GML Validation Based on Norwegian Standard

Heng Min

Master's Thesis
Master of Science in Media Technology
30 ECTS
Department of Computer Science and Media Technology
Gjøvik University College, 2010

Avdeling for
informatikk og medieteknikk
Høgskolen i Gjøvik
Postboks 191
2802 Gjøvik

Department of Computer Science
and Media Technology
Gjøvik University College
Box 191
N-2802 Gjøvik
Norway

GML Validation Based on Norwegian Standard

Heng Min

1st July 2010

Abstract

Geographic information plays an important part in people's life. Especially with the development of computer science and Internet. GIS also went into a new era, from paper works to digital formats. The digital geographic data files can be spread and shared fast and widely via Internet. However, establishing new geographic data is expensive, thus the data users are not always the data establisher. In the hight speed developing GI society, many private sectors have participated in geographic data production, and also many datasets are from unclear sources with unclear quality information. In addition, errors can be generated during the transform and transfer processes. Therefor proper geographic data quality control and management is in urgent demand.

GML is an XML-formed document for geographic information. It is a young but fast developing geographic data format, with its natural advantages, it is now being considered as standard geographic data format world wide. This project is concerned with the data quality control for GML files. Data validation is one kind of data quality measurement method in the data quality control process. The thesis aims to build up a validation framework for GML files based on the Norwegian standard.

In the report the geographic data modeling is first described, including the Norwegian road network production specification, simple feature specification, and the related GML knowledge. The data quality issues are introduced after the modeling, the data quality issues are elaborated from two levels: general inconsistency level and specific ISO data quality element level. Following is the introductions, discussions and comparisons to the existing relevant geographic data quality tools. And at the ene the final GML validation framework is determined.

Preface

I would like to thank my supervisors Sverre Stikbakke and Rune Hjelsvold at Gjøvik University College, they have been encouraging and helping me through the whole master thesis period. I also want to thank Erling Onstein who suggested this project idea, and helped me clearing some questions during the thesis process.

Heng Min, 1st July 2010

Contents

1 Introduction

1.1 Background

Geographic information plays an important part in people's life. Especially with the development of computer science and Internet, GIS also went to a new era, from paper works to digital formats. There are more than 200 geographic data formats, so exchange of geographical information is an important activity in the geographical society [1]. GML is one of many geographical data formats. It is an XML-based document made to express geographic features. With its natural advantages that inherit from XML, GML is being considered as the standard exchange for geographical data formats world wide. However, data is not enough, in the real life's geographic information business, discovering, reusing and sharing geographic data is vital. Because establishing new geographic data is expensive, established data should be used by more than the data owner. In the latest years, there has been an increasing focus on SDIs [1]. SDI is often used to denote the relevant base collection of technologies, policies and institutional arrangements that facilitate the availability of and access to spatial data. The core elements in a SDI are *Portal*, *Metadata*, *Framework*, *Geodata*, *Standards* and *Partnerships* [2]. So with the help of SDI, the users can have a market place for searching for the data in a standard way, or by a catalog service. The users can also decide if the data is suitable for their applications by reading the metadata.

Data quality is an important part in the metadata. Different people have various standards about data quality, for some industries, extremely precise geographic data is required, while for others, the errors can be accepted under certain tolerance. By measuring the data quality and publish the measured result as metadata in a standard way will be of great help for reusing and sharing existed geographic data.

From a different angle, when modeling the real world, the first step is to simplify the real world, because the real world is too complex. Due to the quality problem of the modeling process itself, the final dataset can be discourse with the real world or the simplification of real world. Despite the possible quality problem of conceptual models the compliant and conformance between the dataset and the conceptual model, the specification of a geographical data product, is the most important part of the geodata validation.

The validation framework's goal is to validate the GML dataset against the GML schema extended with extra rules to make sure that the dataset conforms to corresponding product specification, while conforming with relevant ISO standards.

1.2 Project Extent

In this part the extent of this project is claimed.

- This project aims on vector data quality control, so the research of raster data quality is not

[1] Spatial data infrastructures

included

- Due to the time limitation, I chose road network domain as study point, thus the 2 dimension geometry objects, such as polygon, surface, are referred to that much
- The validation mechanism does not include the possibility of correcting the errors
- The road network product specification is based on Norwegian standard, different countries may have different models, meaning the requirements may differ
- The GML validation should be applied when the quality information of a dataset is changed. The quality of information can be affected by three conditions [3].
  - when any quantity of data is deleted from, modified or added to a dataset
  - when a dataset's product specification is modified
  - when the real world has changed

The thesis expects the readers to have certain relevant knowledge, such as XML and Geographical Information System. Next, Section 1.3 states the research questions of the thesis.

1.3 Research Questions

This section states the research question of the thesis.

1. What are the quality issues of geographic data?
2. Why are the existing tools not sufficient?
3. How can the GML validation framework be developed?

1.4 Motivation and Benefits

Geographic data quality is an essential parameter in GIS, it is the guarantee of usable product and services. Building up the GML validation system based on Norwegian standard can be used for controlling the data quality so that it optimizes the further usage of the geographical data. This can greatly improve the GI market, thus make the geographical information industry more attractive. On the specific level, the possible benefit objects could be data maintainers, standard associations, software developers and end users.

Geographical information infrastructure's purpose is to build an organization ensuring access and funding for high-quality geographic datasets for everybody, and develop and implement proper technological solutions to support [2]. The data maintainer has the responsibility for maintaining high-quality geographical datasets. This project can be helpful to data maintainers to have more control on the geodata, and therefore supply good quality geographical data to its users. This builds up a fire wall to all the data sources, and reduces the risk of users downloading inappropriate data. In Norway, Norwegian Mapping Authority is the national geographical data maintainer. For decades SOSI has been the standard geographical data format in Norway. One software, SOSI-Control is the system for controlling the quality of the SOSI datasets. However, now as GML is developing fast and there is an intension that GML will be adapted as supplement

to SOSI, or might even replace SOSI as the standard for exchanging geodata formats in Norway. The Norwegian Mapping Authority built a workshop for this, they will face the same challenges of GML data quality control. The thesis aims to build a GML validation system that can validate the GML dataset against the GML schema and some extra rules, which ensures that the dataset is conformed to the product specification. It can be helpful for taking GML in official use in Norway. The geographical activities in Norway are connected to the other parts in the world, therefore the project can stimulate the Norwegian geographical activities going to the international stage indirectly [1].

For standard associations the data quality of GML is a common domain that all countries will face if they adapt to GML as their standard exchange format. It is the intention that GML will be a world wide standard format; like XML validation, because its widely used and the challenges of its quality were not just for one or a few web users, but for all. Thus to standardize what are the XML quality problems, and the validation requirements was a good solution for the issue. Even though there are many different XML validators, what the validators do all conform to the standard requirements. In this project, the main aim is to find the GML validation requirements that are common and implementable, by analyzing the geodata quality issues addresses in ISO standard and Norwegian product specification.

For software developers to develop a software, certain user requirements and problem analyzing are necessary before doing programming and further work. This project is doing the job of analyzing the problem domain, and comes to the proper requirements and realizing methodology. So by this thesis report, software developer can fast understand the problem area and start work on the practical part easily and quickly.

For end users the GML validation will be a helpful tool to first detect possible errors in their GML files so that they can correct errors, improve data quality. Second they can avoid using unqualified data from unknown sources that does not validate their data before publishing them.

1.5 Contribution

The thesis can form the following contributions.

1. The thesis investigates the geographical data quality issues, states and discusses the current research in the geographical data quality area. Describe the relevant ISO standards content and their connections by putting them in context.
2. Discuss the capabilities, use cases of the existing geographic data quality tools, and their conformance level to the ISO standards. How to utilize these tools for developing the GML validation system is also addressed.
3. Develop a GML validation framework based on Norwegian standard, while conforming to the ISO/TC 211 standard body.

2 Knowledge description

2.1 Modeling the geographical world

Our real world is complicated, it includes all aspects that may or may not be perceived by individuals, or deemed relevant to a particular application [4]. The computer systems are not able to understand and describe the real world. A data model is a set of constructs for describing and representing parts of the real world in a digital computer system [4]. In a model certain properties important for the purpose of the model is included while the other properties are ignored [5]. So the geographical data model is a simplification of the geographic phenomena in the real world.

We can start with discussion of the role of data models in GIS. Data models works as a technical and practical approach to realize the computer implementation of geographical reality. Data models are vitaly important to GIS because they control the way that data are stored, and have a major impact on the type of analytical operations that can be performed [4]. Then let us get a closer look at how geographic data modeling represents the spatial part of the real world in a computer. First data models can be divided into several levels according to abstraction. The reality, which is human-oriented, is on the least abstract level and the physical model is the most abstract one; in between are conceptual models and logical models [4]. Data models can also be divided into different catalogs. For example, CAD, graphical and image GIS data models, raster data models and vector data models. As mentioned in the introduction chapter this project only focus on vector data, meaning only vector data models is taken into concern. This chapter is structured by the models abstraction, but the data type is vector, these two are not int conflict, but mixed together.

Reality is all aspects of the real world phenomena, no matter if it gets perceived or not. For example, trees, buildings, mountains, people, etc. Geographic reality is the reality that is relevant to geographic applications. Geographic reality can be sorted into many catalogs according to what the application focus on, for instance, transport network, property, forest usage and so on.

Conceptual models is human-oriented, often partially structured, model of selected objects and processes that are thought relevant to a particular problem domain [4]. The most well known conceptual modeling language is UML. It seems be regarded as the only modeling language that is used in almost all the cases. Conceptual model can be various for the same domain of interest. Because different people may have different understanding of the same phenomena, therefor, people may extract different elements and structure them in various ways for a same domain. In the section Conceptual Model, Norwegian road network specification will be introduced as the example of conceptual model in this project.

Logical/operational model is an implementation-oriented representation of reality that is often expressed in the form of diagrams and lists [4]. Logical modelling is done upon the conceptual model, which means it explains the conceptual model which is still human-oriented to computer. After this step, the computer can understand and implement the geographic information.

