Public Service Provision

![SlideShare](slideimage.png)

**Figure 9:** A question embedding a SlideShare with statistical data

---

**How do I get data citations to the Protein Database**

see [http://data.nature.com/query](http://data.nature.com/query)

**Endpoint:** http://data.nature.com/query

**Query:**

```sql
prefix npgd: <http://ms.nature.com/terms/>

select ?doi ?data
where {
  ?doi a npgd:Article ;
  npgd:hasDataCitation [ npgd:hasLink [ ?_ ?data ] ; npgd:type npgd:type ] .
  filter(regexp(?type, "pdb"))
} limit 25
```

<table>
<thead>
<tr>
<th>doi</th>
<th>data</th>
</tr>
</thead>
<tbody>
<tr>
<td>ncbi293</td>
<td>pdb:2WXH</td>
</tr>
<tr>
<td>ncbi792</td>
<td>pdb:2GZI</td>
</tr>
<tr>
<td>nature07307</td>
<td>pdb:3CLI</td>
</tr>
<tr>
<td>genome1163</td>
<td>pdb:4DY5</td>
</tr>
</tbody>
</table>

**Figure 10:** SPARQL Queries as Answer in 42-data
Benchmarking the Extraction and Disambiguation of Named Entities - Data pulled out

Raphael, nice work and nice overview. I have pulled out the data on Named Entity Linking in your paper and added it to a Google Spreadsheet (see below). Would be great to have the data also for the other experiments. Maybe others could add more results in this thread? What is the influence of parameters, data sets etc.?

Named Entity Linking Data from Rizzo et. al. 2014, Benchmarking the Extraction and Disambiguation of Named Entities on the Semantic Web

Named Entity Linking Comparison. Please find the details of the data in Rizzo, Erp, Troncy, Benchmarking the Extraction and Disambiguation of Named Entities on the Semantic Web, LREC 2014.

<table>
<thead>
<tr>
<th>Data Set</th>
<th>System (for NEL)</th>
<th>Source</th>
<th>Venue</th>
<th>Year</th>
<th>Precision</th>
<th>Recall</th>
<th>F1</th>
</tr>
</thead>
<tbody>
<tr>
<td>AIDA YAGO</td>
<td>AichemyAPI</td>
<td>Rizzo et. al.</td>
<td>LREC</td>
<td>2014</td>
<td>70.63</td>
<td>14.05</td>
<td>23.43</td>
</tr>
<tr>
<td>AIDA YAGO</td>
<td>dataTXT</td>
<td>Rizzo et. al.</td>
<td>LREC</td>
<td>2014</td>
<td>39.2</td>
<td>54.93</td>
<td>45.75</td>
</tr>
<tr>
<td>AIDA YAGO</td>
<td>DBPedia</td>
<td>Rizzo et. al.</td>
<td>LREC</td>
<td>2014</td>
<td>26.93</td>
<td>42.21</td>
<td>32.88</td>
</tr>
<tr>
<td>AIDA YAGO</td>
<td>Luepedia</td>
<td>Rizzo et. al.</td>
<td>LREC</td>
<td>2014</td>
<td>57.98</td>
<td>29.9</td>
<td>39.45</td>
</tr>
<tr>
<td>AIDA YAGO</td>
<td>Textrazor</td>
<td>Rizzo et. al.</td>
<td>LREC</td>
<td>2014</td>
<td>49.21</td>
<td>51.66</td>
<td>50.41</td>
</tr>
<tr>
<td>AIDA YAGO</td>
<td>THD</td>
<td>Rizzo et. al.</td>
<td>LREC</td>
<td>2014</td>
<td>32.5</td>
<td>40.1</td>
<td>35.9</td>
</tr>
</tbody>
</table>

Figure 11: An answer on Named Entity Linkage and an embedded Google Spreadsheet
2.2 Annotator Tool

Much of the work on the enrichment of scientific articles and mind maps is not directly visible to the end user. In order to make the results from the enrichment service more visible, a showcase web application has been developed, named the CODE Annotator Tool. The development has been steered to allow a rich interaction with the user, simulating the scenario of users who wish to extract and model specific knowledge. To accomplish this task, a tight integration with other tools from the project partners has been one of the goals.

The CODE annotator tool is available at: http://code-annotator.know-center.tugraz.at/

2.2.1 Use Case

The following section gives a brief overview of the typical work flow of a user interacting with the CODE annotator tool.

