

EXPLORING SUBSET PROFILE AND VALIDATION PROCEDURES OF GEOGRAPHICAL MARKUP LANGUAGE (GML) FOR 3D AREAL PLAN INFORMATION.

Assoc. Prof Dr Erling Onstein¹

Assist. Prof Sverre Stikbakke¹

¹ Norwegian University for Science and Technology, **Norway**

ABSTRACT

In the paper we explain our experiences in the work towards a GML subset profile for Norwegian Areal Plan data sets, and how this GML subset profile together with requirements form the UML based conceptual application schema that can be used for data validation.

GML is about to be the preferred exchange language for geographical information in Norway. One challenge using GML, slowing down the implementation of GML in Norway, is the complexity, e.g. the possibilities of representing the same information in multiple ways.

In all digital information handling, automated validation is important. For data validation, both the digital data and the requirements to the data are needed in suitable languages/formats.

ISO 19136 define the “ISO-certified” version of GML. The Annex G of ISO19136 have guidelines for defining GML subset profiles. Using GML subset profiles the unneeded complex parts and alternative solutions in GML can be removed, and data still be conformant to the full GML. Selected activities considering GML profiles are investigated: INSPIRE, CityGML and OGC GML SimpleFeatureProfile. The surprising finding from this investigation is the lack of formal GML profiles to be used for data validation. In all the three activities, subsets of GML are explained only using text for human reading, to some degree pointing to recommended UML modelling practice. None of the three activities have made available official XSD schema files for information validation following their text-based GML subset profiles.

For the XML-based GML, the natural data validation start is XSD-based data structure validation against the formal XSD application schema. For validation of geometry (e.g. closure of polygons and solids) additional requirements to the “pure” XSD data structure rules are needed. Some geometry requirements can be derived directly from the GML semantics, e.g. a GML Ring should be a closed curve and a GML CompositeCurve should have only connected curve subparts. Some other geometry requirements are connected to the semantics of the defined feature types in the user application schema, e.g. the spatial extent of a county should be inside the spatial extent of the country.

For geographic information described using UML and based on the ISO19109 General Feature Model, UML classes representing feature types are important modelling elements. We have added geometry requirements to UML-based feature types and made

this available for validation on the dataset (GML) level. The geometry requirements are defined using the ISO19157 Data Quality / Data Quality Measure (DQM) principles. In the context of this paper, only quality category Logical Consistency DQMs are relevant. In the paper we explain the experience with defining and validating rather general geometry rules connected to polygon geometry; polygon/polygon relationships and polygon set/tessellations rules.

The paper ends with conclusions and recommendations for further work.

Keywords: GML, validation, areal plan, spatial relations, PostGIS

INTRODUCTION

In geospatial data production and data maintenance, digitalization of processes is one main activity. Automated data quality control and validation are important activities within digitalization. To be able to automate these processes, the requirements to the data must be available in a formal way, possible for computers to interpret.

In Norway, digital procedures for handling areal plans have been used for several years. Areal plans are produced by private and public actors. The municipalities are responsible for the political approval of the plans. The responsible ministry (Ministry of Local Government and Modernisation) have, with legal basis in the Planning and Building Act, decided on the rules: All areal plans must be sent to the municipality as a digital geospatial dataset conformant to a ISO19131-based product specification, the AP (Areal Plan) Product Specification. Approved areal plans are made available in digital form in the national geospatial infrastructure. This is a national implementation of the principles described by Aydinoglu [1], and a step towards what is described in [2]. The product specification has a mandatory application schema to be used. The application schema is based on ISO 19109 General Feature Model (GFM) implemented in the national Norwegian SOSI standard. The file format to be used is the national SOSI format. One important reason for this procedure to be accepted by the involved actors and has been successful, is a freely available validation tool. Using this tool, everybody producing an areal plan can validate the resulting dataset before sending the dataset over to the municipality.