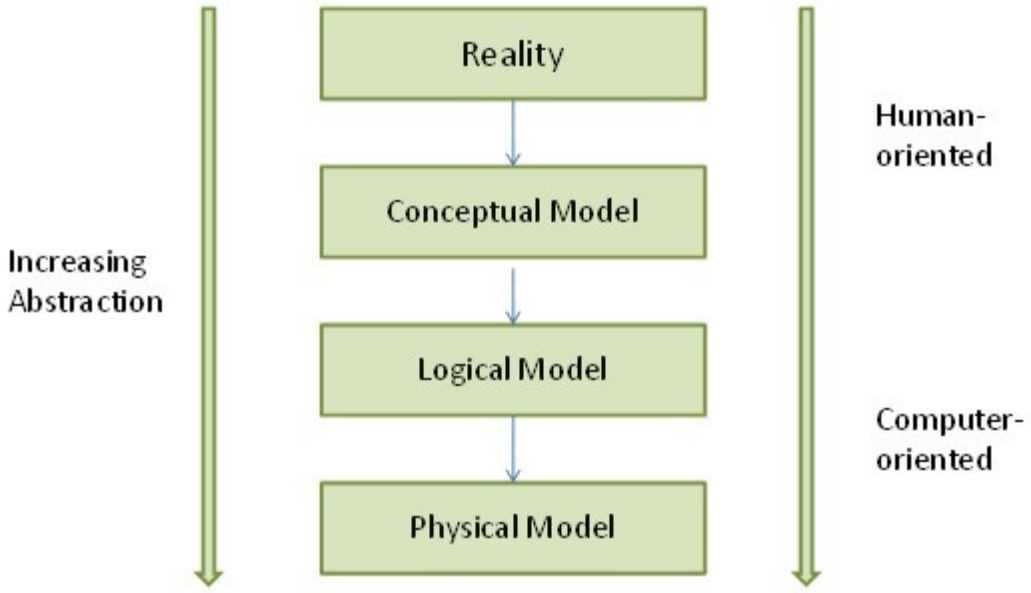


Figure 1: Levels of abstraction relevant to GIS data models. [4]

Like the conceptual model, there is not only one way to express a UML model. Over 200 geographical datase formats exist in GIS, each of the formats often has its own logical model to transform the UML. GML, one of the most popular geodata format nowadays, has its logical model, GML schema. SOSI, the Norwegian standard for exchanging geodata formats, has a what is called a 'SOSI-database' which translates UML model. The rules of transforming UML model to implementation-oriented depends on the realizing dataset format.

Lastly, the physical model portrays the actual implementation in a GIS, and often comprises tables stored as files or databases [4]. In other words, the physical model is the actual dataset. The datasets are the instances of the logical model, for example, a GML file which contains all the roads in Gjøvik, and the datasets are what people are sharing and building services upon.

2.1.1 Conceptual Model

Conceptual models, as mentioned earlier, can be subjective. There can be many models in one object domain, however usually for one country or union of countries, there is only one standard model. In Norway, the organization that defines these standards is Norwegian Mapping Authority. This organization created the standard geospatial data format in Norway, SOSI. Even this project is focusing on GML format, as mentioned earlier, all different data formats are just different ways to realize the product specification, so the product specification is in common. In this section, the Norwegian road network product specification is introduced. We can go straight to the point by looking at the UML models. These can be seen in Figures 2, 3 and 4.

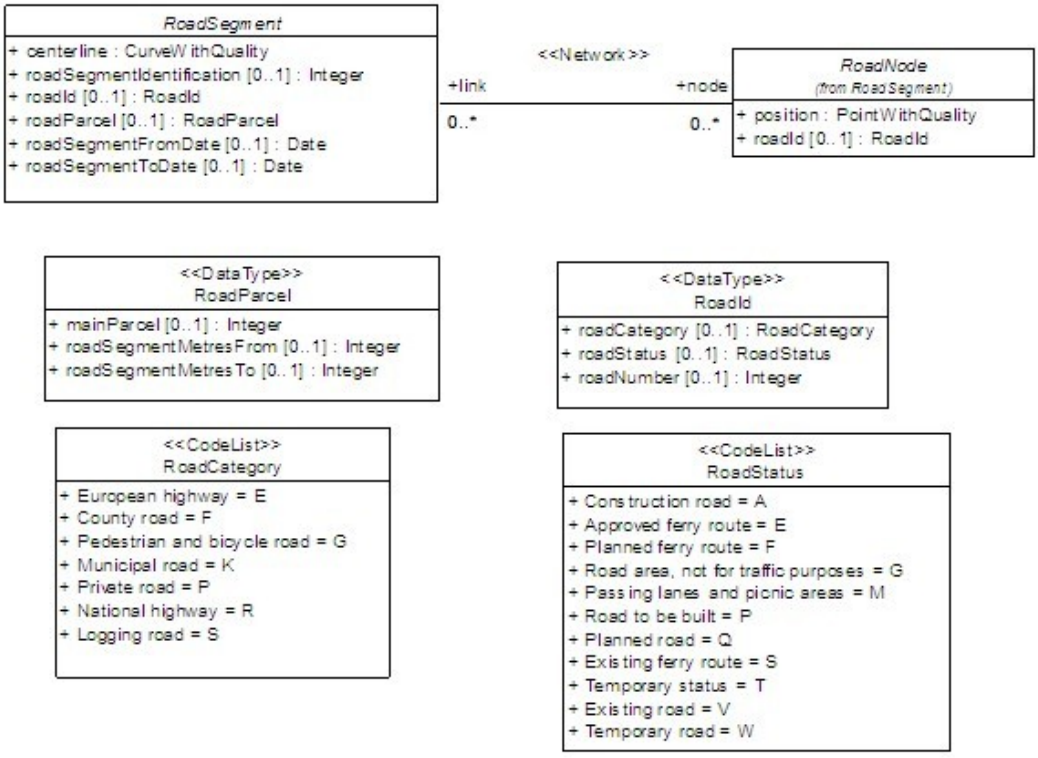


Figure 2: Road Network UML model. [6]

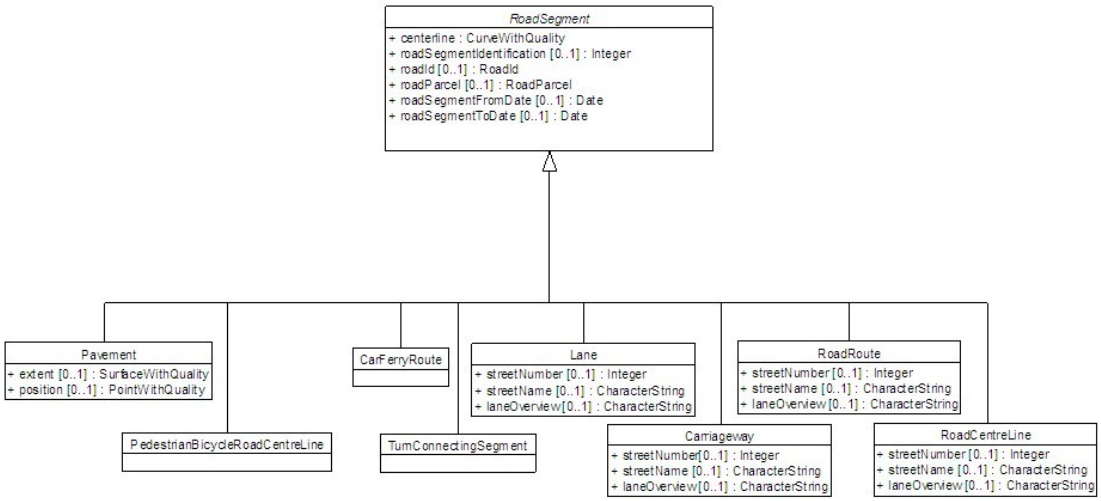


Figure 3: Road Segment UML model. [6]

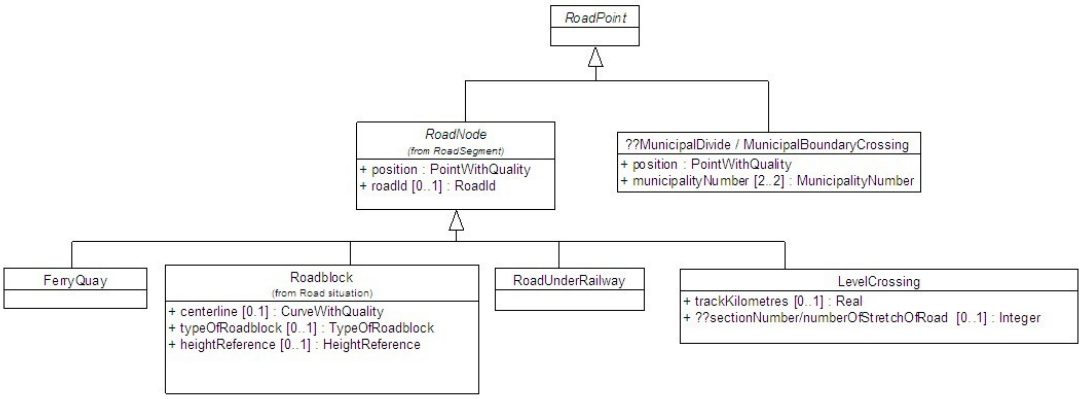


Figure 4: Road Point UML model. [6]

From the UML models, we can see that in the road network domain, polygon and surface feature types are not involved. The two abstract object types are RoadSegment and RoadNode. Several object types under each abstract object type. Seeing the overview, the road network in Norway is expressed by two kinds geometries: Line and Point. The road network phenomenon in the real world is modeled as either a line or a point. There are eight object types of line type, and five of point type. The line objects are Pavement, PedestrianBicycleRoadCentreLine, CarFerryRoute, TurnConnectingSegment, Lane, Carriageway, RoadRoute and RoadCentreLine. And the five Point objects are MunicipalDivide, FerryQuay, RoadBarrier, RoadUnderRailway and LevelCrossing. These 13 object types, with corresponding attributes and constraints, cover the road network phenomenon in real world. In other words, any phenomenon that concerns road network in real life must be one of the 13 object types, except the road shorter than 50 meters and some short private blind road [7] as claimed in the standard specification.

Nr	Navn/Rollenavn	Definisjon	-	+	Type	Restriksjon
3	Objekttype VegSenterlinje	Linje midt mellom vegkanter.				Subtype av veglenke
3.1	Gatenummer	Nummerering av alle veger, som sammen med kommunenummer danner en gateiden som er en unik ident for gater	0	1	Integer	
3.2	Gatenavn	Gatenavn på addressepunkt	0	1	CharacterString	
3.3	feltoversikt	Kjørefeltnummer angir stedfesting i vegens tverretning	0	1	CharacterString	

Figure 5: Discription of the roadcenterline object in Vegnett object catalog [6]

The conceptual rules defined in the production specification can be divided into explicit rules and implicit rules. The explicit rules include the object structure (UML model), object definition, attributes to the object, value constraints, geometry type, etc. For instance Figure 5 describes the definition of the roadcenterline object and also the structure restriction, subtype of road-segment, as well as its three attributes with definition descriptions, occurrence limits and value types. However not all the rules are stated in a clear specific way, the implicit rules need to be discovered through the understanding of the product specification, for example, road node should be covered by road segment.

