DELIVERABLE

Project Acronym: **SDI4Apps**
Grant Agreement number: **621129**
Project Full Title: **Uptake of Open Geographic Information Through Innovative Services Based on Linked Data**

D4.1.2 HARMONISATION AND MULTILINGUAL TOOLS - 2

Revision no. 10

Authors: Karel Charvát (Czech Centre for Science and Society)
Ondrej Bojar (Czech Centre for Science and Society)
Roman Sudarikov (Czech Centre for Science and Society)
Otakar Čerba (University of West Bohemia)
Tomáš Mildorf (University of West Bohemia)
Martin Tuchyňa (Slovak Environment Agency)
Tomáš Kliment (ePro)
Stein Runar Bergheim (Asplan Viak Internet)

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

REVISION HISTORY

Revision	Date	Author	Organisation	Description
01	20/09/2015	Karel Charvát	CCSS	Initial draft
02	25/10/2015	Otakar Čerba	UWB	Data transformation
03	07/10/2015	Tomáš Mildorf	UWB	Interoperability, data harmonisation tools and processes
04	16/10/2015	Martin Tuchyňa	SAZP	LOD harmonisation tools
05	22/10/2015	Stein Runar Bergheim	AVINET	Final changes, Executive Summary
06	27/03/2017	Ota Cerba	UWB	Update for last year
07	27/03/2017	Roman Sudarikov	CCSS	Update for last year
08	27/03/2017	Karel Charvat	CCSS	Final draft
09	31/03/2017	Martin Tuchyna	SAZP	Update from pilot 6
10	31/03/2017	Karel Charvat	CCSS	Final

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This deliverable contains original unpublished work except where clearly indicated otherwise.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

Disclaimer:

Views expressed in this document are those of the individuals, partners or the consortium and do not represent the opinion of the Community.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

EXECUTIVE SUMMARY

This report contains the results of the data harmonisation activities conducted as part of the SDI4Apps project, work package 4, tasks T4.1-3. The document was done in incremental way. It is the final version of document, which is extension of D4.1.1

This document must be understood in context of the objectives of the project as well as in alignment with the stages in which the platform, Open API and client-side JavaScript library will be implemented.

In order to build applications on the SDI4Apps platform and APIs, it is necessary to upload data onto the Cloud. Some of these data are application specific and unique to each user and use case; other data are shared across several applications running in the platform. The latter type of datasets include background maps and generic thematic layers that are useful in a wide range of professional domains. When making such data available, harmonization and multilingualism becomes an issue, for data to be useful they have to be combined in ways that make sense to a broad group of users -- for data to be understandable they have to be multi-lingual.

SDI4Apps selected three key data sets for this purpose:

- Transport network
- Points of interest (POIs)
- Open land use (OLU)

In addition to these three data sets, the SDI4Apps platform will come 'preconfigured' with background maps that are composed of OpenStreetMap (OSM) data – and based on OSM cartographic templates. In the case of OSM, the harmonization is a community effort – and while consistency cannot be guaranteed, the history of the data sets has demonstrated its continuous improvement.

Data harmonisation is an expert discipline. It is necessary for shared and re-usable data – but regular users should not attempt such tasks without undergoing specific training; SDI4Apps has therefore placed the harmonisation process 'outside' the platform. The project assumes harmonization to take place prior to loading data into the platform.

While harmonization processes requires expert knowledge, such processes may have to be repeated; as foundation data are updated, the same integration processes will have to be re-run. Thus, in addition to describing the harmonization process, this deliverable also studies a selection of technologies for automating data harmonization processes. In this respect, the approaches that have been assessed include:

- HALE – an expert tool for data harmonization
- Relational database tools – data manipulation using low-level database languages like SQL
- GIS tools – software like QGIS, uDig and similar
- pyWPS – data harmonization as a web processing service

The deliverable deals with both spatial data in a broad sense as well as Linked Data specifically. The deliverable presents possible approaches to transforming enterprise data into RDF and publishing them as Linked Open Data.

The final subject of the deliverable is technologies to handle issues related to multilingualism. In addition to modifying data models, it is necessary to deal with the issue of content language when preparing datasets for exploration and discovery. The document discusses implementations of statistical translation as means for real-time query expansion or auto-generation of multilingual indexes on single language content.

1 HARMONISATION

In collaboration with the OpenTransportNet project, the data interoperability, harmonisation processes and harmonisation tools were described and are included in this report as well as in the OpenTransportNet report D4.4 Data Harmonisation and Integration. These tools and processes are based on the experience from previous projects of the SDI4Apps project partners.

1.1 Interoperability

The SDI4Apps platform will act as a spatial data infrastructure (SDI). SDI, sometimes referred to as spatial information infrastructures, is generally understood as a computerised environment for handling data that relate to a position on or near the surface of the earth (CEN/TR 15449:2011).

There exist many definitions of SDI. The authors use the definition adopted by the INSPIRE directive. INSPIRE defines SDI as “the metadata, spatial data sets and spatial data services; network services and technologies; agreements on sharing, access and use; and coordination and monitoring mechanisms, processes and procedures, established, operated or made available in an interoperable manner.” (European Parliament 2007).

Interoperability is defined by the International Organisation for Standardization (ISO) as "capability to communicate, execute programs, or transfer data among various functional units in a manner that requires the user to have little or no knowledge of the unique characteristics of those units." (ISO/IEC 2382-1:1993).

Recent activity of the European Commission brought to an attention a document describing the European Interoperability Framework (EIF) for European public services. The document highlights the needs and benefits of interoperability. Interoperability enables (European Commission 2010):

- cooperation among public administrations with the aim to establish public services;
- exchanging information among public administrations to fulfil legal requirements or political commitments;
- sharing and reusing information among public administrations to increase administrative efficiency and cut red tape for citizens and businesses. (p. 2)

EIF distinguishes four levels of interoperability including legal, organisational, semantic and technical.

1.2 How to Achieve Interoperability

Interoperability on all levels can be achieved through common standards, specifications and other agreements. If all data, services, legislation, technologies etc. share the same set of agreements, the interoperability can be achieved. The most important international and well-respected standards in the geospatial domain are from the Technical Committee 211 of the International Organization for Standardization (ISO/TC 211) and Open Geospatial Consortium (OGC). Together with national standards, they create the core for SDI implementation. It is highly recommended to keep the national standards compliant with ISO and OGC standards to enable interoperability across national borders.

ISO/TC 211 Geographic information/Geomatics is responsible for the ISO geographic information series of standards. These standards may specify, for geographic information, methods, tools and services for data management (including definition and description), acquiring, processing, analysing, accessing, presenting and transferring such data in digital/electronic form between different users, systems and locations. The ISO/TC 211 standards provide a framework for the development of sector-specific applications using geographic data.

The Open Geospatial Consortium (OGC) is an international industry consortium of 467 companies, government agencies and universities participating in a consensus process to develop publicly available interface standards. OGC standards support interoperable solutions that "geo-enable" the Web, wireless and location-based services and mainstream IT. The standards empower technology developers to make complex spatial information and services accessible and useful with all kinds of applications. OGC standards are developed in a unique consensus process supported by the OGC's industry, government and academic

members to enable geoprocessing technologies to interoperate, or "plug and play". (Open Geospatial Consortium 2012).