Figure 12: Data import in the Annotator Tool
Import the Data

Every user can create and store an arbitrary number of data sets. The most elegant way to create a new data set is to import a collection of scientific articles from a Mendeley account. One can import the whole library, a single folder or sub-folder, or all documents shared within a group (see Figure 12).

Import a Concept Hierarchy

Once a data set is selected, one or more models can be applied to this data set. For example, a model in the form of a concept hierarchy can be imported from Mindmeister (see Figure 13). Such a model consists of a set of entity classes and relations between them. If the model already contains example names for the concepts, the data set is automatically annotated by the tool.

Figure 13: Import a concept hierarchy in the Annotator Tool.
The document structure of scientific articles in PDF format is analysed by a number of tailored information extraction algorithms. The extracted document structure, table of contents, tabular data, and references are displayed (Figure 14). One can navigate through the pages of the document, highlighting the labelled components of the document.

2.3 Query Wizard

2.3.1 Description

There are very few people who are proficient in SPARQL, the W3C standard language to query Linked Data. As a consequence, the vast amounts of knowledge contained in the Linked Open Data Cloud remains underutilized.

People without background in semantic technologies are accustomed to simple search user interfaces, such as the one from Google. Also, when it comes to working with data, many people know how to use spreadsheet applications, such as Microsoft Excel. The CODE Query Wizard provides a web-based interface that dramatically simplifies the process of displaying, accessing, filtering, exploring, and navigating the Linked Data that’s available through a SPARQL endpoint. The main innovation of the interface is that it turns the graph structure of Linked Data into tabular form and
provides easy-to-use interaction possibilities by using metaphors and techniques that the end users are already familiar with.

The CODE Query Wizard offers two entry points: Users can either initiate a keyword search over a Linked Data repository, or they can select any of the already available datasets, represented as RDF Data Cubes. In both cases, the CODE Query Wizard presents a table containing the results. The users can then choose which columns they are interested in, and they can set filters to narrow down the displayed data. Additionally, they can explore the data by “focusing” on an entity, or they can aggregate a dataset to get a summary of the data.

The CODE Query Wizard is available online at http://code.know-center.tugraz.at/search

2.3.2 Search, Filtering and MindMap Generation

Figure 15: Searching for “toy story” in the Wikidata (CODE Edition) endpoint.

In Figure 15 the user typed in the search query “toy story”. After pressing the “Search” button, a full text query will be executed and all labels containing the search terms will be returned.

Figure 16 shows the first 10 out of 21 search results, which are displayed in the form of a table. The table initially contains two columns: “Label”, which contains a short textual description (and where the full text search was performed in), and “Type”, which describes the type of the found entities. The user can load additional search results (“Load 10 more results”, “Load all results” buttons) or add additional columns (“Add column…”).
Figure 16: 10 out 21 results found for “toy story” shown as a table consisting of two columns: “Label” and “Type”.

The user can also refine the displayed results by setting filters. Figure 17 shows the menu for setting “film” as a filter in the “Type” column. Filters can be set for different column types (URIs, text, numbers, dates and times), and multiple filters can be combined for advanced filtering.
When the user is happy with the displayed data, they can easily create a nicely looking mind map based on the data using the integrated MindMeister mind mapping service (see Figure 19). This feature is especially useful for getting a quick overview over a certain topic with only a handful of results.
Figure 18: Result of the filtering operation shows the three Toy Story films.

Figure 19: Information on the three Toy Story films, visualized as a MindMeister mind map.
2.3.3 Data Set Aggregation and Visualization

Instead of searching for relevant Linked Data themselves, users can also load already existing data sets (in the form of an RDF Data Cube). Figure 20 shows such a data set, in this example the “% of basic public services for citizens, which are fully available online”, provided by the EU Open Data platform. Again users can easily set filters if they are only interested in parts of the available data.

