Abstract: This deliverable introduces the RDF Data Cube Validation tool for assessing the quality of statistical data represented using the RDF Data Cube vocabulary, which will provide quality assessment services for ESTA-LD. In addition to providing information on how to set up and use the validation tool, the deliverable will also introduce the first prototype of the ESTA-LD component, the GUI that allows online access and analysis of spatio-temporal data.

The information in this document reflects only the author's views and the European Community is not liable for any use that may be made of the information contained therein. The information in this document is provided "as is" without guarantee or warranty of any kind, express or implied, including but not limited to the fitness of the information for a particular purpose. The user thereof uses the information at his/ her sole risk and liability.
History

<table>
<thead>
<tr>
<th>Version</th>
<th>Date</th>
<th>Reason</th>
<th>Revised by</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.0</td>
<td>07/11/2014</td>
<td>First version and section outline</td>
<td>Vuk Mijović</td>
</tr>
<tr>
<td>0.1</td>
<td>14/11/2014</td>
<td>First draft with Validation tool description</td>
<td>Vuk Mijović</td>
</tr>
<tr>
<td>0.2</td>
<td>18/11/2014</td>
<td>Added description of ESTA-LD</td>
<td>Valentina Janev</td>
</tr>
<tr>
<td>0.3</td>
<td>20/11/2014</td>
<td>Technical description of Validation tool</td>
<td>Vuk Mijović</td>
</tr>
<tr>
<td>0.4</td>
<td>21/11/2014</td>
<td>Technical description of ESTA-LD</td>
<td>Dejan Paunović</td>
</tr>
<tr>
<td>0.5</td>
<td>24/11/2014</td>
<td>Interim version prepared for peer review</td>
<td>Vuk Mijović</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Valentina Janev</td>
</tr>
<tr>
<td>0.6</td>
<td>28/11/2014</td>
<td>Improvements based on peer review</td>
<td>Vuk Mijović</td>
</tr>
<tr>
<td>1.0</td>
<td>28/11/2014</td>
<td>Final version</td>
<td>Vuk Mijović</td>
</tr>
</tbody>
</table>

Author List

<table>
<thead>
<tr>
<th>Organisation</th>
<th>Name</th>
<th>Contact Information</th>
</tr>
</thead>
<tbody>
<tr>
<td>IMP</td>
<td>Vuk Mijović</td>
<td><a href="mailto:vuk.mijovic@pupin.rs">vuk.mijovic@pupin.rs</a></td>
</tr>
<tr>
<td>IMP</td>
<td>Valentina Janev</td>
<td><a href="mailto:valentina.janev@pupin.rs">valentina.janev@pupin.rs</a></td>
</tr>
<tr>
<td>IMP</td>
<td>Dejan Paunović</td>
<td><a href="mailto:Dejan.paunovic@pupin.rs">Dejan.paunovic@pupin.rs</a></td>
</tr>
</tbody>
</table>
Executive Summary

The overall objective of task T4.6 is to demonstrate the applicability of the GeoKnow Generator for advanced spatio-temporal analysis of data and statistics collected by two relevant institutions – the National Statistical Office and Business Registers Agency of Serbia. Since statistical data will be represented using RDF Data Cube vocabulary, it will be necessary to ensure that the data abides to the principles of the vocabulary. These principles are formalized as a set of integrity constraints which are validated by executing SPARQL ASK queries. The quality service (RDF Data Cube Validation Tool) described in this deliverable validates against these constraints, provides information about resources which caused the violation and in some cases also suggest a way to fix the dataset. The RDF Data Cube Validation Tool has been used to assess the quality of the Linked Data collected from Serbian government institutions for the GeoKnow project. Also included in this deliverable is an overview of the first prototype of the ESTA-LD component aimed for business users interested in spatial analysis of different socio-economic indicators. The component demonstrates the use of Linked Data technologies for interactive analysis, visualization and drill-down style online access to highly granular spatio-temporal data.
### Abbreviations and Acronyms

<table>
<thead>
<tr>
<th>Abbreviation</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>LOD</td>
<td>Linked Open Data</td>
</tr>
<tr>
<td>QB</td>
<td>RDF Data Cube vocabulary</td>
</tr>
<tr>
<td>IC</td>
<td>Integrity Constraint</td>
</tr>
<tr>
<td>GUI</td>
<td>Graphical User Interface</td>
</tr>
<tr>
<td>SPARQL</td>
<td>SPARQL Protocol and RDF Query Language</td>
</tr>
<tr>
<td>SBRA</td>
<td>Serbian Business Register Agency</td>
</tr>
<tr>
<td>SORS</td>
<td>Statistical Office of the Republic of Serbia</td>
</tr>
<tr>
<td>DSD</td>
<td>Data Structure Definition</td>
</tr>
</tbody>
</table>
Table of Contents

1. INTRODUCTION..........................................................................................................................7

2. EXPLORATORY SPATIO-TEMPORAL ANALYSIS OF LINKED DATA (ESTA-LD) ........8
   2.1 USE CASE ANALYSIS..................................................................................................................8
   2.2 CHARACTERISTICS OF SPATIO-TEMPORAL DATASETS .............................................................9
   2.3 REQUIREMENTS FOR THE ESTA-LD COMPONENT ..................................................................10
   2.4 ESTA-LD SCENARIOS ...............................................................................................................11

3. RDF DATA CUBE VOCABULARY - MODELLING STATISTICAL DATA .........................13
   3.1 MODELING HIERARCHICAL DATA WITH SKOS .......................................................................14