The application schema for areal plans have until now been 2D, i.e. spatial positions are given using only northing and easting. For areas for building activities, height regulations are also used, given as feature type attributes.

To be able to automate the use of areal plans, mainly in building permit applications [3] it is decided to update the AP Product Specification. The main change is to introduce 3D features for selected parts. One consequence of this is to replace the SOSI format (not covering 3D solid geometry) by GML (Geography Markup Language according to ISO19136) as exchange format. The SOSI-format-based validation tool must also be replaced by a GML-based tool. This paper describes selected parts of this change.

GML PROFILE ACTIVITIES

GML is a well-established language for geospatial information [4]. The Norwegian SOSI standard the replacement of the original SOSI-format with GML started in 2006. GML

Schematron schema document can be used to ascertain whether a GML document instance complies to the rules specified in this profile.” The decision of not including a formal GML subset profile makes formal conformance validation of GML datasets impossible.

The investigation of the three activities above (CityGML, INSPIRE and OGC GML Simple Features Profile) ended up with no good examples of how GML subset profiles can be defined. This is a rather surprising finding.

DEFINING FORMAL GML PROFILE

An overview of the different parts of the ISO19136 GML standard is found in Figure 2. The figure shows that the Annex D define the full GML schema. Using the rules in ISO19136 Annex G GML subset profiles can be defined. A user application schema, as the one needed for the Norwegian areal plans, should use either the full GML schema or a GML areal plan subset profile (the AP profile) defined as specified in annex G.

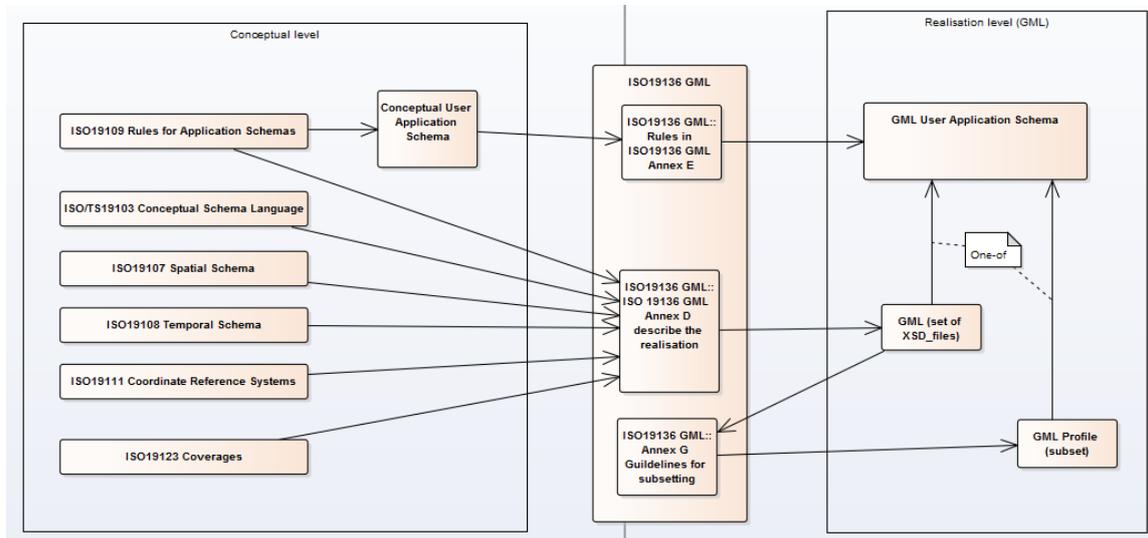


Figure 2 Standards and parts in GML

The Figure 3 is a review of the parts in GML as found in GML 19136. The figure shows that some parts should be fully included and some should be completely removed. Only one part, the GML Geometry, will be modified into a less complex part.

In the following, only the modifications in the geometry part will be explained.