2.1.2 Operational Model

One conceptual model can be realized in many different ways. This project is about GML validation, so in this section, how to realize the conceptual model in GML will be introduced. There are two subsections: Simple Features, which introduces the geometry classes for describing the geospatial element of the real world, and topological spatial relations; GML Specification, which introduces the different aspects of GML, including benefits, schemas structure, capabilities.

Simple Features

Firstly the geometry class hierarchy will be introduced to discuss the structure and overview of the geometry classes. Follow with that, a figure that show the operations on geometry classes will be given. And last but not least, I will put weight on the Egenhofer point-set topological spatial relations.

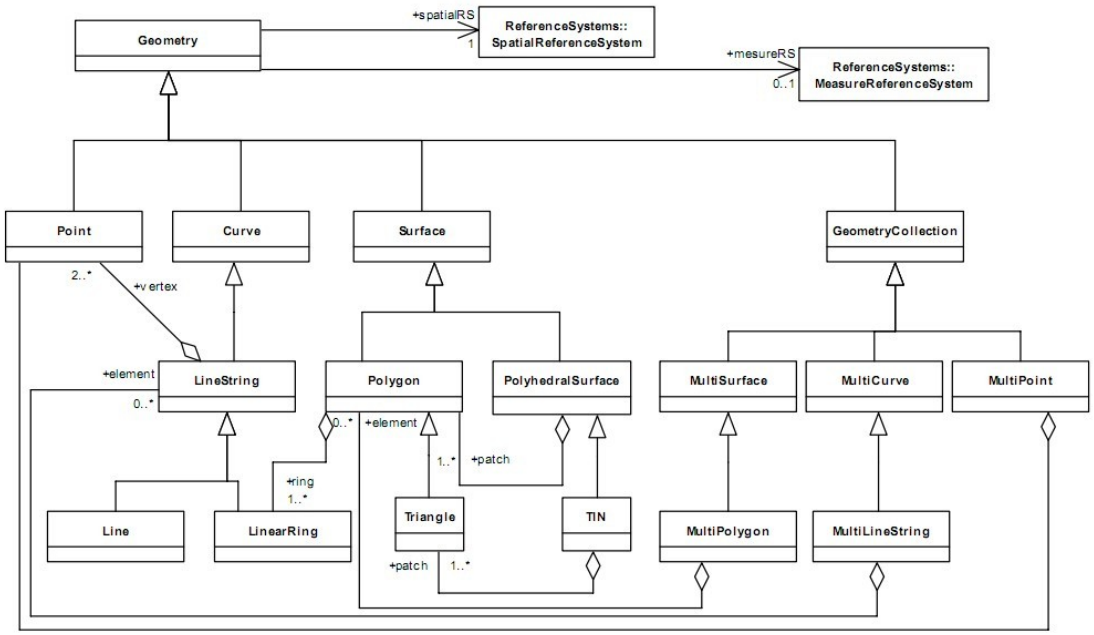


Figure 6: Geometry class hierarchy [8]

From the figure 6 we can see that the root class of this hierarchy is Geometry. It is an abs-

tract class, and can be divided into two parts: base geometry and extended geometry. The base geometry class has subclasses for Point, Curve, Surface and GeometryCollection. The extended geometry class has subclasses named MultiPoint, MultiLineString and MultiPolygon for modeling geometries corresponding to collections of Points, LineStrings and Polygons, respectively. MultiCurve and MultiSurface are not included in the extended geometry class because they are abstract classes. They are superclasses that generalize the collection interfaces to handle Curves and Surfaces. In the model, the triangle symbol represents the 'super-sub class' relationship, and the diamond symbol represents the 'consist of' relationship. The Point is the smallest element of geometry, it doesn't consist of anything, but two geometry classes directly consist of it: LineString and MultiPoint. The LineString class is the only subclass of the class Curve, and it has two subclasses: Line and LinearRing; The MultiLineString class consists of LineString, and it is the instantiation of the subclass of the MultiCurve abstract class. A Polygon is one subclass of Surface class, the other subclass is PolyhedralSurface. Polygon class consists of LinearRings and it has subclass called Triangle. PolyhedralSurface is a special collection of polygons, and it has subclass called TIN which consists of Triangles. And MultiPolygon consists of Polygons and it is the instantiation of the subclass of MultiSurface class. MultiPoint, MultiCurve and MultiSurface are all subclasses of GeometryCollection class. The geometric objects have dimension, where point is 0 dimension, line/curve is 1 dimension, and polygon/surface is 2 dimension. For all the instantiations of the geometry classes, the specific descriptions (the geometry dimension, definition, simple type, boundary, etc.), methods are defined, seeing at OGC Simple Feature specification. For example [8], A curve is a 1 dimentional geometric object usually stored as a sequence of Points, with only one subclass LineString. A lineString is a Curve with linear interpolation between Points, a Line is a LineString with exactly 2 Points. The LineString(curve) is simple if it doesn't pass through the same Point twice with the possible exception of the two end points. It is closed if its start Point is equal to its end Point. It has methods: NumPoints(), PointN(N:Integer).

Figure 7 is a list of possible operations to the geometry class. The first part is the basic methods on geometric objects, the methods can implement on the geometric objects to get back the information of the objects. The second part is the methods for testing spatial relations between geometric objects, geometric objects are connected and their relations are very important. The third part are the methods that support spatial analysis.

Among the geometry class operations, the ones for the spatial relations are significant. The topology validation will be a big percentage of the system, and the spatial relations are the key to understand the topology issues.

Point-Set Topological Spatial Relations

Point-set topology is based upon point-set geometry, and is concerned about how these sets relate to each other [5]. The most famous point-set topological spatial relations was proposed by M. J. Egenhofer and R. D. Franzosa, it is mathematical theory, designed to categorize binary spatial relations. Before the Egenhofer point-set topological spatial relations, most of the previous work describe the spatial relations as the results of binary point-set operations. But none of them has been performed systematically enough to be used as a means to prove that the relations defined provide a complete coverage for the topological spatial relations between two spatial objects [9]. And those previous works more or less all have some drawbacks, such as incomplete, or relation

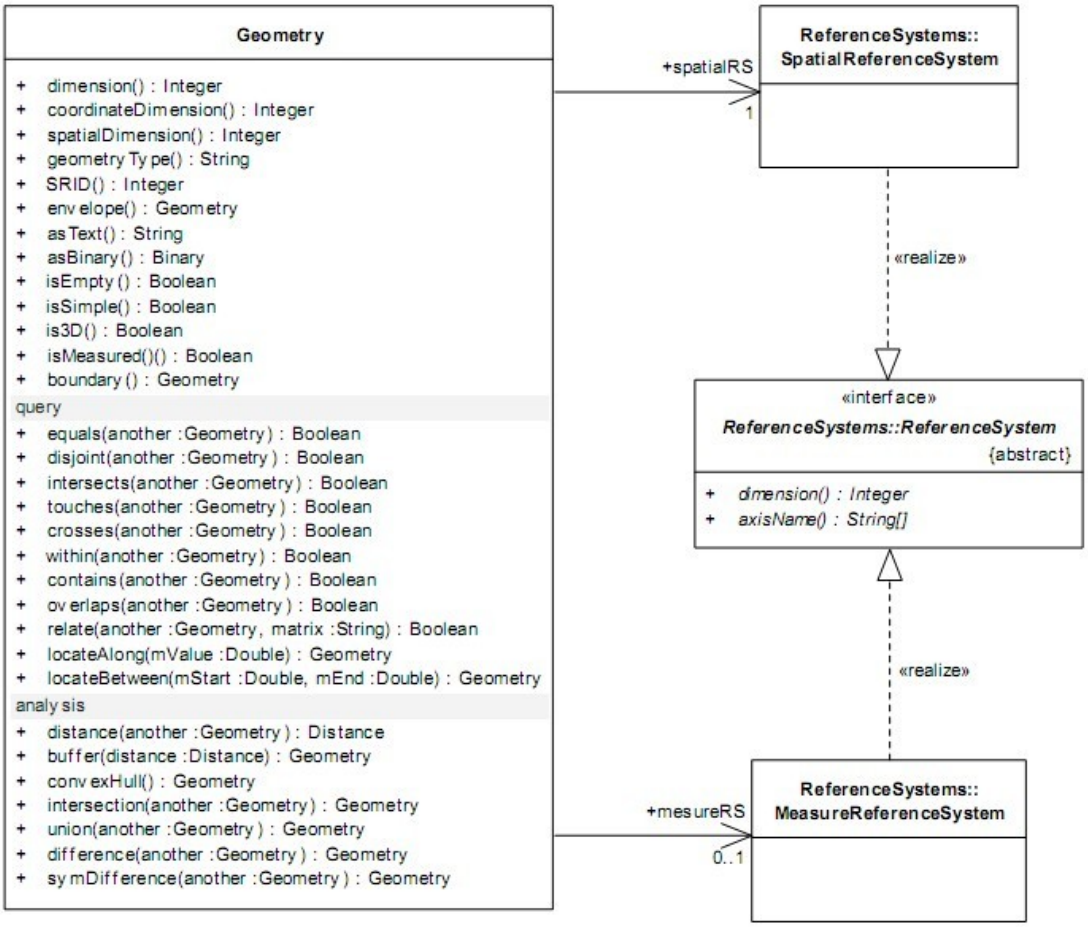


Figure 7: Geometry class operations [8]

categories intersect. Egenhofer created the 9 intersection model, which is largely used in GIS for developing of software and spatial database systems.

In the OGC Simple Feature specification, the dimensionally extended nine-intersection model is addressed. In this model, the spatial relations are based on the intersections of interior, boundary, and exterior between two geometric objects. Using *I(), B(), E()* to represent interior, boundary, and exterior respectively, and *dim(x)* to return the maximum dimension (-1, 0, 1, or 2) of the geometric objects in x, with a numeric value of -1 corresponding to empty set. The general form of the dimensionally extended nine-intersection matrix is shown in Figure 8, with graph illustration in Figure 9.