As OTN is dealing also with non-spatial data and Web platforms, the World Wide Web Consortium (W3C) standardisation organisation should be mentioned. W3C is developing a set of standards for semantic Web. The W3C and the OGC announced in January 2015 a new collaboration to improve interoperability and integration of spatial data on the Web.[1]

1.3 Harmonisation Tools and Processes

Data harmonisation is necessary for combining data from heterogeneous sources (e.g. regional datasets) into integrated, consistent and unambiguous information products (e.g. European datasets). Such datasets can be then easily used in combination with other harmonised data for viewing as well as querying and analysing. Data harmonisation is a complex task that has not a universal solution that can cover all possible scenarios. Ideal technical solution (system architecture, software) is always determined by many specific facts such as the way in which the original data are stored, data volume and the type of harmonisation. The harmonisation process is a best practice in the geoinformation domain and therefore following chapters firstly describes harmonization experiences of the UWB team, acquired during previous projects (Humboldt, Plan4all and Plan4business) and formulated in Janečka et al. (2013) into a 5-step harmonisation approach.

All relevant steps to perform data harmonisation are depicted in Figure 1.The first three steps are common steps for all scenarios. The theory of spatial data harmonisation within the framework of INSPIRE is based mainly on the INSPIRE conceptual models. The understanding of both source and target data is based mainly on particular data specifications, documentation and metadata.

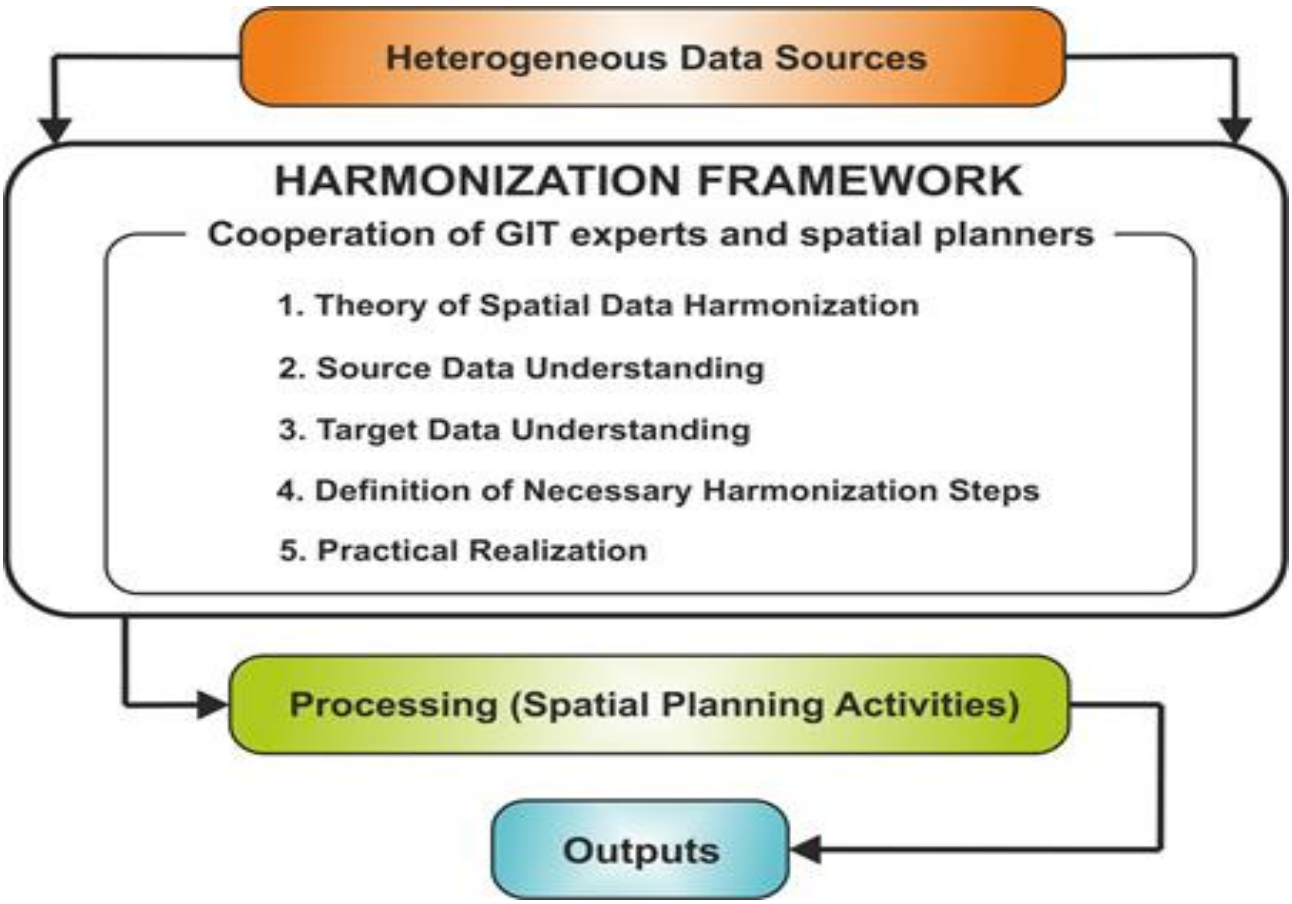

Figure 1 Data harmonisation process (Janečka et al. 2013)

[1] http://www.w3.org/2015/01/spatial.html.en

From the technical point of view there exist several ways how to handle data harmonisation. Relevant approaches are based on ETL tools (e.g. FME) or specialised software designed for the harmonisation (e.g. HALE or Shape Change). Other solution is to use the capabilities provided by relational database management systems (RDBMS) like PostgreSQL with PostGIS. Another solution could be to use a geographic information system such as ArcGIS. In the upcoming sections, we mention the tools most relevant for the use within the SDI4Apps project. Adapted from Janečka et al. (2013). Important role in the harmonisation processes play also Web processing services (http://www.opengeospatial.org/standards/wps), allowing design and execution of the complex geospatial processes within the environment of the internet (e.g. pyWPS).

1.3.1 HALE

The Humboldt Alignment Editor (HALE) is an open source software framework that was designed in the scope of the Humboldt project. "HALE is a tool for defining and evaluating conceptual schema mappings. The goal of HALE is to allow domain experts to ensure logically and semantically consistent mappings and consequently transformed geodata. Furthermore, a major focus is put on documentation of the schema transformation process and its impacts, e.g. in the form of lineage information attached to the resultant transformed data."[2]

For advanced harmonisation projects, where a collaboration over a large community is required, there is a platform where professionals develop and share data transformation processes. The platform is available at www.wetransform.to.

1.3.2 Relational Database Tools

Harmonisation frameworks focus on setting up the harmonisation rules and maintaining the harmonisation lifecycle. Data harmonisation can be also performed by using a database technique and build in functions.

When we focus on the sub-process that deals with conceptual schema transformation, we can find that RDBMSs can offer capabilities to deal with this issue. Once we are able to import our data to RDBMS, we can utilise existing functions and SQL language to fulfil harmonisation processes focused on change of taxonomies and any related harmonisation of attributes (rename, retype) as well as geometry processing (depending on available spatial functions of particular RDBMS).

1.3.3 GIS Tools

The primary objective of GIS tools is not on data harmonisation. However, GIS tools can be used for this purpose. On the one hand, GIS tools are more suitable for harmonisation of geometries than the relational database tools. On the other hand, typical database tasks such as attribute mapping are more difficult in a GIS tool than in a database. GIS tools also follow the geoprocessing principle during the data processing: "input -> operation -> output" and therefore, they produce many temporary layers. Therefore, the user has to take care about naming conventions and data management during the process. ArcGIS (with Arc Toolbox, Model Builder and Python) can serve as a GIS example of harmonisation tools.

