OpenCube

Publishing and Enriching Linked Open Statistical Data for the Development of Data Analytics and Enhanced Visualization Services

Deliverable 1.2

OpenCube Requirements and Conceptual Models

Editor(s):	Efthimios Tambouris (CERTH) Evangelos Kalampokis (CERTH)
Responsible Organisation:	CERTH
Version-Status:	V2 Re-submitted
Submission date:	14/11/2014
Dissemination level:	PU

This project has been funded with the support of the **FP7 Framework Programme** of the European Union

.

Deliverable factsheet

Project Number:	FP7 - 611667
Project Acronym:	OpenCube
Project Title:	Publishing and Enriching Linked Open Statistical Data for the Development of Data Analytics and Enhanced Visualization Services

Title of Deliverable:	D1.2 – OpenCube Requirements and Conceptual Models
Work package:	WP1 – Requirements Analysis and Conceptual Models
Due date according to contract:	30/04/2014

Editor(s):	Efthimios Tambouris (CERTH) Evangelos Kalampokis (CERTH)
Contributor(s):	Areti Karamanou (CERTH) Konstantinos Tarabanis (CERTH)
Reviewer(s):	SWIRRL
Approved by:	All Partners

Abstract:	This document includes the results of the second phase of the user requirements elicitation process. In specific, it presents (a) the OpenCube lifecycle, which describes the steps that raw multi-dimensional statistical data should go through in order to create value, (b) the refined user requirements categorized per lifecycle step and (c) a gap analysis between identified requirements and existing tools.
Keyword List:	Requirements, functionalities, linked data, statistical data.

Consortium

	Role	*Name*	*Short Name*	*Country*
1.	Coordinator	Centre for Research and Technology - Hellas	CERTH	Greece
2.	R&D partner	National University of Ireland, Galway	NUIG	Ireland
3.	SME partner	Fluid Operations AG	FLUIDOPS	Germany
4.	SME partner	SWIRRL IT LIMITED	SWIRRL	UK
5.	SME partner	ProXML bvba	ProXML	Belgium

Revision History

Version	*Date*	*Revised by*	*Reason*
0.1	1/3/2014	CERTH	ToC
0.2	1/4/2014	CERTH	OpenCube Lifecycle
0.3	15/4/2014	CERTH	Requirements refinement
0.4	18/04/2014	CERTH	Gap analysis, conclusions
1.0	30/04/2014	CERTH	Submission to the EC
1.1	30/10/2014	CERTH	Update of the OpenCube lifecycle
2.0	20/11/2014	CERTH	Submission to the EC (v2)

Statement of originality:

This deliverable contains original unpublished work except where clearly indicated otherwise.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

Table of Contents

List of Figures

List of Tables

List of Abbreviations

The following table presents the acronyms used in the deliverable in alphabetical order.

Abbreviation	*Description*
API	Application Programming Interface
CMS	Content Management System
CSO	Central Statistics Office
CSV	Comma Separated Values
DCLG	Department for Communities and Local Government
EC	European Commission
EU	European Union
GUI	Graphical User Interface
LOD	Linked Open Data
LOSD	Linked Open Statistical Data
LOV	Linked Open Vocabularies
OECD	Organization for Economic Co-Operation and Development
OGD	Open Government Data
OLAP	Online Analytical Processing
RDBMS	Relational Database Management System
RDF	Resource Description Framework
UK	United Kingdom
UN	United Nations
URI	Uniform Resource Identifier
WP	Work Package

Executive Summary

OpenCube project aims at developing software tools that facilitate (a) publishing of raw multi-dimensional statistical data (data cubes onwards) as high-quality Linked Open Data and (b) reusing distributed Linked Open Data in data analytics and visualizations.

WP1 is responsible for eliciting user requirements that will guide the development of the OpenCube components. In specific, it aims at (a) identifying and documenting the requirements related to publishing of linked data cubes and reusing them in visualizations and data analytics, and (b) designing and specifying the OpenCube conceptual models in the terms of a lifecycle for linked data cubes.

The present deliverable is the second deliverable of WP1, D1.2 - OpenCube Requirements and Conceptual Models. Its purpose is to document the final list of requirements as well as the methodology for eliciting and assessing the requirements that the OpenCube components and tools will address. Information that is included in this deliverable is valuable to all partners for ensuring the appropriate development of the OpenCube architecture and OpenCube components and tools.

More specific, the methodology followed in this deliverable capitalizes and extends the results of the first deliverable (i.e. D1.1) of WP1. The deliverable includes the following results:

- The OpenCube lifecycle that describes the steps that raw data cubes should go through in order to create value. The lifecycle comprises 8 steps: (i) Discover & pre-process raw data, (ii) Define, structure & create cube, (iii) Annotate cube, (iv) Publish cube, (v) Discover & explore cube, (vi) Transform cube, (vii) Analyze cube, (viii) Communicate results. The first four steps regard linked data cubes publishing, while the last four linked data cubes consumption.
- The refined list of user requirements categorized according to the steps of the lifecycle. The final list includes 48 functional and 13 non-functional requirements.
- Gap analysis between existing tools and the identified functional requirements. The analysis indicates that requirements that are related to (a) Annotate cube, (b) Discover & explore cube, and (c) Transform cube are not fully addressed by the existing tools.

1 Introduction

The aim of this section is to introduce the background of the work pursued with Task 1.2 “Refinement of requirements and conceptual models”. The scope and the objective that the current document has set out to achieve are presented in sub-section 1.1. The intended audience for this document is described in sub-section 1.2 while sub-section 1.3 outlines the structure of the rest of the document.

1.1 Scope

The present document is the Deliverable 1.2 “D1.2 - OpenCube Requirements and Conceptual Models” (henceforth referred to as D1.2) of the OpenCube project. The main objective of D1.2 is to refine the initial list of requirements included in D1.1 “Initial OpenCube Requirements” and document the final set of user requirements that will be addressed by the OpenCube platform. These requirements will feed into the development of the OpenCube main artifacts, i.e. the OpenCube Reference Architecture and the OpenCube toolkit and platforms extensions.

1.2 Audience

The intended audience for this document is the OpenCube consortium, the European Commission, and the public interested in developing software tools for managing linked data cubes.

1.3 Structure

The structure of the document is as follows:

- Section 2 presents the methodology used to refine the initial list of requirements and concludes with the final methodology.
- Section 3 provides background information on the results of the first phase of user requirements elicitation that was described in D1.1.
- Section 4 presents the OpenCube lifecycle that describes the process that raw data cubes should go through in order to create value.
- Section 5 presents the final list of OpenCube user requirements categorized in the steps of the OpenCube lifecycle.
- Section 6 presents the gap analysis between existing tools and identified functional requirements.
- Section 7 draws conclusion and sets future goals.

2 Methodology

In this section, the methodology that was followed in order to come up with the final list of OpenCube requirements is described. The methodology is closely related to the first phase of the requirements elicitation process that was documented in D1.1. Figure 1 presents the activities of both phases and how the activities and the phases are related to each other. In particular, the Requirements Refinement phase capitalized on the following:

- The initial prioritized list of requirements along with the UML use case diagrams that was documented in D1.1.
- The OpenCube lifecycle that describes the process that raw data cubes should go through in order to create value.
- The OpenCube architecture that is documented in both D2.1 and D2.2. Specifically, the steps of the OpenCube lifecycle are unambiguously mapped to different parts of the architecture.
- The feedback of the pilot partners on the initial list of requirements.
- The review of existing linked data management tools.