	Interior	Boundary	Exterior
Interior	*dim(I(a)∩I(b))*	*dim(I(a)∩B(b))*	*dim(I(a)∩E(b))*
Boundary	*dim(B(a)∩I(b))*	*dim(B(a)∩B(b))*	*dim(B(a)∩E(b))*
Exterior	*dim(E(a)∩I(b))*	*dim(E(a)∩B(b))*	*dim(E(a)∩E(b))*

Figure 8: The DE-9IM. [8]

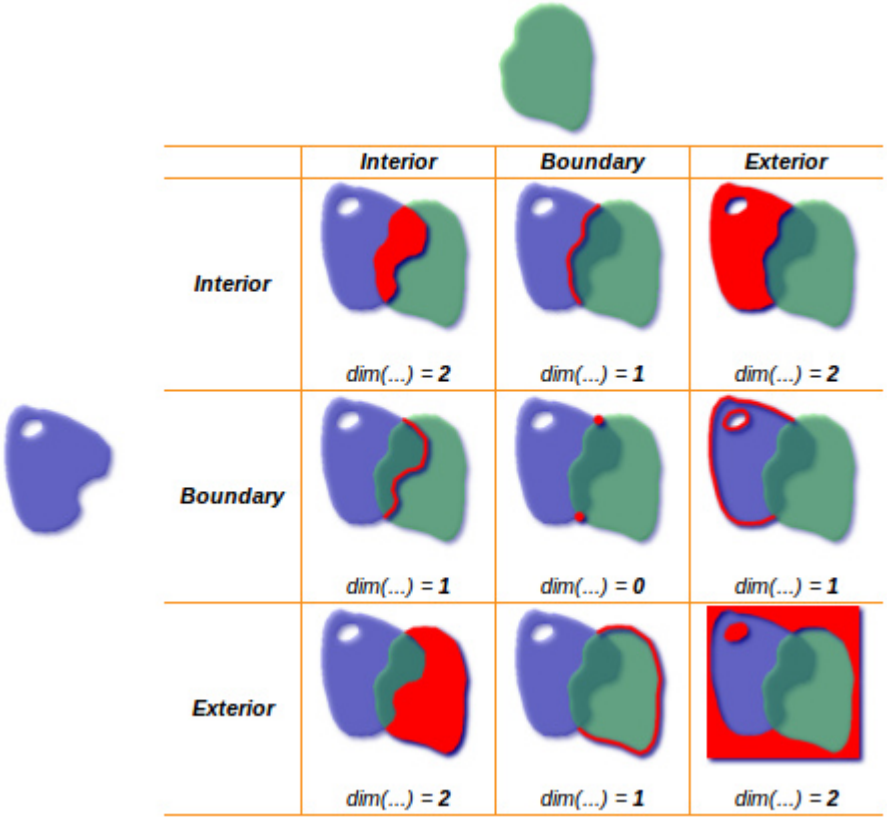


Figure 9: Nine-Intersection Model. [8]

The spatial relations mentioned earlier in the geometry class operations, equals, disjoint, intersects, touches, crosses, within, contains, overlaps, and relate, can be equivalent to different combinations of the intersection terms based on DE-9IM.

For example, $objecta.Equals(objectb) \Leftrightarrow$
$((I(a) \cap I(b) \neq \emptyset) \wedge$
$(I(a) \cap B(b) = \emptyset) \wedge$
$(I(a) \cap E(b) = \emptyset) \wedge$
$(B(a) \cap I(b) = \emptyset) \wedge$
$(B(a) \cap B(b) \neq \emptyset) \wedge$
$(B(a) \cap E(b) = \emptyset) \wedge$
$(E(a) \cap I(b) = \emptyset) \wedge$
$(E(a) \cap B(b) = \emptyset) \wedge$
$(E(a) \cap E(b) \neq \emptyset))$
$\Leftrightarrow a.Relate(b,'TFFFTFFFT')$

The spatial relations are between two geometric objects, the methods will return boolean values, true if the relation between the two objects is correct. For example, a.overlaps(b) will return true if a does overlap b, and false if a doesn't overlap b. For the topologies which are about the geometric object itself, then the methods of geometry itself would be applied. For instance, to check if a line 'A' is self-intersected, we can apply A.isSimple(), and it will return boolean value.

Importance of topology

In a Geographic Information System (GIS), topology is a set of rules which define the relationship between points, lines, and polygons [10].The word topology has several meaning when discussed in the GIS context [11]. It can refer to the following:

- Theory or mathematical model of features in space
- Mechanism that allows features in the same or differnt feature classes to share geometry
- Set of editing tools that works with features in an integrated fashion
- Physical data model for feature data
- Set of validation rules for geographic features
- Mechanism for navigating between features using topological relationships

Topological geometry does not depend on a coordinate system. It is concerned about geometrical property which are unaffected by continuous change of shape or size of figures [5]. It is significant for data processing and spatial analysis. For the road network application, topology reflects the logic structure between the features, which is important for the connectivity between road features, and therefor for the path selections and navigation. Additionally,the topology relationship is helpful for building the new features and for decision making, such as route planning.

GML Specification

Geography Markup Language is an XML grammar defined by the Open Geospatial Consortium (OGC) to express geographical features. GML serves as a modeling language for geographic systems as well as an open interchange format for geographic transactions on the Internet [12]. This chapter will discuss different aspects of GML format, including 'GML and XML', 'the benefits of using GML', 'GML schemas',and 'GML capacities'.

GML and XML

Because GML is based on XML, it leverages a wealth of standards, tools and practices for data exchange being developed by several consortia around the world. The XML technology family includes the following [13]:

- for encoding and data modeling expression (DTD, RDF AND XSD)
- for linking and associating resources (XLink)
- for selecting and pointing (XPath, XPointer)
- for transforming content (XSLT)
- for graphic rendering (SVG, VML, X3D)

All these standard technologies can be directly adapted by GML. The GML employs the schema, an XSD file, for expressing the data structure of a spatial domain. In both schema and the instance files, XLink and XPath/XPointer are used. The XSLT can be applied for styling the GML file, and SVG for visualizing. Besides these, the external technologies developed for the XML can be straightly used by GML, such as parsing, querying, and validation, even though sub-differences may exist.

The benefits of using GML

Being a subset to XML, the GML inherits the advantages of XML. This can bring many benefits of using GML. Some most important ones are stated below [14].

- Better quality maps. GML encodes information about geographic features or objects, and these can be displayed as fine a resolution as required.
- Works on a browser, without the need to purchase client-side software.
- Editable maps
- More sophisticated linking capabilities
- Service chaining, such WFS [1]

Geospatial applications can benefit from GML's robust functionality and the technologies that enhance it [15]. And that is why GML is becoming the standard exchanging worldwide. Several obstacles hindering GML usage, however, must be overcome so that maximum usability can be achieved, such as the challenging to extract information from GML documents due to time constraints and application complexity [15]. But GML is a promising language, its full potential have not yet been fully developed.

[1] Web Feature Service

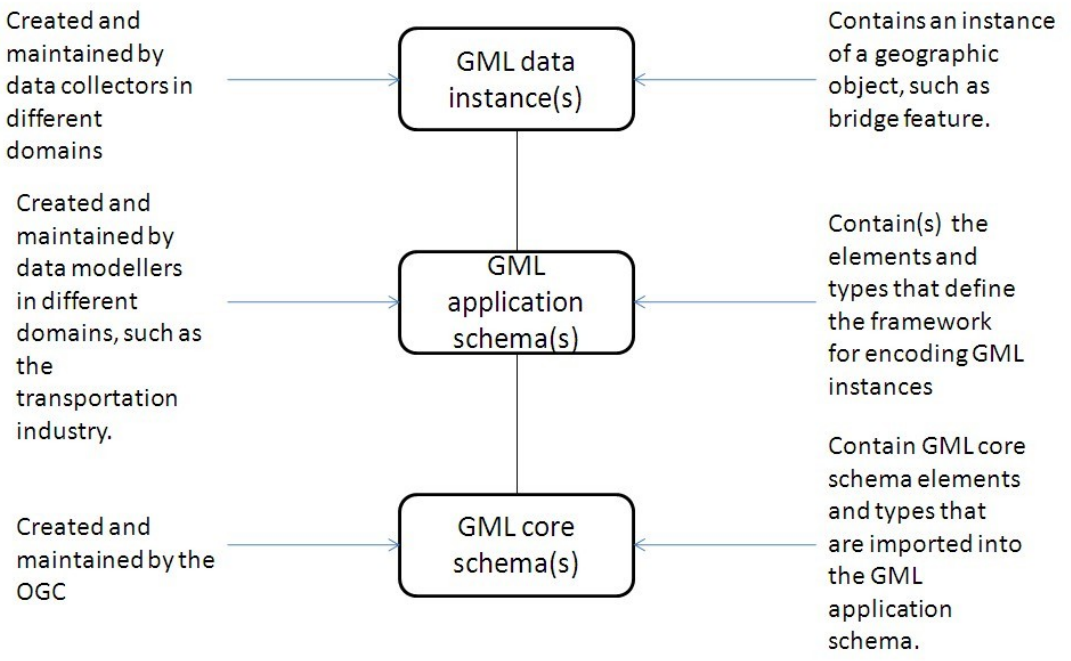


Figure 10: GML instance, application schema, and core schema. [16]

GML schemas and capability

There are two parts to the XML grammar - the schema that describes the document and the instance document that contains the actual data. Figure 10 shows the structure and relations among GML instance, application schema, and core schema.

GML core schema includes the schemas on subjects of feature, datums, coordinateReferenceSystem, coverage, dynamicFeature, measures, temporal, etc. These schemas are the common geospatial constructs. Core schema contains the necessary elements for creating application schema. In other words, no matter what domain the application schema is trying to express, it always need to select certain core schemas for structuring. And the core schema contains relatively complete common subjects that the application schema may need.

The application schema built on GML core schema. The core schema has defined the common constructs and constraints, and the application schema then chooses relevant core schemas, describe more specific feature structure, relationships and constraints according to the application. In GML 3.0.0, there are 27 core schemas. So for example, to create a road domain application schema, assume that 10 of the 27 core schemas are relevant to this application, then these 10 core schemas will be included in the application schema, in addition, the application schema will describe the structure and constraints according to the road domain specification, such as road feature must have roadName, coordinate, roadID attributes.

The GML instance is the file with concrete information encoded complying with application

schema. The GML instance according to the road application schema will contain the real road information, for example, 'roadID:1001, roadName:Wall street, coordinates: 10 10 10 11 10 12 10 13'.

So the GML instance follows the rules defined in the application schema, and it is more concrete than the application schema; the application schema imports the core schema for expressing the data model of a domain, it is more concrete than the core schema; and the core schema defines the most common subjects to construct all the application schema.