1.3.4 pyWPS

Python Web Processing Service (http://pywps.wald.intevation.org) is an implementation of the Web Processing Service standard from the Open Geospatial Consortium. It offers an environment for programming own processes (functions or models) which can be accessed from the public. The main advantage of PyWPS is that it has been written with native support for GRASS GIS. Access to GRASS modules via web interface should be as easy as possible. pyWPS also supports the data model transformations (e.g. INSPIRE) which are foreseen to be used to support the Slovakian Ecosystem Services pilot.

1.3.5 Short Comparison of Harmonisation Tools

All above mentioned harmonisation tools can handle both spatial and non-spatial data - even if first two are primarily focused on attribute schema mapping and the third one is more focused on geometry. What do all the tools have in common is that once the data harmonisation schema is set up, then it could be written as

[2] http://community.esdi-humboldt.eu/

a batch and run in an automated way - with little or no knowledge of the inside of the routine. Practical example of data harmonisation in SDI4Apps is described in the next section.

1.4 Harmonisation Examples

1.4.1 Harmonisation of Transport Data Model

The Transport Network Schema, as depicted in Fig. 1, is a result of a collaboration between four European projects: SDI4Apps, FOODIE (http://foodie-project.eu/), OpenTransportNet (http://opentnet.eu/) and SmartOpenData (http://www.smartopendata.eu/).

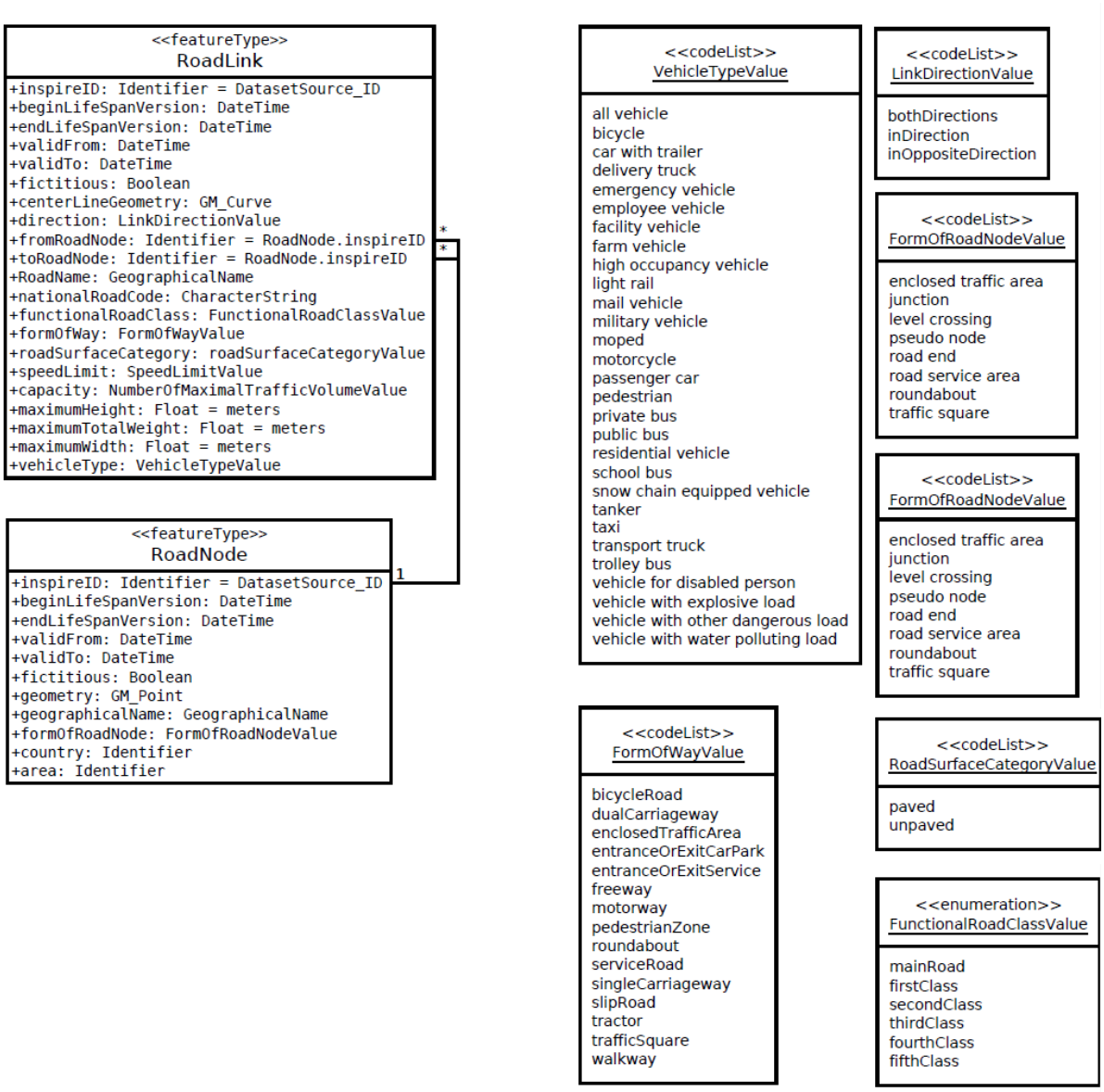

Figure 2 The UML component diagram of theTransport Network Schema as a basis for the navigation application in (precision) agriculture.

Such collaboration enabled in 2015 to create a very efficient data model for navigation purposes based on the open network data. The basic source of the open network data is the OpenStreetMap (http://www.openstreetmap.org) which represents a volunteer-based project aiming at achieving a free editable map of the world. OpenStreetMap has also been intended for the navigation purposes, however such application is still not directly feasible (September 2015). For that reason, four above-mentioned projects have unified their forces in order to create:

1. the Consolidated Transport Network Schema, as depicted in **Error! Reference source not found.**, as a basic data model for the navigation purposes in all four projects;
2. mapping between the OpenStreetMap as a source and the Consolidated Transport Network Schema as a target in order to transform data;
3. the PostgreSQL + PostGIS database defined according to the Consolidated Transport Network Schema;
4. population, verification and correction of the OpenStreetMap data for navigation purposes, such as to eliminate undershoots and/or overshoots in order to get topologically clean dataset;
5. export the corrected geometry back into the OpenStreetMap.

The crucial requirements of the FOODIE project on the FOODIE Transport Network Schema have been following (see also the Listing 1):