The main change in the geometry part is the reduction of geometry primitives in the one- and two- dimensional parts of GML geometry, the curve and the surface parts. This is shown in Figure 4. Only the classes shown in yellow and green are found useful for the AP profile, and thus included. The rest of the primitives are removed from the profile.

The restrictions to the three-dimensional parts (the volume/solid parts) are only connected to the restrictions in surface patch geometry types, i.e. the restrictions given in Figure 4. According to the procedure in ISO19136 Annex G, the resulting GML profile is only one XSD file. To avoid mixing the suggested AP profile with the full GML when it comes to “real data”, the AP profile is assigned a new xml namespace:

`xmlns:gmlap="http://gml.arkitektum.no/KMD_3D_Arealplan/GMLProfiltest/1.0"`

19136- chapter	Name of part	Realisation of	Relevant in the AP profile
7	7.2.6 Metadata	ISO 19115	All
8	8.1.XLink		All
	8.2 Basic types, nil handling		
9	GML Feature	ISO19109, clause 7	All
10,11	GML Geometry	ISO19107	Profile
12	GML Coordinate reference systems	ISO19111	All
13	GML Topology	ISO19107	Remove
14	GML Temporal information	ISO19108	Remove
15	GML Definitions and dictionaries		Remove
16	GML Units, measures and values		All
17	GML Directions		Remove
18	GML Observations		Remove
19	GML Coverages	ISO19123	Remove

Figure 3 GML parts to be included in the GML AP profile.

instead of the common namespace for full GML:

`xmlns:gml=http://www.opengis.net/gml/3.2`

The defined AP profile is still only a candidate profile. The decision whether to include it in Norwegian SOSI standard is expected late 2017.

GML DATA REQUIREMENTS

The requirements to a GML dataset come from several sources. The most important are:

- The XML structure requirement, given in the application XSD file. Implementation of XSD-based structure validation is well known. Both commercial tools (e.g. Altova XML Spy, Oxygen XML Editor), web services [10], and plug-in open-source software libraries for development of own software, e.g. [11] are available.
- Requirements directly connected to the GML geometry model. Example of this are 1) all geometry of type GML Ring must be a closed ring, 2) GML Solids must be correctly closed [12] 3) Parts of a GML CompositeCurve must be connected, validation implementation shown in [10].
- Geometry-related rules in the user application schema.

Using Model Driven Architecture (MDA) the geometry-related rules should be stated in the conceptual application schema, and automatically transformed to the realisation platform, in our case GML. When using the GFM principles, the conceptual model should be Unified Modelling Language (UML).

DECLARATION OF USER APPLICATION SCHEMA RULES

Within the context of this paper (application areal plan) relevant geometry-related rules are rules constraining the location of the feature instances belonging to feature types. The conceptual UML model for the test case is selected parts from INSPIRE LandUse [13]: the two feature types ZoningElement and SpatialPlan. As all information in an areal plan at least in Norway only have 2D geometry, relevant examples are limited to 2D. Example of rules are found in Figure 5.