The core schema also implies the GML capabilities. GML objects can represent profound geospatial information, including 'features', 'geometric primitives', 'geometric complex', 'geometric composites' and 'geometric aggregates', 'coordinate reference system', 'topology', 'temporal information and dynamic features', 'definitions and dictionaries', 'units, measures and values', 'directions', 'observations' and 'coverages' [17]. The figure 11 shows the hierarchy of these GML classes.

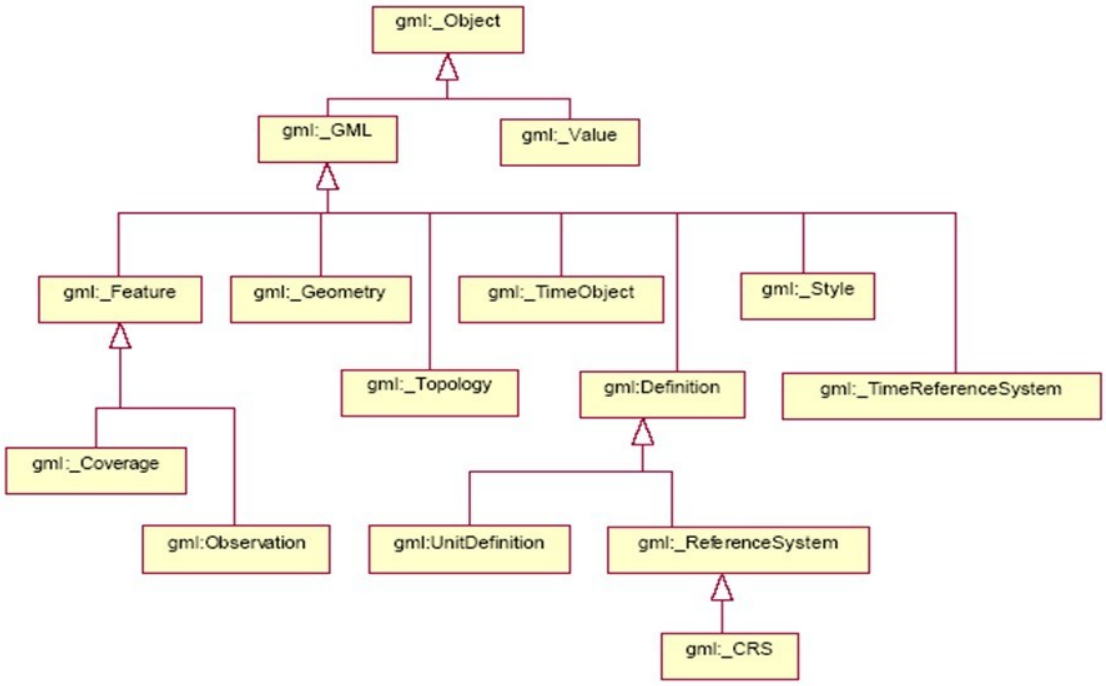


Figure 11: GML Class Hierarchy

The specific encoding rules for each GML class can be seen at OpenGIS Geography Markup Language Encoding Standard.

2.2 Geographic Data Quality Issues

In our real life almost everything that happens, happens somewhere. Knowing where something happens can be critical [4]. GIS can be used in almost all the cases that have something to do with location. GIS is meant to solve real world problems, with GIS many of our day-to-day

working and living arrangements can be improved. In the book Geographic Information Systems and Science by P. A. Longley et al., the reasons that GIS is used by more and more individuals and organizations are given.

- Wider availability of GIS through the Internet, as well as through organization-wide local area networks.
- Reductions in the price of GIS hardware and software, because economies of scale are realized by a fast-growing market.
- Greater awareness that decision making has a geographic dimension.
- Greater ease of user interaction, using standard windowing environments.
- Better technology to support applications, specifically in terms of visualization, data management and analysis, and linkage to other software.
- The proliferation of geographically referenced digital data, such as those generated using Global Positioning System (GPS) technology or supplied by value-added resellers (VARs) of data.
- Availability of packaged application, which are available commercially off-the-shelf (COTS) or 'ready to run out of the box'.
- The accumulated experience of applications that work.

However, without good quality geographic data, none of above can be well served. Data quality is the guarantee of the efficiency and convenience that GIS brings to solve problems. Bad quality data will directly lead to the final product being unusable, or wrong decisions and plans, thus the initial purpose of GIS won't be fulfilled. On the other hand, data quality as one primary parts in SDI, it can improve the efficiency of data sharing. The users or organizations search the data through the portal by SDI and build up services based on the data. If the data quality of the datasets is unknown or bad, it will largely influence all the further usages of the datasets, waste time and human efforts and have bad results. So data quality is essentially vital to GIS, and geographic data quality control is necessary.

Clarifying the geographic data quality issues is the first and important step for developing the validation framework and controlling data quality. The quality for data is much more difficult to define than for a manufactured product, because the data has no physical characteristics that allows the quality to be easily assessed. Data quality is thus a function of intangible properties, such as 'completeness' and 'consistency' [18]. The data quality can be divided into quantitative and non-quantitative quality issues. Information about the purpose, usage, and linage of a dataset is non-quantitative quality information [3]. This thesis is mainly about the quantitative geographic data quality, so the next two sections 2.2.1 2.2.2 are also with regard to the quantitative quality information.

2.2.1 General inconsistency problem

From the general view, all the geographic data quality issues can be classified as the discord problem between the geographic data and the geographic part of real world.

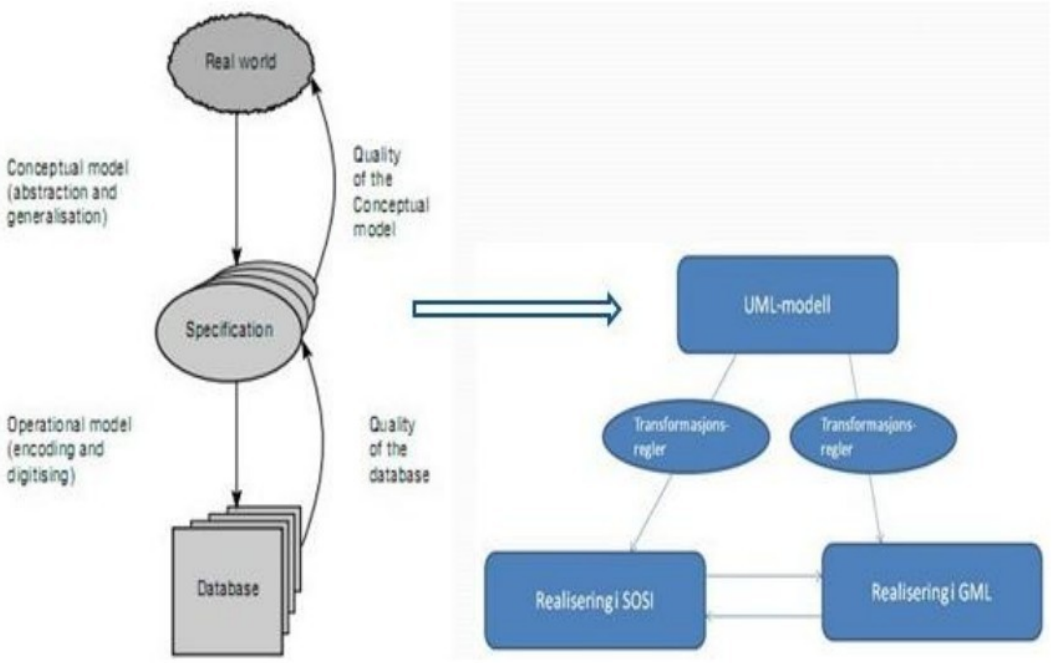


Figure 12: Consistency among reality, conceptual model and dataset.

From the introduction of geographic data modeling, we know that the digital geographic data was not the real geographic world, but the part of it which is considered essential. In Figure 12, we can see a three levels structure of geographic information: real world, specification and dataset. For a dataset, being consistent to the real world means that the dataset is consistent to the specification, and the specification is consistent to the real world. This project is about the conformance between specification and the dataset, as shown in the right part of the figure. The specification can have errors comparing with the real world by modeling, however that is another problem domain concern good modeling. So the pre-assumption of this validation system is that the specifications cover the real world of interests perfectly, that only the consistency between the dataset and the specifications is on concern.

The middle file, that is between specification and dataset, describe UML model's content and structure in a way that dataset can understand. For GML, the middle file is GML schema. GML schema is also a document that based on XML grammar, thus the GML file can naturally understand the schema. The mechanism for validating XML file against its schema has already been built for years, which means, if a GML file is validated against its GML schema, xsd file, then that GML file is conformant to the schema, the middle file. However, there is no insurance

that the middle file can cover all the information, constraints in the specification. Due to the limit of the language itself, some information is difficult or impossible to express. For example, one specification can define that there should not be an intersection on roads if the roads don't have intersection in real life. This constraint is to avoid fake intersection, one typical instance can be the overline bridge, two roads may share the same location coordinates, but they are with different heights and don't intersect. Constraints or information like this are usually omitted in GML schema.

So there is no solution to structure, include and handle all the requirements in an GML schema. Therefor the GML file is not completely conformed to the specification, even thought it is conformed to its schema. That is also the main reason why the existing XML validation system is not enough for GML validation. Besides the structure and definition consistency problem, the discord between the dataset and the real world can be the attribute value. Even if the objects element is correctly modeled, its value can be an error.

2.2.2 ISO geographic data quality elements

The section 2.2.1 addressed the general quality problem for a geographic dataset, however the general inconsistency problem can include many different aspects problems. For example, accuracy, precision, consistency, and completeness are regarded as four components of data quality, with each of the component being differentiated in space, time and theme [18]. In this thesis, I will introduce the geographic data quality elements defined by ISO/TC 211 standards. ISO/TC 211 is a committee of the ISO concerned with geomatics standards. The data quality elements that will be introduced are about the quantitative quality. Clarifying and classifying the geographic data quality problems with the data quality elements defined by ISO standard can naturally lead to that the GML validation requirements will conform to the ISO standard, since the validation requirements must be accorded to what are the quality problems. ISO standards are mostly universally accepted, and might lead to the GML validation framework to be more widely useful.

Figure 13 shows the components of the geographic data quality, and their sub-components, with corresponding definitions. Below a short explanations to all the data quality elements with examples is given. The complete and specific description can be seen at ISO/TC 19113 geographic information – quality principles, ISO/TC 19114 geographic information – quality evaluation procedures, and ISO/TC 19138 geographic information – data quality measures.