1. *RoadLink* feature type as the core of the topological graph when composing the edges of such graph;
2. attributes *inspireId* (unique identifier of the edge), *fromRoadNode* and *toRoadNode* (connection to the beginning and ending nodes), *roadName* (unique human designation of the edge), *formOfWay* (a classification based on the physical properties of the edge), *roadSurfaceCategory* (specification of the state of the surface of the associated *RoadElement*, indicates whether a road is paved or unpaved), *maximumHeight* (maximum height of a vehicle for an explicit edge), *maximumTotalWeight* (maximum total weight of a vehicle for an explicit edge), *maximumWidth* (maximum width of a vehicle for an explicit edge) and *vehicleType* (list of possible types of vehicles);
3. *RoadNode* feature type as the core of the topological graph when composing the nodes of such graph;
4. attributes *inspireId* (unique identifier of the node) and *formOfRoadNode* (functions of road nodes).

```
=======                                          ======
«featureType»
RoadLink                                         OpenStreetMap source
========                                         ======
+ inspireID: Identifier [1]                 OSM.roads.osm_id
+ beginLifeSpanVersion: DateTime [1]              <date of import>
+ endLifeSpanVersion: DateTime [0..1]             <null>
+ validFrom: DateTime [1]                        <date    of    import/<null>    if
                                                 OSM.roads.type=planned/proposed/c
                                                 onstruction>
+ validTo: DateTime [0..1]                  <null>
+ fictitious: Boolean = false [1]           false
+ centerlineGeometry: GM_Curve [1]               OSM.roads.geometry  (topologically
                                                 cleaned)
+ direction: LinkDirectionValue «codelist»       OSM.roads.oneway
+ fromRoadNode: forein key [1]                   RoadNode.inspireID {FK}
+ toRoadNode: forein key   [1]                   RoadNode.inspireID {FK}
+ RoadName: CharacterString [0..*]              OSM.roads.name {street names}
+ nationalRoadCode: CharacterString [0..1]       OSM.roads.ref {FK}
+ functionalRoadClass:FunctionalRoadClassValue «enumeration»
                                                 OSM.roads.type
+ formOfWay: FormOfWayValue «codelist»     OSM.roads.type
+ roadSurfaceCategory: RoadSurfaceCategoryValue «codelist»
                                                 OSM.roads.surface
```

```
+ speedLimit: SpeedLimitValue (km/h)            OSM.roads.maxspeed
+ capacity: NumberOfMaximalTrafficVolumeValue [0..1]  <null>
+ maximuHeight: Float         (meters)          OSM.roads.maxheight
+ maximumTotalWeight: Float   (tons)      OSM.roads.maxweight
+ maximumWidth: Float         (meters)          OSM.roads.maxwidth
+ vehicleType: VehicleTypeValue     «codeList»  null


=============
«featureType»
RoadNode
=============
+ inspireID: Identifier [0..1]                  generated
+ beginLifeSpanVersion: DateTime [1]            <date of import>
+ endLifeSpanVersion: DateTime [0..1]           <null>
+ validFrom: DateTime [1]                       <date of import>
+ validTo: DateTime [0..1]                 <null>
+ fictitious: Boolean = false [1]          false
+ geometry: GM_Point [1]                        generated from RoadLink
+ geographicalName: CharacterString [0..1]
                                                OSM.highway=motorway_junction.name
+ formOfRoadNode: formOfRoadNodeValue «codeList»       <null>

=============================================================================«codeList»
Network::LinkDirectionValue               OSM.roads.oneway
============================
+ bothDirections                                0
+ inDirection                                   1    (follows    the    way    of
                                                vectorization)
+ inOppositeDirection                           1 (opposite)

========================
«enumeration»
FunctionalRoadClassValue                       OSM.roads.type
========================
mainRoad                                       motorway, motorway_link, trunk,
                                               trunk_link
firstClass                                      primary, primary_link
secondClass                                    secondary, secondary_link
thirdClass                                      tertiary, tertiary_link
fourthClass                                    residential, living_street,
                                               unclassified
fifthClass                                      <all other values>

==============
«codeList»
FormOfWayValue                                 OSM.roads.type
==============
+ bicycleRoad                                  cycleway
```

+ dualCarriageway	motorway_link, trunk, trunk_link, primary_link, secondary_link, tertialy_link
+ enclosedTrafficArea	raceway
+ entranceOrExitCarPark	<not a corresponding value>
+ entranceOrExitService	<not a corresponding value>
+ freeway	<not a corresponding value>
+ motorway	motorway
+ pedestrianZone	<not a corresponding value>
+ roundabout	<not a corresponding value>
+ serviceRoad	<not a corresponding value>
+ singleCarriageway	<all other values>
+ slipRoad	<not a corresponding value>
+ tractor	<not a corresponding value>
+ trafficSquare	<not a corresponding value>
+ walkway	pedestrian, footway, steps, path
==========	
«codeList»	
VehicleTypeValue	<not a corresponding value>
==========	
«codeList»	
RoadSurfaceCategoryValue	OSM.roads.surface
==========	
+ paved	paved, asphalt, cobblestone, cobblestone:flattened, sett, concrete, concrete:lanes, concrete:plates, paving_stones, paving_stones:30, paving_stones:20, metal
+ unpaved	<all other values>
==========	
RoadLink-osm_extension	
==========	
+ z_order: Int = 0 [1]	if OSM.roads.bridge=1 then z_order=1 if OSM.roads.tunnel=1 then z_order=-1

Table 1 Mapping between the OpenStreetMap structure and the Transport Network Schema

1.4.2 Smart Tourist Data Model

As Part of Smart Tourist Data, there were prepared harmonised database Smart Point of Interest (SPOI). SPOI is the seamless and open resource of POIs available for other users to download, search or use in applications and services. The data model (see Figure 3) of SPOI comes from review of literature, existing data (for example OpenPOIs), and recommendations of W3C and OGC and user requirements. The current version of the data set has been created as a harmonized combination of selected OpenStreetMap data, GeoNames.org data, experimental ontologies developed in the Section of Geomatics of the University of

West Bohemia, local data provided by the Uhlava region (Czech Republic) and other data available on the Internet (for example selected files from POI Plaza). The transformation was realized by Bash scripts, PHP, XSLT templates and Saxon processor. Data are stored in the Virtuoso tool as RDF triples. SPOI is published via map client and SPARQL endpoint that enables comfortable, efficient and standardized querying of data.