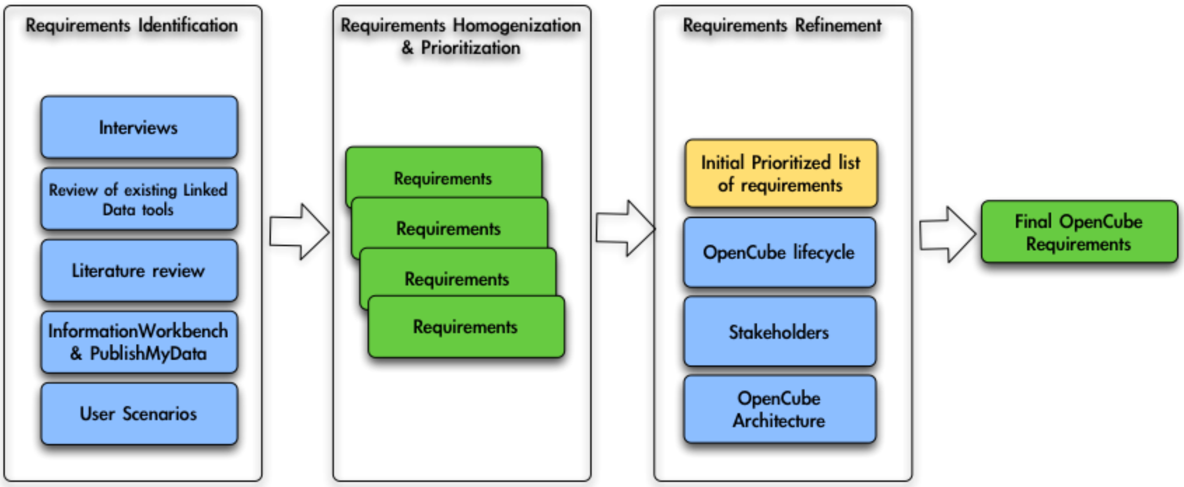
Requirements Identification
Interviews
Review of existing Linked Data tools
Literature review
InformationWorkbench & PublishMyData
User Scenarios
Requirements Homogenization & Prioritization
Requirements
Requirements
Requirements
Requirements
Requirements Refinement
Initial Prioritized list of requirements
OpenCube lifecycle
Stakeholders
OpenCube Architecture
Final OpenCube Requirements

Figure 1 The requirements elicitation approach

3 Background on OpenCube requirements

This section aims at providing the background information required for understanding the work presented in this deliverable. In particular, it describes typical linked data lifecycles that have been proposed in the literature. It also presents the RDF data cube vocabulary, which is a W3C standard for modelling multi-dimensional data as RDF graphs. Finally, the section presents the preliminary OpenCube lifecycle, which was included in D1.1.

3.1 Linked Data Lifecycle

A few Linked Data lifecycles have been recently proposed in the literature to describe the steps that data go through in publishing and reusing Linked Data. These lifecycles focus on data that describes things in the world (e.g. people, locations) and how they are related to other things and thus do not fully address the needs of multi-dimensional data.

Hyland & Wood (2011) have specified a Linked Data lifecycle comprising seven steps: (i) identify the actual data to be transformed to Linked Data, (ii) define the model that will be used, (iii) name objects using URIs, (iv) describe data using existing, widely used vocabularies, (v) convert data to RDF, which can be performed using triplification, partial script conversion or expert modelling followed by scripted conversion, (vi) publicize data through initiatives such as CKAN or LOD cloud, and (vii) maintain data.

Moreover, Auer et al. (2011) defined eight steps in the Linked Data lifecycle: (i) transform data to RDF, which includes the extraction of data from sources (structured or unstructured) and its mapping to an RDF data model, (ii) store and index data efficiently and using appropriate mechanisms, (iii) manual revise, extend and create new structured information according to initial data, (iv) establish links to different sources that refer to the same entities but are published by different data publishers (v) enrich data with high-level structures so as to be more efficiently aggregated and queried (vi) assess data quality using data quality metrics available for structured information such as accuracy of facts and completeness, (vii) repair data quality problems identified in the previous step, and (viii) search, browse and explore the data in a fast and user friendly manner.

Finally, Villazon-Terrazas et al. (2011) suggest that the process of publication of government Linked Data comprises of the following steps: (i) identification of government sources and the design of the URIs of the resources, (ii) determination of the ontology to be used for modelling the domain of the data identified sources (including identification of suitable vocabularies to reuse), (iii) transformation of the whole data source content into RDF as well as cleansing of the datasets and their linking with other datasets, (iv) storing of data, publishing of metadata and data publicizing, and, finally, (v) exploiting data in order to enable transparency and encourage data reuse.

3.2 Data Cube vocabulary

Statistical data should be modelled as data cubes that describe data in a multi-dimensional fashion. Towards this end, the RDF data cube vocabulary can be employed. This vocabulary is a W3C standard for modelling multi-dimensional data, such as statistics, as RDF and thus adhering to the linked data principles.

Data cubes are characterised by their *dimensions*, their *measures* and possibly by additional *attributes* (Figure 2). The dimensions in the cube define what each observation is about. For instance, a cube related to unemployment might have the following dimensions: location, time period, gender and age group. An individual observation refers to a unique value along each of these dimensions. In the unemployment example, a single observation would define the unemployment rate of women (*gender*) between 22 and 30 years old (*age group*) that live in Kensington (*location*) for the first six months of 2009 (*time period*). The possible values for each dimension are taken from a *code list*. A code list is a controlled vocabulary such as a list of constituencies in the UK or possible age groups. The measure defines what kind of quantity is being measured e.g. the unemployment rate. An observation could also have attached attributes, which facilitate the interpretation of the observation value e.g. the unemployment rate is measured as percentage and is an estimation of the Ministry of Labour.