4. RDF DATA CUBE VALIDATION TOOL .....................................................................................16
   4.1 QB INTEGRITY CONSTRAINT CHECKING ...................................................................................17
   4.2 IMPLEMENTATION ......................................................................................................................19
   4.3 INSTALLATION AND DEPLOYMENT ...........................................................................................20
   4.4 CONFIGURATION ......................................................................................................................20
   4.5 USER GUIDE .............................................................................................................................21
   4.6 EXAMPLE SCENARIO ...............................................................................................................23

5. ESTA-LD GUI (FIRST PROTOTYPE) .........................................................................................25
   5.1 ESTA-LD ARCHITECTURE ..........................................................................................................25
   5.2 IMPLEMENTATION OF THE FIRST PROTOTYPE .......................................................................26
   5.3 USING ESTA-LD ......................................................................................................................26

6. CONCLUSION AND FUTURE WORK ......................................................................................28

7. REFERENCES ...............................................................................................................................29
List of Figures

Figure 1: The Spatio-Temporal Analysis Use case ..............................................................8
Figure 2: Geographical map of Serbia displayed on different granularity levels ..............10
Figure 3: Guided analysis of spatio-temporal data ..............................................................10
Figure 4: Typical scenario ..................................................................................................12
Figure 5: Integration with other components .....................................................................19
Figure 6: RDF Data Cube Validation Tool - Configuration Window ..............................21
Figure 7: RDF Data Cube Validation Tool - User Interface .............................................22
Figure 8: RDF Data Cube Validation Tool - Failed execution of a fix ..............................23
Figure 9: RDF Data Cube Validation Tool - Configure the endpoint and graph ............24
Figure 10: RDF Data Cube Validation Tool - Apply quick fix ........................................24
Figure 11: ESTA-LD Architecture ....................................................................................25
Figure 12: ESTA-LD GUI (run chart) ...............................................................................26
Figure 13: ESTA-LD GUI (run chart) ...............................................................................27

List of Tables

Table 1: Basic characteristics of spatio-temporal data published by two Serbian government agencies .................................................................9
Table 2: Integrity constraint explanations and solutions ..................................................19
Table 3: Integrity constraint icons/indicators ................................................................22
1. Introduction

The objectives of the WP4 are to develop (1) new exploration, (2) visualisation and (3) authoring & curation interfaces for the GeoKnow Generator, which integrate heterogeneous information from various sources and support the evolution of both instance data as well as information structures over time.

The main goal of Task 4.6 is to demonstrate the applicability of the GeoKnow Generator for advanced spatiotemporal analysis of business and public data that appear at different levels of granularity, thus, imposing challenges for data processing, presentation and exploration. This goal will be achieved by

- Developing a specialized component called ESTA-LD (Exploratory Spatio-Temporal Analysis for Linked Data) [1, 2].
- Developing the required quality assessment services that will be used in the preprocessing step in order to ensure that the input data is properly modelled prior to using ESTA-LD.
- Evaluating the ESTA-LD functionalities with data and statistics collected by two relevant Serbian government institutions - the National Statistical Office and the Business Registers Agency. The LOD is currently available via the Serbian CKAN[1], while the original data is publicly available at the Statistical office databases[2] and the Register of Regional Measures and Incentives[3].

Nowadays, the available Linked Data is modelled using various W3C standard vocabularies such as the RDF Data Cube vocabulary (QB) and eGovernment Core vocabularies. However, it has to be ensured that the vocabularies are used as it is prescribed. To ensure the quality of statistical data we will use integrity constraints defined by the QB vocabulary while the data captured with other vocabularies will be validated by using RDFUnit, a test-driven data debugging framework that can run automatically generated (based on a schema) and manually generated test cases against an endpoint.

This deliverable will report on the first release of

- the Validation tool for assessing the quality of statistical data represented using the QB vocabulary (see Section 4).
- the first prototype of the ESTA-LD component (see Section 5), the GUI that allows online access and analysis of spatio-temporal data.

The evaluation of the ESTA-LD component will be done with data and statistics collected by two relevant institutions – the National Statistical Office and Business Registers Agency of Serbia and will be reported in Deliverable 4.6.2 due to Month 36.

This deliverable is organized as follows. In the Section 2 we will briefly introduce the business requirements of ESTA-LD and provide insights about data that will be used. Section 3 will showcase the use of RDF Data Cube vocabulary for modelling of statistical data, while Section 4 will describe the validation tool and serve as a user guide. First prototype of ESTA-LD will be introduced in Section 4, while Section 5 will provide conclusions and give insights into future steps.

---

1 http://rs.ckan.net
2. Exploratory Spatio-Temporal Analysis of Linked Data (ESTA-LD)

This section will provide the context of Task 4.6 driven by the needs of Serbian government organizations.

2.1 Use case analysis

In the last five years, the global Open Government Data (OGD) initiatives, such as the Open Government Partnership\(^4\), have helped to open up governmental data for the public, by insisting on opening non-sensitive information, such as core public data on transport, education, infrastructure, health, environment, etc. The vision for ICT-driven public sector innovation \([3]\) refers to the use of technologies for the creation and implementation of new and improved processes, products, services and methods of delivery in the public sector.