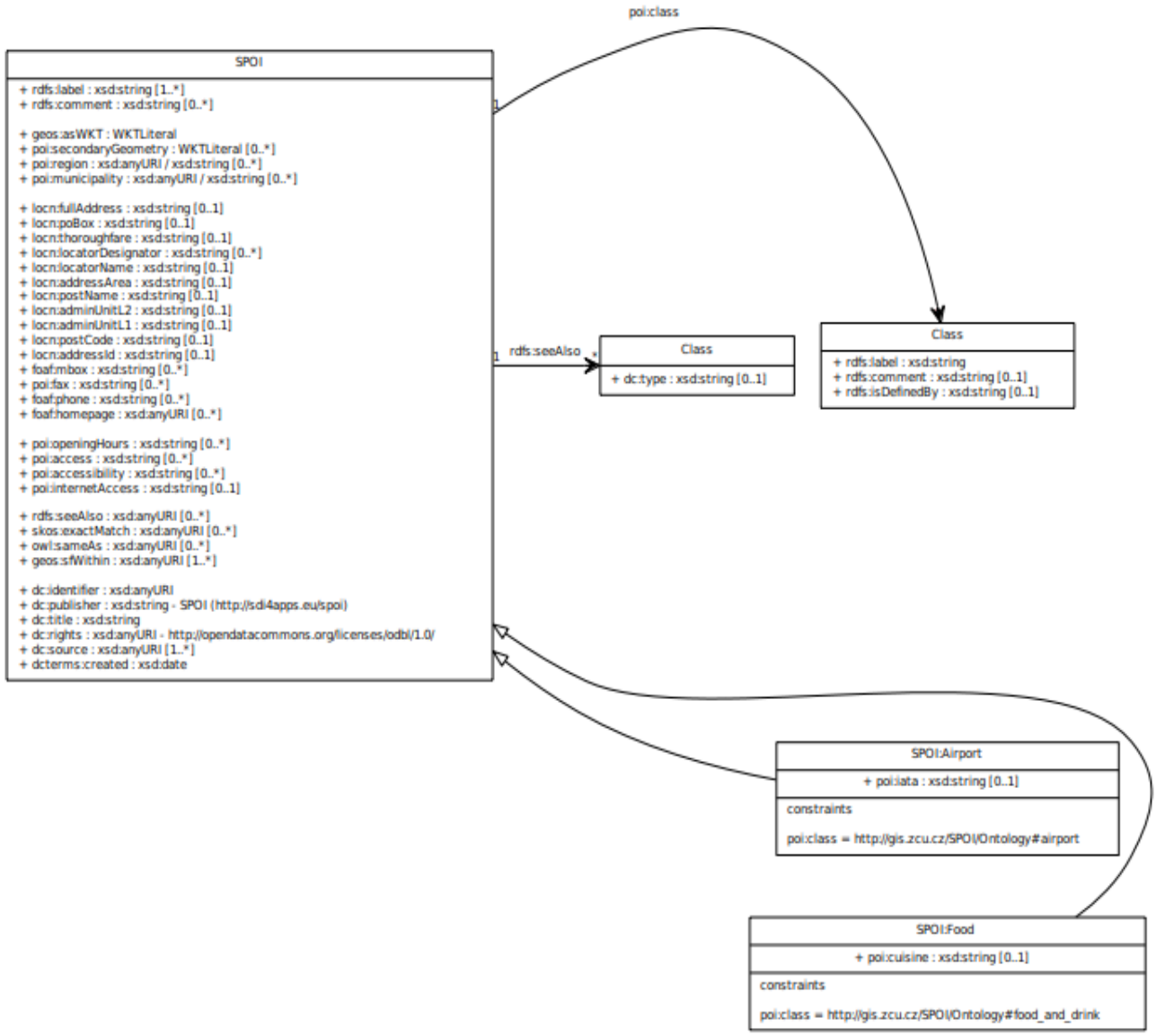

Figure 3 SPOI data model

SPOI uses mainly respected and open web standards such as RDF format, SPARQL query language or several vocabularies (for example FOAF). The data storage and SPARQL endpoint is implemented in Virtuoso tool.

Data harmonization (Annex 1) includes following steps (their concrete implementation depends on particular input data resources):

1. Transcription to structured data – several data are provided by users as tables or texts, which have to be transformed to an XML structure that is then processed by XSLT templates.
2. Transformation to common data model (Figure 3) – this harmonisation step is realized by XSLT templates (developed for every source data set) and Saxon processor or by Bash script or by PHP.
3. Preparation of common vocabularies – SPOI Classification Ontology using Waze and OSM categories as well as links to GEMET or EuroVoc.
4. Mappings and reclassification between categories used in source data (OSM, GeoNames.org...) to SPOI Ontology.
5. Preparation of topological links to relevant countries and identity links to equivalent or similar features.
6. Filtering of original information.
7. Export to common data format (RDF).

1.4.3 Open Land Use

The Open Land Use data set was initially harmonised for the Czech Republic as an example. This approach will be applied in other countries. The land use classification is based on the Hierarchical INSPIRE Land Use http://inspire.ec.europa.eu/documents/Data_Specifications/INSPIRE_DataSpecification_LU_v3.0rc2.pdf) as Classification System (HILUCS) (see well as colour scheme for the visualisation.

The translation tables that are used to translate original land use/land cover classes into HILUCS are legacy of the Plan4business project.

Data sources for the pilot area (Czech Republic):

1. Cadastral data (RUIAN)
2. Land Parcel Identification System (LPIS)
3. Spatial plans (functional areas)
4. Urban Atlas (European Environmental Agency)
5. Corine Land Cover (European Environmental Agency)

The data for the project is downloaded and stored in PostgreSQL database. The concept of the database is influenced by the huge volume of highly detailed spatial data that we are dealing with (there are about 17 000 000 parcels available in vector format for the whole country - and as vectorization/mapping of cadastral map is ongoing - this number will increase, also there are many thousands of features in other layers).

Database structure:

1. Master table
2. Child tables that inherit from the master table

It was decided to divide the data by the LAU2 administrative units. Every LAU2 table with land use features is a separate child table that inherits from the master table that has the following structure:

```
CREATE TABLE olu_master
(
Idbigserial NOT NULL,
-- unique id
geom geometry NOT NULL,
-- geometry of feature
```

hilucs_land _use numeric(3,0) NOT NULL,

-- hilucs landuse

id_original bigint NOT NULL,

-- id from the child table

id_adm_unit numeric(6,0) NOT NULL,

-- administrative unit id (to which feature belongs)

CONSTRAINT parcely_master_pkey PRIMARY KEY (id)

);

Table 2 OLU scheme

The features inside each LAU2 table are made up based on the following rules:

- Initially, each table gets filled in with land parcels (most detailed level),
- if vector land parcels don't exist in that given LAU2 - or don't cover the whole LAU2 region - we go to the next level: LPIS data - and fill with it what is not filled with land parcels (geometrically LPIS - Cadastre) - after that, again, if not everything in LAU2 is filled with with Cadastre + LPIS we go to available spatial plans and fill gaps with these data, if again some gaps are left - we use Urban Atlas to fill it - if not - Corine Land Cover (this dataset is the least detailed but covers the whole Czech Republic). So in the end we have the database covering the whole pilot area - Czech Republic.

2 LINKED OPEN DATA HARMONISATION

2.1 SK Pilot Linked Open Data Harmonisation

In order to demonstrate the potential of linked open data harmonisation and further re-use, set of activities was undertaken within the Slovakian pilot 6 focused on Ecosystem services evaluation[3].

As described on D 4.1.1. set of spatial linked open data has been harmonised according the SmOD INSPIRE Vocabularies[4] and published via Open Data Node portal of Slovak environmental agency[5].

To demonstrate the possibility to combine this data with additional data resources created via SDI4Apps project, selected linked open data has been integrated into the SK Pilot Ecosystem Services Evaluation App[6].

Following two INSPIRE datasets has been integrated and made available for the further queries:

- SK LD INSPIRE Species distribution[7]
- SK LD INSPIRE Bio-geographical regions[8]

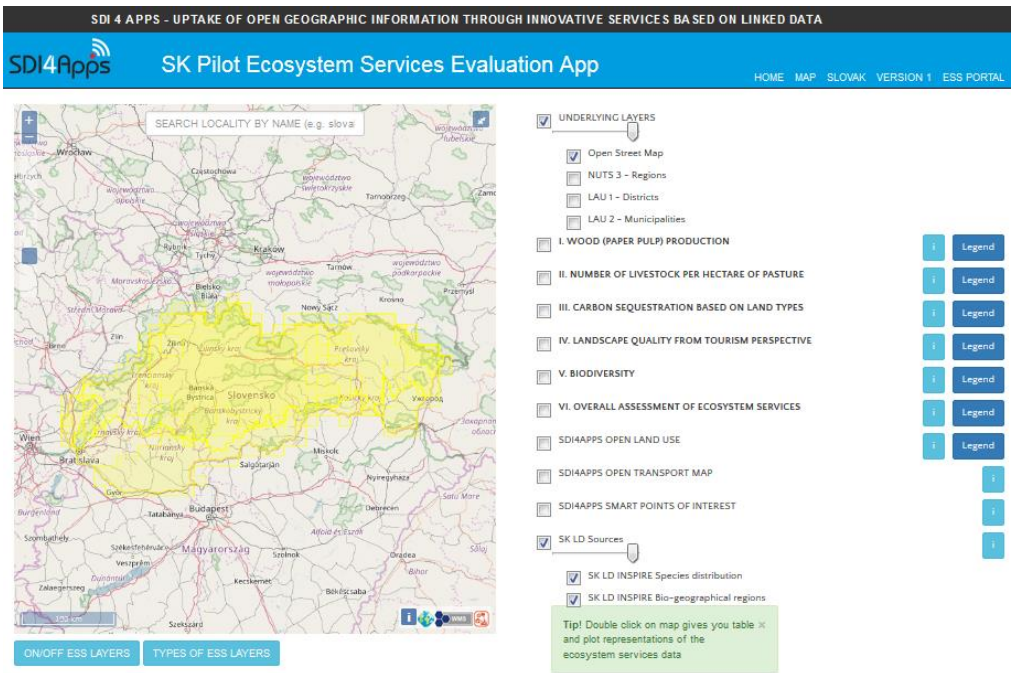

Figure 4 SPOI SK Linked Data in ESS Evaluation App

Integration has been made via Map application of the SK Pilot Ecosystem Services Evaluation App, combining these datasets with another Ecosystem services related datasets as well as datasets from SDI4Apps, Open Transport Net projects on top of underlying open data resources (Open street map, NUTS regions).