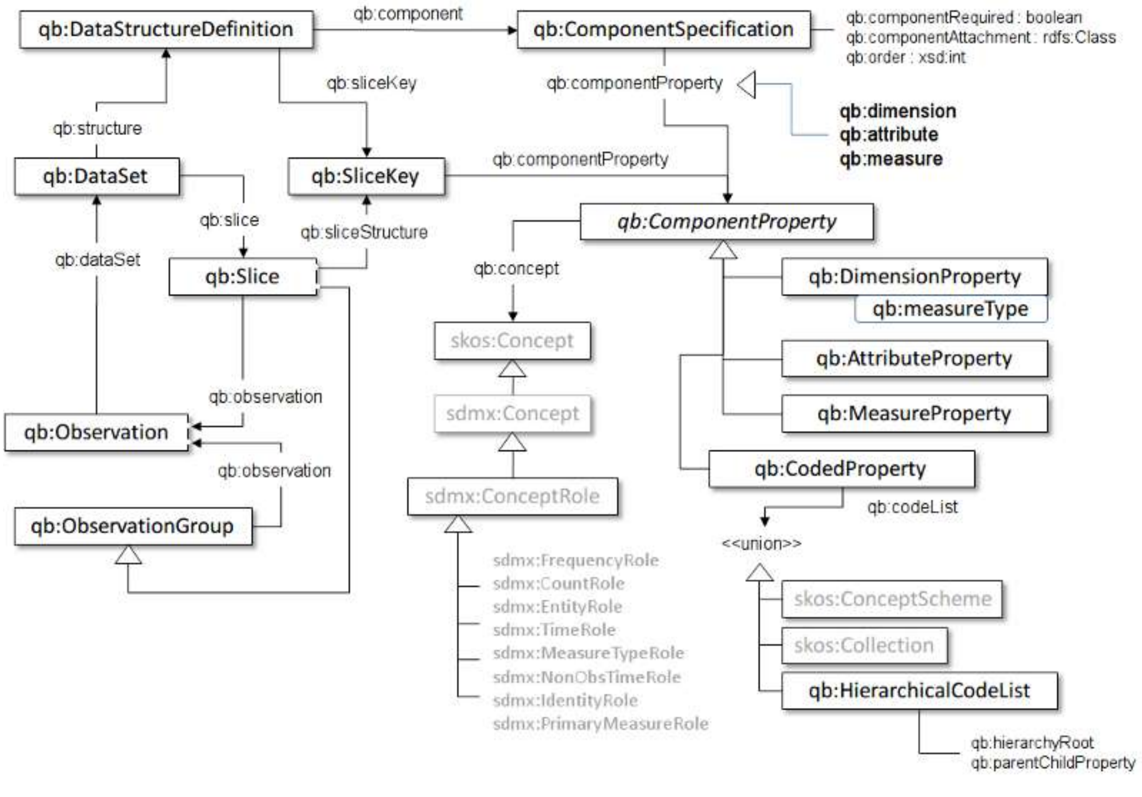

Figure 2 The RDF Data Cube Vocabulary

In RDF Data Cube, the qb:DataSet represents the resource of the entire data set, a data set that corresponds to the defined structure of the RDF Data Cube. The data sets are allowed to be organized in several slices. The structure of qb:DataSet or of a slice of the actual data is defined by the class qb:DataStructureDefinition. Qb:DataStructureDefinition is associated with the qb:component property in order to specify the component(s) of the datasets structure. The qb:ComponentProperty is the super class property of the properties that represent dimensions, measures and attributes namely qb:DimensionProperty , qb:MeasureProperty and qb:AttributeProperty respectively.

3.3 LOSD lifecycle

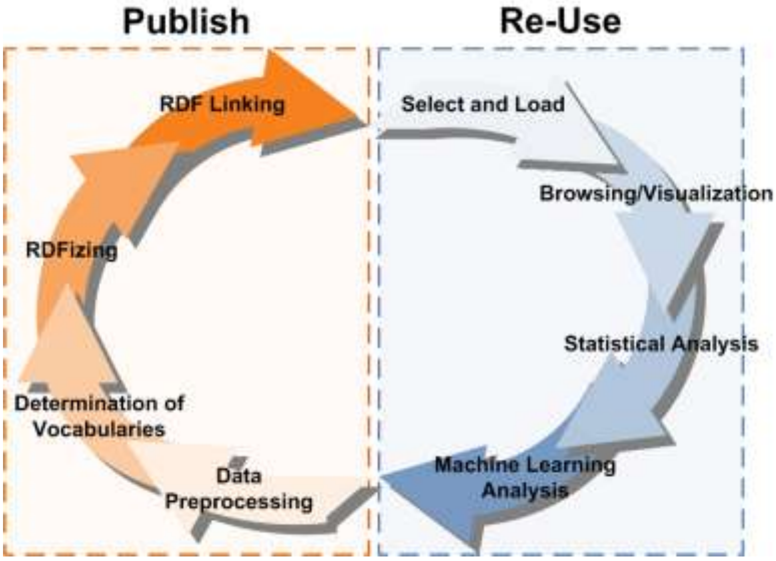

Figure 3 The preliminary OpenCube lifecycle

In D1.1 a linked data cube lifecycle was presented. In particular, by adapting the generic Linked Data lifecycles we come up with this new lifecycle that comprises eight phases:

- **Data pre-processing**: Data pre-processing regards the transformation of initial data into a form that will facilitate the data publishing process and improve the quality of the data. Data pre-processing functionalities may regard the data itself or the structure of the data. Data pre-processing phase can be decomposed into four steps, namely
    - data import,
    - data filtering (i.e. removing unwanted data),
    - data cleaning (i.e. identifying and fixing possible mistakes), and
    - data transformation (i.e. create new data from the initial one).
- **Determination of ontology:** Determination of Ontology is the process that defines the ontology that will be used to model data. The most important step of the ontology determination phase regards the definition of the structure of the ontology that data will be mapped to. As reusing of current vocabularies is considered to be of high importance in this phase (Bizer et al., 2007), defining the structure of the ontology also requires importing and reusing existing RDF vocabularies and their properties.
- **RDFizing:** RDFizing is the process of transforming data to RDF. RDF instances and properties should be uniquely characterized by an RDF URI and strictly conform to the structure of an RDF vocabulary.
- **RDF linking:** The RDF Linking step concerns the discovery of relationships and the establishment of links between entities in related RDF data.

- **Browsing/visualization:** Browsing/Visualization is the process of presenting RDF data in a comprehensive, meaningful and effective way.
- **Statistical analysis:** Statistical Analysis is the process of performing descriptive statistics on data.
- **Machine learning:** Machine learning is the process of learning from data in order to perform reasoning on large datasets, map specific inputs to desired outputs and predict future outputs based on specific inputs.

4 The OpenCube Lifecycle

In this section the OpenCube lifecycle is presented. This lifecycle presents the steps that raw multi-dimensional data (onwards data cubes) should go through in order to create value. It includes the publishing of raw data cubes as Linked Data and the consumption of Linked Data cubes in data analytics. The lifecycle comprises 8 steps: (i) Discover & pre-process raw data, (ii) Define structure & create cube, (iii) Annotate cube, (iv) Publish cube, (v) Discover & explore cube (vi) Transform cube, (vii) Analyze cube, (viii) Communicate results. The first four steps regard Linked Data Cubes publishing, while the last four Linked Data Cubes consumption.

We should note here that the specification of the OpenCube lifecycle takes into account the need to develop an architectural design for the OpenCube solution. This means that there should be a direct mapping between the steps of the lifecycle and the parts of the architecture.

The steps of the lifecycle are presented in detail in the following sub-sections. The special requirements of data cubes processing in each step of the lifecycle are presented and the differences with typical Linked Data lifecycles are highlighted.

