GIMA

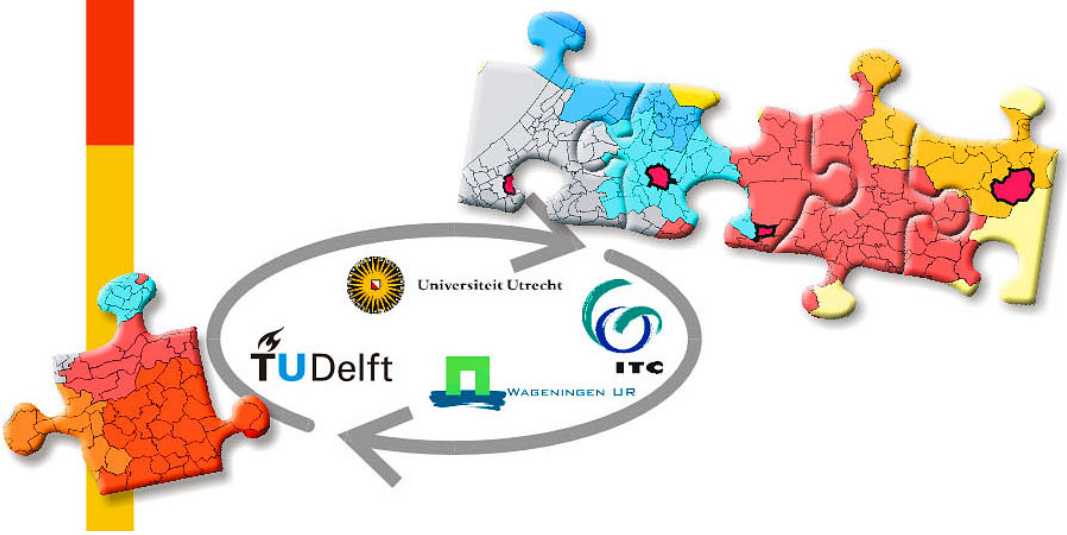
Geographical Information Management and Applications

Automated Metadata Generation

Internship Report – August 2012

Author: Nart Tamash (UU: 3569225)
E-mail: nart.tamash@gmail.com

GIMA Supervisor: Ir. John Stuiver
Host Organisation: University College London
Host Org. Supervisor: Dr. Claire Ellul / Patrick Rickles



Abstract

Geographic information metadata has a very important role in the GIS data management workflow of any project or organisation. Moreover it is one of the main components of any spatial data infrastructure (SDI), assisting in the discovery, evaluation, and acquisition of spatial data. Traditionally, metadata is very often neglected by severely under-prioritising its creation and maintenance. To efficiently overcome this issue, automated metadata generation and update methods have to be implemented in the GIS data management workflow. This is even more important when considering the increasing number of non-GIS specialists that create and use spatial data in inter-disciplinary (research) projects. The internship project aimed at researching a database trigger mechanism to implement automated metadata generation based on the standard of one of the most dominant SDI initiatives at the moment: INSPIRE. The main advantage of this approach is that the automation functionality is not dependent on specific GIS software or file format, and considering the free and open source spatial database options (i.e. PostgreSQL/PostGIS), it can prove to be a very cost-efficient solution. A theoretical background is provided on metadata automation approaches and concepts researched so far. Development and implementation of the database trigger mechanism is briefly described on an element by element basis, followed by discussion of results, which include the successful automation of 15 of the 18 mandatory INSPIRE metadata elements. Recommendations for further work are also provided, as well as the main conclusions of the project.

Keywords: metadata, automation, INSPIRE, database, trigger, PostgreSQL, PostGIS, PL/pgSQL

Table of Contents

Introduction

This document is a report related to an internship as part of the GIMA (Geographical Information Management & Applications) master programme. The internship was carried out within the Department of Civil, Environmental & Geomatic Engineering at University College London between 16th April and 7th September, 2012. Responsible supervisors at the host organisation were Dr. Claire Ellul and Patrick Rickles, while from the GIMA programme the internship was supervised by Ir. John Stuiver. The topic of the internship was based on developing an automated metadata generation system based on a database trigger mechanism. The report provides a general background to the topic, by presenting the project framework and carrying out a literature review. It then describes the implementation phase and processes carried out, and finalises with a discussion on results and recommendation for future work, as well as conclusions of the entire internship project.

It is also important to note that the work is based on a European metadata standard (i.e. INSPIRE). The detailed requirements of this standard are presented in Annex A of the report. Moreover, the internship work represented the 'back-end' side (i.e. database based) of a 2-part project. The other (front-end) part of the project is further presented in the report.

1. Project framework

This chapter will provide a general overview over the framework of the internship project by giving a brief introduction into the problem statement in Section 1.1, and clearly defining the goals of the internship in Section 1.2. Finally, Section 1.3 will shortly describe the steps to be taken to achieve the proposed goals.

1.1 Problem statement

In the geo-information context metadata is used to capture basic characteristics about the spatial data ('data about data') such as standalone GIS files, geospatial databases, or earth imagery. It can also be used to document geospatial resources including data catalogues, mapping applications, data models and related websites. Moreover, it has been considered to be one of the main components of any Spatial Data Infrastructure (SDI), as metadata availability is crucial in data discovery and understanding the suitability of a given dataset for a specific task, understanding its quality, as well as tracking its lineage.

Nowadays spatial data is not exclusively created by expert users anymore, as there are an increasing number of non-GIS specialists that create and use this type of data as part of their work. When also taking into consideration the fact that GIS metadata is traditionally considered to be 'boring' and time-consuming to produce, the result is that metadata is often not created at all, or it is created at the end of a project for data publication purposes. Unfortunately, this will not reflect the actual quality and lineage of a dataset (Ellul *et al.*, 2012), and it will ultimately result in a poor data management workflow, which might eventually affect the end results of a project and the reuse of the dataset. Furthermore, metadata records are generally not linked to the datasets they represent,

which means that when a dataset is changed or updated, the metadata has to be updated separately as well.

These aspects are extremely relevant in contexts ranging from specific organisations to much broader adoptions like national and internationals policies. In this sense, many metadata standards have been developed and adopted to be used at enterprise, local, regional, national or international scales. One of the most representative example is the INSPIRE Directive, which sets up a framework for the creation of an European Spatial Data Infrastructure (ESDI), which will enable the sharing of environmental spatial information among public sector organisations and better facilitate public access in general to spatial information across Europe. Inevitably, the INSPIRE framework addresses metadata as one of its main elements. Therefore, metadata generation needs a more integrated approach with minimum manual input that would first of all save on the costs associated with metadata creation and maintenance, and would also ensure a more efficient data management workflow by addressing the metadata issue at the right time and not just as a very last project task. This is especially relevant in projects that involve non-GIS specialists that use and create spatial data, and that are unaware and unwilling to be aware on the importance of metadata, an issue that is traditionally often ignored even by specialists.

1.2 Internship objective

Based on the presented problem statement in the previous section, the main objective of the internship was to develop an automated metadata generation system based on a database trigger mechanism, and to integrate it within the general GIS data management workflow of a given project. The work mainly falls under the scope of two research projects, one at University College London (i.e. Adaptable Suburbs – http://www.ucl.ac.uk/adaptablesuburbs/) and one at London Metropolitan University (i.e. SECOA – http://www.projectsecoa.eu/), where spatial data is used and/or created by non-GIS specialists and where metadata is extremely important for data documentation, especially giving that under the EPSRC (Engineering and Physical Sciences Research Council) Policy Framework, which covers UK academic work, documenting data used in research projects will be a requirement of the funding in many cases.

To give the internship work a broader relevance, the metadata elements are based on the specifications of the INSPIRE Directive regarding metadata. The INSPIRE metadata standard itself is derived from the international ISO 19115 (Geographic Information – Metadata) standard. It is therefore very relevant to explore the possibility of automating the generation of metadata elements of such a standard, which is already being used by many national mapping agencies across Europe, and will also start to be used by other data providers in the future. Currently, INSPIRE-based metadata is created manually and maintained separate of the datasets it describes. Since the INSPIRE standards are strongly aimed at data providers, some elements of the metadata standard might not be relevant or applicable in an academic research project. Nevertheless, the aim is to use the INSPIRE metadata standard as a foundation, which will then be possible to customise based on the needs of any individual project. Additional metadata elements can be added as required on a project by project basis, and in the same time, elements of the INSPIRE standards can be dropped (i.e. not used) or given default values if and where applicable.

Finally, based on the achieved results, it is also important to discuss the limitations of metadata automation based on the given standard and chosen methodology, as it is quite clear that not all elements will be possible to be automated (e.g. abstract), at least not within the scope of this internship and the timeframe allocated for it.