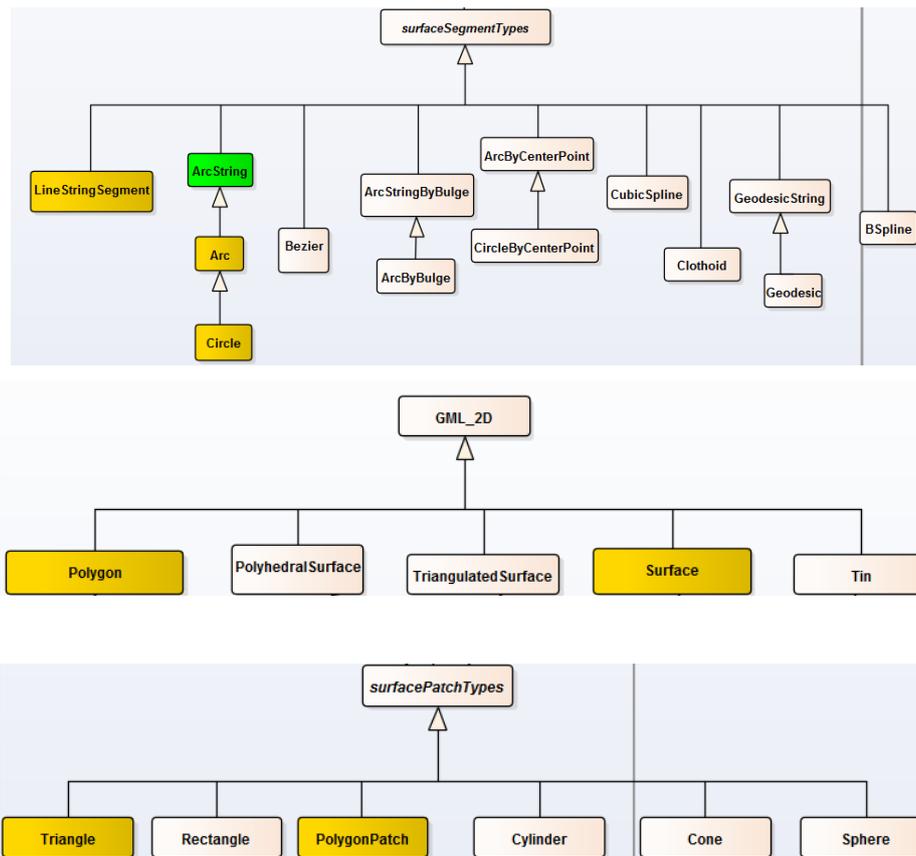


Figure 4 GML geometry, 1D part (top), 2D part (middle) and 2d/surface patch types (bottom)

These three rules share common semantics:

- They are valid for all datasets following the application schema
- The rules should be declared as part of the conceptual (UML) model, and transformed following MDA principles to all representations (GML, databases...)

In the conceptual UML model, the three constraints can be expressed in several ways:

- Using associations between UML classes representing the relevant feature types.
- Using the recommended constraint language for UML, Object Constraint Language (OCL), a standard language for defining constraint, developed by Object Management Group (<http://www.omg.org/spec/OCL/2.4>). OCL expressions can be transformed to XML/Schematron rules, and used for validation of XML data. The main use of OCL is to express rules that is not easy to express using e.g. UML Class diagrams. Examples of this is dependency between attribute values for one class. OCL can also compare attribute values, e.g. for defining a constraint telling that for existing data the data capture date must be before current date. But unfortunately, we have not found geometry operators in OCL.
- Using UML Tagged Values. Tagged values can be considered as user-extensions to UML classes. Tagged values can be passed to over to the XSD application schema as well. However, the tagged values are not easily visible within an UML Class diagram.

Requirement Name	Description	Explanation
Inside	All ZoningElements must be INSIDE the related SpatialPlan.	<ul style="list-style-type: none"> Required parameters: One parent feature instance which the tested feature must be inside Computation procedure: One feature instance at a time
NO_Overlap	No ZoningElement must OVERLAP another ZoningElement	<ul style="list-style-type: none"> Required parameters: A set of feature instances which the tested feature must not overlap Computation procedure: One feature instance at a time
CompleteTesselation	A SpatialPlan must be completely covered with the union of all ZoningElements	<ul style="list-style-type: none"> Required parameters: A set of feature instances (set of childs) which completely covers the tested feature (the parent) Computation procedure: One feature instance (parent) at a time

Figure 5 The selected geometry requirements

- Using ISO19157/19131 Data Quality Measures. ISO19157 Data Quality describe how data quality requirements to datasets can be described. Data quality is divided into three main data quality categories: Accuracy, completeness and logical consistency. In our context, the logical consistency is the relevant one. ISO19157 introduces Data Quality Measures (DQM), and lists examples of DQMs for all quality elements. The purpose of DQM is that data quality requirement given as a threshold value (e.g. accuracy must be better than 30cm) should be followed by procedures describing how to compute the corresponding values from a dataset, and statistical test procedures for deciding whether the datasets fulfils the requirement or not. Within this DQM environment, all three example rules could be defined, and related to correct features. The main disadvantage defining rules this way, is that the quality requirements are not part of the conceptual schema, but stored in separate product specification chapters.