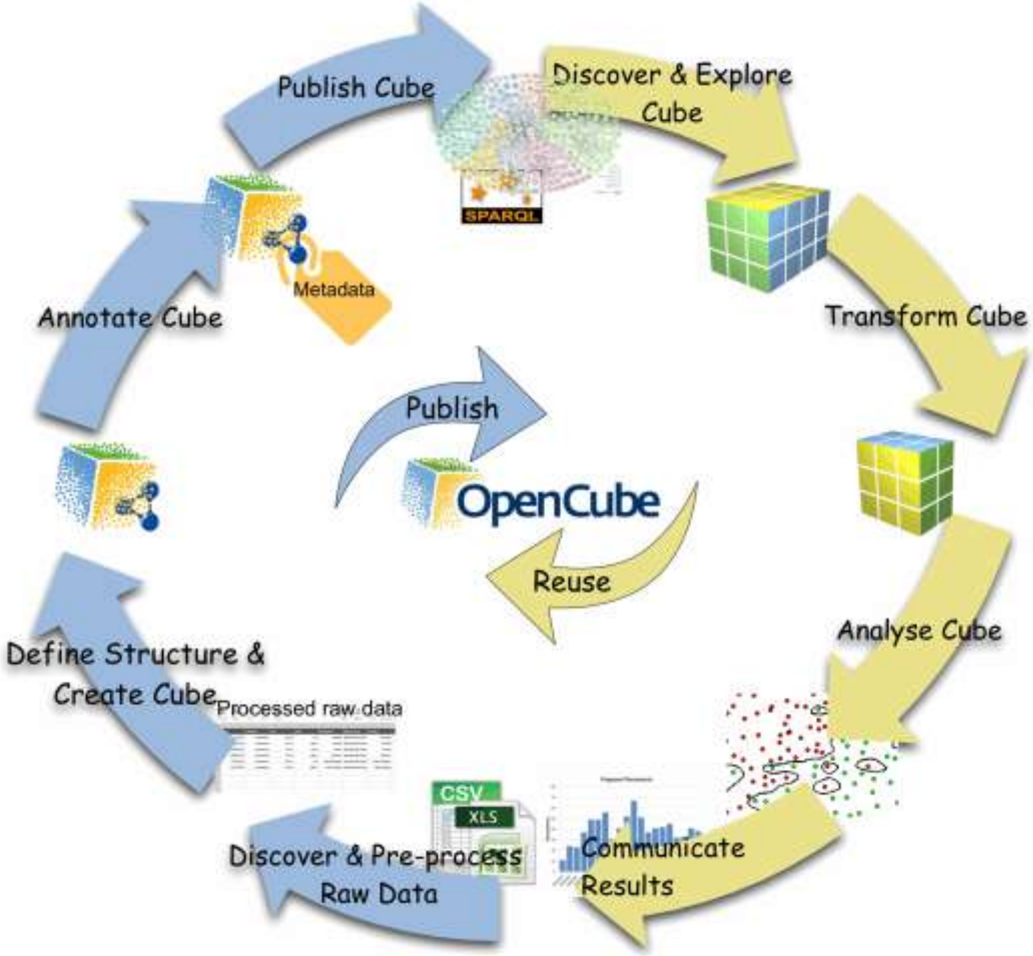

Figure 4 The OpenCube Lifecycle

4.1 Discover & pre-process raw data

The step enables users to discover, access, view and process raw data cubes. At this step data cubes come in various data formats such as CSV files, XLS files, RDBMS or RDF files. In addition, cubes can be formatted in various structures:

- Rectangular data structure, which includes columns containing variables for a set of cases, contained in the rows. This type of data is often collected by survey, although in some cases may come from administrative sources, sensors, or registers.
- Tree data
- Graph data

Finally, data cubes can be imported from different data sources such as local files, files pointed by URLs, html tables, APIs etc.

In this step, users are able to browse raw data and also to perform various activities that aim at improving the quality of the raw data. These activities may include:

- Data sorting. The user can sort imported data with a variety of sorting services including alphabetical sorting, numerical sorting and sorting by date
- Data filtering e.g. removing entire columns or rows.
- Data cleansing. This includes identifying and fixing possible mistakes such as spelling mistakes or removing duplicates.
- Data transformation. This includes creating new data out of existing data in order to (a) create common data formats (e.g. in the case of time or geographical data), (b) handle null values, (c) handle missing values, (d) convert character encoding, etc.
- Data aggregation from multiple files or sources. For example, consider the case of the unemployment rate at European countries in different years and age groups. This case involves three dimensions (i.e. countries, years and age groups) and thus it is difficult to include all the information in one file. So, the data could be divided in multiple files. For example, the unemployment rate per country and age group in a specific year could be described in a single CSV file. So, aggregating data from multiple files would provide information for more than one year.

This step could also include raw file or raw data storage in a local repository or database system. In this case, metadata regarding the provenance of the raw data may be also stored along with the actual data.

4.2 Define structure & create cube

An important step in Linked Data creation regards the definition of the structure of a model that the data will be mapped to. At this step of the lifecycle the raw data cube is further processed in order to create the structure of the Linked Data Cube. Initially, a conceptual model that drives the development of the structure of the Linked Data Cube is created. This specifies:

- The dimensions of the cube, which define what the observation applies to.

- The measured variables (i.e. what has been measured) along with details on the unit of measure or how the observations are expressed.

As reusing widely accepted vocabularies is considered to be of high importance in linked data, defining the structure of the model also requires importing and reusing existing linked data vocabularies. In the case of data cubes, however, the *RDF Data Cube vocabulary* is a W3C standard and thus always used as the main framework to model data cubes as RDF graphs. So, the dimensions and measures that formulate the conceptual model should be used to define dimensions, measures and attributes of the RDF data cube vocabulary.

Although the RDF Data Cube vocabulary is used to structure a data cube as an RDF graph, other linked data vocabularies can be also used to define the values of the dimensions, measures and attributes of the cube. Common statistical concepts can be reused across datasets e.g. dimensions regarding age, location, time, sex etc. or the values of a specific dimension (e.g. the countries of Europe). These concepts are defined in linked data vocabularies that standardize dimensions, attributes and code lists. The most widely accepted is the SDMX-RDF vocabulary[1], which is based on the statistical encoding standard SDMX.

As a result, publishing linked data cubes mainly require search and discovery of controlled vocabularies. The reuse of these vocabularies could enable *the linking of disparate cubes*. This would in turn facilitate the performance of comparative analytics and visualizations on top of cubes that share at least one common dimension. In particular:

- By linking the dimensions of different cubes one could perform comparative analyses or visualizations of multiple measures across the same dimensions. For example, one could compute the correlation between unemployment and crime rate by analyzing observations from two data cubes that share the dimension countries in a specific year.
- By linking the measures of different cubes one could create larger cubes that describe the same measure.

We should also note that reusing controlled vocabularies could be considered as reconciling against such collections. In general, in data cubes the establishment of links between different datasets is realized at the schema level (dimensions, attributes and measures) and not at the instance level.

As a result, this peculiarity of data cubes introduces an extra need that is related to the management of controlled vocabularies that could be reused across different datasets. This includes the creation, store, search, discovery and reuse of existing controlled vocabularies. This is a very challenging process that should be addressed in order to enable controlled vocabularies reuse at a Web scale. In the creation of these vocabularies popular datasets such as DBpedia or GeoNames could be reused.

In this step a number of design issues regarding the structure of the cube should be taken into account. These include:

[1] https://code.google.com/p/publishing-statistical-data/

- The use of sdmx-measure:obsValue as a dimension of the cube along with indicator dimensions and lists with all measures instead of defining and using a measure property. This approach is followed in the Eurostat - Linked Data dataset[2].
- The use of more than one dimension properties to define a single cube dimension. For example, both the qb:dimension sdmx-dimension:freq and the qb:dimension dcterms:date could define a single time dimension that describes yearly values of a measure.
- Attribute as a qb:DimensionProperty which takes values from a code list e.g. propoerty:unit in the Linked Data dataset of Eurostat.

This step also includes the creation of the actual RDF data out of the raw data based on the structure definition that was created at the previous step. This step includes the following activities:

- URI design.
- Definition of mapping between raw and RDF data. This activity introduces complexity in the case of data cubes because it presents different needs than the same activity in typical linked data scenarios, i.e. linked data that describe properties of some entities. This activity is presented in more detail below.
- Data storage to an RDF store.
- Validation for compliance with schema or values constraints.

The definition of a mapping between the raw and the RDF data is the most challenging activity in this step. The mapping is directly related to the electronic format and the structure of the raw data. For example, one common way of mapping spreadsheet to RDF data is as follows:

- The spreadsheet describes a set of entities belonging to a certain class.
- The rows of the spreadsheet are mapped to different entities constituting the subject of a triple
- The columns of the spreadsheet are mapped to the predicates of the triples.
- The values of the cells are mapped to the value of the triples.

In the specific case of data cube data represented in table form, often each cell in a table is mapped to a qb:Observation, with the contents of the cell becoming the value of the observation measure, and a column header and a row header corresponding to dimensions of the data. In some examples, a further data dimension is represented by multiple worksheets in the spreadsheet.

In the case of a relational database, the most common mapping process is as follows:

- Database tables are mapped to classes of entities.
- The rows of a database table are mapped to different instances of the class.
- The columns of a database are mapped to predicates of the triples.
- The values of the tables are mapped to the value of the triples.

[2] http://eurostat.linked-statistics.org

However, in the case of data cubes dimensions, measures and attributes should be extracted from the raw data. So, there is a need for appropriate mappings that will define which piece of data describes a value of a dimension or the actual measure and how these are related, i.e. which value of a dimension refers to which measure.