1.3 Methodology

The main idea about the mechanism behind automating the generation of metadata was clearly established from the beginning by the internship provider. The proposed method has a strong database focus as it involves coding inside a database to track when data is inserted, updated or deleted, and automatically create and update the metadata accordingly. This way, a permanent link between the datasets and metadata is maintained. The proposed database to use is PostgreSQL/PostGIS giving the main advantage of being a free and open source-software (FOSS), and used by more and more GIS packages in their workflow.

Essentially, the methodology to automatically generate metadata within a database is based on a 'trigger system'. A database trigger is a procedural code that is automatically executed in response to certain events on a particular table or view in a database, in this case the events being uploading, deleting, or updating data in a spatial table. Therefore, the procedural code, which in this case is mainly written in PL/pgSQL (procedural language for the PostgreSQL database system), that will be executed as defined by the triggers, will have to create, populate and update the metadata elements as required.

Based on these aspects, methodology is described in an iterative approach taken to design and trigger development, as depicted in the methodology diagram below:

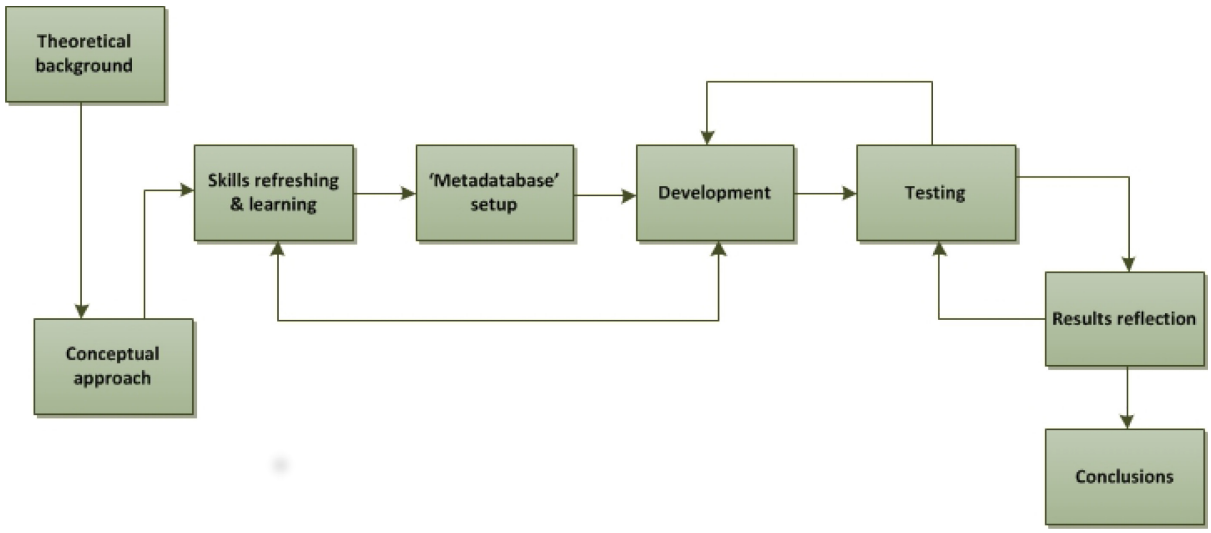


Fig. 1: Methodology diagram

- Theoretical background – in order to get a better insight in the underlying aspects of metadata automation and the methods that have been researched so far, a literature review was performed to have a clear and complete overview of the relevant theories.
- Conceptual approach – possible courses of action are researched before proceeding to the actual implementation stage.
- Implementation – development of the actual metadata automation systems, which has itself distinct work packages:
    - Skills refreshing and learning – it was necessary to utilise general (spatial) database concepts, some of which were also covered during one of the GIMA modules. In addition, generic SQL skills were required (i.e. database definition language, database manipulation language, queries), as well as getting acquainted to more advanced features like database procedural languages, functions, stored procedures, and triggers. The results obtained are nevertheless influenced by this process, as the entire internship period was a very intense learning cycle.
    - 'Metadatabase' setup – based on the studied database functionality and after analysing the INSPIRE metadata standard requirements, a conceptual approach was outlined to identify the possibilities to implement the metadata automation process directly in the PostgreSQL/PostGIS database.
    - Development and testing – the research into, and the creation of the actual (trigger) functions that would populate the different metadata elements, followed concomitantly by a thorough testing to assure a faultless functionality.
- Results discussion – presentation of the achieved results taking into consideration the main limitations, and possibly suggestions for better alternatives. Another aspect discussed here are the capabilities of different open source or commercial tools to automatically generate metadata, and how those capabilities can be compared to the achieved results and possibly, future work.
- Conclusions – a clear link is presented between the proposed objective(s) and results obtained, with the identification of the added value of the internship project.

2. Theoretical background

This chapter explores various relevant theories found in the literature to provide a better insight on the underlying aspects of metadata generation, and especially the concepts and possibilities of automated metadata generation.

Metadata is commonly defined as 'data about data' and it is very important in ensuring that resources are well documented and continue to be accessible in the future (Olfat *et al.*, 2010; NISO, 2004). Nowadays there is an increase of spatial datasets being used, created and exchanged between people or organisations who are not necessarily GIS specialised. This is in contrast to the traditional approach where geographical information was provided top-down by data providers such as National Mapping Agencies (NMA). Given this increase, but also the reduction in expertise of end-users due to increasing interdisciplinary use of GIS, it is very important to have information in the form of metadata, to allow these users to understand and integrate data coming from various sources, and identify any issues, omissions, data capture methods, or previously carried out analysis (Ellul *et al.*, 2012).

Furthermore, metadata plays a very important role in Spatial Data Infrastructure (SDI) initiatives, being one of the very first things that needs to be implemented (i.e. metadata catalogues), partly for the same reasons mentioned above, but also to perform vital functions that make spatial datasets discoverable, interoperable and capable to be shared between different systems (Olfat *et al.*, 2010).

The process of using metadata can be divided into four components as depicted in Figure 2 below. Discovery and evaluation of metadata can be integrated into the wider context of searching for a specific dataset (Winer, 2011).

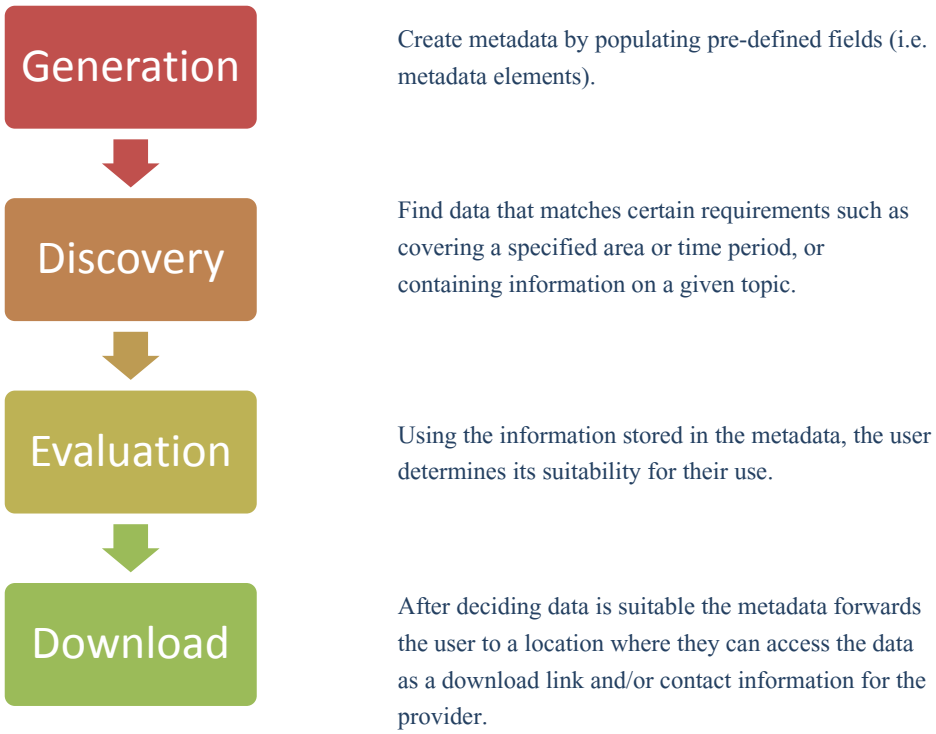


Fig. 2: The metadata process (after Winer, 2011)

The most important stage, which is also the main aspect under the scope of the internship project, is the metadata generation. According to Olfat *et al.* (2010), the generation of geographic information metadata can be separated into automatic (computerisation methods), semi-automatic (automatic and manual methods), and manual methods (human reasoning and decision-making). Moreover, these approaches were formed and have evolved based on technological initiatives over time as can be depicted in Figure 3 below.