![Figure 20: Prepared dataset (RDF Data Cube) displayed in the Query Wizard.](image)
2.4 Visualization Wizard

2.4.1 Description

To support users in the visual analysis of Linked Data, the CODE project provides another wizard, called the CODE Visualization Wizard (or Vis Wizard for short). Vis supports the user by automating the visualization process. This means that after analyzing the structural and semantic characteristics of the provided Linked Data (we rely on the RDF Data Cube vocabulary\(^3\)), the Wizard automatically suggests any of the 10 currently available visualizations that are suitable for the provided data. Furthermore, the Vis Wizard automatically maps the data on the available visual channels of the chosen visualization. If the users wish to adjust the mapping, they can do so with a few simple clicks. Usually, more than one visualization is suitable for any given dataset. In that case, multiple visualizations can be displayed side by side. When certain parts of the data are selected in one of the visualizations, they are automatically highlighted in all of the others as well. This can provide quick insights into complicated data, taking advantage of the powerful human visual perception system. Furthermore, the user is supported by the Vis Wizard to organize, manage, refine and inspect the visualized data with the following methods:

1. Brushing in multiple coordinated views
2. Filtering
3. Aggregation

\(^3\) [http://www.w3.org/TR/vocab-data-cube](http://www.w3.org/TR/vocab-data-cube)
Data manipulations techniques like aggregation and filtering, and interactive techniques coordinated brushing, allow the user to refine the visual representation of the data providing powerful visual analysis functionality.

Visualisation Wizard is available at [http://code.know-center.tugraz.at/vis](http://code.know-center.tugraz.at/vis)

### 2.4.2 Use Cases

For a given data set, a list of meaningful visualization is suggested (see Figure 22). The user can choose from the suggested charts by clicking on the corresponding button, the visualization will be automatically generated. If for a given data set multiple visualizations are possible, these can be generated and viewed side by side.

![Possible Charts](http://code.know-center.tugraz.at/vis)

**Figure 22: Five suggested charts, shown as enabled buttons.**

Supported visualizations currently include:

1. Pie chart
2. Line chart
3. Bar chart
4. Grouped bar chart
5. Bubble chart / scatterplot
6. Stream graph
7. Geo visualization
8. Scatterplot matrix
9. Parallel coordinates
10. Table view
When user selects one of the suggested visualizations Vis Wizard automatically maps the data on the available visual components (e.g. axes) of the selected visualization. As already mentioned if user wishes to adjust the mapping, for example by changing what is shown on the x-axis and what is represented by colours, this can be achieved with a few simple clicks. An example is shown in in Figure 23, where in the upper image years are shown on the x-axis and countries are encoded through colour, while in the lower image it is vice-versa.
The user is supported by the Vis Wizard to organize, manage, refine and inspect the visualized data for instance with brushing functionality in multiple coordinated view. The idea behind brushing is to select a subset of data in one visualization (the "base chart") and see the distribution of the selected data in one or more other visualizations. All visualizations integrated in the Visualization Wizard support coordinated brushing over multiple views. Visually, the data elements selected by the brush retain their original color, while data elements not selected by the brush are shown in gray. Figure 24 shows a parallel coordinates visualization where a brush over the counties is applied, while the Figure 25 shows the result of the brushing operation in the scatterplot (up) and grouped bar chart (bellow).
Figure 24: Brushing over countries in parallel coordinates.
Figure 25: Brushing in multiple coordinated views (scatterplot and grouped bar chart) with Parallel coordinates (previous figure) as the base chart
2.4.3 Filtering

To filter data in the Vis Wizard the following filtering operations are supported:

- Filtering by data element: for example in a dataset on project funding per county the amounts for Austria shall be shown (see Figure 26).
- Range filtering for numeric data and dates: When setting a numeric range filter only data elements within the selected data interval will be visualized.
- Filtering by numeric observation categories (e.g. planned funding, consumed resources etc.): once a whole category of numeric observations is removed the data set a different set of visualizations will be suggested by the Vis Wizard.

The CODE Query Wizard is used to set the filters functions (see Section 2.3).

![Figure 26: Filtering a dataset using filtering by data element.](image-url)
2.4.4 Aggregation

The aim of the aggregate function is to group the values of multiple rows together to form a single value. The following aggregation functions are currently included in the Vis Wizard: Average, Count, Maximum, Minimum, and Sum.

Figure 27: Base (up) and aggregated (bellow) data set derived by grouping by year and using maximum value as function.
After the aggregation, the original and the aggregate cube are displayed together. Since the structure of the aggregated data set is different than the original one, different visualizations will be suggested for the latter. Also, the Visualization Wizard provides coordinated brush functionality between the visualization for base cube and the visualizations for aggregated dataset. In Figure 27 an example is shown where EU project funding for countries is aggregated over years using the maximum function.

2.4.5  Mind Mapping

The MindMeister mind mapping service is used to turn the generated visualization into a nice mind map. Each visualization contains a static preview image and a link back to the interactive version of the chart in the Vis Wizard (see Figure 28).