Completeness consistency data quality element

The data quality element, completeness, has two subelements which are commission and omission. They concern about if there are excess data present in a dataset and if there are data absent from a dataset respectively. How can one know that if there is completeness consistency problem in the dataset? The answer is to compare the dataset on questions with a dataset that contains data which is regarded to be true or correct. For this kind of data quality element, an external dataset is needed to evaluate. By comparing the dataset and the external dataset, we can count the number of excess data and absent data, and report them.

For instance, there is one dataset that contains the correct number of the roads in Gjøvik. The dataset with unknown description of completeness consistency, can be compared with the true dataset, using any proper method, such as graphic detection. And then count the excess data

data quality element	data quality subelement	definition
completeness	commission	excess data present in a dataset
	omission	data absent from a dataset
logical consistency	conceptual consistency	adherence to rules of the conceptual schema
	domain consistency	adherence of values to the value domains
	format consistency	degree to which data is stored in accordance with the physical structure of the dataset
	topological consistency	correctness of the explicitly encoded topological characteristics of a dataset
positional accuracy	absolute or external accuracy	closeness of reported coordinate values to values accepted as or being true
	relative or internal accuracy	closeness of the relative positions of features in a dataset to their respective relative positions accepted as or being true
	gridded data position accuracy	closeness of gridded data position values to values accepted as or being true
temporal accuracy	accuracy of a time measurement	correctness of the temporal references of an item (reporting of error in time measurement)
	temporal consistency	correctness of ordered events or sequences, if reported
	temporal validity	validity of data with respect to time
thematic accuracy	classification correctness	comparison of the classes assigned to features or their attributes to a universe of discourse (e.g. ground truth or reference dataset)
	non-quantitative attribute correctness	correctness of non-quantitative attribute
	quantitative attribute accuracy	accuracy of quantitative attributes

Figure 13: ISO data quality elements[19]

(the road doesn't exist in Gjøvik) and the absent data (the road exists in Gjøvik, but not included in the dataset). Assume that there are 5 excess road objects and 3 absent road objects, with total 100 road objects in the dataset, then result can be reported that 5 excess, 3 absent; or 5 percent excess, 3 percent absent.

Logical consistency data quality element

The data quality element logical consistency, has four subelements which are conceptual consistency, domain consistency, format consistency and topological consistency. Logical consistency is an important aspect of geographic data quality. Consistency is the basis and precondition to valid implementations on the dataset.

Conceptual consistency means the compliance between the dataset and the conceptual model, the dataset has to follow the rules explicitly or implicitly defined in the conceptual schema, for example, duplication of features and invalid overlap of features.

Domain consistency refers to the adherence of values to value domains. The value domains are the acceptable attributes defined, for example, if the road feature is defined having the attributes: road ID, road name, start point, end point, and road surface; and in one dataset, one road object is give attribute 'leading to', then this is a violation of domain consistency. The GML schema can describe the attribute domain.

Format consistency indicates the degree to which data is stored in accordance with the physical structure of the dataset. The structure of the objects within an domain can also be described in the GML schema. E.g. the physical structure of an object is 8 bits integer, if it is given float type in the dataset, then it goes against the format consistency.

Topology consistency means the correctness of the topological characteristics of a dataset. Topology has been stated in section 2.1.2. It is about certain constraints on the geometry relations, for example, two lines cannot overlap, the start point and the end point of a polygon must be the same. Topology rules cannot be covered by the GML schema.

Positional accuracy data quality element

Accuracy refers to the conformance between encoded and actual value of a particular attribute for a given entity [18]. When talking about positional accuracy, it refers to conformance between encoded and actual value of the positional attribute for a given entity. The most important part in the definition is 'encoded and actual value'. The encoded value means the value in the dataset we want to validate, and the actual value means the values accepted being true. And the accuracy is based on the comparison between the data in the dataset for measuring and the data in the real world or being accepted as true. The positional accuracy data quality element includes three subelements: absolute or external accuracy, relative or internal accuracy, and gridded data position accuracy. But no matter which subelement, the external dataset is demanded for comparing the positional accuracy.

One result reporting example (Table D.4 given in ISO/TC 19114) can be referred [20]. The positional accuracy checking is about absolute or external accuracy of all the nodes forming road boundaries in one dataset. The percentage evaluation is that for each node, measure the error distance between absolute coordinates values of the node in the dataset and those in the universe of discourse, count the number of the nodes whose error distance exceeds the specification limit

(e.g. 1m), divide the number of the non-conforming nodes by the number of the nodes in the data quality scope, multiply the result by 100, the result can be for example 10 percent. And the other evaluation can be just measure the error distance between absolute coordinate values of the node in the dataset and those in the universe of discourse, and the result can be for example 1.5m.

Temporal accuracy data quality element

Temporal is about time. Temporal accuracy refers to three aspects: accuracy of a time measurement, temporal consistency, and temporal validity. The temporal information is often omitted. And the implications of this omission are potentially quite significant, especially for the features with a high frequency of change over time [18]. Being able to check if the time is valid or time sequence is correct, we need to know when is the correct time and the time sequence. By analyzing the dataset itself, we will not know if the time of an item is accurate or not. Thus external temporal information is needed.

If a bridge was built in 1998, and in the dataset it was given value to 1999, then there is a discord between the value in dataset and the true value in the universe of discourse. In addition if the value was given by mistake that it is 2998, which is in the far future, it is not even valid. If one road was built in 1950, and after that, it was repaired and rebuilt several times, then the time sequence of each change should be correct.

Thematic accuracy data quality element

Thematic accuracy has three subelements: classification correctness, non-quantitative attribute correctness, and quantitative attribute accuracy. The thematic accuracy refers to a wide various sides of a dataset, both quantitative and non-quantitative attributes. Temporal and positional attributes are subsets of quantitative attribute, we can say that if the dataset has positional accuracy problem or temporal accuracy problem, the dataset also has thematic accuracy problem, but not in opposite.

Here two examples are given. If one dataset is about the property information, but was classified as land registration, then this belongs to thematic accuracy quality problem (non-quantitative). If one road is under status of 'road to be built', but was set attribute of 'Existing road', then this is also a thematic accuracy problem (quantitative).

3 Related Work

This chapter consists of 4 sections, which introduce four existing programs in the data quality control field. The descriptions and discussions will place the emphasis on the capacities, user cases of each program, as well as the reasons why they are not sufficient for the requirements of the GML validation, and also how can they contribute to further develop the GML validation.

3.1 XML validation

XML[1] is a set of rules for encoding documents in machine-readable form. It is defined in the XML 1.0 Specification produced by the W3C[2] [21]. XML validation is the process of checking if a document is written in XML to confirm that it is both 'well-formed' and also 'valid' in that it follows a defined structure [22]. The definition implies the two main capacities of XML validation: well-formed and valid structure.

A 'well-formed' document follows the basic syntactic rules of XML, which are the same for all XML documents [22]. And these basic syntactic rules of a well-formed XML document are defined by the W3C.

The official syntax rules of well formed XML documents [23]:

- Every open tag has close tag
- The tags must nest correctly
- The name of elements must be exactly correct, because XML is case sensitive
- XML attribute value must be quoted
- Must have root element

A valid structure respects the rules dictated by a particular DTD or XML schema [22]. In the XML technology family, DTD[3] and XML schema are the files that define and control the structure of an XML document. DTD is a non-XML document, while XML schema is written in XML grammar. There are no explicit descriptions on what specific aspects can be validated against a DTD or schema. But by personal testing, some important abilities are listed below:

- Check if all relevant schemas are included
- Only the element that has been defined in the schema can appear in the instances
- Only the attributes to one element that has been defined in the schema can appear in the instances

[1] Extensible Markup Language
[2] World Wide Web Consortium
[3] Document Type Definition

- The range and type of the value to an attribute must conform to the definitions in the schema
- The number of occurrences of a element must conform to the constrain
- The elements must have correct relations as defined
- Check coordinate reference system

In addition rules that the well-formed validation can check against were found out by the testing, and worth to mention here:

- Undefined namespace can't be used
- No multi-headers, no extra content
- Must have correct XML title, version and character set

All XML documents have the same requirements for being valid. The XML validation is the basic quality guarantee to XML files. GML, as an XML grammar document, can directly use the XML validation. But XML validation is not enough to validate GML files. As mentioned in section 2.1, the schema can not cover all the requirements the conceptual model defines, thus extra rules must be checked in addition to the schema validation.

According to the geographic data quality elements defined in ISO/TC 211 standards, the logic consistency element consists of four sub-elements, domain consistency and format consistency are two of them. The concepts and explanations were described in section 2.2.2, and XML validation can do the job of validating these two quality elements. The XML validation can be utilized to the GML validation system completely and directly. Currently, almost all programming languages have existing solutions for XML validation, so that the extra GML constrain functions can be built as extension to the XML validation. The XML validation has the characteristic that when an error is detected by the XML validation, the program will stop. So one needs to fix the error and run the XML validation again until there is no error. Therefor, in the GML validation process, the user still needs doing so before running the extra functions.

3.2 SOSI-Control

SOSI-Control is the current software used in Norway for controlling the quality of SOSI file against certain product specification. It is made from scratch as a student project in Gjøvik in 1992. After years of further development, The SOSI-Control software is a necessary tool for data quality control and maintainance. It is regarded that it is sufficient to control the conformance between SOSI files and corresponding product specification.

The SOSI-Control has 6 main controlling catalogs: format checking, content checking, node checking, surface control, object control and statistics. Each catalog consists of a number of sub controlling aspects [24]. A simple description about the SOSI-Control requirements is given below, the detail can be seen in the user help document of SOSI control software.