In the GeoKnow framework, we study the possibilities for adoption of Linked Data technologies for improving the integration and enhancing the analysis of data coming from different sources (e.g. Register of the Regional Development Measures and Incentives\(^5\), Dissemination database of SORS - Statistical Office of the Republic of Serbia\(^6\)). At government agencies (e.g. SBRA - Serbian Business Registers Agency), instead of the current way of collecting and visualizing data, innovative approaches could be applied (see Figure 1) that:

- extend the existing public service and standardize the data collection process;
- ensure interoperability and improve transparency of data on European level;

---

\(^4\) [http://www.opengovpartnership.org/](http://www.opengovpartnership.org/)


allow advanced spatio-temporal analysis of available indicators;
offer user-friendly services in a secure and flexible manner allowing personalisation for different types of users (public administration, businesses and citizens).

As is shown in Figure 1, the data can be exposed for exploration in different ways, e.g. via a SPARQL endpoint at the publishers site or downloadable dump file registered with a government catalogue (e.g. the Serbian CKAN). In the Task 4.6 framework we are especially interested in possibilities for integrating Linked Open Data for advanced analyses of socio-economic indicators in official government statistical systems as statistical data are often used as foundations for policy prediction, planning and adjustments, having a significant impact on the society (from citizens to businesses to governments).

2.2 Characteristics of spatio-temporal datasets

Many data objects in real world have attributes related to both space and time, thus imposing challenges for visualizing both dimensions on a geographical map, e.g. environmental and meteorological data in real time navigational systems, socio-economic indicators in official government statistical systems, geo-positioning data, etc. Moreover, these data objects are often multi-dimensional in nature meaning that the information can be represented on different granularity levels in space and time, as well as the type of information. Examples of indicators with attributes related to both space and time are given in Table 1.

| Table 1: Basic characteristics of spatio-temporal data published by two Serbian government agencies |
|---------------------------------|---------------------------------|---------------------------------|---------------------------------|
| Publisher                      | Domain                          | 1st dimension: Space Granularity levels | 2nd dimension: Time period, granularity | Other dimensions |
| SBRA                           | Regional development measures and incentives by financial type | Country-region-area-municipality | 2010-2013, yearly | Financial type |
| SORS                           | Demographic statistics, Population projections | Country-region | 2009-2010, yearly | Sex, age |
| SORS                           | Labour market                   | Country-region | 2008-2013, yearly-monthly | Sex, age, occupation, type of settlement |
| SORS                           | Tourism (overnight stay)        | Country-region | 2010-2013, yearly-monthly | Tourist type |

Due to multidimensionality, in order to ensure efficient exploration and analysis, hierarchical data structures are needed for modelling the space and time dimensions (e.g. see Figure 2).
2.3 Requirements for the ESTA-LD component

The selection of GeoKnow tools for analysis of Linked Data depends on the characteristics of data and the goals of analysis. Facete, for instance, provides faceted navigation of the model and visualization of points of interest (described with WGS84 vocabulary) on a geographical map. Additionally, CubeViz can be used for visualization of statistics without possibilities to relate the indicators with geographical entities. ESTA-LD aims at overcoming the limitations of CubeViz and Facete and offering generic interface for exploring spatio-temporal data. The overall vision, how ESTA-LD can be used is presented on Figure 3:

1. The user uploads a graph into the local RDF data store (Virtuoso);
2. The user checks for validation of the integrity constraints. If errors are identified, automatic or semi-automatic repair of the multidimensional model is available;
3. The user can proceed with the exploration using Facete, CubeViz or ESTA-LD.
ESTA-LD answers the GeoKnow requirements:

- **FR-E1** GeoKnow Generator has to support statistical and temporal geospatial information management including advanced analysis of spatio-temporal data.
- **FR-E2** GeoKnow Generator has to integrate quality analysis services.

Taking into consideration the GeoKnow requirements, as well as the recommendation of the EC 'Interoperability Solutions for European Public Administrations' (ISA) program, we have further refined these recommendations as following:

- **FR-E1.1** ESTA-LD has to support data retrieval from arbitrary SPARQL end points
- **FR-E1.2** ESTA-LD has to provide drill down based on 2 dimensions (temporal and space), see also requirement FR-S8 (Visual analysis where I could intuitively select features and analyse them without having to dig into datasets)
- **FR-E1.3** ESTA-LD has to provide user-friendly filtering options for selecting values of interest (related to the requirement FR-S8)
- **FR-E1.4** ESTA-LD has to provide different visualization/analysis options (bar chart, run chart, histogram)
- **FR-E2.1** ESTA-LD has to provide quality assessment of business data and statistics modelled with standard W3C vocabularies (QB vocabulary, core business vocabulary)
- **FR-E2.2** ESTA-LD has to provide support to the user in the quality assessment process (see also requirement FR-U8: GeoKnow Generator has to be able to identify conflicts and provide solutions for solving them).

### 2.4 ESTA-LD scenarios

ESTA-LD will be realised in the third year of the GeoKnow project and it will be evaluated with datasets from two relevant Serbian institutions: the National Statistical Office and Business Registers Agency of Serbia. The statistical data provided by these organizations will be transformed to RDF datasets based primarily on the QB vocabulary, while the information related to the business organizations will be based on eGovernment Core vocabularies.

In order for ESTA-LD to be able to perform the processing and visualize the data, the input data will have to satisfy certain quality criteria related to the vocabularies that are used. QB vocabulary, which allows modelling of statistical data, and is therefore the basis for ESTA-LD, will be introduced in Section 3. Data that is modelled using this vocabulary will be validated with the RDF Data Cube vocabulary (see Section 4), while business data, which will be based on Core vocabularies will be validated by using RDFUnit, a test driven data-debugging framework that can run automatically generated (based on a schema) and manually generated test cases against an endpoint.