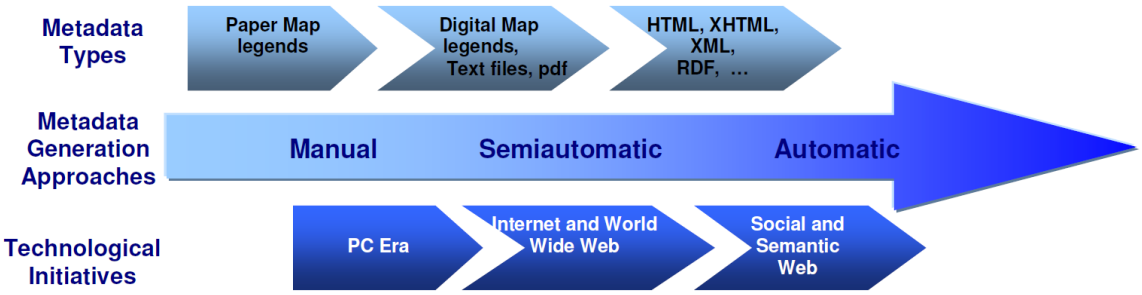


Fig. 3: Geographic information metadata generation approaches, types, and technological initiatives (Olfat et al., 2010)

The current manual and semi-automatic methods for metadata generation are considered monotonous, time-consuming, and labour-intensive (Olfat *et al.*, 2012a). Even though, nowadays, metadata types have evolved and generally the XML format is used, the generation methods are still manual in most cases, aided by various input tools. Moreover, the existent semi-automatic approaches only extract some of the metadata elements related to spatial references (e.g. extent, coordinate reference system) from the datasets, with non-spatial elements still being added manually by an operator who is very often unaware of the dataset lifecycle. This will lead in most cases to the metadata being incomplete, incorrect, imprecise, and in some cases, even missing. Another important observation related to the current manual, and in some cases semi-automatic metadata generation, is that metadata is commonly created and stored separately from the actual spatial datasets it relates to, and INSPIRE specifications is a very representative example in this sense. This separation of storage means that the metadata and the spatial datasets it represents have to be managed and updated separately; this will inevitably lead to redundancy and inconsistency (Olfat *et al.,* 2012b).

Kalantari *et al.* (2009) introduces a framework for automated metadata generation which separates the metadata generation into three components as illustrated in Figure 4.

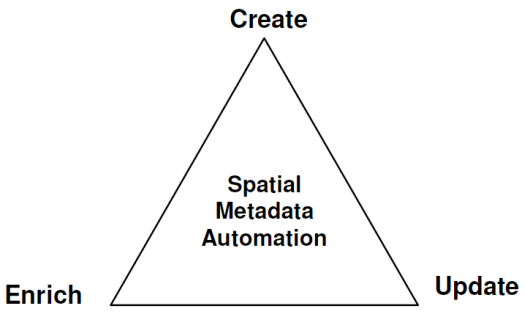


Fig 4: Metadata automation framework (Kalantari et al., 2009)

Automatic creation refers to the initial creation of metadata when there is no existing metadata associated with a dataset, while automatic enrichment involves improving the content of the metadata through monitoring tags that are used by users for finding datasets. Finally, automatic updating will update the metadata at the same time with its related spatial dataset update process.

On the other hand, Ellul *et al.* (2012) identified two main approaches to metadata automation and several attempts for each approach. First approach is to automate data quality assessments and therefore populate metadata elements based on the results. So far, this has been attempted by comparing the datasets with 'better/higer' quality datasets (Koukoletsos *et al.*, 2011), through modelling (de Bruin, 2008), and thorugh examining the different values of nominal, ordinal, ratio, and interval data (van Oort, 2006). The second approach, and probably more relevant, is through direct automated metadata creation. Attempts in this case include harvesting existing metadata (Batcheller, 2008), automated tagging (Kalantari *et al.*, 2010), title and location information extraction (Olfat *et al.*, 2010), and format, number and types of geometry, resolution, bounding box, use constraints (Manso-Callejo *et al.*, 2009). However, certain metadata elements cannot be completely eliminated and still require user input (e.g. abstracts). For example, Winer (2011) has attempted an automated metadata extraction system but this was limited to ESRI Shapefiles and only extracting the bounding box and projection.

One important limitation to most of the direct automated metadata extraction mentioned above is the tie to a specific software package or a specific data format. This may prove to be commercially unsustainable and too dependent on certain technologies. There is a need therefore for a more independent and transparent approach, and one that is easier to integrate with web-based applications/services, as this seems to be the current trend in what spatial data is concerned. Ellul *et al.* (2012) also concludes that generating and storing metadata with the data in an integrated single environment such as a spatial database would greatly assist in having metadata generated and maintained directly into the user's workflow. It would also be possible to automatically update the metadata when the underlying dataset is edited/modified. Some spatial databases also allow the logging of GIS operations, while some 'extreme' functionality might include voice recording and transcription services to facilitate the population of elements that are almost impossible to automate, like abstract and title. It is also important to note that a database approach is independent of any GIS software package or file format.

In general, when considering ready to use standard-based metadata generation solutions, there are several options, and some of them also automatically generate some of the elements. One example in this sense is the ArcCatalog (part of the ESRI ArcGIS package), which can organise its internal metadata editor based on various standards, INSPIRE being one of them. Several elements can be automatically generated, like bounding box and lineage for instance. However, this solution is not as complete as the database trigger mechanism developed, and in addition, the same limitation of proprietary software and file format applies.

3. Implementation

This chapter will present the main steps taken to develop the automated metadata system. Section 3.1 briefly describes the pre-development aspects and used software, while Section 3.2 goes through the selected metadata standard and its documentation. Section 3.3 describes the base setup of the metadata 'repository', and finally, Section 3.4 describes the main development aspects, the actual trigger mechanism that automatically generates the metadata.

3.1 Preparation and setup

As already mentioned the database that was used is PostgreSQL (version 9.1.4) with its spatial extension PostGIS (version 2.0.1). An Amazon EC2 (Elastic Compute Cloud) server was provided running Linux (Amazon Linux AMI), where the database had to be installed. This itself was a challenging task as some generic Linux command line skills had to be employed. In addition, there were many 'tweaks' that had to be discovered and performed in order to get PostgreSQL and PostGIS installed on the specific Linux system, which is very different to the smooth installation on a Windows system via an executable file for example. These tweaks were mainly related to various packages and plugins that needed to be installed before PostgreSQL/PostGIS could be installed (e.g. GEOS, PROJ, GDAL, LIBJSON, LIBXML, etc.) but also some PostgreSQL installation files that had to be edited after the installation to activate certain functionality like making the database accessible from an external IP address.

The software used to access the server was mainly PuTTY (release 0.62), a free and open source terminal emulator application, which can act, among others, as a client for the SSH computing protocol, an option used for connecting to the server. In addition, WinSCP (version 4.3.7) a free and open source SFTP, SCP, and FTP client was used to securely perform file transfers between the local machine and the server. Finally, pgAdmin III (version 1.14.2) was used as the main database design and management system due to its friendlier graphical interface. Besides the server and database administration tools, Quantum GIS (version 1.7.4) was used for front-end testing.

3.2 Metadata standard

The metadata standard that was used, as stated and explained in the first chapter, was the INSPIRE standard, which itself is based on the ISO 19115 (Geographic Information – Metadata) standard. This standard is based on the 'Commission Regulation (EC) No 1205/2008 of 3 December 2008 implementing Directive 2007/2/EC of the European Parliament and of the Council as regards metadata' and on the technical guidance document 'INSPIRE Metadata Implementing Rules: Technical Guidelines based on EN ISO 19115 and EN ISO 19119' (version 1.2). Table 1 below contains the metadata elements list applicable for spatial datasets. For spatial data services (e.g. WMS, WFS) this list is slightly different. The metadata elements are grouped in 10 categories according to the standard. Annex A of this report contains a more complete description of each element and sub-elements where applicable (e.g. identifier, bounding box), as well as code list values of certain elements (e.g. topic category).

Table 1: INSPIRE metadata elements for spatial datasets

1. Identification	6. Quality and validity
1.1 Resource title	6.1 Lineage
1.2 Resource abstract	6.2 Spatial resolution (optional)
1.3 Resource type	7. Conformity
1.4 Resource locator (optional)	7.1 Degree
1.5 Unique resource identifier	7.2 Specification
1.6 Resource language (optional)	8. Constraints related to access and use
2. Classification of spatial data and services	8.1 Limitations on public access
2.1 Topic category	8.2 Conditions for access and use
3. Keyword	9. Responsible organisation
3.1 Keyword value	9.1 Responsible party
3.2 Originating controlled vocabulary (optional)	9.2 Responsible party role
4. Geographic location	10. Metadata on metadata
4.1 Geographic bounding box	10.1 Metadata point of contact
5. Temporal reference (at least one element)	10.2 Metadata date
5.1 Temporal extent (optional)	10.3 Metadata language
5.2 Date of publication (optional)	
5.3 Date of last revision (optional)	
5.4 Date of creation (optional)	

3.3 Metadata setup in the database

After a thorough study of the INSPIRE documentation to develop an understanding the requirements of each of the metadata elements, a conceptual approach was taken to establish how the metadata records will be established in the database and how the main mechanism of triggers will operate. After researching and trying to implement several alternatives, the conclusion was that the best approach is to have a standalone metadata table, where the columns would represent the different metadata elements and each table record would stand as a metadata record describing one dataset.