Using OCL seems like the best solution, but since OCL have no support for geographical operators, this is not a solution until such operators are added. The DQM method have options for defining complex computation rules useful for this kind of consistency rules. For our further testing, we have, in lack of OCL support, used DQMs with parameters pointing to involved elements in the datasets.

The structure of a DQM is defined in the ISO19157 standard, given as a list of components. Our third requirement (see Figure 5), the complete tessellation, can be defined as in Figure 6.

APPLICATION SCHEMA GEOMETRY VALIDATION

The 4-intersection model for spatial relations is a well-known basis for spatial queries. The model operates on simple regions in R^2 and describes possible spatial relations between two such regions. These topological relations are named disjoint, contains, inside, equal, meet, covers, covered by and overlap. The 4-intersection model are later

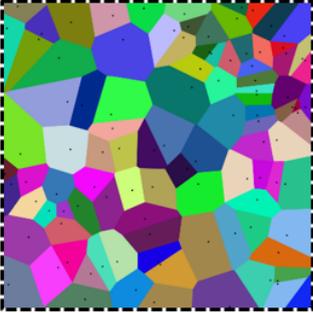
Line	Component	
1	Name	CompleteTessellation
2	Alias	Tessellation complete
3	Element name	Conceptual consistency
4	Basic measure	Either Error indicator, Error count or Error rate
5	Definition	Total number of erroneous polygons within the data
6	Description	All child polygons must be located inside the parent polygon. Reports the number of child polygons not inside the parent polygon. A set of feature instances (set of childs) which completely covers the tested feature (the parent)
7	Parameter	Parameter: Identification of child polygons
8	Value type	Either Boolean, Integer or Percentage
9	Value structure	
10	Source reference	
11	Example	 <p>Parent polygon (dashed quadrat) filled correctly with child polygons (coloured). Errors when gaps between childs, and also error when overlap between childs.</p> <p>Reporting by parent polygons.</p>
12	Identifier	NTNU/ConceptualConsistency03

Figure 6 DQM Complete Tessellation

extended by the Dimensionally Extended Nine-Intersection Model (DE-9IM) model to deal with points and lines in addition to simple regions.

As part of standardization efforts within OGC and ISO, the semantics in DE-9IM is included in standards for geographic information. Of particular interest is the ISO 19125-1:2004 Geographic Information, Simple features access, Part 1: Common Architecture.

The spatial relations from the 4-intersect model are helpful means to describe the semantics of spatial requirements. This semantic is also implemented in software procedures.

The requirements of geometry parts of areal plans in Norway comply with ISO 19125-1. It should then be possible to validate the individual zoning divisions using the relational predicates found in ISO19125-1.

Requirement Inside

This requirement can be described by the relations inside and covered by. Inside is the relation where the geometry of object A is completely within the geometry of object B. However, this is not sufficient, as the case where object A has one or more points common with the border of object B is also allowed. This condition can be described by the Covered By relation.

Requirement NoOverlap

The relation overlap describes the condition where there is an overlap.

Requirement CompleteTessellation

An areal plan is not allowed to have holes or gaps. Every spot within the plan should have a purpose, described by zoning subdivisions. There is no 4-intersect model type relation who describe this situation as they operate on simple regions, i.e. regions without holes. Some intermediate procedures are needed to find potential holes. Areal plan area is the encompassing region for the zoning divisions as each subdivision is owned by an areal plan. By merging the zoning divisions (ZoningElements) and subtract the resulting area from the areal plan area (SpatialPlan) we should get the geometry of the holes as a result.