The actual mapping needs to be represented in a mapping language. For example, in the case of relational databases, the R2RML language[3] is a W3C standard that enables the expression of customized mappings from relational databases to RDF datasets. However, in the case of spreadsheet documents the community lacks a similar initiative. Current activity of the W3C 'CSV on the Web' working group may lead to relevant standards and the Open Cube project is monitoring the outputs of that working group.

4.3 Annotate Cube

This step refers to the enrichment of RDF data cubes with metadata to facilitate discovery and reuse. Sources of metadata include raw data files, the cube's structure and/or standard thesaurus of statistical concepts.

The RDF data cubes that were created at the previous step are now enriched with metadata in order to facilitate discovery and reuse. The metadata can be related to:

- Provenance information of the raw data files.
- Generic metadata about the data cube. This metadata are similar to those described by models such as DCAT[4] and could provide a categorization of the cube or provenance related information.
- The collection or production process of the actual data based on which the cube has been created. For example, the unemployment rate of the countries was collected from national public authorities in each country or it was estimated by Eurostat.
- The content of the cube. In statistical agencies it is common to have a standard thesaurus of statistical concepts (time, geographic area, currency), which underpin the components used in multiple different data sets. Towards this end, the qb:concept is used and is linked to controlled term list or thesaurus.

4.4 Publish cube

At this step, the generated data cubes are made available to the public through different interfaces e.g. Linked Data API, SPARQL endpoint, downloadable dump etc. In addition, during this step the datasets are publicized in data catalogues such as Europe's public data portal[5] or other national portals (e.g. data.gov.uk or data.gov.gr), the datahub platform[6] or the Linking Open Data cloud[7].

[3] http://www.w3.org/TR/r2rml/
[4] http://www.w3.org/TR/vocab-dcat/
[5] http://publicdata.eu
[6] http://datahub.io

Metadata that describe the dataset should be also published along with the actual data. The produced metadata are usually shared across multiple platforms and implementations. So, users need to be able to import or export metadata related to data cubes.

4.5 Discover & explore cubes

At this step, the users that aim to consume data from data cubes exploit the mechanisms set up at the previous step in order to discover the appropriate data cubes for a task at hand. For example, we consider a researcher that needs to study the relation between unemployment and criminality. So, the researcher needs to analyze data that describe unemployment and criminality in different geographic areas or time periods.

The discovery of specific RDF data cubes will be done based on:

- The measured variable of the cube. This can be denoted through either the description of the measure or generic description of the cube.
- The dimensions of the cube. This can be also denoted through either the description of the measure or generic description of the cube.
- The attributes of the cube.

In general, the discovery of RDF data cubes could be done through:

- A data catalogue that allows exploring the available data cubes based on:
    - Generic metadata records stored inside the catalogue platform that describe the cube as a whole.
    - Cube specific metadata that provide information about the concepts that formulate the cube.
- Full-text search that enables discovery of data cubes not only by metadata but also by the actual content of the cubes.

At an example we suppose that the researcher identifies two cubes:

- A cube presenting the number of unemployed people in three dimensions, i.e. countries, years and age groups.
- A cube presenting crime incidents in two dimensions, i.e. countries and time (quarters of the year).

At this stage we consider that the user is also able to **browse** the data in order to better understand the data cube and proceed with the following steps. In particular, it enables users to view data based on different dimensions or measures. For example, if the data describes unemployment rate at different European countries in different years then users could view either unemployment rate of a particular country throughout the years or unemployment rate of a specific year across different

[7] http://lod-cloud.net

countries. This would enable users also to sort or to filter the data based on the values of the dimensions or the actual values of the observations.

4.6 Transform cube

After the discovery of the RDF data cubes the user should decide whether or not to keep the whole content of the cube for the task at hand. So, the user could select only a part of the data cube at this stage.

Commonly the selection of subsets of RDF is performed using the SPARQL query language[8] where a query specifies part of the RDF graph. However, data cubes insert more complexity at this step. In particular, selection of a data cube adopting OLAP-style terminology could be one of the following.

- *Slice* enables creating a new sub cube with one fewer dimension by choosing a single value for one of the dimensions of the initial cube. For example, if a cube describes unemployment rate at different countries in different years and for different age groups then by selecting only a specific age group e.g. (18-25 years old) a new sub cube is defined with two dimensions (countries and years). The RDF Data Cube vocabulary enables the definition of slices through the *qb:Slice* concept. In particular, slice in RDF Data Cube vocabulary is defined as a selection that "fixes all but one (or a small subset) of the dimensions".
- *Dice* allows picking specific values of multiple dimensions. In the unemployment example, we could perform the following dice operation {dice for (country="Greece" or "Ireland") and (year="2011" or "2012") and (age group="18-25" or "25-32")} in order to define a new sub cube with a limited number of countries, years and age groups. The RDF data cube vocabulary includes the *qb:ObservationGroup* for these situations. This concept can contain an arbitrary collection of observations. A *qb:Slice* is a special case of a *qb:ObservationGroup*.

Data selection from multiple and distributed data cubes in order to select specific parts of the graph. In this case, it is very important that data cubes have been created by exploiting widely accepted controlled vocabularies as described in the second step of the lifecycle. This will enable the performance of queries across multiple data cubes, e.g. through using federated SPARQL queries.

Following the example of the previous stage the researcher decides to select:

- A slice of the first cube by fixing year dimension in 2010. So, after this step the Cube contains data about unemployed people in all European countries for 2010 and for all age groups.
- A dice of the second cube by picking the quarters of 2010. So, after this step the Cube contains data about crime incidents in European countries for the quarters of 2010.

We should finally note that we differentiate between Cube Selection as a step of the OpenCube lifecycle and Cube Selection as a user requirement. The former refers only to data processing that provides as output part of the initial data while the latter refers to a "conceptual" processing of the

[8] http://www.w3.org/TR/sparql11-query

Cube that might need data transformation. For example, *conceptually* a user might select part of a cube by removing one of the dimensions. However, this selection cannot be performed without transforming the data.

At this step the actual values of the observations and thus the whole data cube are transformed. This enables users to perform a number of more advanced operations on top of the RDF data cubes. In particular, the transformations of the data include:

- Aggregating values across a dimension. For example, aggregate the unemployed people in different age groups in order to identify the total number of unemployed people.
- Averaging values across a dimension. For example, calculate the average unemployment rate within the last 10 years.
- Normalizing values across a dimension. For example, this could also enable the normalization of the unemployment rates of different years that refer to the same country based on the general unemployment rate in Europe.

The processing of RDF data cubes at this step will enable users to perform the following actions:

- *Dimension reduction:* This would enable users to select part of a data cube by removing one of the dimensions. In the unemployment rate example this would enable to remove the age group dimension and thus keep only the years and countries dimensions.
- *Roll-up* operation: This OLAP-style operation performs aggregation on a data cube either by climbing up a concept hierarchy for a dimension or by dimension reduction. In our example, this OLAP operation would enable removing the country dimension and thus summarizing unemployment rate in Europe throughout the years. Again in this case the RDF data cube vocabulary does not include concepts or properties to enable explicitly defining this type of relation between datasets. Hence, concept schemes and hierarchies should be used towards this end.
- *Drill Down:* This is an OLAP-style operation that allows the user to navigate among levels of data by stepping down a concept hierarchy from less detailed data to highly detailed data. Following the previous example, stepping down a concept hierarchy for the dimension time could perform this OLAP operation. If we consider the concept hierarchy “month<quarter<year” then drill down would present unemployment rate of different age groups at different countries for any quarter. This operation is not possible to be performed at this step because it requires the use of existing cubes.

Following the example of the previous step the researcher selects to transform the two cubes in the following ways:

- The age groups dimension of the first cube is now removed. So, the cube after this step presents the number of unemployed people in 2010 across all European countries.
- The time dimension of the first cube was rolled-up so that the quarters become year. So, the cube after this step presents the number of crime incidents in 2010 across all European countries.