![Figure 28: MindMeister mind map created from Linked Data via the Vis Wizard](image)

2.5  Mendeley Research Paper Mining

Mendeley Research Paper Mining is a prototype which extracts facts in form of tables and figures from research papers, links them to Open Data and enables their utilization through the Mendeley Desktop Client. The new features developed for Mendeley Desktop as part of the CODE project are accessible via two new tabs that appear when the user selects a supported paper, either from their library or by doing a search of Mendeley’s catalog of the world’s research.

To view the content extracted by CODE services for academic papers, the user first selects the ‘Literature Search’ view in Mendeley Desktop and enables the ‘Open Access’ filter to restrict searches to open access papers (see Figure 29).
The user can then perform a search on a topic of interest. For open access papers where Mendeley has a copy of the paper available, a PDF icon is displayed next to the results. The user then selects one of these results and in the right-hand pane two new tabs, “Contents” and “Enrichments”, are displayed.

Selecting the “Contents” tab (see Figure 30) displays an overview of the structure of the paper, automatically extracted by CODE services developed by the Know-Center. The user can then select an entry in the Contents view to jump directly to that part of the paper. This is a useful addition to Mendeley since only a small proportion of existing papers in PDF format have such a structured overview embedded in them.

The user can also view a list of tables (see Figure 31) and figures (see Figure 32) that appear in the paper by selecting the ‘Enrichments’ tab. This tab contains thumbnails and captions of figures along
with tables of data found in the paper. This allows the user to quickly browse the results of experiments given in the paper. Selecting a table or figure causes Mendeley’s PDF viewer to scroll to that section of the paper.

![Figure 31: Tables extracted from a paper.](image)

In the original paper, the tables appear only as text and consequently performing any kind of analysis or manipulation of the data requires laborious work. In the Enrichments view however the tables have been converted to structured data which can easily be copied to the clipboard via the drop-down menu in the table heading. The copied data can then be imported into spreadsheet tools such as Excel or other tools for further analysis or visualization.

The layout of the extracted structured tables is often imperfect but can usually be cleaned up in Excel or another application quickly and this is much easier than working with the text representation of the table in the original paper.
The figure list provides an easy way for the user to see all of the figures found in the paper together.

Figure 32: Figures extracted from a paper.

The user can report errors in the data extraction by clicking a ‘Report Problem’ link from the drop-down menu for each extracted table or figure. This feedback can then be used in future work to improve the extraction quality.

Finally, the Enrichments tab displays a notification on first use with a link to more information about the features developed as part of the CODE project.
2.6 MindMeister Semantic Mind Maps

2.6.1 MindMeister Wunderkind

2.6.1.1 Description

The idea behind Wunderkind was to allow for automated, contextual content suggestion by providing MindMeister users a virtual partner to collaborate with while building mind maps. In the first version, the virtual partner would provide suggestions only when triggered by user interaction, although this can be expanded in the future to a more intelligent behaviour of showing relevant data as the user is adding new content to the mind map in a “Experimental Features” mind map mode.

Figure 33: Current flow of the Wunderkind interaction
The figure above shows the data flow of the Wunderkind feature. At most, there are three calls made to the respective endpoint that provides the suggestions based on the selected topic of the mind map.

### 2.6.1.2 Use Case

A typical request for Wunderkind is detailed below.

![Image of Open Settings dialog](image)

**Figure 34: Open Settings dialog**

To enable the MindMeister CODE showcases, an existing user needs to activate the “Experimental Features” setting. This is done by first opening the settings dialog.

![Image of Experimental Features setting](image)

**Figure 35: Experimental Features setting**

There, the user clicks the respective checkbox on the first tab and presses OK. Experimental features are now active.
To get WunderKind suggestions, first open the WunderKind footer bar in the mind map editor. The bar will display found matches for the currently selected topic.

WunderKind results are retrieved when a user:

- Enters a new topic
- Edits an existing topic
- Clicks the blue WunderKind button on a selected topic
- Presses Shift+Space on a selected topic
The footer bar shows the number of results, a panel with the first result which reveals more results on hover, as well as two dropdown menus: one labelled “Type” for disambiguation of the selected topic, and one labelled “Category” for the respective Freebase category of the results. Both selections are automatically prefilled using disambiguation and named entity recognition mechanisms (the latter uses contextual map information such as map title and parent topic to determine a fitting category).