- A. Format Checking
  - A-1. Header
    - Header should include all obligatory information
    - The syntax of the header information should be correct
    - No undefined basic element can occur
  - A-2. Multiple Headers: Only one header can be in a dataset
  - A-3. Character set
  - A-4. Ending: Must have an ending; Give warning to the content after ending tag
- B. Content Checking
  - B-1. Unique serial number
  - B-2. Boundary: The coordinates of the elements in all groups should in the boundary defined in the header
  - B-3. SOSI level
  - B-4. Accuracy and unit
  - B-5. Height information: three kinds of coordinates: coordinates with same height, different location; coordinate with height information; coordinate without height information, have to be in correct format respectively, and can't mix in one group.
  - B-6. SOSI-syntax
- C. Node checking
  - C-1. Nodes
    - Statistic information about the points marked with KP(node) in the current file
    - Group the nodes in team by node type, coordinate type, and other catalogs
    - Check all the blind nodes, which are the nodes only connect to one feature
  - C-2. Group the nodes that can join together
    - Search all the 2'er (connect to two features) nodes and groups that have similar characteristics
    - Present a list about all the groups that can be join in one group
  - C-3. Connecting only points: present a list with all the points that can be marked as node by giving different tolerance value.
- D. Surface control: leave out
- E. Object control
  - Classify the group elements of SOSI file in object classes (2.0->) or domain code (1.4) and output the statistics

- Give error message if a group element was not given to any object class
- Count the sum of the group elements that are distributed to more than one object classes
- The objects contains obligatory basic elements
- The basic elements have legal values
- Basic elements only occur once in the group element, if the definition is not multiple.
- The objects can't contain illegal basic elements according to the object definitions
- Geometry type checking
- Multiplicity and cardinality (minimum and maximum)
- If there are double lines and curves (line or curve been defined twice)
- Surface (polygon) boundary

- F. Statistic
  - Objects
  - The amount of points per line or curve
  - Average line length
  - Point density
  - Point specifics on domain code
  - Quality code: combination of different information
  - DEK indicator

So from the capacities view, SOSI-Control is good enough for consistency control under Norwegian standard. But there are some reasons leading to that the GML validation can't direct copy the requirements from SOSI-Control.

Different geographical data formats

The SOSI data format was invented before the XML technology, and it has been developed with several versions. Due to the unique encoding way, some requirements in the SOSI-Control are only for the characteristics of the SOSI file, for example SOSI version checking, and coordinate format that with same height value, different location value. So even though different geographic data formats share the same conceptual model, product specification, the quality control requirements must consider the features of the format itself.

GML format, as one kind of XML format, has the natural properties of an XML file. XML has its own characteristics as well, for example the definitions of well-formed can be different between SOSI file and XML file. And XML has already developed a standard way of validating the formed and structure. So it is not a good idea to abandon the existing property advantages of a GML file, and copy the requirements from SOSI-Control that are made according to features of SOSI format.

SOSI-Control lacks of certain conformance to ISO standards

As mentioned SOSI format was created before the current relevant technologies. So the SOSI-Control classify the quality problem in its own way, in the light of the definitions from product specification and some extra important rules. There was no international standard to conform with. So the function catalogs wrote earlier are not from the view of data quality in ISO standard, but in a more specific level, such the object types node, surface. Thus the historical reason leads to that the SOSI-Control can accomplish the mission of controlling the consistency between SOSI dataset and its product specification, but lack of certain conformance to ISO standards. Then we can't generally say that SOSI-Control can cover which data quality element in ISO standard.

The requirements for the GML validation must first be adjusted by the own features of the GML format, and also be conformed to the ISO standards. In addition, even though SOSI-Control and GML validation are both defined as geographic data quality control tools, they do differ in some way. SOSI-Control can check the dataset against some rules, as well as searching and reporting some useful information for further manual implement and decision. For example, 'group the nodes in team by the node type' and 'present the point density', they are not about data mistakes, but a clearer display of the concerned content, and this can be handy for later analysis and implementations. But for a validation system, the focus is that if the data comply with defined rules. So these kind of requirements defined in SOSI-Control may not be necessary as requirements for GML validation.

But the SOSI-Control can still be utilized for defining the GML validation requirements, as well as good experiences of how a quality control tool works. As the SOSI-Control has been developed for around 20 years, the requirements for checking SOSI datasets have been improving, we can consider that the requirements defined in SOSI-Control equal to the requirements for consistency control between Norwegian product specification and dataset. The GML validation is also aiming to control the inconsistency problem, but also need to take some extra parameters into consideration, such as the ISO standards. So the SOSI-Control is a shortcut to determine the necessary requirements from the product specification. The GML validation requirements can use the SOSI-Control requirements by some implementations, such as resorting, deleting, and complementing. For example, the requirement that 'basic elements only occur once in the group element, if the definition is not multiple', defined as object control, can be resorted to XML validation; the SOSI level checking can be deleted; and extra topology constrains can be complemented.

3.3 ESRI ArcGIS

ESRI is a software development and services company providing Geographic Information System software and geodatabase management applications. It was founded as Environmental Systems Research Institute in 1969 as a land-use consulting firm [25]. Now ESRI is a leading commercial company globally in geospatial area. ArcGIS is an integrated collection of GIS software products. It provides a standards-based platform for spatial analysis, data management, and mapping [26]. In ArcGIS, the *Validate Topology* capability will ensure data integrity by validating the features of a geodatabase against a set of topology rules [11].

How does it work

To be able to create the topology rules in arccatalog, you first have to make sure that the data format is geodatabase, and the feature classes for validation need to be in the same feature dataset, because all participating feature classes must have the same spatial reference [11]. In Geodatabase, the geospatial data is organized as data objects, and these data objects store in feature class, object class or feature datasets. The object class is used to store the non-spatial information. The feature class is used to store the spatial information and corresponding attribute information, in one feature class. The spatial geometry must be the same, for example, all the points, all the lines and the feature dataset is used to store the feature classes with the same spatial reference [27]. After importing the feature classes in the feature dataset, the user can create certain topology rules, and do the topology validation. Then the user can use the topology toolbar in ArcGIS to modify and correct the files according to the topology mistakes returned by the validation process, for instance, delete duplicated lines.

The topology rules

Compare with the example topology consistence rules given in the ISO/TC 211 standards, the topology rules that ESRI concludes are more complete. They have created a set of topologies covering different application cases [28]. The topology rules with descriptions and examples can be seen in Appendix A.

The ESRI ArcGIS includes complete implementations to topology, includes editing features in topology, create topology rules, topology validation and part topology correction and more. It is more for analyzing and further implementation purpose than dataset quality control. It only concerns on topology elements, the other data quality elements defined in ISO standard are not touched. So the ESRI ArcGIS software is only designed for the circumstance of topology problems, while the GML validation will cover a wider quality problems. In addition, one more difference between the ArcGIS topology functions and the GML validation is that the former one demands certain manual operations during the process such as the works before creating the topology rules and doing topology validation, and the later one will give almost all the works to the validation system, the user only needs to assign the GML dataset and select the relevant topology rules. The huge contribution from the ESRI ArcGIS to the GML validation system is the relatively complete set of topology rules, which can be included in the GML validation requirements on topology consistency element.

3.4 Geoconnections: Validating Web Feature Server

Geoconnections is recognized around the world for its prominent role in building the Canadian Geospatial Data Infrastructure. The CGDI [4] provides services for viewing, retrieving and analyzing spatial information. It is built using open standards from the OpenGIS Consortium that allow the various CGDI systems to share data and maps transparently [29]. The OGC standards conformed by Geoconnections include GML, WFS [5], simple features, and so on. And one project of Geoconnections is validating Web Feature Server. The project has two main goals, which are to enhance GeoServer, the free, open-source server software that allows users to access geogra-

[4] Canadian Geospatial Data Infrastructure
[5] Web Feature Service

phic information over the web, and to provide a mechanism for checking, or validating, on-line edits to geospatial databases to make sure the edits are clean and error-free [30]. The OGC Web Feature Service provides an interface allowing requests for geographical features across the web using platform-independent calls [31]. Unlike the Web Map Service which returns the user an image, the WFS returns an editable file, GML file, to the user. Thus the user can use the GML file returned from WFS for editing, spatially analyzing, and such implementations. And WFS-Transaction can even allow the user send the edited file back and restore it in the server. The validating Web Feature Server is an extension to the existing GeoServer reference implementation of the Open GIS Consortium's Web Feature Server Specification, the extension consists of the addition of a Validation Processor to the Transaction operation [32]. To some degree, the validation to WFS is actually the validation to the GML file, because the validation is on the changes, such as update, query, delete and modify, the users did to the GML files, and before storing the changes.

The VWFS [6] requirements include [32]:

- the specification of attribute constraints
- the specification of topology constraints
- verification of geospatial database consistency

Methodology for validating on these requirements will make use of the existing tools. For example database integrity issues, such as consistency across updates and delete operations, the traditional database techniques will be used; the high-level geospatial query operations will use the geotools2 library [32]. The attribute constrains and topology constraints are two subelements of logic consistency data quality element defined in ISO/TC 211 standards, while the database consistency is also significant, but not included in the ISO data quality classification. In the GML validation system, the attribute constraints is decided to utilize the XML validation system. So the topology validating is of most interest.

The validating WFS is designed as an enhancement to GeoServer, therefor the functionalities are fitted as plug-ins, refercing the figure 14. The project has done several validation test inplementations, including feature validation and integrity validation [33]. The feature validation are mostly the topology validation, and it complies with the ESRI's ArcGIS Geodatabase and topology rules. This leads the seamless utilization of the VWFS by the GML validation system, since the GML validation system also adapt the ESRI ArcGIS's topology rules as topology constraints.

Since the VWFS is closely connected with server, the developing case is more complicated than the GML validation framework, such as the consideration of web based configuration. The GML validation framework is initially designed as desktop software, but integrating the GML validation with a server is should be explored as future work. Anyway, the GML validation framework and the validating WFS have quite a few requirements, usage, and functionalities in common. The VWFS framework has been delivered and integrated into the GeoServer Web Feature Server, released as 'open source' software [34]. It means that the internals can be accessed, changed, and redistributed by software developers. So the topology realization with the Geo-

[6] Validation Web Feature Server

tools2 library can be extremely useful and helpful for developing the topology validation part of the GML validation framework.

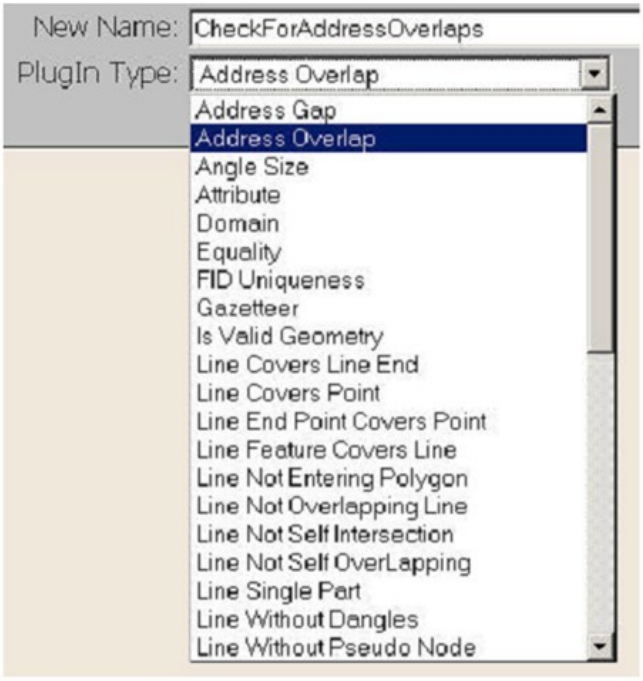


Figure 14: Selecting a Plug-In using GeoServer 1.2.0 [34]

4 The GML validation framework

The GML validation framework consists of three parts: determining the GML validation requirements, methodology and approach, and result handling. The GML validation requirements comply with two standards: the corresponding product specification under Norwegian standard, and the relevant ISO/TC 211 standards about geographic information. The requirements determining will utilize and integrate the requirements from existing geographic data quality control tools introduced in Chapter 3. The methodology and approach invents a new structure of XML validation extended with extra rules. So the XML validation is the basic part, and the validation against the extra rules will be developed as extension to the XML validation. The result handling includes the general statistic report, and ISO quality evaluation report.