4.7 Analyze cube

In this step the data cubes that were resulted from the previous step are employed in order to compute simple summaries of the data. The user could select to produce either quantitative (i.e. summary statistics) or visual (i.e. simple to understand graphs) summaries.

As regards the quantitative summaries, a user at this step will be able to describe the observations across a dimension using statistics such as mean, median, standard deviation, variance, coefficient of variation, etc. For example, this step would enable the calculation of the mean and standard deviation of the unemployment rate of European countries in a particular year. Moreover, users could calculate statistics (e.g. Pearson's correlation coefficient) that estimate dependence between paired measures described in disparate data cubes. Paired here is used to denote that the measures share at least one common dimension and thus can be compared.

Finally, the types of visualization charts that can be used in this step include scatter plots, bar charts, pie charts, histograms, geo charts, timelines, etc.

Following the example of the previous steps, the researchers use the cubes created after the last step in order to perform the following:

- Create a scatter-plot presenting unemployed people against crime incidents across European countries.
- Calculate Pearson's correlation coefficient between number of unemployed and number of crimes.

At this step the cubes that were created at the previous steps could be also used in machine learning and predictive analytics in order to produce learning or predictive models. Machine learning is the process of learning from data in order to perform reasoning on large datasets, map specific inputs to desired outputs and predict future outputs based on specific inputs. In general, in machine learning two main actions take place: learning model creation and learning model evaluation.

In the context of quantitative empirical modeling, the term predictive analytics refers to the building and assessment of a model aimed at making empirical predictions using data and statistical or data mining methods. In general, the goal of predictive models is to predict the output value (Y) for new observations given their input values (X). The inputs are often called the predictors, and more classically the independent variables while the outputs are called the response, or classically the dependent variables.

At this step even the models that were created could also be published into the Linked Data Web and thus feedback the lifecycle at the first step. Publishing descriptions of statistical models on the Web following the Linked Data principles could have the following benefits[9]:

- Discovery of variables that an empirical model has suggested a predictive relationship between them. For example, it will be possible to discover that X number of models show a predictive

[9] http://www.purl.org/limo-ontology/limo

relationship between product sales and advertising budget while Z number of models show a negative or no relationship between them.

- Discovery of all predictor variables that are connected to product sales through successful empirical predictive models.
- Discovery of statistical or data mining methods that have been used to identify relationships between variables. For example, most of the models that are able to accurately predict product sales from advertising budget have used linear regression methods.
- Discovery of datasets that have been used to identify predictive relationships between variables. For example, models that show a strong predictive relationship between product sales and advertising budget have employed data from the U.S. in the period between 1975 and 2004.
- Discovery of a specific predictive model that shows a relationship between variables based on aspects such as its creator, the affiliation of the creator, the journal that the results have been published in, etc.
- Discovery of new datasets in order to reuse existing models. For example, identification of datasets in Europe from the last ten years in order to reuse a predictive model produced with data from the U.S.

Following the example of unemployment and criminality, we consider that the researcher now wants to create a model in order to be able to estimate future crime rates based on unemployment. Towards this end, the researcher exploits the results of the previous step and the data cubes in order to select an appropriate data mining method (e.g. Support Vector Machines) and build a model. The researcher goes back to previous steps in order to also identify data to evaluate the model.

4.8 Communicate results

This step involves the presentation of the results of the previous steps to the end users in an easy to consume way. It mainly regards non-expert users that require using data to answer specific questions regarding their domain of interest. The “communicate results” step includes the development of visualizations but also of customized data products to specific audiences.

This step may also feed back to the first step of the lifecycle as the results of the analyses that were performed in the previous steps indicate that there is a need for further analyses that require additional data. Towards this end, the analysis proceeds with the first step of the lifecycle in order to discover new raw data, transform them to RDF and eventually perform a comparative analysis with existing RDF data cubes.

5 Refined OpenCube Requirements

In this section the final list of the OpenCube requirements is presented based on the steps of the OpenCube lifecycle that was presented in the previous section. In addition, the refined requirements are presented in comparison to the initial list of requirements that was included in D1.1.

The sub-sections below correspond to the steps of the OpenCube lifecycle. In each step the initial requirements are presented along with the refined requirements that emerged during the refinement process.

5.1 Discover & pre-process raw data

5.1.1 Initial OpenCube requirements

Table 1 The initial requirements regarding "Discover & pre-process raw data" step

Number	Initial Requirement	Priority
1.	Import data from file	High
2.	Import data from OLAP	High
3.	Enable data filtering	High
4.	Enable data transformation (e.g. to handle missing values or to modify the format of fields when importing).	High
5.	Enable data cleaning	Medium
6.	Import data from HTML (e.g. tables in web pages)	Low
7.	Import data from databases	Low

5.1.2 Refined OpenCube requirements

Table 2 The refined requirements regarding "Discover & pre-process raw data" step

Number	Refined Requirement	Priority
1.1	Import data from file. Data importing allows the access of raw data from files in various electronic formats. These include spreadsheet (XLS, CSV) and PC-Axis files.	High
1.2	Import data from different data sources. A raw data source may be a local file or a web file that can be accessed by URL.	High
1.3	View raw data either in table or tree.	High
1.4	Enable data sorting. The user can sort imported data with a variety of sorting services including alphabetical sorting, numerical sorting and	High

	sorting by date.	
1.5	Enable data filtering. Data filtering services include removing entire rows or columns from the initial data. For example, user may want to discard the header rows of a csv file or discard a number of columns with useless data.	High
1.6	Enable data cleansing. Data cleansing services include identifying and fixing errors including spelling mistakes and removing duplicates or blanks.	High
1.7	Enable data transformation. The transformation regards the creation of new data out of existing data. Data transformation services include the creation of data common formats such as time or geographical data formats, the handling of missing values and the character encoding conversion.	High
1.8	Enable data aggregation from multiple files or sources. This is useful, for example, in cases that information cannot be included in one CSV file and is hence divided in multiple files.	High
1.9	Enable syntax quality assessment. This allows the syntactical validation of raw data files for selected formats of raw data. Examples of syntax quality assessment include CSV parsing.	High
1.10	Enable discovery of data files	High

5.2 Define structure & create cube

5.2.1 Initial OpenCube requirements

Table 3 The initial requirements regarding “Define structure & create cube” step

Number	Initial Requirement	Priority
1.	Use of RDF Data Cube vocabulary	High
2.	Define RDF Structure	High
3.	Discover required code lists and concept schemes	High
4.	Import code lists and concept schemes in the ontology definition	High
5.	Reuse existing vocabularies	High
6.	Update data with reconciled data	High

7.	Reconcile against concept schemes and code lists	High
8.	Design URIs	High
9.	Define mapping	High
10.	Create metadata to enable datasets categorization	High
11.	Download data	High
12.	Create SDMX-ML from data cube	High
13.	Create code lists and concept schemes	Medium
14.	Store and share code lists and concept schemes	Medium
15.	Search for existing vocabularies	Medium
16.	Reconcile against popular datasets such as Freebase, Wordnet or DBpedia	Medium
17.	Import data into an RDF store	Medium
18.	Edit the produced RDF data	Medium
19.	Validate the structure of the data based on the constraints of the vocabulary and show the mistakes along with possible solutions	Medium
20.	Validate the actual data and show the mistakes along with possible solutions	Low
21.	Faceted view of the produced data	Low