Next, the user moves his mouse over the results field to reveal more results. The list is usually scrollable, and by clicking onto one or more results they are selected and added as children to the currently selected topic in the mind map.

The inserted topics link back to the respective Freebase resource page, in the case of “Regent’s Park” in London [http://www.freebase.com/view/m/01lp6].
Figure 39: The user selects matching results to add to the mind map

Figure 40: Type and Category selection can be overridden
Of course, the user can override the automatic selection of Type and Category using the dropdowns in the WunderKind footer.

During testing, we experimented with a number of data sources, such as Google, Wikipedia, and MindMeister Public Maps. In the tests conducted to far, Freebase yields the best results, which is why it was chosen as default source in the final user interface.

2.6.2 RDF / SKOS Publisher

2.6.2.1 Description

Within the CODE project, it is highly desired to publish the extracted as well as user generated data in a Linked Open Data aware manner. Following this premise, an endpoint has been implemented for the industrial partner MindMeister and an automated service process has been implemented to trigger a periodic update of changed data to the LOD cloud.

In detail, the following information is stored in the respective endpoint for MindMeister. As shown in other CODE deliverables, mind maps support the user to visually manage the extracted and generated data, e.g., produced by the CODE Query Wizard. Furthermore, the encompassed information can be easily changed and shared with others. To follow the idea of Linked Open Data, the MindMeister API has been extended to dump each public mind map using RDF.

2.6.2.2 Use Case

Exporting a mind map in RDF can be done in two ways. First, any public mind map can be accessed at:

http://www.mindmeister.com/[map_id].rdf

where [map_id] is the public ID of the specific mind map. Secondly, it can be downloaded as RDF via the MindMeister export dialog (see Figure 41). For the export dialog to show the option, “Experimental Features” need to be enabled (see 2.6.1.2),
Figure 41: Export dialog in MindMeister with the option "Export as RDF file"

The entire structure of the mind map is encoded in RDF and the given format is Notation 3 (N3) (Figure 42). In particular the following data is included:

Map:
- title
- description
- date of creation
- date of last update
- is shared

User (creator and collaborators):
- name
- user id
- url to channel page on MindMeister

Node in the map:
- parent node
- label
- creator
Further information on the creation of the endpoint can be found in D3.1 and D3.3 accordingly. The endpoint can be accessed at:

http://zaire.dimis.fim.uni-passau.de:8080/bigdataMindmeister/

Currently, the input routines are up and running. After a short revision phase, both endpoints will be registered at the CKAN platform with their specific information.

2.6.3 Research Presentation Creation

2.6.3.1 Description

Finally, the MindMeister presentation mode provides an attractive way of visualizing interlinked information as effect-driven slideshow and differentiates itself from classic PowerPoint-style presentations by maintaining structure and context for the user during the course of the
presentation. We aimed at providing easy ways for researchers to turn papers into online presentations, using automated extraction tools and enrichment with LOD visualizations.

At this stage in the project we’re proud to say that we’ve successfully implemented modules for all three scenarios and included them in the beta version of the CODE platform.

The capability of importing PDFs on MindMeister allows the automatic creation of mind maps from the table of content of an arbitrary scientific paper (Figure 43).

**Figure 43: Data flow of mind map creation from table of content of scientific papers**

The PDF file is sent to MindMeister via an import dialog in the web interface. The next step includes the extraction of the table of content of the paper. This is done by the Enrichment Service. The input is the binary data of the PDF and the table of content is returned in JavaScript Object Notation (JSON) back to MindMeister.

**2.6.3.2 Use Case**

After enabling “Experimental Features” (see 2.6.1.2), the MindMeister import dialog accepts also PDF files as input.
First, a user selects one or more PDF research papers to import into MindMeister using the Import dialog (Import button in map listing).

**Figure 44: Import dialog with "PDF" option**
The Know-Center service for table-of-content extraction is then called on the uploaded documents. And a mind map is created for each document so that each header corresponds to a node in the mind map and its level represents the level of the node in the mind map.

Using MindMeister’s auto-create presentation function, the mind map can now be turned into an online slideshow at the click of a button (Figure 45). Researchers can use this functionality to quickly create an overview of a research paper and publish it online, in turn making it available as RDF linked data through the feature explained above.
3 References

[1] CODE Description of Work, Version Date 20012-01-23