As some of the metadata elements are composed of several sub-elements, (e.g. identifier is composed of 'code' and a 'namespace'), the first approach attempted was to create custom composite data types in the database. Using this approach, the identifier column has a composite type formed by two sub-columns (i.e. code and namespace). Unfortunately, although this is implementable in the database, third-party applications like Quantum GIS do not recognise such 'composed column' types. Therefore, each element/sub-element had to have allocated its own column. Annex B of this report contains the complete SQL code for the creation of the metadata table. As also mentioned in the first chapter, the INSPIRE metadata elements are used as the base elements to which additional elements can be added as needed by the various projects where the automated metadata system might be used. One of these elements a geometric representation of

the bounding box, in addition to the descriptive information containing the maximum and minimum latitude and longitude. Therefore the metadata table becomes a spatial dataset containing the bounding boxes of the various datasets uploaded to the database. This feature is extremely useful, especially for non-GIS specialists, as it allows better understanding of data by visualising its location in addition to the descriptive information that the metadata table contains. Additional metadata elements implemented will be discussed in the next section.

3.4 Metadata elements

This section will briefly describe the development of the trigger functions that were created to populate and update the various metadata elements.

3.4.1 Metadata records creation

The first element of required functionality was to monitor when new (spatial) tables are added to the database, and create a new (empty) record in the metadata table for that dataset. The only way to truly do this automatically would be to create a trigger on a certain 'index' table (that would keep a record of all tables in the database). Such an 'index' table exists in the PostgreSQL system catalogs, but it is not possible to create triggers on the tables stored in these catalogs. Another possibility would have been to create a table view[1] based on the mentioned catalog table, but the trigger functionality on such a view (table) is limited. After further research and prototyping, the implemented solution was to 'manually' (or by means of an external operation) insert a record in the metadata table containing the content of at least one column. By using this approach, besides having a new metadata record created when a new spatial dataset is inserted in the database, the system can also trigger the other functions that were due to be created to populate the other metadata elements.

This external metadata record insertion operation provides an ideal solution due to the ability to implement it as part of another project. The front-end functionality comes out of an MSc thesis of another student related to uploading spatial datasets to the database. This functionality is based on the development of a Quantum GIS plugin that would manage the PostgreSQL/PostGIS database (i.e. mainly uploading data to the database), by taking into consideration the automated metadata functionality of the database. This provides maximum benefit to the non-GIS specialist who does not need to worry about creating metadata manually.

Therefore, it was decided that the plugin (still under development at this time) would incorporate the required hidden functionality. Besides uploading the actual dataset to the database, it would also insert a new record in the metadata table by populating the 'dataset_name' element with the actual name of the spatial dataset file being uploaded (i.e. in case of uploading a shapefile 'uk_counties.shp', the value inserted would be 'uk_counties'). The 'dataset_name' is an additional metadata element that was implemented for exactly this reason. In addition, the plugin will also incorporate in its user interface the manual insertion of other metadata elements that proved to be impossible to automate. For general testing purposes of the trigger functions, until the plugin is developed, the 'dataset_name' element was inserted manually by SQL commands.

[1] In database theory, a view consists of a stored query accessible as a virtual table in a relational database.

3.4.2 Resource title

The title of the spatial dataset is an element that is difficult to automate; therefore it should be integrated within the Quantum GIS plugin to be manually typed by the user uploading the dataset to the database. However, typing the 'resource title' would be optional. In case it is not provided by the user, the database was programmed to assign a default value to this element, equivalent to the value of 'dataset_name', which is inserted as explained earlier without the user knowing. Therefore, it could be said there is a certain degree of automation to the 'resource title' element as well.

3.4.3 Resource abstract

This element is practically impossible to automate and will be integrated in the Quantum GIS plugin to be filled in by the user.

3.4.4 Resource type

According to the standard the resource type is a code list value, and there is a single value that covers all spatial datasets: 'spatial dataset'. Other values refer to spatial data services and spatial dataset series. Therefore, the database was programmed to implement a default value (i.e. 'spatial dataset') for each metadata record in the 'resource type' column.

3.4.5 Resource locator

This element is optional and should provide a URL, if available, to obtain more information and/or to the resource. Due to it being optional, it was not tackled as it has been established that mandatory elements have priority. However, automation of this element has been considered and will be discussed in the recommendations section.

3.4.6 Unique resource identifier

According to the INSPIRE documentation, this element is composed of two components: namespace and code. Therefore, in the metadata table is represented by two different columns. The namespace represents the value domain of the dataset (assigned by the data owner) while the code uniquely identifies the metadata record/dataset in the context of the value domain. Therefore, the database was programmed to assign a default value to the namespace: ucl.ac.uk_CEGE_metadata (i.e. UCL's web domain + department of Civil, Environmental and Geomatic Engineering acronym + metadata). Populating the code component has a more complex function behind it. It looks in the PostgreSQL catalog tables and fetches the system unique object identifier assigned by PostgreSQL to all tables in a database. This assures the uniqueness of the identifier value and in addition, it links each metadata record to the spatial dataset/table it represents.

3.4.7 Resource language

This is an optional element, being conditioned by the existence of textual information in the resource descriptive (i.e. attributes) information. This element is similar to 'metadata language', which is mandatory, and could be implemented using the same approach (described in the metadata language section). However, there are certain differences that might complicate the automation of 'resource language'. This is discussed further in the results section.

3.4.8 Topic category

According to the standard, this element is based on given code list values. More than one value can be assigned from the code list if the dataset can be assigned to multiple categories. As this element would be difficult to automate, especially giving that the values it can take are fixed, it has been

decided to incorporate the topic categories specified in the code list in the Quantum GIS plugin as a multi-selection list where the user can pick the categories he thinks the data falls into. Of course, these are then inserted in the metadata table once the dataset is uploaded to the database.

3.4.9 Keyword value

This was probably the most difficult metadata element to automate as significant effort was spent trying to create the trigger function. In principle, the methodology behind automating this element was to scan through all character string attributes/columns of a dataset/spatial table, merge all of them into one column, and separate the words in different rows. After that, the top occurring words would have been considered as keywords. Unfortunately, due to some limitations in the pgSQL procedural language, certain functionality needed to build the function was not possible to implement. This will be researched further in the remaining weeks of the internship. Keywords are very important because they open many other options for additional metadata elements that might be implemented (e.g. word clouds).

Optionally, if the keyword originates from a controlled vocabulary (thesaurus, ontology), this has to be provided as part of an additional sub-element of the of keyword category. In addition a reference date (i.e. creation, revision, publication) of the controlled vocabulary can be provided. This aspect was not tackled as keywords generated by the above explained methodology do not have any links to controlled vocabularies. However, it might be possible to compare the generated keywords to keywords from controlled vocabularies, and when there is a match, the controlled vocabulary details could also be automatically inserted in the metadata record.

3.4.10 Geographic bounding box

The geographic bounding box is expressed with westbound and eastbound longitudes, and southbound and northbound latitudes in decimal degrees. The metadata table holds a separate column for each of these elements. Here, the trigger function created to populate these columns makes use of some of the PostGIS spatial functions that calculate the minimum and maximum latitudes and longitudes. Whenever the data is edited directly in the database and the spatial extent modified, the coordinates update accordingly in the metadata table. This requires a more complex trigger function that creates itself a new trigger function for every spatial table added to the database, which monitors any changes happening on the respective table, and sends updates back to the metadata table when applicable.

3.4.11 Temporal reference

As part of the temporal reference categories there are four elements defined (i.e. temporal extent, date of creation, date of last revision, date of publication) and at least one has to be provided. The choice was to automate ‘date of last revision’. This date is taken as the date of the upload of the dataset to the database, and thereafter, the date on which any edits are made on the dataset.

3.4.12 Lineage

As the INSPIRE documentation defines the content of this element as a ‘statement’ of ‘free text’, similar to the ‘resource abstract’ for that matter, it has been concluded that it is not possible to automate and will be integrated in the Quantum GIS plugin as a field the user can fill in, based on his knowledge of the dataset lineage. Alternatives are further discussed in the results discussion chapter.

3.4.13 Spatial resolution

This is an optional element, which was not tackled and automated. However, an approach to automate this element is discussed in the results reflection section.