[3] http://sdi4apps.eu/project-information/pilot-applications/pilot-6-ecosystem-services-evaluation/
[4] https://www.w3.org/2015/03/inspire/
[5] https://data.sazp.sk/
[6] http://skpilot-viewer.virt.ics.muni.cz/ol3/eng/map-dev.html
[7] https://data.sazp.sk/dataset/sk-ld-inspire-species-distribution
[8] https://data.sazp.sk/dataset/sk-ld-inspire-bio-geographical-regions

2.1.1SK LD INSPIRE Species distribution

This datasets provides the sample data about the species distribution in Slovakia. Harmonisation exercise via ESS SK pilot shows the possibility to visualise the linked open data via external web map application. This app provides also the possibility to query attribute information via Map feature info.

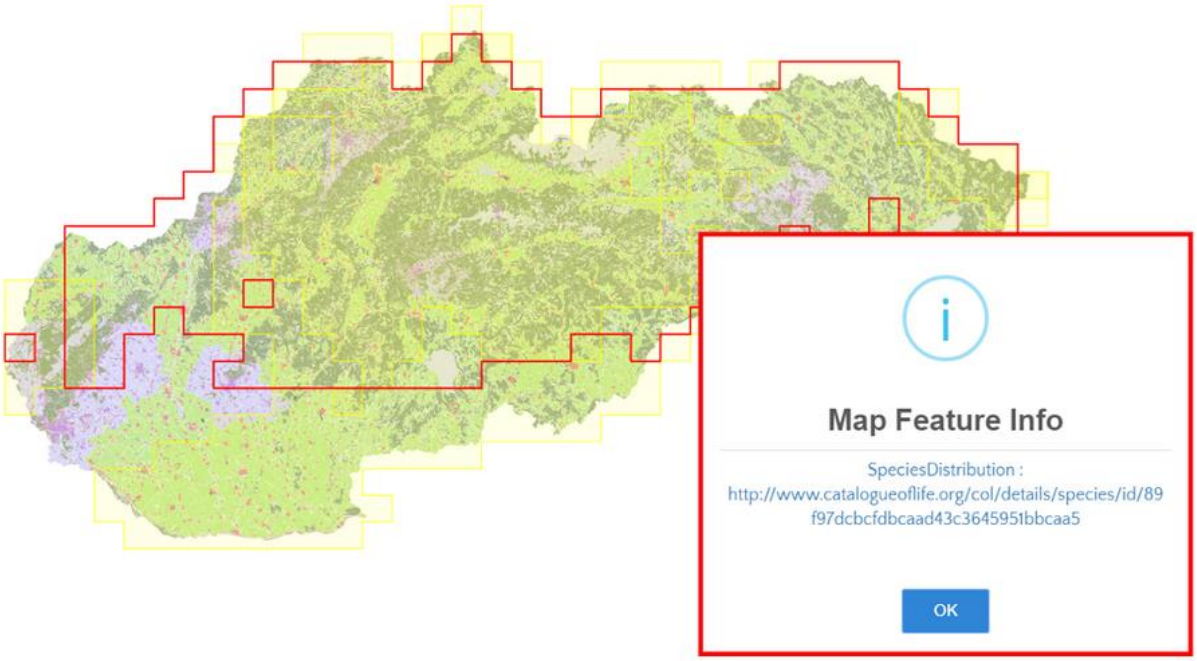

Figure 5 SK Open Linked Data for INSPIRE Slovakian species distribution

When requesting Map feature info, users can access additional linked information from external resources (

- INSPIRE Registry[9]
- Catalogue of life[10]

[9] http://inspire.ec.europa.eu/registry/

[10] http://www.catalogueoflife.org

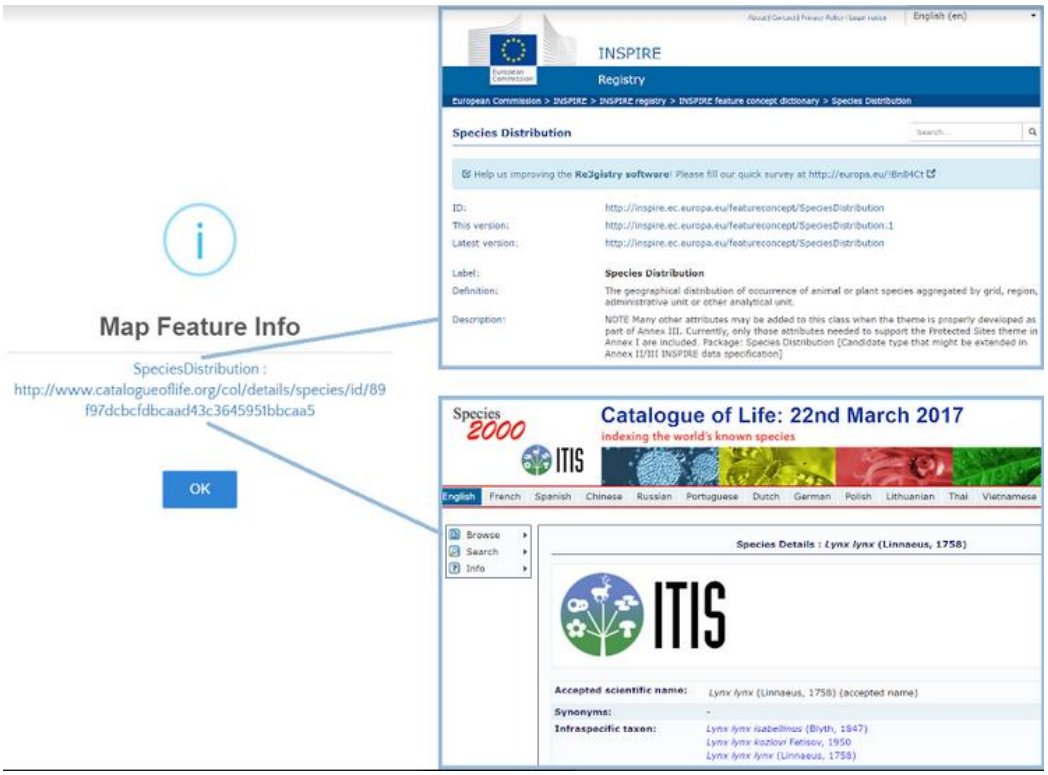

Figure 6 Additional external map feature links from Slovakian species distribution

2.1.2SK LD INSPIRE Bio-geographical regions

With this dataset, information about the location of Bio-geographical regions has been integrated into the SK Pilot Ecosystem Services Evaluation App.