Validation and repair of the datasets is an essential building block of practically every scenario for ESTA-LD. Let's take for example the following typical scenario. Data that is originally available in CSV and XML files is transformed to RDF in order to be processed and visualized with ESTA-LD. However, as stated above, ESTA-LD will not be able to process and visualize the data if it is not of good quality, which may be the case due to missing data in the original files, or errors in the transformation process. Therefore, a validation and repair cycle should be performed before ESTA-LD is put to use (see Figure 4).

7 Deliverable 1.4.2 Intermediate release of the GeoKnow Generator
8 http://ec.europa.eu/isa/
9 Since input files are in XML, transformation will be done using XSLT.
10 http://aksw.org/Projects/RDFUnit.html
This scenario shows the importance of validating the data and improving its quality, i.e. validation and repair steps, in order to ensure that ESTA-LD will be able to produce desired outputs (or any other tool that is based on the QB vocabulary). RDF Data Cube Validation tool supports both of these steps. It is explained in more detail in Section 4, while the demonstrator is available at http://jpo2.imp.bg.ac.rs/rdf-data-cube-validation-demo.

Demonstrator of the first prototype of ESTA-LD is available at http://jpo2.imp.bg.ac.rs/ESTA-LD, and the additional information about the component and how to use it is given in Section 5.
3. RDF Data Cube vocabulary - modelling statistical data

In this section we will give an overview of modelling statistical data with the RDF Data Cube vocabulary along with examples taken from datasets that were published in cooperation with the Statistical Office of the Republic of Serbia.

In January 2014, W3C recommended the RDF Data Cube vocabulary [5], a standard vocabulary for modelling statistical data, see http://www.w3.org/TR/vocab-data-cube/. The vocabulary focuses purely on the publication of multi-dimensional data on the Web. It builds upon the core of the SDMX 2.0 Information Model [6]. In 2001, the Statistical Data and Metadata Exchange (SDMX'11) Initiative was organised by seven international organizations (BIS, ECB, Eurostat, IMF, OECD, World Bank and the UN) to release greater efficiencies in statistical practice.

A statistical data set comprises a collection of observations made at some points across some logical space. The collection can be characterized by a set of dimensions that define what the observation applies to (e.g. time, country) along with metadata describing what is measured (e.g. economic activity, prices), how it is measured and how the observations are expressed (e.g. units, multipliers, status). We can think of the statistical data set as a multi-dimensional space, or hyper-cube, indexed by those dimensions. This space is commonly referred to as a cube for short; though the name shouldn’t be taken literally, it is not meant to imply that there are exactly three dimensions (there can be more or fewer) nor that all the dimensions are somehow similar in size. Cubes are modelled with a set of dimensions, attributes and measures, where dimensions serve to identify the observations, measures are used to describe phenomena being observed, while attributes allow to qualify and interpret the observed values. Dimensions, measures, and attributes are component properties and they represent the core of the DSD (Data Structure Definition), which describes structure of the dataset and can be reused. Furthermore, it is possible to group subsets of observations together by creating slices through the cube in which one or more dimension values are fixed. Structure of a slice is given by associating it with a slice key, much like DSDs are used to describe structure of datasets.

An example how to use the vocabulary to represent one single observation is given in the example bellow. The observed socio-economic phenomenon is described using several dimensions modelled with the SKOS vocabulary.

**Example: Modeling socio-economic data with QB**

The observation is a coarse-grained representation of the indicator "Tourists arrivals" for the territory of country Serbia (geo:RS) and for year 2005 (time:Y2005). Additionally the indicator represents the "Total" number of tourists including "Domestic" and "Foreign". One single observation using the QB entity (qb:Observation) is as follows:

```ruby
@prefix rs: <http://elpo.stat.gov.rs/lod2/RS-DIC/rs/>  
@prefix qb: <http://purl.org/linked-data/cube#>
http://elpo.stat.gov.rs/lod2/RS-DATA/Tourism/Tourists_arrivals/data/obs1>  
a qb:Observation ; 
rs:geo geo:RS ; 
rs:time time:Y2005 ; 
rs:dataType datatype:number ; 
rs:obsIndicator "Tourists arrivals" ; 
rs:obsTurists "Total" ; 
sdmx-measure:obsValue "1988469" .
```

The example observation is identified through dimensions $rs$-geo, $rs$-time, $rs$-obsIndicator, and $rs$-obsTourists, while the measured value given through measure property smdx-measure:obsValue is described through attribute property $rs$-dataType.

3.1 Modeling hierarchical data with SKOS

In order to formalize the conceptualization of hierarchical dimensions (space, time), we can use the Simple Knowledge Organization System (SKOS), see http://www.w3.org/TR/2005/WD-swbp-skos-core-spec-20051102/. SKOS Core is a model and an RDF vocabulary for expressing the basic structure and content of concept schemes such as thesauri, classification schemes, subject heading lists, taxonomies, ‘folksonomies’, other types of controlled vocabulary, and also concept schemes embedded in glossaries and terminologies. Concepts represented as skos:Concept are grouped in concept schemes (skos:ConceptScheme) that serve as code lists from which the dataset dimensions draw on their values. Semantic relation used to link a concept to a concept scheme is skos:hasTopConcept. Herein, we will present an example of coding the space and time dimension in RDF.