5.2.2 Refined OpenCube requirements

Table 4 The refined requirements regarding “Define structure & create cube” step

Number	Refined Requirement	Priority
2.1	Make use of RDF Data Cube vocabulary. The RDF Data Cube vocabulary is a W3C standard used as the main framework to model multi-dimensional data as RDF graphs.	High
2.2	Define and manage the structure of the cube. The definition of the structure of a data cube according to the RDF Data Cube vocabulary requires the selection and description of the dimensions (which describe what an observation applies to) and measures (that refer to	High

	what has been measured).	
2.3	Discover controlled vocabularies (i.e. code lists, concept schemes, taxonomies, etc.). For example, a widely accepted hierarchical code list can be used to encode the values of statistical datasets and facilitates the easy identification of the overall code list.	High
2.4	Reuse controlled vocabularies. This can be also considered as linking to or reconciling against such collections.	High
2.5	Create controlled vocabularies. The creation of controlled vocabularies is required when existing controlled vocabularies do not satisfy the needs of the user. The new controlled vocabularies can be created using popular datasets such as Dbpedia and Geonames.	High
2.6	Store and share controlled vocabularies. The created controlled vocabularies can be stored and shared in order to allow their re-use at a Web scale.	High
2.7	Design URIs. This refers to the creation of the proper reference URIs that will be used in the creation of the RDF cube.	High
2.8	Define mapping from raw data to cubes. This facilitates the mapping between the raw data models (in table or tree view) and the RDF Data Cube model.	High
2.9	Store the mapping of data. The mapping of data is required for future re-use.	High
2.10	Apply the mapping to the raw data to create the RDF data cube.	High
2.11	Validate data for compliance. The validation of data can be done with schema constraints and values constraints.	High

5.3 Annotate cube

5.3.1 Initial OpenCube requirements

Table 5 Initial requirements regarding “Annotate cube” step

Number	Initial Requirement	Priority
1.	Create metadata to enable datasets categorization	High
2.	Create metadata regarding provenance	Medium

5.3.2 Refined OpenCube requirements

Table 6 Refined requirements regarding "Annotate cube" step

Number	Refined Requirement	Priority
3.1	Create metadata regarding provenance. These metadata hence refer to provenance of the raw and RDF data cube files.	High
3.2	Create metadata regarding the process of raw data production. For example, these metadata can include information describing the cases used in the raw data.	High
3.3	Create generic metadata, which are similar to metadata described by DCAT. Metadata can be enhanced with domain-dependent concept schemes to support custom annotations of datasets.	High
3.4	Create metadata about the content of the cube for example using the standard thesaurus of statistical concepts that statistical agencies employ.	High
3.5	Quality assessment of metadata. The validation of the created metadata is significant for the production of high quality metadata	High

5.4 Publish Cube

5.4.1 Initial OpenCube requirements

Table 7 Initial requirements regarding "Publish cube" step

Number	Initial Requirement	Priority
1.	Publicize data to CKAN	Medium

5.4.2 Refined OpenCube requirements

Table 8 Refined requirements regarding "Publish cube" step

Number	Refined Requirement	Priority
4.1	Publish RDF data cubes using Linked Data API, SPARQL endpoint and dump files.	High
4.2	Publicize RDF data cubes. This includes the publicizing of datasets in data catalogues such as the public data portal, data.gov.uk, data.gov.gr, data hub platform, and the Linked Open Data cloud.	High
4.3	Publish metadata along with cubes. The publishing of data cubes is	High

	accompanied with the publishing of the relevant metadata so as to facilitate the discovery of the data cubes.	

5.5 Discover & explore cubes

5.5.1 Initial OpenCube requirements

Table 9 Initial requirements regarding "Discover & explore cubes" step

Number	Initial Requirement	Priority
1.	Search for datasets based on their metadata.	High
2.	Receive recommendations for related or comparable datasets preferably from within CKAN.	High
3.	Browse datasets.	High
4.	Configurable display of datasets as tables.	High
5.	Receive recommendations for interesting related data.	Medium

5.5.2 Refined OpenCube requirements

Table 10 Refined requirements regarding "Discover & explore cubes" step

Number	Refined Requirement	Priority
5.1	Discover cubes based on the measured variable.	High
5.2	Discover cubes based on the dimensions of the cubes.	High
5.3	Discover cubes based on the attributes of the cube.	High
5.4	Discover cubes based on provenance information (e.g. publisher, date, etc.)	High
5.5	Browse and pivot data. The user is able to browse the data from the discovered cubes in order to select the one(s) of his interest. The user is also allowed to pivot data so as to rotate data to axes and provide an alternative presentation of them.	High
5.6	Receive recommendations. Recommendations can be used to inform the user about relevant or comparable datasets or for interesting related data.	High

5.6 Transform cube

5.6.1 Initial OpenCube requirements

Table 11 Initial requirements regarding "Transform cube" step

Number	Initial Requirement	Priority
1.	Browse data.	High
2.	Filter datasets to specific subsets (e.g. only specific years and geographies).	High
3.	Select external data sources.	High
4.	Establish links to other datasets	High
5.	Perform OLAP operations.	Medium

5.6.2 Refined OpenCube requirements

Table 12 Refined requirements regarding "Transform cube" step

Number	Refined Requirement	Priority
6.1	Select a slice of the data cube. A slice enables the creation of a new sub cube with one fewer dimension by choosing a single value for one of the dimensions of the initial cube.	High
6.2	Select a dice of the data cube. A dice allows the selection of specific values of multiple dimensions.	High
6.3	Select combined data from multiple and distributed cubes. In this case it is important that the cubes have been created by exploiting widely available controlled vocabularies	High
6.4	Enable dimension reduction operation. Dimension reduction enables users to select part of a data cube by removing one of its dimensions and keeps the rest of the data cube.	High
6.5	Enable roll-up operation. Roll-up is an OLAP-style operation that allows the aggregation on a data cube either by climbing up a concept hierarchy for a dimension or by dimension reduction.	High
6.6	Enable drill-down operation. Drill-down is also an OLAP-style operation that allows the user to navigate among different levels of data by stepping down a concept hierarchy from less detailed data to highly detailed data.	High

5.7 Analyze cube

5.7.1 Initial OpenCube requirements

Table 13 Initial requirements regarding "Analyze cube" step

Number	Initial Requirement	Priority
1.	Compute descriptive statistics	High
2.	Perform basic mathematical and statistical operations on data.	High
3.	Select appropriate visualization chart format.	High
4.	Visualize data.	High
5.	Create reports with simple tables and charts aimed at the general public (e.g. one per area/industry/occupation)	Medium
6.	Create forecasting models.	High
7.	Evaluate forecasting models with new data.	Medium
8.	Store information to describe the model (e.g. in PMML).	Low

5.7.2 Refined OpenCube requirements

Table 14 Refined requirements regarding "Analyze cube" step

Number	Refined Requirement	Priority
7.1	Compute quantitative summaries of the data. The user is able to describe the observations across a dimension using statistics such as mean, median, standard deviation, variance, coefficient of variation etc.	High
7.2	Estimate dependency between measures. This refers to the calculation of statistics (e.g. Pearson's correlation coefficient) that estimate the dependency between paired measures (aka measures that measures share at least one common dimension) described in disparate data cubes in order to compare them.	High
7.3	Create learning models. In machine learning, learning models are the product of the process of learning from data in order to perform reasoning on large datasets, map specific inputs to desired outputs	Medium

	and predict future outputs based on specific inputs.	
7.4	Create predictive models. In predictive analytics, the goal of predictive models is to predict the output value (Y) for new observations given their input values (X).	Medium
7.5	Evaluate models. The produced models are evaluated in order to understand and assess their performance.	Medium
7.6	Publish models into the Linked Data Web. The publishing of learning models facilitates the discovery of predictor variables of the model, the discovery of statistical or data mining methods that have been used, etc.	Low
7.7	Store models. The information described by the learning model can be stored in PMML.	Low

5.8 Communicate results

5.8.1 Refined OpenCube requirements

Table 15 Refined requirements regarding "Communicate Results" step

Number	Refined Requirement	Priority
8.1	Visualize data. Data can be visualized using different types of visualization charts scatter plots, bar charts, pie charts, histograms, geo charts, timelines, etc.	High
8.2	Configure data visualizations. The user is able to configure the type of the visualization of the data, the way of its presentation, etc.	High
8.3	Create customized data products	High

5.9 Initial OpenCube Requirements that cannot be mapped

The table below presents the initial requirements that were not mapped to the new lifecycle and were not mapped to the refined requirements. The majority of these requirements are related to the linking activity. In the OpenCube lifecycle we consider that the linking activity is part of the "Transform Cube" that involves merging disparate data cubes.