A special case of this requirement is when there are no subdivisions in the areal plan area altogether. This is not a normal situation, but it might occur as an error condition. These areas can be detected by spatial join on areal plan area and zoning subdivisions. Empty areas will be identified by empty result sets.

Software implementations

Several software products and libraries are compliant with the ISO 19125-1 standard. Well-known is the Java Topology Suite. GEOS is a C++ port of JTS that are used in the PostGIS spatial database extension for PostgreSQL [14]. It also contains GIS operations like overlay, areal calculations and so forth.

We assumed PostGIS to be a suitable tool for validation of spatial requirements. This assumption is partly based on these factors:

- the software is open sourced and available for inspection
- it is easy to integrate with other open source software, e.g. GDAL software libraries for format transformations
- the software is used in production settings with good results
- test procedures can be written and documented as SQL scripts

PostGIS methods used in validation procedures are shown in Figure 7.

Spatial requirement	ISO19125:1 method	GEOS/PostGIS method	Used for
Inside	Covered By	ST_CoveredBy	Test for inside condition
No Overlap	Overlaps	ST_Overlaps	Test for overlap
	Intersect	ST_Intersect	Identify and visualize overlapping areas
Complete tessellation	Union	ST_Union	Merge zone divisions inside regulatory plan areas
	Difference	ST_Difference	Create polygons of areas without zone divisions

Figure 7 Relevant ISO 19125-1 Methods and PostGIS equivalent terms

Sequence of operations when testing spatial requirements:

1. Create empty PostGIS database
2. Import GML file to database

3. Check for inside condition
4. Check for overlaps
5. Merge all zone subdivisions
6. Do difference overlay with areal plan area and merged subdivisions
7. Export GML files with features representing areas with failed spatial requirements

Test case

Gjøvik municipality has an area of 672 km² and a population of 30 000. Area plans are currently published in SOSI-format which is soon to be replaced by GML. Areal plan data from 2016 were used for testing. It consists of 330 individual areal plans, and 6143 zoning elements within the plans.

Plan data was imported from SOSI format to PostGIS database using FME software. This step will later be replaced with import from GML files when areal plans in GML format become available.

Inside requirement

Initial test with the PostGIS ST_CoveredBy method revealed 429 violations of the inside requirement, i.e. 429 zoning elements had one or more points outside their encompassing areal plan. This seemed as an unlikely number, and we suspected this to be caused by rounding effects. The input data file has a coordinate resolution of 1 cm, while the PostGIS database geometry type has double precision floating point representation of coordinates. To mitigate rounding effects, we decided to buffer each encompassing areal plan with 5 cm before the ST_CoveredBy test was performed. By this procedure, the number of violations dropped from 429 to 17. Further tests with even larger buffer distance had some effect, e.g. with 50 cm buffer distance, the number of violations was 4.

To locate and investigate these violations we used some overlay procedures. Some of the violations were of the sliver type polygons, and some were inclusion of larger areas outside the actual plan. One single case was caused by wrong attribute coding of areal plan id.

No overlap requirement

This test was performed with the ST_Overlaps method on the zoning element parts. Again, to overcome rounding effects, we used an area limit of 0.01 m² on this test. 7 violations were identified, but in all cases the areas were less than 0.05 m². We concluded that these violations were negligible.

Complete tessellation requirement

Tessellation test based on procedure described above with an area limit on violations of 0.01 m² detected 155 violations. By increasing the limit to 1 m², the number of violations were 31. Some of these violations were due to lack of zoning data for older areal plans, and in these cases the entire plan area was empty. Most of the remaining violations had no obvious cause, and may come from erroneous omissions in digitalization process.