**Example: Spatial dimension**

```rdfs
geo:RS21
rdf:type geo:Region;
owl:sameAs <http://dbpedia.org/page/%C5%A0umadija_and_Western_Serbia>;
skos:broader geo:RS;
skos:narrower geo:RS218, geo:RS214, geo:RS217;
skos:notation "RS21"^^xsd:string;
skos:prefLabel "Region of Sumadija and Western Serbia"@en, "REGION ŠUMADIJE I ZAPADNE SRBIJE"@sr.
```

SKOS properties skos:broader and skos:narrower can be used for relating concepts of same type, in our case geographical area (geo:Region). However, if the concepts are not of the same type (e.g. to regions and municipalities), the skos:relatedAlignment can be applied.

```rdfs
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
@prefix geo: <http://elpo.stat.gov.rs/lod2/RS-DIC/geo/> .
@prefix skos: <http://www.w3.org/2004/02/skos/core#> .

geo:_70980
rdf:type geo:Municipality;
owl:sameAs <http://dbpedia.org/resource/Prijepolje>;
skos:notation "70980"^^xsd:string;
skos:prefLabel "Prijepolje"@en;
skos:related geo:RS211.
```

**Example: Time dimension**

Observed data can be described with time information using different formats (e.g. seconds, day-time, day, month, year). One way to specify the frequency of data (or time granularity) in a dataset is to use the SDMX CONTENT-ORIENTED GUIDELINES [7].

```rdfs
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
@prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#> .
@prefix skos: <http://www.w3.org/2004/02/skos/core#> .
@prefix time: <http://elpo.stat.gov.rs/lod2/RS-DIC/time#> .

time:P1D
rdf:type owl:Class;
rdfs:subClassOf skos:Concept;
skos:prefLabel "Daily (or Business)"@en.
```
time:P1M
    rdf:type owl:Class ;
    rdfs:subClassOf skos:Concept ;

time:P1Y
    rdf:type owl:Class ;
    rdfs:subClassOf skos:Concept ;
    skos:prefLabel "Annual"@en .

time:Y1980
    rdf:type time:P1Y ;
    skos:notation "Y1980"^^xsd:string ;

time:Y1980M1
    rdf:type time:P1M ;
    skos:broader time:Y1980Q1 ;
    skos:notation "Y1980M1"^^xsd:string ;
4. RDF Data Cube Validation tool

The Validation tool answers the following GeoKnow requirements:

- FR-E2 GeoKnow Generator has to integrate quality analysis services.

Unlike tools such as RDFUnit and CROCUS\(^\text{12}\) which aim at improving the quality of datasets in general (any dataset can be assessed), this tool is focused on the QB vocabulary and is based on integrity constraints defined by the experts\(^\text{13}\). Therefore, in the context of data modelled with QB vocabulary, this tool can precisely detect all errors and even help repair the data in some cases. These constraints are defined as SPARQL ASK queries that provide an answer whether the constraint is violated or not. For the purpose of this tool, these queries were slightly modified. Namely, based on these ASK queries, we defined SELECT queries, which also provide information about resources that violate the constraint, as well as additional information which might be useful to the user or needed in order to identify a solution and apply a fix. After the query is executed, the tool points to the problematic resources, provides additional information, and tries to find a solution.

First developments started during the LOD2 project. At the time it was embedded in the Statistical Workbench [4], and therefore relied on the Workbench for the configuration and connection to the endpoint. In order to be able to more easily integrate it in the GeoKnow Generator and with ESTA-LD, we separated it from the Statistical Workbench and turned it into a standalone web application. Additionally, the tool was enhanced to provide more flexibility, and now a user can choose exactly which integrity constraints will be evaluated and when they are to be re-evaluated. It is also possible to validate a dataset with RDFUnit in order to assess the quality of non-statistical data that is encoded using Core vocabularies.

Following is a current list of features:

1. Configuration: endpoint, graph, basic authentication, OntoWiki instance
2. QB integrity constraint checking
3. Explanation about the problem and identification of offending resources for violated integrity constraints (for details, see Information/Explanations column in Table 2)
4. Automatic repair of violated integrity constraints (available for 13 integrity constraints, for details see column Solutions in Table 2)
5. Opening of offending resources in OntoWiki (if OntoWiki is configured)
6. Validation with RDFUnit

At the moment, RDFUnit and the Validation tool are yet to be a part of the GeoKnow Generator, so the integration with RDFUnit is very basic. Namely, it is integrated as a library where the Validation tool just calls RDFUnit with default parameters and shows the results to the user. In the future, we aim to redirect to GeoKnow Generator's instance of RDFUnit with initial parameters set up, thus enabling a user to further tweak parameters and utilize full potential of RDFUnit.

---

\(^{12}\) CROCUS was described in D3.5.1

\(^{13}\) Integrity constraints are defined by the QB vocabulary
### 4.1 QB Integrity Constraint Checking

For a full list of QB integrity constraints along with explanations and the solutions provided by the Validation tool, see Table 2. Basic terms from the QB vocabulary which appear in this table are described in Section 3.