3.4.14 Conformity

In the context of INSPIRE, datasets have to be conformant to certain data specifications. This category aims to establish conformity to the specifications by two elements: 'degree' and 'specification'. While 'degree' is of Boolean value and simply states if the dataset is conformant to a specification or not, 'specification' has to contain the title and a reference date of the specification. These elements are completely out of the scope of the data that might make use of the metadata automation system, as this data will be generated by academic researchers. Therefore 'degree' is programmed to have a default value of 'notConformant', while 'specification' is left empty.

3.4.15 Conditions for access and use / Limitations on public access

These are very 'data provider' oriented metadata elements. It is obviously hard, if not impossible to concretely automate such elements, especially giving that, as the 'resource abstract' and 'lineage', they are supposed to be filled with a 'statement of free text'. However, there are several code list values like options, which can be assigned by default for every new metadata record. The most neutral ones and the ones that were also implemented as default values are 'Conditions unknown' and 'No limitations'. Of course, these may be highly dependent on the type of data that is uploaded to the database and can, and should be changed accordingly by the user.

3.4.16 Responsible organisation

The description of the responsible organisation shall include the name of the organisation as well as a contact email address. These elements were automated based on the different database roles (i.e. accounts). A role can be thought of as either a database user, or a group of database users, depending on how the role is set up (http://www.postgresql.org).

Therefore, whenever the database-based automated metadata system is implemented within a project, database groups and roles are set up based on the users that will have access to the database. Database groups will be assigned to the different organisations while individual roles will be assigned to individual users. The PostgreSQL catalog tables identify the database roles and groups by unique identifiers, which have been used in a lookup table. The lookup table is manually populated by the database administrator when new roles and groups are created with the organisation name and contact email address, while a trigger function automatically populates the records with the PostgreSQL group identifier based on the group name. Based on this information, the metadata table has been programmed to automatically fetch the responsible organisation details, based on the user that uploads data to the database and the group he is assigned to.

Another element that is provided within the responsible organisation metadata category is the 'responsible party role', which is based on a code list that covers most of the situations a given organisation can find itself in relation to the data. The value of this element has been set by 'default' to 'User' (i.e. party who uses the resource), as this role would be the most applicable (and neutral) within an academic project where the used data usually originates from various sources.

3.4.17 Metadata point of contact / Metadata date

The metadata point of contact has been automated the same way as the 'responsible organisation', since it contains the same sub-elements (i.e. organisation name and contact email address), and in principle it will also contain the same values, as most of the metadata elements are created automatically, and some added manually by the same user when data is uploaded to the database.

Metadata date is automatically filled with the date when the dataset has been uploaded to the database and the metadata record created, and updated when metadata is modified.

3.4.18 Metadata language

To automate the metadata language element, an open source language identification module for Perl was used. This is a different approach from all the other elements because it uses PL/Perl procedural language (allows to write PostgreSQL functions in the Perl programming language) instead of PL/pgSQL. The module was customised to receive as an input the text from the 'resource abstract' element, which should normally be long enough to allow the detection of the language. The output is an acronym of the identified language (i.e. English – ENG) as required by the metadata standard.

3.4.19 Additional elements

As presented in the previous chapters there are several metadata elements that have been proposed to be added to the standard, as needed by the two projects the system will be used on (i.e. Adaptable Suburbs & SECOA).

The bounding box of each dataset has been implemented as an actual geometry in the metadata table, which becomes an actual spatial dataset that can be imported in any GIS application. This allows for a very useful map visualisation of the metadata and consequently, the stored data in the database. The geometry of the metadata table is also automatically updateable when datasets are edited directly in the database, using a similar approach as the update functionality implemented in the regular bounding box metadata elements (i.e. latitude/longitude coordinates).

A rating system has also been implemented, allowing the user to rate the quality of both the metadata and data, from the front-end (i.e. Quantum GIS plugin). This is implemented by means of lookup tables that are programmed to receive certain rating scale values, which are then translated into a description of the rating inserted in the metadata table.

Another element that is still under development is creating and storing in the metadata table word cloud images based on the keywords count. This would be a very useful feature as it allows a very visual description of the data. As soon as the described methodology for generating keywords is fully implemented, the word cloud generation will also be attempted in the remaining weeks.

Fig. 5: Word cloud example
(https://wiki.carleton.edu/download/attachments/12226100/uni_tag_cloud_wordle.png)

4. Results Discussion

This chapter will reflect upon the results obtained for the database-based automated metadata system, especially reviewing issues encountered and their impact on the project and the concept of metadata automation. Alternative solutions are also provided for some of the metadata elements, and recommendations are made for further work.

First of all, the INSPIRE metadata standard has been especially created for data providers which makes automation of many elements difficult as they are based on information that is almost impossible to extract from the dataset itself in an independent environment like a database, and most of the times that kind of information is only held by the data providers themselves (i.e. date of creation, date of publication, etc.). Despite these facts, 15 of the 18 mandatory elements were automated.

In addition, the entire metadata automation mechanism is highly dependent on a parallel project as described in the previous chapter, which involves developing a Quantum GIS plugin to load datasets into the PostgreSQL/PostGIS database, and which incorporates several functionalities to aid and complete the metadata generation. The two projects were planned from the beginning to merge at a certain stage and to deliver a unique tool for spatial data management (database and metadata wise) that would especially help the non-GIS specialists. The Quantum GIS plugin is still under development and will be fully tested with the database trigger mechanism in the remaining weeks of the internship. However, no major issues are expected as the trigger mechanism was thoroughly tested by manually performing the SQL queries that will be integrated in the Quantum GIS plugin.

As it has been expected, not all mandatory metadata elements were possible to be automated based on the database trigger mechanism approach. Elements such as abstract and lineage are maybe the most representative in that sense. Since these are mandatory elements in the standard, they will be integrated (together with the 'topic category' code list values) in the Quantum GIS plugin for the user to fill in upon loading the dataset in the database.

Some of the metadata elements were assigned default values (originated in the code lists provided by the standard) because these values would reflect best the data used in an academic research project. However these values cannot be considered fixed at all time and should be changed accordingly. The default values can also vary based on the responsible organisation, if it's known that a certain organisation has a fixed relation to the data it uploads to the database. This can be implemented similarly to the 'responsible organisation' contact details; therefore there would be a different default value with each different database group. Nevertheless, this is a limitation as default values, even if variable by organisation, cannot be applied in all cases.

Overall, the obtained results are promising and they definitely provide a solid base for further work. In this sense it is important to re-emphasize the scope of the standard-based automated metadata system, and that is to be mainly used in academic research projects. With further research of the database trigger mechanism, the functionality and applicability can definitely be increased. Moreover, to enforce the fact that this approach is feasible for further work and can turn into a completely compliant solution, it is important to note that PostgreSQL for instance has native functionality of mapping tables to XML records. As XML is the traditional file format for spatial

metadata, the metadata table maintained in the database can be used to generate compliant XML metadata documents whenever needed.

4.1 Recommendations

Although some of the mandatory and many of the optional metadata elements have not yet been implemented, there have been several ideas (at a conceptual level at least) to automate these elements, some of which will be researched further for the remaining weeks of the internship. 'Resource locator' for instance could be implemented by further extending the functionality of the Quantum GIS plugin. Therefore, whenever data is uploaded to the database via the plugin, the original data file can be zipped (i.e. archived) and uploaded to a web server where it would be accessible to the project team by a URL. This URL would then be inserted in the metadata table as the 'resource locator' value.

'Lineage' does have several alternatives as well to be automated using other approaches. For instance, Oracle provides a logging functionality that creates copies of the datasets at specified intervals of time or every time the dataset is edited/processed. ArcGIS also has a similar functionality by storing a log of all performed operations on a dataset. The disadvantage of both these methods though, is that they rely on certain software (i.e. ArcGIS, Oracle) or file formats (i.e. shapefiles in case of ArcGIS), and on top of that, on commercial software. As already mentioned in this report, the idea is to have a platform as independent as possible from any file format or software package. Alternatively, a voice recording and transcription service could be implemented in the front-end, to allow users to dictate the text to be inserted in the 'lineage' element.

'Resource language' on the other hand could be automated using the same Perl language identification module that was used for 'metadata language'. The only problem here would be that the Perl module needs to have as input a character string column that contains text which is 'coherent' enough have its language identified. In the case of 'metadata language' the 'resource abstract' satisfies that requirement as it is always structured as a statement of at least one phrase. Moreover, the input column for 'resource language' would always be different. One solution would be to concatenate all character string columns from a dataset (approach similar to the automation of keywords generation) and provide that as input to the language identification module. It is then a matter of testing the efficiency of such an approach, and of course this is highly dependable on the availability and actual content of the character string columns of the dataset.