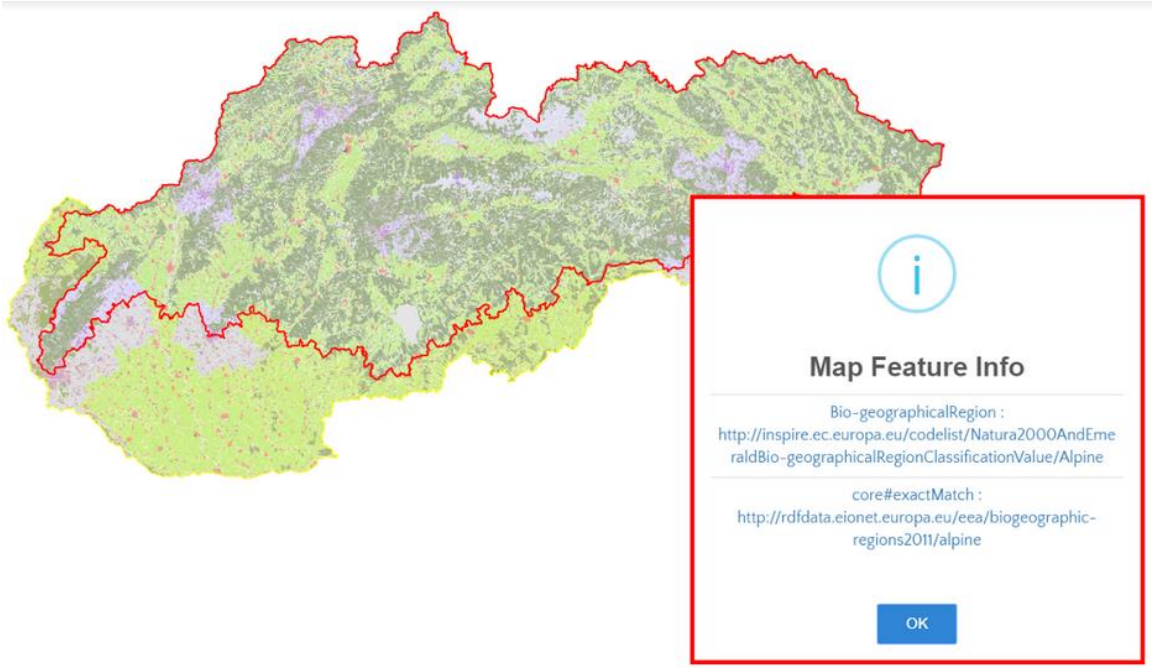

Figure 7 SK Open Linked Data for INSPIRE Slovakian Bio-geographical regions

When requesting Map feature info, users can access additional linked information from external resources via:

- INSPIRE Registry[11]
- SKOS
- Eionet[12]

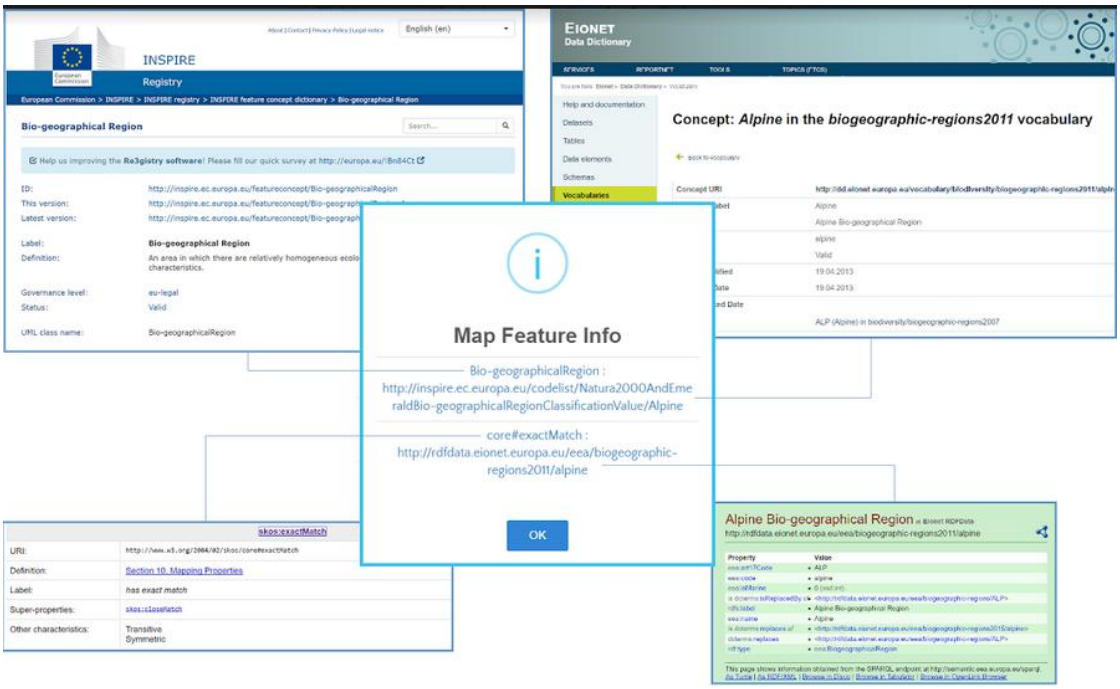

Figure 8 Open Linked Data for INSPIRE Slovakian Bio-geographical regions

[11] http://inspire.ec.europa.eu/registry/

[12] http://www.eionet.europa.eu/

3 MULTILINGUAL TOOLS

3.1 Statistical Translation

Machine translation is a subfield of computational linguistics that investigates the use of software to translate text or speech from one language to another. On a basic level, MT performs simple substitution of words in one language for words in another, but that alone usually cannot produce a good translation of a text because recognition of whole phrases and their closest counterparts in the target language is needed. Solving this problem with corpus and statistical techniques is a rapidly growing field that is leading to better translations, handling differences in linguistic typology, translation of idioms, and the isolation of anomalies. Current machine translation software often allows for customization by domain or profession (such as weather reports), improving output by limiting the scope of allowable substitutions. This technique is particularly effective in domains where formal or formulaic language is used. It follows that machine translation of government and legal documents more readily produces usable output than conversation or less standardised text. Statistical machine translation tries to generate translations using statistical methods based on bilingual text corpora. Where such corpora are available, good results can be achieved translating similar texts, but such corpora are still rare for many language pairs. Generally, the more human-translated documents available in a given language, the more likely it is that the translation will be of good quality.[13]

3.2 MOSES

Moses is an implementation of the statistical (or data-driven) approach to machine translation (MT). This is the dominant approach in the field now, and is employed by the online translation systems deployed by the likes of Google and Microsoft. In statistical machine translation (SMT), translation systems are trained on large quantities of parallel data (from which the systems learn how to translate small segments), as well as even larger quantities of monolingual data (from which the systems learn what the target language should look like). Parallel data is a collection of sentences in two different languages, which is sentence-aligned, in that each sentence in one language is matched with its corresponding translated sentence in the other language. It is also known as a bitext. The training process in Moses takes in the parallel data and uses concurrences of words and segments (known as phrases) to infer translation correspondences between the two languages of interest. In phrase-based machine translation, these correspondences are simply between continuous sequences of words, whereas in hierarchical phrase-based machine translation or syntax-based translation, more structure is added to the correspondences. For instance a hierarchical MT system could learn that the German ‘hat X gegessen’ corresponds to the English ‘ate X’, where the ‘Xs’ are replaced by any German-English word pair. The extra structure used in these types of systems may or may not be derived from a linguistic analysis of the parallel data. Moses also implements an extension of phrase-based machine translation know as factored translation that enables extra linguistic information to be added to phrase-based systems.[14]