Table 16 Initial requirements that did not included in the refined list

Number	Initial Requirement	Priority

1.	Select datasets to link the data.	High
2.	Define the entities to be linked.	High
3.	Define the links to be established.	High
4.	Define properties to be checked.	High
5.	Store the links.	High
6.	Select similarity metrics.	Medium

5.10 OpenCube non-Functional Requirements

Table 17 List of OpenCube non-functional requirements

No	Requirement	Priority
1.	Software components should be designed to allow automation	High
2.	Easy-to-use and intuitive UI based on different target users.	High
3.	Easy access to a wide range of different sources of data.	High
4.	Software components should be designed or selected to allow integration into an overall platform.	High
5.	Where possible, interfaces of software components should be designed or selected to allow integration into software in a range of programming languages.	High
6.	Analysis of large volumes of data in acceptable time.	High
7.	Low maintenance of the tools.	Medium
8.	Minimal training or need for new skills to use the tools.	Medium
9.	Most web pages generated by the software should be usable at a range of screen sizes, down to a minimum width of pixels.	Medium
10.	Contribute to the professional image of the organization.	Low
11.	Data can be published at a precisely controlled time.	Low

6 Gaps Analysis: Requirements and Existing Tools

In this section we map the identified requirements to existing tools that work with linked data in order to identify gaps. We employ the results of deliverable D1.1 where 35 tools that process linked data were identified and analysed. In the gap analysis we use 11 out of the 35 tools that are more related to multi-dimensional statistical data. The tools that we have included are the following:

1. OpenRefine[10]
2. PoolParty[11]
3. CSVImport (LOD2 project)[12]
4. TabLinker[13]
5. Computex[14]
6. SILK[15]
7. Pubby[16]
8. CubeViz (LOD2 project)[17]
9. SPARQL R[18]
10. RapidMiner (LOD)[19]

Tables 17 and 18 present the mapping between these tools (horizontal axes) and the requirements (vertical axes). In particular, Table 17 considers the requirements that are included in the publishing related steps and Table 18 in the reusing related steps. The cells that are coloured in dark green denote that the requirement is fully addressed by the tool while the cells with the light green denote that the requirement is partially addressed by the tool.

Table 18 Mapping existing tools (horizontal) to Publishing related requirements (vertical)

	1	2	3	4	5	6	7	8	9	10
R1.1	▨		▨	▨						
R1.2	🟩		🟩							
R1.3	🟩									
R1.4	🟩									
R1.5	🟩									
R1.6	🟩									
R1.7	🟩									
R1.8										

[10] http://openrefine.org
[11] http://www.poolparty.biz
[12] https://github.com/AKSW/csvimport.ontowiki
[13] https://github.com/Data2Semantics/TabLinker
[14] http://computex.herokuapp.com
[15] http://wifo5-03.informatik.uni-mannheim.de/bizer/silk/
[16] http://wifo5-03.informatik.uni-mannheim.de/pubby/
[17] http://cubeviz.aksw.org
[18] http://cran.r-project.org/web/packages/SPARQL/
[19] http://dws.informatik.uni-mannheim.de/en/research/rapidminer-lod-extension/

R1.9										
R1.10										
R2.1			🟩	🟩						
R2.2			🟩							
R2.3		🟩								
R2.4						🟩				
R2.5		🟩								
R2.6		🟩								
R2.7			🟩	🟩						
R2.8			▨	▨						
R2.9			🟩							
R2.10					🟩					
R3.1										
R3.2										
R3.3										
R3.4										
R3.5										
R4.1							▨			
R4.2			▨							
R4.3										

Table 19 Mapping existing tools (horizontal) to Reusing related requirements (vertical)

	1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**
R5.1										
R5.2										
R5.3										
R5.4										
R5.5										
R5.6										
R6.1								🟩		
R6.2										
R6.3									🟩	🟩
R6.4										
R6.5										
R6.6										
R7.1									🟩	🟩
R7.2									🟩	🟩
R7.3									🟩	🟩
R7.4									🟩	🟩
R7.5									🟩	🟩
R7.6										
R7.7									🟩	🟩
R8.1								🟩	🟩	🟩
R8.2								🟩	🟩	🟩

The tables indicate that requirements that are related to (a) Annotate cube, (b) Discover & explore cube, and (c) Transform cube are not addressed by the existing tools. These steps provide opportunities for further development in order to fully support the OpenCube lifecycle and thus enable end users to publish and reuse linked data cubes.

We should also note that this gap analysis does not take into account other characteristics of the tools such as licences, easiness of use, dependencies with other software or platforms, etc. This type of analysis takes place in WP2. So, the results of this deliverable should be read in conjunction with deliverables D2.1 and D2.2 in order to enable better understanding of the existing needs regarding linked data cubes publishing and reuse. Even if a tool has been assessed to meet the requirements in isolation, we also need to consider how easy it would be to integrate the tool into a platform, and whether the tool would fit into a coherent platform, with consistent workflows and user interfaces.

7 Conclusion

The purpose of this deliverable is to elicit and present the final list of user requirements that will be addressed by the OpenCube project. Information included in this deliverable is valuable to the consortium for specifying the OpenCube architecture and tools.

The deliverable includes the following results:

- The OpenCube lifecycle that describes the steps that raw statistical multi-dimensional data should go through in order to create value. The lifecycle comprises 8 steps: (i) Discover & pre-process raw data, (ii) Define structure & create cube, (iii) Annotate cube, (iv) Publish cube, (v) Discover & explore cube (vi) Transform cube, (vii) Analyze cube, (viii) Communicate results.
- The refined list of user requirements categorized according to the steps of the lifecycle. The final list includes 49 functional and 10 non-functional requirements.
- Gap analysis between existing tools and the identified functional requirements. The analysis indicates that (a) Annotate cube, (b) Discover & explore cube, and (c) Transform cube have the biggest gaps. Even where existing tools meet the functional requirements, further evaluation is required, in conjunction with the System Architecture work package, to assess whether an existing tool meets our non-functional requirements and is suitable for integration in the platform.

References

Auer, S., Lehmann, J., Ngomo, A-C. N., Zaveri, A. (2013) Introduction to Linked Data and Its Lifecycle on the Web, Lecture Notes in Computer Science, Vol. 8067, pp 1-90.

Bizer, C., Cyganiak, R., & Heath, T. (2007). How to publish linked data on the web. Retrieved January, 3, 2014.

Etcheverry L. and Vaisman A. A. (2012). Enhancing OLAP Analysis with Web Cubes. In ESWC 2012, pages 53–62.

Hyland, B., Wood, D. (2011) The Joy of Data - A Cookbook for Publishing Linked Government Data on the Web. in Linking Government Data. Springer, pp 3-26.

Villazón-Terrazas, B., Vilches-Blázquez, L. M., Corcho O., Gómez-Pére A. (2011) Methodological Guidelines for Publishing Government Linked Data. in Linking Government Data. Springer, pp 27-. 49