'Spatial resolution', another optional element that was not automated, refers to the level of detail of the dataset. This is a very difficult to automate element for vector data, the only solution that was considered is to set up a function that compares the geometry of the dataset with several geometries of base topographic at different scales, and then try to match or give a scale range based on the comparison.

5. Conclusion

The internship project aimed to implement an automated metadata system based on a database trigger chain mechanism. To increase the relevance of the project the metadata structure was based on the INSPIRE framework requirements, this being one of the most dominant initiatives in the geographic information domain in the last few years, and which will receive a constantly increasing attention in the years to come. Despite using the INSPIRE standard for metadata, the main scope was to aid the GIS management workflow of academic research projects where non-GIS specialists that make use of, and create spatial data, are involved. Therefore, the structure of the metadata has been extended with additional elements as required by the two projects it has been designed for (i.e. SECOA & Adaptable Suburbs).

It has been demonstrated that metadata can definitely be automated, independently from any software or file format, directly in a spatial database, by developing trigger functions that generate and update the metadata content. However, there are certain limitations to this approach that have been encountered during the course of the internship. Nevertheless, some of these limitations can be overcome with further work, as the achieved results during the internship timeframe have covered and successfully automated many of the metadata elements.

The main challenge was the metadata standard itself. Most standard-based metadata elements are very 'data provider' oriented and some of the required information is sometimes held only with data providers and not possible to be automatically extracted from the dataset itself. In addition, based on the standard, some of the metadata elements only accept code list values, which is again, very complicated to automate. Finally, as was expected, elements that should contain 'free text statements', like abstracts, were also not possible to automate. However, certain approaches have been proposed that could automate to a certain extent these elements, like an embedded voice recording and transcription service. Of course, these are solutions that would particularly benefit the non-GIS specialists.

It is clear that not all metadata elements can be automated by only using the trigger mechanism approach in the database, but it is certain that many other technologies can be incorporated either in the back-end (i.e. database) and/or in the front-end (i.e. desktop/web GIS application) to automate most, if not all, of the remaining elements, and still keep software and/or file format independence. This can be achieved primarily by integrating any front-end functionality trough the Python programming language, which is becoming a native programming language to an increasing number of GIS applications (ArcGIS included). The question that remains is whether a system like this will be embraced by software providers. To answer this question, further research into the problem is needed, and besides implementing a complete automation of metadata (at least for the mandatory elements), a very detailed testing has to be performed, as well as a business case analysis taking into consideration the efficiency in terms of a cost-benefit analysis. Nevertheless, for academic research projects, where a metadata standard doesn't necessarily have to be as formal and strict, the already developed functionality during the internship project can be implemented and used to relief the (non-)GIS specialists of the metadata creation task, considered to be a real burden.

References

Batcheller, J. (2008), *Automating Geospatial Metadata Generation – An Integrated Data Management and Documentation Approach*. Computers & Geosciences, 34: 287-398.

De Bruin, S. (2008), *Modelling Positional Uncertainty of Line Features for Stochastic Deviations from Straight Line Segments*. Transactions in GIS, 12(2), pp. 165-177.

Drafting Team 'Metadata', (2010), *INSPIRE Metadata Implementing Rules: Technical Guidelines based on EN ISO 19115 and EN ISO 19119 – Version 1.2*. Available at http://inspire-.jrc.ec.europa.eu/documents/Metadata/INSPIRE_MD_IR_and_ISO_v1_2_20100616.pdf

Ellul, C., Winer, D., Mooney, J., Foord, J. (2012), *Bridging the Gap between Traditional Metadata and the Requirements of an Academic SDI for Interdisciplinary Research*. Proceedings of GSDI 13 World Conference: Spatially Enabling Government, Industry and Citizens (14-17 May), Québec City, Canada.

European Commission, (2008), *Commission Regulation (EC) No 1205/2008 of 3 December 2008 implementing Directive 2007/2/EC of the European Parliament and of the Council as regards metadata.* Available at http://eur-lex.europa.eu/LexUriServ/LexUriServ.do?uri=CELEX:-32008R1205:EN:NOT

Kalantari, M., Rajabifard, A., Olfat, H. (2009), *Spatial Metadata Automation: A New Approach*. In Ostendorf B., Baldock, P., Bruce, D., Burdett, M. and P. Corcoran (eds.), Proceedings of the Surveying & Spatial Sciences Institute Biennial International Conference, Adelaide, Surveying & Spatial Sciences Institute, pp. 629-635.

Kalantari, M., Olfat, H., Rajabifard, A. (2010), *Automatic Spatial Metadata Enrichment: Reducing Metadata Creation Burden through Spatial Folksonomies*. GSDI 12 World Conference: Realising Spatially Enabled Societies, Singapore.

Koukoletsos, T., Haklay, M., and Ellul, C. (2011), *An automated method to assess Data Completeness and Positional Accuracy of OpenStreetMap*. Proceedings of the GIS Research UK 19th Annual Conference GISRUK 2011, University of Portsmouth, Portsmouth.

Manso-Callejo, M.A., Wachowicz, M., and Bernabé-Poveda, A. (2009), *Automatic Metadata Creation for Supporting Interoperability Levels of Spatial Data Infrastructures*. GSDI 11 World Conference – Spatial Data Infrastructure Convergence: Building SDI Bridges to Address Global Challenges, Rotterdam (15-19 June), The Netherlands.

NISO (2004), *Understanding Metadata.* National Information Standards Organisation, USA. Available at http://www.niso.org/publications/press/UnderstandingMetadata.pdf

Olfat, H., Kalantari, M., Rajabifard, A., Senot, H., Williamson, I. (2012a), *Spatial Metadata Automation: A Key to Spatially Enabling Platform*. Proceedings of GSDI 13 World Conference: Spatially Enabling Government, Industry and Citizens (14-17 May), Québec City, Canada.

Olfat, H., Kalantari, M., Rajabifard, A., Williamson, I. (2012b), *Towards a Foundation for Spatial Metadata Automation*. Journal of Spatial Science, 57(1), pp. 65-81.

Olfat, H., Rajabifard, A., Kalantari, M. (2010), *Automatic Spatial Metadata Update: a New Approach*. XXIV FIG International Congress (11-16 April), Sydney, Australia.

Van Oort, P. (2005), *Spatial Data Quality: From Description to Application*. PhD thesis, Wageningen University, The Netherlands.

Winer, D. (2011), *Making Spatial Metadata Easier for Inexperienced Users – Automated Extraction and Improved Search*. MSc Thesis, University College London, London, United Kingdom.

Websites

Adaptable Suburbs – http://www.ucl.ac.uk/adaptablesuburbs/

INSPIRE – http://inspire.jrc.ec.europa.eu/

PostgreSQL – http://www.postgresql.org

SECOA – http://www.projectsecoa.eu/

Annex A – INSPIRE Metadata requirements

NOTE 1: The metadata implementing rules in this Annex are based on the *'Commission Regulation (EC) No 1205/2008 of 3 December 2008 implementing Directive 2007/2/EC of the European Parliament and of the Council as regards metadata'*.

NOTE 2: Directive 2007/2/EC is referred to in this Annex represents *'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://eur-lex.europa.eu/LexUriServ/LexUriServ.do?uri=CELEX:32007L0002:EN:NOT

I. METADATA ELEMENTS

1. IDENTIFICATION

The following metadata elements shall be provided:

1.1. Resource title

This a characteristic, and often unique, name by which the resource is known.

The value domain of this metadata element is free text.

1.2. Resource abstract

This is a brief narrative summary of the content of the resource.

The value domain of this metadata element is free text.

1.3. Resource type

This is the type of resource being described by the metadata.

The value domain of this metadata element is defined in part II.1 of this Annex.

1.4. Resource locator

The resource locator defines the link(s) to the resource and/or the link to additional information about the resource.

The value domain of this metadata element is a character string, commonly expressed as uniform resource locator (URL).

1.5. Unique resource identifier

A value uniquely identifying the resource.

The value domain of this metadata element is a mandatory character string code, generally assigned by the data owner, and a character string namespace uniquely identifying the context of the identifier code (for example, the data owner).

1.6. Resource language

The language(s) used within the resource.

The value domain of this metadata element is limited to the languages defined in ISO 639-2.

2. CLASSIFICATION OF SPATIAL DATA AND SERVICES

2.1. Topic category

The topic category is a high-level classification scheme to assist in the grouping and topic-based search of available spatial data resources.

The value domain of this metadata element is defined in part II.2 of this Annex.

3. KEYWORD

If a resource is a spatial data set or spatial data set series, at least one keyword shall be provided from the general environmental multilingual thesaurus (GEMET) describing the relevant spatial data theme as defined in Annex I, II or III to Directive 2007/2/EC.

For each keyword, the following metadata elements shall be provided:

3.1. Keyword value

The keyword value is a commonly used word, formalised word or phrase used to describe the subject. While the topic category is too coarse for detailed queries, keywords help narrowing a full text search and they allow for structured keyword search.