<table>
<thead>
<tr>
<th>Constraint Description</th>
<th>Information/explanations</th>
<th>Solutions</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>IC-1. Unique Dataset</strong></td>
<td>Observations that violate the constraint</td>
<td>Manually edit problematic observations in OntoWiki</td>
</tr>
<tr>
<td><strong>IC-2. Unique DSD</strong></td>
<td>Datasets that violate the constraint</td>
<td>Choose which of the available datasets the observation will belong to</td>
</tr>
<tr>
<td><strong>IC-3. DSD includes measure</strong></td>
<td>DSDs that do not contain at least one measure</td>
<td>Manually edit problematic DSD in OntoWiki</td>
</tr>
<tr>
<td><strong>IC-4. Dimensions have range</strong></td>
<td>Dimensions that do not have a defined range.</td>
<td>Choose which of the appropriate attributes to transform to measure</td>
</tr>
<tr>
<td><strong>IC-5. Concept dimensions have code lists</strong></td>
<td>Dimensions with range skos:Concept that do not have a code list.</td>
<td>Choose a code list for the dimension from the list of matching code lists.</td>
</tr>
<tr>
<td><strong>IC-6. Only attributes may be optional</strong></td>
<td>Optional components that are not attributes.</td>
<td>Manually edit problematic components in OntoWiki.</td>
</tr>
<tr>
<td><strong>IC-7. Slice keys must be declared</strong></td>
<td>Slice keys that are not associated with a DSD.</td>
<td>Choose which DSD the slice key will be associated to.</td>
</tr>
<tr>
<td><strong>IC-8. Slice keys consistent with DSD</strong></td>
<td>Slice keys that are not consistent with its DSD.</td>
<td>Choose which DSD the slice key will be associated to.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Key</th>
<th>Description</th>
<th>Actions</th>
</tr>
</thead>
<tbody>
<tr>
<td>IC-9.</td>
<td>Slices that are not associated with exactly one slice key. Slice keys that the above slices can be associated to.</td>
<td>Manually edit problematic slices in OntoWiki. Choose slice key that the selected slice will be associated to.</td>
</tr>
<tr>
<td>IC-10.</td>
<td>Slices that do not have a value for every dimension in its slice key.</td>
<td>Manually edit problematic slices in OntoWiki. Choose slice key that the selected slice will be associated to.</td>
</tr>
<tr>
<td>IC-11</td>
<td>Observations that do not have a value for each dimension.</td>
<td>Manually edit problematic observations in OntoWiki.</td>
</tr>
<tr>
<td>IC-12</td>
<td>Observations that have the same value for all dimensions as at least one other observation from the same dataset. Duplicates of the above observations.</td>
<td>Remove duplicates.</td>
</tr>
<tr>
<td>IC-13</td>
<td>Observations that do not have a value for any attribute marked as required.</td>
<td>Manually edit problematic observations in OntoWiki. Remove qb:componentRequired property for the attribute.</td>
</tr>
<tr>
<td>IC-14</td>
<td>Observations that do not have a value for every measure. Missing measures for the observations from above.</td>
<td>Manually edit problematic observations in OntoWiki.</td>
</tr>
<tr>
<td>IC-15</td>
<td>Observations that do not have a value for the measure corresponding to its given qb:measureType. Missing measures for observations from above.</td>
<td>Manually edit problematic observations in OntoWiki.</td>
</tr>
<tr>
<td>IC-16</td>
<td>Observations that have a value for more than one measure.</td>
<td>Manually edit problematic observations in OntoWiki. Remove all excess measures.</td>
</tr>
<tr>
<td>IC-17</td>
<td>Observations that belong to measure dimension cube and do not have values for all measures.</td>
<td>Manually edit problematic observations in OntoWiki.</td>
</tr>
<tr>
<td>IC-18</td>
<td>Observations, slices and datasets for which the dataset</td>
<td>Manually edit problematic observations in OntoWiki.</td>
</tr>
</tbody>
</table>
4.2 Implementation

Implementation-wise, main goals were (1) easy integration with ESTA-LD and GeoKnow Generator, and (2) easy modification and addition/removal of integrity constraints. Since the tool uses SPARQL endpoints to get input and provide the output (as recommended by the GeoKnow Generator contribution guide), integration will not be an issue.

![Figure 5: Integration with other components](image)

In order to address the second requirement, integrity constraints are implemented in a simple and straightforward manner. Namely, their implementation requires only a class that provides the following:

- Title of the constraint
- SPARQL query (or queries)
- Function which tells if the constraint was violated or not based on results of the above query
- Method that generates the user interface for the particular integrity constraint

These classes are added through a single method call, after which the core of the application handles user’s clicks, executes queries, sets the status of integrity constraints and, when needed, triggers generation of the user interface for the selected integrity constraint, thus ensuring that integrity constraints are easy to add, remove and maintain.
The tool is implemented as a Java web application based on the following frameworks:

- Vaadin\textsuperscript{14}: a Java framework for building web applications,
- Sesame\textsuperscript{15}: an open-source framework for querying and analysing RDF data.

Vaadin is used to implement the GUI (graphical user interface), while Sesame is used to execute SPARQL queries on the specified endpoint.

4.3 Installation and Deployment

The tool is open source and the code is available on GitHub\textsuperscript{16}, at [https://github.com/GeoKnow/DataCubeValidation](https://github.com/GeoKnow/DataCubeValidation). It can be downloaded from the link or cloned by using git:

```
git clone https://github.com/GeoKnow/DataCubeValidation.git
```

To build the tool, install Maven\textsuperscript{17}, go to project root directory and type:

```
mvn install
```

After that, war file will be generated in the target directory. You should be able to deploy it in any servlet container which supports Servlet 3.0 specification (it was tested on Apache Tomcat 7). After deployment, application will be available under context rdf-data-cube-validation (i.e. reachable under URL http://servlet-url:servlet-port/rdf-data-cube-validation).

Alternatively, you can run the tool locally in Jetty by opening terminal in project root and typing:

```
mvn jetty:run
```

After this, the tool will be accessible at [http://localhost:8080](http://localhost:8080).