Technical requirements and limitations

The procedures described here have limitations regarding geometry types. An initial attempt to import to PostGIS database from GML/XML files failed due to incompatible geometry types. With the GDAL library, the import procedure for XML files created curvepolygon geometry for the areal plan data. Unfortunately, none of the PostGIS procedures accepted this geometry type. The FME import procedure created geometry type multipolygon, but the relational predicates like `ST_Overlaps` and `ST_CoveredBy` will not operate correctly on geometry collection datatypes like multipolygon. To overcome this, the geometry field was converted to polygon type. However, overlay procedures may create the multipolygon or geometry collection datatypes. This phenomenon occurred in some rare cases where the areal plan itself consisted of multipolygons. These plans were excluded from further analysis.

CONCLUSION AND FURTHER WORK

In a fully digital data information workflow, it is important to have generally defined requirements to all parts of the data. We have explained how a formal GML subset profile can be defined and used. We have shown one way to define a proper foundation for full validation of GML-based datasets, based on MDA principles. We have also shown one way of defining user-application geometry requirements using ISO19157 Data Quality Measures.

Although our method is a possible way of handling these kind of requirements, we recommend further testing on geographical operators within the Object Constraint Language.

The DQM-based requirements fit well to PostGIS implementation, and validation of datasets against the defined requirements can be done.

We consider this work as a step towards an automated validation test for GML datasets. So far, we have identified a possible path for validation of spatial requirements. The procedure we have presented here is not fully automated, and more work is needed to develop the wanted fully automated validation tool for all GML information requirements.

ACKNOWLEDGEMENTS

This paper is based on activity funded by Norwegian Ministry of Local Government and Modernisation, Department of Planning.

REFERENCES

- [1] Aydinoglu, Arif: Open Data Management Issues for Geographic Information Systems, in Proceedings 16th International Multidisciplinary Scientific Geoconference, Book 2 Volume 1, pp 789 – 794, 2016
- [2] Istomin, E.P at al: Geoinformation Management as a modern Approach to the Management of Spatially Distributed Systems and Territories, in Proceedings 15th

International Multidisciplinary Scientific Geoconference, Informatics, Geoinformatics and Remote Sensing Volume 1, pp 607 - 623, 2015

[3] Onstein E, Tognoni M: Case study “Building Permits” as proof of concepts in merging GIS and BIM information. Accepted for publication at Wessex Institute BIM 2017 Conference, 2017

[4] Chojka A: GML – does it really work in Practice?, in Proceedings 15th International Multidisciplinary Scientific Geoconference, Informatics, Geoinformatics and Remote Sensing Volume 1, pp 633 - 639, 2015

[5] <https://www.citygml.org/>

[6] INSPIRE: Guidelines for the Encoding of Spatial Data, 2014. Available as d2.7_v3.3.pdf from <http://inspire.ec.europa.eu/documents/guidelines-encoding-spatial-data>

[7] Gózdź K, Pachelski W: The Use of CityGML in the context of creating Smart Cities, in Proceedings 15th International Multidisciplinary Scientific Geoconference, Informatics, Geoinformatics and Remote Sensing Volume 1, pp 883 - 889, 2015

[8] Open Geospatial Consortium: OGC GML Simple Features Profile, 2012, available from <http://www.opengeospatial.org/standards>

[9] Inspire: INSPIRE Generic Conceptual Model, version 3.4, 2014, available from <http://inspire.ec.europa.eu/documents/inspire-generic-conceptual-model>, downloaded version: D2.5_v3.4rc3.pdf

[10] Open Geospatial Consortium: OGC Compliance Program, available on <http://cite.opengeospatial.org/>

[11] <https://wiki.python.org/moin/PythonXml>

[12] Hugo Ledoux: On the validation of Solids Represented with the International Standards for Geographic Information, 2013. Available from <http://dx.doi.org/10.1111/mice.12043>

[13] Inspire: D2.8.III.4 Data Specifications on Land Use – Technical Guidelines, 2013, available from <http://inspire.ec.europa.eu/Themes/129/2892>

[14] PostGIS 2.3.3dev Manual (2017) Available at: <http://postgis.net/docs/manual-2.3/> (accessed: 1 May 2017)