The value domain of this metadata element is free text.

3.2. Originating controlled vocabulary

If the keyword value originates from a controlled vocabulary (thesaurus, ontology), for example GEMET, the citation of the originating controlled vocabulary shall be provided.

This citation shall include at least the title and a reference date (date of publication, date of last revision or of creation) of the originating controlled vocabulary.

4. GEOGRAPHIC LOCATION

The requirement for geographic location referred to in Article 11(2)(e) of Directive 2007/2/EC shall be expressed with the metadata element geographic bounding box.

4.1. Geographic bounding box

This is the extent of the resource in the geographic space, given as a bounding box.

The bounding box shall be expressed with westbound and eastbound longitudes, and southbound and northbound latitudes in decimal degrees, with a precision of at least two decimals.

5. TEMPORAL REFERENCE

This metadata element addresses the requirement to have information on the temporal dimension of the data as referred to in Article 8(2)(d) of Directive 2007/2/EC. At least one of the metadata elements referred to in points 5.1 to 5.4 shall be provided.

The value domain of the metadata elements referred to in points 5.1 to 5.4 is a set of dates. Each date shall refer to a temporal reference system and shall be expressed in a form compatible with that system. The default reference system shall be the Gregorian calendar, with dates expressed in accordance with ISO 8601.

5.1. Temporal extent

The temporal extent defines the time period covered by the content of the resource. This time period may be expressed as any of the following:

— an individual date,

— an interval of dates expressed through the starting date and end date of the interval,

— a mix of individual dates and intervals of dates.

5.2. Date of publication

This is the date of publication of the resource when available, or the date of entry into force. There may be more than one date of publication.

5.3. Date of last revision

This is the date of last revision of the resource, if the resource has been revised. There shall not be more than one date of last revision.

5.4. Date of creation

This is the date of creation of the resource. There shall not be more than one date of creation.

6. QUALITY AND VALIDITY

The requirements referred to in Article 5(2) and Article 11(2) of Directive 2007/2/EC relating to the quality and validity of spatial data shall be addressed by the following metadata elements:

6.1. Lineage

This is a statement on process history and/or overall quality of the spatial data set. Where appropriate it may include a statement whether the data set has been validated or quality assured, whether it is the official version (if multiple versions exist), and whether it has legal validity.

The value domain of this metadata element is free text.

6.2. Spatial resolution

Spatial resolution refers to the level of detail of the data set. It shall be expressed as a set of zero to many resolution distances (typically for gridded data and imagery-derived products) or equivalent scales (typically for maps or map-derived products).

An equivalent scale is generally expressed as an integer value expressing the scale denominator.

A resolution distance shall be expressed as a numerical value associated with a unit of length.

7. CONFORMITY

The requirements referred to in Article 5(2)(a) and Article 11(2)(d) of Directive 2007/2/EC relating to the conformity, and the degree of conformity, with implementing rules adopted under Article 7(1) of Directive 2007/2/EC shall be addressed by the following metadata elements:

7.1. Specification

This is a citation of the implementing rules adopted under Article 7(1) of Directive 2007/2/EC or other specification to which a particular resource conforms.

A resource may conform to more than one implementing rules adopted under Article 7(1) of Directive 2007/2/EC or other specification.

This citation shall include at least the title and a reference date (date of publication, date of last revision or of creation) of the implementing rules adopted under Article 7(1) of Directive 2007/2/EC or of the specification.

7.2. Degree

This is the degree of conformity of the resource to the implementing rules adopted under Article 7(1) of Directive 2007/2/EC or other specification.

The value domain of this metadata element is defined in part II.3 of this Annex.

8. CONSTRAINT RELATED TO ACCESS AND USE

A constraint related to access and use shall be either or both of the following:

— a set of conditions applying to access and use (8.1),

— a set of limitations on public access (8.2).

8.1. Conditions applying to access and use

This metadata element defines the conditions for access and use of spatial data sets and services, and where applicable, corresponding fees as required by Article 5(2)(b) and Article 11(2)(f) of Directive 2007/2/EC.

The value domain of this metadata element is free text.

The element must have values. If no conditions apply to the access and use of the resource, 'no conditions apply' shall be used. If conditions are unknown, 'conditions unknown' shall be used.

This element shall also provide information on any fees necessary to access and use the resource, if applicable, or refer to a uniform resource locator (URL) where information on fees is available.

8.2. Limitations on public access

When Member States limit public access to spatial data sets and spatial data services under Article 13 of Directive 2007/2/EC, this metadata element shall provide information on the limitations and the reasons for them.

If there are no limitations on public access, this metadata element shall indicate that fact.

The value domain of this metadata element is free text.

9. ORGANISATIONS RESPONSIBLE FOR THE ESTABLISHMENT, MANAGEMENT, MAINTENANCE AND DISTRIBUTION OF SPATIAL DATA SETS AND SERVICES

For the purposes of Article 5(2)(d) and Article 11(2)(g) of Directive 2007/2/EC, the following two metadata elements shall be provided:

9.1. Responsible party

This is the description of the organisation responsible for the establishment, management, maintenance and distribution of the resource.

This description shall include:

— the name of the organisation as free text,

— a contact e-mail address as a character string.

9.2. Responsible party role

This is the role of the responsible organisation.

The value domain of this metadata element is defined in part II.4 of this Annex.

10. METADATA ON METADATA

For the purposes of Article 5(1) of Directive 2007/2/EC the following metadata elements shall be provided:

10.1. Metadata point of contact

This is the description of the organisation responsible for the creation and maintenance of the metadata.

This description shall include:

— the name of the organisation as free text,

— a contact e-mail address as a character string.

10.2. Metadata date

The date which specifies when the metadata record was created or updated.

This date shall be expressed in conformity with ISO 8601.

10.3. Metadata language

This is the language in which the metadata elements are expressed.

The value domain of this metadata element is limited to the official languages of the Community expressed in conformity with ISO 639-2.

II. VALUE DOMAINS (i.e. code lists)

1. RESOURCE TYPE

1.1. Spatial data set series (series)

1.2. Spatial data set (dataset)

1.3. Spatial data services (services)

2. TOPIC CATEGORIES IN ACCORDANCE WITH EN ISO 19115

2.1. Farming (farming)

Rearing of animals and/or cultivation of plants.

This category applies to Directive 2007/2/EC spatial data theme Annex III(9) Agricultural and aquaculture facilities.

2.2. Biota (biota)

Flora and/or fauna in natural environment.

This category applies to the following Directive 2007/2/EC spatial data themes: Annex III(17) Bio-geographical regions, Annex III(18) Habitats and biotopes, Annex III(19) Species distribution.

2.3. Boundaries (boundaries)

Legal land descriptions.

This category applies to the following Directive 2007/2/EC spatial data themes: Annex I(4) Administrative units, Annex III(1) Statistical units.

2.4. Climatology / Meteorology / Atmosphere (climatologyMeteorologyAtmosphere)

Processes and phenomena of the atmosphere.

This category applies to the following Directive 2007/2/EC spatial data themes: Annex III(13) Atmospheric conditions, Annex III(14) Meteorological geographical features.

2.5. Economy (economy)

Economic activities, conditions and employment.

This category applies to the following Directive 2007/2/EC spatial data themes: Annex III(20) Energy resources, Annex III(21) Mineral resources.

2.6. Elevation (elevation)

Height above or below sea level.

This category applies to the following Directive 2007/2/EC spatial data theme: Annex II(1) Elevation.

2.7. Environment (environment)

Environmental resources, protection and conservation.

This category applies to the following Directive 2007/2/EC spatial data theme: Annex I(9) Protected sites.

2.8. Geoscientific Information (geoscientificInformation)

Information pertaining to earth sciences.

This category applies to the following Directive 2007/2/EC spatial data themes: Annex III(3) Soil , Annex II(4) Geology, Annex III(12) Natural risk zones.

2.9. Health (health)

Health, health services, human ecology, and safety.

This category applies to the following Directive 2007/2/EC spatial data theme: Annex III(5) Human health and safety.

2.10. Imagery / Base Maps / Earth Cover (imageryBaseMapsEarthCover)

Base maps.

This category applies to the following Directive 2007/2/EC spatial data themes: Annex II(3) Orthoimagery, Annex II(2) Land cover.

2.11. Intelligence / Military (intelligenceMilitary)

Military bases, structures, activities.

This category does not apply specifically to any Directive 2007/2/EC spatial data themes.

2.12. Inland Waters (inlandWaters)

Inland water features, drainage systems and their characteristics.

This category applies to the following Directive 2007/2/EC spatial data theme: Annex I(8) Hydrography.

2.13. Location (location)

Positional information and services.

This category applies to the following Directive 2007/2/EC spatial data themes: Annex I(3) Geographical names, Annex I(5) Addresses.