3.3 Implementation

The implementation was provided as subcontract of Charles University in Prague, Institute of Formal and Applied Linguistics (UFAL in the following) for Czech Centre for Science and Society. In the first phase of the collaboration between CCSS.cz, one partner in the SDI4Apps project, and UFAL, UFAL implemented a baseline machine translation (MT) system for Czech-to-English translation. The system is based on the Moses open-source toolkit and as such relies on parallel and monolingual data coming from the domain in question as much as possible. For this, we collected parallel and monolingual texts from EUR-Lex. EUR-Lex was accessed in two ways, directly browsing the files and also by its search facility. The respective entry points are:

- http://eur-lex.europa.eu/oj/direct-access.html

[13] https://en.wikipedia.org/wiki/Machine_translation
[14] http://www.statmt.org/moses/?n=Moses.Overview

- http://eur-lex.europa.eu/search.html

The size obtained corpora is detailed in Table 2.

	Segments	English Words	Czech Words
Direct Access, Parallel	5.19M	36.10M	28.75M
Search Access, Parallel	10.38M	91.98M	122.76M
Direct Access, Monolingual	20.96M	--	141.12M
Direct Access, Monolingual	20.93M	158.74M	--
Search Access, Monolingual	40.10M	--	304.49M
Search Access, Monolingual	45.24M	633.99M	--

Table 3 Corpora

After de-duplication at the segment level, we have, approximately figures as indicated in Table 3.

	Segments	English Words	Czech Words
Direct Access, Parallel	1.54M	26.83M	20.42M
Search Access, Parallel	2.92M	63.25M	95.68M
Direct Access, Monolingual	5.36M	--	97.38M
Direct Access, Monolingual	5.15M	107.83M	--
Search Access, Monolingual	10.77M	--	169.32M
Search Access, Monolingual	13.94M	462.76M	--

Table 4 After de-duplication

The data were divided into training and development set ourselves and UFAL had a baseline system ready, although not optimized for the intended domain. Ideally, the development set of at least 2000 sentences, each equipped with its translation, would come exactly from the domain we want to translate.

All the reported work has been carried out on the computer cluster at UFAL. After this Moses was installed on the supercomputer of the Masaryk University.

Further, clean-up of the corpora is desirable, but we first need to see the translation quality on some real input documents, to assess if better translation quality will be reached more likely by additional data or by data clean-up.

3.4 Machine Translation as a Service

During the second phase of the project, UFAL created a system for automatic training and deployment for statistical machine translation models. The work was initially done on computer cluster at UFAL and Metacentrum and later migrated to the supercomputer of the Masaryk University. The system consisted of two major parts: model creation and training subsystem and model deployment server.

3.4.1 Model creation and training

Model creation and training was performed on the basis of eman-lite[15] toolkit. This toolkit builds upon on Moses statistical machine translation system, GIZA++ implementation of IBM models for word alignment and SALM - an efficient implementation of suffix arrays. The toolkit takes prepared data in the form of a parallel corpus for phrase extraction and monolingual corpus for language model construction, processes it and creates a Moses model with language model and phrase tables extracted from the given corpora. After that it performs the training process to obtain correct parameters (weights) for the model using Minimum Error Rate Training (MERT) method. The system attempts to optimize parameters using BLEU[16] evaluation metric, based on predefined parallel development corpus. The closer the development corpus is to the data to be translated during the later usage of the system, the better results would be, that's why it is necessary to have at least a small parallel development corpus with data from actual domain of interest.

In the end, the system produces the model with optimized weights for both phrase tables and language model, which is passed to the next step – the deployment of the model on a server.

3.4.2 Model deployment server

Model deployment server integrates MTMonkey[17] system. MTMonkey is an adaptable system for Machine Translation services, written in Python. It allows clients to contact Machine Translation servers using JSON-encoded messages. The system consists of several parts. The first part is an application server, which receives requests and manages them between workers, based on requested parameters such as source and target languages. The workers are the second part of the system and they perform the actual translation, each worker having its own language pair to handle. The rest are different supporting scripts and tools to perform text preprocessing and system's self-checks.

3.4.3 Service infrastructure

The integration of the two components (model creation and training, and model deployment) described above is done using a newly-created system EmaNkey[18]. It wraps both parts and provides means for complete automatic run from parallel corpus to deployed model. The system uses a predefined directory structure for easy navigation and convenient control of the translation flow.

For example, if one wants to create a new model from a parallel corpus, one has to create a new folder in “models/” directory, put there the preprocessed corpus (with names train.tgt and train.src for target and source sides, resp.) and then launch “process_models.sh” and wait till the model is processed and deployed. If the “process_models.sh” is placed in a cron scheduler, then the whole thing would happen automatically. Once the model is trained, it is automatically deployed. That means that a new MTMonkey worker is created and MTMonkey server configuration is updated, so this new worker also can get requests. To provide the distinction between several models with same source and target languages, each model appends date-stamp to it's name, so clients can specify, which model to use for the translation.

[15] http://www.kconnect.eu/sites/default/files/docs/toolkit_and_report_for_translator_adaptation_to_new_languages.pdf
[16] http://www.aclweb.org/anthology/P/P02/P02-1040.pdf
[17] https://github.com/ufal/mtmonkey
[18] https://github.com/pompomon/EmaNkey

The whole system was tested on cluster machines at UFAL and Metacentrum and then was installed on the supercomputer of the Masaryk University.

4 CONCLUSION

The harmonisation processes and their performance are for experts only. A team of experts from the SDI4Apps consortium dedicates most effort to data harmonisation and integration with other data sources. These data include spatial data, non-spatial data and linked data.

The three data sources that will serve as a basis for the SDi4Apps platform are being harmonized and the harmonisation will continue until the end of the project. Data are key components of the platform and represent more than 80% of the platform value. The rest belongs to software and hardware.

The multilingual tools were implemented and being trained to perform translation for the platform. The following aspects were taken into account:

- some relevant sources crawled for training data,
- further sources are still needed,
- MT system was installed on the MU facilities.

Harmonisation process was recognised as one from potential sustainable economic activities for SDPI4Apps partners

REFERENCES

European Parliament, 2007. DIRECTIVE 2007/2/EC OF THE EUROPEAN PARLIAMENT AND OF THE COUNCIL of 14 March 2007 establishing an Infrastructure for Spatial Information in the European Community (INSPIRE). Available at: http://eurlex.europa.eu/JOHtml.do?uri=OJ:L:2007:108:SOM:EN:HTML [Accessed January 14, 2015].

International Organization for Standardization, 2011. CEN/TR 15449 Geographic information - Standards, specifications, technical reports and guidelines, required to implement Spatial Data Infrastructures.

International Organization for Standardization, 1993. ISO/IEC 2382-1 Information technology -- Vocabulary -- Part 1: Fundamental terms.

Janecka, K., Cerba, O., Jedlicka, K., Jezek, J. 2013. TOWARDS INTEROPERABILITY OF SPATIAL PLANNING DATA 5-STEPS HARMONIZATION FRAMEWORK. In SGEM2013 Conference Proceedings, ISBN 978-954-91818-9-0 / ISSN 1314-2704, June 16-22, 2013, Vol. 1, pp 1005 - 1016.

European Commission, 2010. European Interoperability Framework (EIF) for European public services.