For those that don't want to bother with the installation and deployment, but rather just want to try it out, there is also a demonstrator available at [http://jpo2.imp.bg.ac.rs/rdf-data-cube-validation-demo](http://jpo2.imp.bg.ac.rs/rdf-data-cube-validation-demo). Default endpoint of the demonstrator contains example graphs in order to showcase various situations:

- [http://validation-test/ic-3/](http://validation-test/ic-3/): IC-3 is violated;
- [http://validation-test/ic-4-5/](http://validation-test/ic-4-5/): IC-4 and IC-5 are violated;
- [http://validation-test/ic-6/](http://validation-test/ic-6/): IC-6 is violated;
- [http://validation-test/ic-7/](http://validation-test/ic-7/): IC-7 is violated.

4.4 Configuration

After the tool is all set up and deployed, it needs to be configured (or one could choose to use the default values which point to the default endpoint and graph). Following configuration options are supported (see Figure 6):

- SPARQL endpoint (required)
- Graph (required)
- Basic Authentication (optional)
- OntoWiki instance (optional)

\textsuperscript{14} [https://vaadin.com/home](https://vaadin.com/home)
\textsuperscript{15} [http://rdf4j.org/](http://rdf4j.org/)
\textsuperscript{16} [https://github.com/](https://github.com/)
\textsuperscript{17} [http://maven.apache.org/](http://maven.apache.org/)
The user must specify which endpoint and graph will be used for validation. Additionally, it is possible to configure basic authentication for querying the specified endpoint, and if there is an OntoWiki instance working on top of the same endpoint, one can provide its URL, which is then used to open problematic resources in OntoWiki. Configuration options can be also be supplied using HTTP parameters which will make it easier to integrate validation tool with other tools in the GeoKnow Generator.

Figure 6: RDF Data Cube Validation Tool - Configuration Window

After the configuration options are filled in and OK button is pressed, the tool checks if the endpoint contains specified graph, thereby also checking if the options for accessing the endpoint are correct. If the graph doesn’t exist or an error occurs (e.g. because the endpoint doesn’t exist), the user is notified and configuration options are not accepted. OntoWiki configuration is not validated since the tool has no way of ensuring that the specified OntoWiki really works on top of the endpoint which is used for validation.

4.5 User Guide

In this section, we will describe the user interface, and provide instructions on how to use the application. User interface consists of three main parts (see Figure 7):

1) **control ribbon**: contains general commands
2) **criteria list**: shows a list of criteria (i.e. integrity constraints) and the results of their validation
3) **Integrity Constraint (IC) pane**: shows details about the selected criteria
Figure 7: RDF Data Cube Validation Tool - User Interface

Control ribbon contains following buttons:

- **Refresh**: re-evaluate the currently selected integrity constraint
- **Evaluate All**: evaluate all integrity constraints and update status in criteria list
- **Settings**: open settings window
- **RDFUnit Validation**: validate the graph with RDFUnit

Criteria list is located on the left side and contains a list of integrity constraints. Status of each integrity constraint is visualized with an icon on the left. When a constraint is selected, related details are shown in the IC pane. Furthermore, if the constraint was not previously evaluated (by selecting it in the criteria list or clicking Evaluate All button), it is first validated and its status icon is set based on the outcome of the validation. If a constraint is not violated, a “thumbs up” icon is shown, if it is violated, constraint is given “thumbs down”, and if an error occurred it is denoted with a red “exclamation mark” icon. Icons and their meanings are also given in Table 3.

<table>
<thead>
<tr>
<th>Icon</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>☑️</td>
<td>Integrity constraint satisfied</td>
</tr>
<tr>
<td>🚫</td>
<td>Integrity constraint violated</td>
</tr>
<tr>
<td>🚨</td>
<td>Error occurred</td>
</tr>
<tr>
<td>🤔</td>
<td>Unknown (was not evaluated yet)</td>
</tr>
</tbody>
</table>

Table 3: Integrity constraint icons/indicators

IC pane shows a detailed view of the chosen integrity constraint. Depending on the outcome of the integrity constraint checking, it shows the following:

- If the constraint is not violated:
  - text that informs the user that everything is in order
- If the constraint is violated:
- list of resources which violate the constraint,
- a description of the problem,
- a shortcut to open a problematic resource in OntoWiki (if OntoWiki is configured), and
- a quick, “one-click” solution to the problem (available for 13 integrity constraints)

- In an error occurred:
  - text that informs the user that an error occurred along with the error message

One click solutions require that the endpoint is writeable, which may not be the case even though basic authentication is supported, e.g. an endpoint might not support basic authentication. In these situations, user is notified about the issue and provided with the query that failed. This way, a user can execute the query manually (see Figure 8).

![Figure 8: RDF Data Cube Validation Tool - Failed execution of a fix](image)

It is possible to validate all constraints at once by clicking the Evaluate All button in the control ribbon. After this, all integrity constraints will be evaluated, and icons in the criteria list will be set based on the outcomes of the evaluation (for the list of icons and their meaning see Table 3). After that you can get details about each constraint by clicking it in the criteria list.

Selecting an integrity constraint from the criteria list will only trigger its evaluation if it was not evaluated before. To re-evaluate an integrity constraint, click the Refresh button.

4.6 Example scenario

In this section, we will describe a scenario where a user configures the endpoint, validates a single integrity constrain and uses the option to automatically fix the problem.