2.14. Oceans (oceans)

Features and characteristics of salt water bodies (excluding inland waters).

This category applies to the following Directive 2007/2/EC spatial data themes: Annex III(16) Sea regions, Annex III(15) Oceanographic geographical features.

2.15. Planning / Cadastre (planningCadastre)

Information used for appropriate actions for future use of the land.

This category applies to the following Directive 2007/2/EC spatial data themes: Annex I(6) Cadastral parcels, Annex III(4) Land use, Annex III(11) Area management/restriction/regulation zones & reporting units.

2.16. Society (society)

Characteristics of society and cultures.

This category applies to the following Directive 2007/2/EC spatial data themes: Annex III(10) Population distribution – demography.

2.17. Structure (structure)

Man-made construction.

This category applies to the following Directive 2007/2/EC spatial data themes: Annex III(2) Buildings, Annex III(8) Production and industrial facilities, Annex III(7) Environmental monitoring facilities.

2.18. Transportation (transportation)

Means and aids for conveying persons and/or goods.

This category applies to the following Directive 2007/2/EC spatial data theme: Annex I(7) Transport networks.

2.19. Utilities / Communication (utilitiesCommunication)

Energy, water and waste systems and communications infrastructure and services.

This category applies to the following Directive 2007/2/EC spatial data theme: Annex III(6) Utility and governmental services.

3. DEGREE OF CONFORMITY

3.1. Conformant (conformant)

The resource is fully conformant with the cited specification.

3.2. Not Conformant (notConformant)

The resource does not conform to the cited specification.

3.3. Not evaluated (notEvaluated)

Conformance has not been evaluated.

4. RESPONSIBLE PARTY ROLE

4.1. Resource Provider (resourceProvider)

Party that supplies the resource.

4.2. Custodian (custodian)

Party that accepts accountability and responsibility for the data and ensures appropriate care and maintenance of the resource.

4.3. Owner (owner)

Party that owns the resource.

4.4. User (user)

Party who uses the resource.

4.5. Distributor (distributor)

Party who distributes the resource.

4.6. Originator (originator)

Party who created the resource

4.7. Point of Contact (pointOfContact)

Party who can be contacted for acquiring knowledge about or acquisition of the resource.

4.8. Principal Investigator (principalInvestigator)

Key party responsible for gathering information and conducting research.

4.9. Processor (processor)

Party who has processed the data in a manner such that the resource has been modified.

4.10. Publisher (publisher)

Party who published the resource.

4.11. Author (author)

Party who authored the resource.

Annex B – Metadata table creation SQL code

```
CREATE TABLE metadata
(
  id INTEGER NOT NULL DEFAULT nextval('metadata_id_seq1'::regclass),
  dataset_oid INTEGER,
  dataset_name CHARACTER VARYING,
  resource_title CHARACTER VARYING,
  resource_abstract CHARACTER VARYING,
  resource_type CHARACTER VARYING DEFAULT 'dataset'::CHARACTER
VARYING,
  resource_locator CHARACTER VARYING,
  identifier_code CHARACTER VARYING,
  identifier_namespace CHARACTER VARYING DEFAULT
'ucl.ac.uk_CEGE_metadata'::CHARACTER VARYING,
  resource_language CHARACTER VARYING,
  topic_category CHARACTER VARYING,
  keyword CHARACTER VARYING,
  vocabulary_title CHARACTER VARYING,
  vocabulary_reference_date DATE,
  vocabulary_date_type CHARACTER VARYING,
  bb_northbound_lat NUMERIC(7,4),
  bb_eastbound_long NUMERIC(7,4),
  bb_southbound_lat NUMERIC(7,4),
  bb_westbound_long NUMERIC(7,4),
  tempext_start_date DATE,
  tempext_end_date DATE,
  creation_date DATE,
  publication_date DATE,
  last_revision_date DATE,
  lineage CHARACTER VARYING,
  resolution_scale INTEGER,
  resolution_distance INTEGER,
  resolution_measure_unit CHARACTER VARYING,
  conformity_degree CHARACTER VARYING DEFAULT
'notConformant'::CHARACTER VARYING,
  confspec_specification CHARACTER VARYING,
  confspec_date DATE,
  confspec_date_type CHARACTER VARYING,
  use_limitations CHARACTER VARYING,
  use_conditions CHARACTER VARYING,
  respparty_name CHARACTER VARYING,
  respparty_email CHARACTER VARYING,
  party_role CHARACTER VARYING,
  metadatacontact_name CHARACTER VARYING,
  metadatacontact_email CHARACTER VARYING,
  metadata_date DATE,
```

```
  metadata_language CHARACTER VARYING(2),
  geom geometry(Polygon,4326),
  CONSTRAINT metadata_pkey1 PRIMARY KEY (id)

);
```

Annex C – Trigger function example

NOTE: The trigger function below implements the automatic update functionality for the bounding box coordinates and geometry. It works by automatically creating additional trigger functions on each of the new datasets inserted to the database and monitoring any change that needs to be transmitted to the metadata table.

```
CREATE OR REPLACE FUNCTION public.create_dataset_update_trigger()
  RETURNS TRIGGER AS
$BODY$

DECLARE
      table_name TEXT;
      func_body TEXT;
      func_cmd TEXT;
      part_of_query TEXT;

BEGIN

      SELECT dataset_name FROM metadata WHERE NEW.dataset_name = dataset_name INTO table_name;

      func_body := '<< variable >>
                  DECLARE
                        xmin REAL;
                        ymin REAL;
                        xmax REAL;
                        ymax REAL;

                  BEGIN';

      part_of_query := '''POLYGON(('' || variable.xmin || '' '' || variable.ymin || '','' ||
                                    variable.xmin || '' '' || variable.ymax || '','' ||
                                          variable.xmax || '' '' || variable.ymax || '','' ||
```

```
                                        variable.xmax || '' '' || variable.ymin || '','' ||
                                              variable.xmin || '' '' || variable.ymin || ''))'',4326)';


func_body := func_body || '

      UPDATE metadata SET bb_westbound_long =
            (SELECT ST_XMin(ST_Extent(ST_Transform(geom,4326))) FROM ' || table_name || ')'
                  || ' WHERE dataset_name = ' || quote_literal(table_name) || ';

      UPDATE metadata
            SET bb_eastbound_long =
            (SELECT ST_XMax(ST_Extent(ST_Transform(geom,4326))) FROM ' || table_name || ')'
                  || ' WHERE dataset_name = ' || quote_literal(table_name) || ';

      UPDATE metadata
            SET bb_northbound_lat =
            (SELECT ST_YMax(ST_Extent(ST_Transform(geom,4326))) FROM ' || table_name || ')'
                  || ' WHERE dataset_name = ' || quote_literal(table_name) || ';

      UPDATE metadata
            SET bb_southbound_lat =
            (SELECT ST_YMin(ST_Extent(ST_Transform(geom,4326))) FROM ' || table_name || ')'
                  || ' WHERE dataset_name = ' || quote_literal(table_name) || ';



      SELECT ST_XMin(ST_Extent(ST_Transform(geom,4326))) FROM ' || quote_ident(table_name) || ' INTO xmin;
      SELECT ST_XMax(ST_Extent(ST_Transform(geom,4326))) FROM ' || quote_ident(table_name) || ' INTO xmax;
      SELECT ST_YMax(ST_Extent(ST_Transform(geom,4326))) FROM ' || quote_ident(table_name) || ' INTO ymax;
      SELECT ST_YMin(ST_Extent(ST_Transform(geom,4326))) FROM ' || quote_ident(table_name) || ' INTO ymin;


      UPDATE metadata SET geom = ST_GeomFromText(' || part_of_query ||
```

```
                '
                WHERE dataset_name = ' || quote_literal(table_name) || ';';

      func_body := func_body || ' RETURN NULL; END;';

      func_cmd := 'CREATE OR REPLACE FUNCTION update_bounding_box_' || table_name || '() RETURNS TRIGGER AS $$'
            || func_body || ' $$ LANGUAGE plpgsql;';

      EXECUTE func_cmd;

      EXECUTE 'CREATE TRIGGER ' || table_name || '_bb_update AFTER INSERT OR UPDATE OR DELETE ON ' ||
quote_ident(table_name)
            || ' FOR EACH ROW EXECUTE PROCEDURE update_bounding_box_' || table_name || '();';

      RETURN NULL;
END;

$BODY$
  LANGUAGE plpgsql VOLATILE
  COST 100;
ALTER FUNCTION public.create_dataset_update_trigger()
  OWNER TO postgres;

-----------------------------------------------------------------------------------------

CREATE TRIGGER create_dataset_update_trigger
  AFTER INSERT
  ON metadata
  FOR EACH ROW
  EXECUTE PROCEDURE public.create_dataset_update_trigger();
```