First, install and deploy the application as described in section 4.3, or go to http://jpo2.imp.bg.ac.rs/rdf-data-cube-validation-demo. Now click the Settings button, and in the endpoint and graph fields type http://jpo2.imp.bg.ac.rs/sparql and http://validation-test/ic-4-5/ respectively (see Figure 9).
Now click on the IC-4 in the criteria list. After this, icon on the left will change indicating that the constraint is violated, while the IC pane will show detailed information about the constraint in question. This integrity constraint states that every dimension should have a defined range, which is missing for one of the dimensions belonging to the cube in question.

Next, select the dimension that caused the violation and the combo box below will suggest what range to assign to it. In the combo box choose `http://pupin.rs/vukm/test-structure/Interval`, and click on the Quick fix button (see Figure 10).

Since the provided endpoint is not writeable, you will get an error. However, the tool will provide a SPARQL Update query which fixes the problem so that it can be executed manually. If you would like to validate other integrity constraints as well click the Evaluate All button, and the status icon of all integrity constrains will change, denoting that IC-5 is also violated.
5. ESTA-LD GUI (first prototype)

ESTA-LD answers the following GeoKnow requirement

- FR-E1 GeoKnow Generator has to support statistical and temporal geospatial information management including advanced analysis of spatio-temporal data.

5.1 ESTA-LD Architecture

ESTA-LD is a web application implemented in Vaadin, while Javascript libraries LeafletJS and Highcharts are used to visualize a map and charts respectively. For querying the SPARQL endpoint we rely on Sesame framework. User selects different parameters such as graph, dataset, dimensions and their values by using various Vaadin components and a LeafletJS map (for selecting the value of geographic dimension). After this, the application generates a SPARQL query which is then executed using Sesame and finally visualized on a chart created with Highcharts and a map using LeafletJS.

![ESTA-LD Architecture Diagram]
5.2 Implementation of the first prototype

The first prototype was developed using HTML5 and JavaScript in order to enable ease of integration with the GeoKnow Generator. Representation and interaction with geographic information were implemented using Leaflet, an open source JavaScript library for mobile-friendly interactive maps (see http://leafletjs.com/). The geographic data (such as region borders), originally created from shape files, is stored in GeoJSON format. It is brought in and programmed with JavaScript and added to maps to create interactive visualizations. On the other side, different statistical indicators, which are the subjects of the spatio-temporal analysis, are stored in the RDF Data Store. This data repository is accessed and queried using SPARQL query language. The actual retrieval of data from the SPARQL Endpoint is implemented using the jQuery library and its standard getJSON function. Finally, the results of the spatio-temporal analysis are visualized using Highcharts, a charting library written in pure HTML5/JavaScript, offering intuitive, interactive charts to a web site or web application (see http://www.highcharts.com/). Since the GeoKnow Generator is a Javascript web application which uses Java web servlets for the integration of Java components, and Virtuoso as an RDF store, the integration of ESTA-LD will be straightforward. ESTA-LD is a JavaScript web application that can be configured to work with any SPARQL endpoint, meaning that its user interface will be easy to integrate with the Generator and that it can be configured to work with the same RDF store as the other components.

5.3 Using ESTA-LD

When launching the ESTA-LD component, the user specifies the SPARQL endpoint, and selects the graph that contains the data to be explored.

![ESTA-LD GUI (run chart)](image)

The data is then retrieved from the specified SPARQL endpoint and visualized on the choropleth map. The choropleth map provides an easy way to visualize how measurement varies across a geographic area. It is an ideal way to communicate spatial information quickly and easily, since the data is aggregated or generalized into classes or categories that are represented on the map by grades of colour. The ranges of data values for different colors are recalculated every time a new set of data is retrieved from the SPARQL endpoint.
After the data is retrieved, the user can utilize different filtering options that are currently implemented (see Figure 12):

- For selecting values from the time dimension,
- For selecting the indicator under study,
- For selecting the granularity level for the space dimension.

Interactive selection of the area of interest on the geographical map (for the selected area a bar-chart or histogram representation of the indicator is displayed, see Figure 13).
6. Conclusion and future work

The goal of the ESTA-LD component is to provide interactive analysis, visualization and drill-down style online access to highly granular spatio-temporal data. It is aimed for business users interested in spatial analysis of different socio-economic indicators modelled using W3C standard vocabularies (RDF Data Cube, eGovernment) and LinkedData principles.

In this deliverable we presented the first prototype of the RDF Data Cube Validation tool that is currently used to check the quality of Linked data to be visualized with the ESTA-LD component. In Section 4, RDF Data Cube Validation tool was described in detail, including installation and deployment instructions, user guide, and an example scenario that shows how to validate a graph and apply solutions suggested by the tool. In the future RDF Data Cube Validation tool will be tested with large datasets. In the next deliverable 4.6.2, we will provide benchmarking results and optimize used queries where needed. Additionally, we will improve the tool and user interface to account for longer execution times, and add support for abbreviated cubes. Also needed for the ESTA-LD is a tool for validating Linked Data modeled with the Core business vocabulary, and the RDFUnit tool is foreseen for this purpose. More results about using RDFUnit tool with ESTA-LD will be provide in Y3 deliverable.

Currently ESTA-LD works with the local Virtuoso RDF triple store. Supported filtering options are selection of indicator under study, selection of the area of interest on the geographical map, selecting geo granularity level, selecting values from the hierarchical dimensions modelled with the SKOS vocabulary, and selecting the chart type. The results of the spatio-temporal analysis are visualized using Highcharts library, while choropleth map is used to visualize how measurement varies across a geographic area. The ESTA-LD will be released during Y3 after thorough evaluation of the component with different datasets from Serbian government institutions.
7. References