4.1 Determine the GML Validation requirements

A qualified geographic dataset is the dataset that fits the users requirements of their applications. The users can use the dataset for different cases, therefor the requirements can be various. But generally, there are always some aspects that people considers when deciding if the dataset fits the intened uses. These aspects are: requirements to the data structure: nice covered by UML class diagram [1]; the dataset is consistent to the conceptual model(despite the quality risk of conceptual model/ UML model itself); how well the data fits the real world: accuracy, completeness, consistency, precision; and also the documentation, availability, acquisition, etc [1]. These requirements form the criteria of a qualified geographic dataset. Note that among those requirements, the quality of the data model (UML model) and the availability and acquisition are beyond a validation system. The earlier chapters have discribed and discusses the data quality issues, and the existing technologies very clearly. But to determine the requirements for the GML validation. Certain analysis and decisions must be discussed.

What ISO data quality element can be measured by the GML validation

Different quality elements require different measuring methods. Talking about the data quality evaluation method, ISO 19114 has defined the classification of data quality evaluation methods, shown in figure 15. The quality evaluation methods can be sorted into direct evaluation method and indirect evaluation method, and the direct evaluation method can be further sorted by internal and external. The external direct evaluation method requires reference data external to the dataset being tested [20]. The GML validation belongs to the internal direct evaluation method, it does validation against the rules by only using the dataset itself. Though due to the property of GML, part of the rules can be written in XML form (GML schema) and be parsed. Thus GML validation, as an internal evaluation method, has a limitation that it cannot measure the data quality elements that require external data. The analysis will be concerned with what kind of evaluation methods the geographic data elements demand, and the conclusion of what ISO data quality element can be measured by the GML validation.

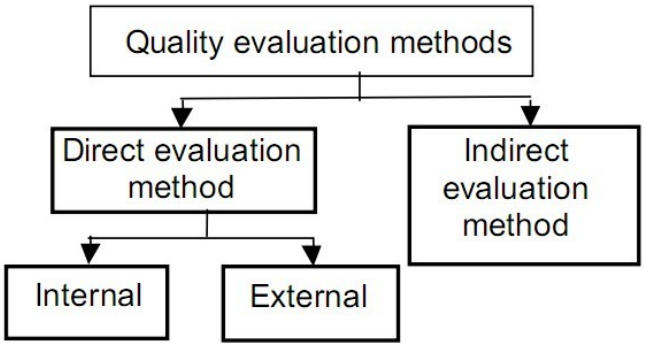


Figure 15: Classification of data quality evaluation methods [20]

Completeness

The data quality element, completeness, concerns if there are excess data present in a dataset and if there are data absent from a dataset respectively. This data quality element needs to use external measurement method, because by one single dataset, we will not be able to decide what data is in the file while it should not be in, and what data this file does not include while it should include. So an external dataset is needed, and by comparing the dataset on question and the external dataset, we can count the number of excess data and absent data, and report them. But the GML validation only implements functions on the dataset that needs to be validated, no extra dataset is included. Thus the data quality element completeness cannot be measured by the GML validation.

Logical consistency

The data quality element, logical consistency, has four subelements which are conceptual consistency, domain consistency, format consistency, and topological consistency. The detail description of each subelement can be seen in Section 2.2.2. Logical consistency concerns the correctness of the dataset against a set of rules. Among the four subelements, conceptual, domain and format consistency all need to adherence to the rules defined for the specific dataset, which means that they demand extra files that describes the correct conceptual definitions, value domain and physical structure of the dataset. Meaning they demand external measuring methods. The GML schema is the kind of extra file that contains the rules, so the validation of GML file against the GML schema covers the domain and format consistency subelements and part of the conceptual consistency subelement of logical consistency. Topology consistency is about the correct relations between the spatial features in an application. To check if the spatial features have correct topology relations, we need certain topology rules. All the geographic objects are modeled by the geometry. In Section 2.1.2 simple features, the geometric classes and the spatial relations between geometries were introduced in detail. Spatial objects can have certain relations between each other, but for some applications in real life, several relations between some kinds of geometries are not allowed, thus we need topology rules to avoid the occurrence of unacceptable spatial relations. For example, a lineString geometry can have overlaps relation with another

lineString geometry, but in an application of road design, this phenomenon will cause a serious error, then the topology rule that define that two lines cannot overlap may be used in this application. The topology rules are beyond the specific dataset, all datasets that belong to the same domain can use the same topology rules for validating topologies. So the solution is to make functions for all the topology rules, and choose certain ones when validating one dataset of one specific domain. Therefor, internal direct measuring method can be used for logical consistency data quality element.

Positional accuracy

Positional accuracy refers to conformance between encoded and actual value of the positional attribute for a given entity. The most important part in the definition is 'encoded and actual value'. The encoded value means the value in the dataset we want to validate, and the actual value means the values accepted as being true. The accuracy is based on the comparison between the data in the dataset for measuring and the data in the real world or being accepted as true. Without the position values accepted as true, the positional accuracy of the data on measuring will not be obtained. So positional accuracy demands the external type of measurement.

Temporal accuracy

The temporal accuracy, in the same way as positional accuracy, needs the true temporal values to decide if the temporal values in the dataset are in conformance with the real world. So temporal accuracy also demand the external type of measurement.

Thematic accuracy

Thematic accuracy refers to almost everything of an entity, include position value, time parameter, other attributes include the non-quantitative attributes. Like the positional accuracy and temporal accuracy, thematic accuracy cannot be measured by the dataset itself, extra information or file is needed. Since only the data quality elements that can be measured by internal direct method, can be taken into consideration to the GML validation, by analyzing the data quality elements, the conclusion is that the logical consistency element can be measured by the GML validation, and the rest data quality elements should be measured by other appropriate methods.

Make use of the requirements from the existing data quality control tools

Conforming to the ISO standards is an essential part, the other mandate part is to conform to the product specification under Norwegian standard. From the discussion above, the conclusion is that the logic consistency data quality element is the requirement part that the GML validation should comply with. The logic consistency includes four subelements: conceptual consistency, domain consistency, value consistency, and topological consistency.

Conceptual consistency actually means the consistency between the dataset and the corresponding product specification (conceptual model). Many rules and constrains in the product specification are defined implicitly, so it is difficult to discover those rules. But the requirements in SOSI-Control can be made good use of, because the SOSI-Control realize the same goal that ensure the consistency between the SOSI dataset and the product specification. Even Dataset formats are different, the requirements or rules in the product specification are the same. So the requirements about road network domain defined in SOSI-Control can be included in the GML

validation.

Domain consistency and value consistency can be well covered by the XML validation requirements. The XML validation can be directly used for GML format. So the GML validation should completely adapt all the XML validation requirements.

Topological consistency is a key element to geographic data. An application may not demand all the topology rules, but supplying a complete set of topology rules for user to choose according to the specific application will be practical and flexible. This complete set of topology rules can adapt the topology rules defined by ESRI ArcGIS. ESRI ArcGIS has defined a number of topology rules that basically cover almost all the topology cases.

The requirements can have intersections, namely, one requirement can be defined more than one time. To avoid duplication, first we need to clear the reasons and resources of the intersections. XML validation requirements and the topology rules does not intersect. But the conceptual model defines the requirements concerning all different sides for a domain, these aspects certainly include the data format, data structure, and topologies. So the requirements intersections are between the conceptual requirements and XML validation requirement, and between the conceptual requirements and the topology requirements. So we need to solve this problem to determine the final requirements for the GML validation.

My proposal is to remove the duplicate requirement from the conceptual requirements, and keep the requirements as the XML validation requirements or topology requirements. The reason is that both the XML validation requirements and the topological requirements have the characteristics of common use, which means that these requirements can be applied to almost all the GML dataset from all kinds of domain. On the contrary, the conceptual requirements are different for different domains. So to extract the common requirements that are for all the domains from the conceptual model will be more efficient and effort saving. According to this solution, we can get the final requirements for the GML validation on road network domain as below.

Final requirements

The detail of the XML validation requirements refers to Section 3.1, the detail of the topology rules refers to Section 3.3, and the detail of conceptual requirements refers to Section 3.2. The requirements are about the road network domain; and the common requirements intersecting with XML validation requirements and topology requirements will be extracted. In addition the SOSI-Control requirements have to be adjusted for GML format. So the final conceptual requirements will be listed out in details, while the XML validation requirements and topology rules refers to the previous sections.

- The GML file is well-formed (XML validation requirements)
- The GML file is valid against its schema (XML validation requirements)
- Conceptual requirements from corresponding production specification: (Norwegian road network product specification)
    - Content Checking
        - 1. Unique serial number

  - 2. Boundary: The coordinates of the elements in all groups should in the boundary defined in the header
  - 3. Coordinate dimension checking: 2D or 3D
- Node checking
  - Check all the blind nodes, which are the nodes only connected to one feature
  - Search all the 2'er (connect to two features) nodes and groups that have similar characteristics (for example, one road is separated into road segments, the end node of one segment should be able to connect the beginning node of the next segment)
- Certain statistic information (how many features, how many lines, and how many points, etc.)

Topological rules can be seen in Appendix A, selective according to different domains.

4.2 Methodology and Approach

Programming Language:

The figure 16 shows the capabilities of several popular programming languages on spatial development. The choice of programming language is really a matter of taste. After deciding the programming language, there are two approaches for developing the validation system. One is pure programming, and the developer writes all the functions, classes, etc. to reach the goal. The other one is to combine the programming with libraries/toolkits. Note that both approaches should have XML validation included, because XML validation is the pre-execution before running the external functions.

Category	Javascript	C/C++	Java	Python
Web mapping clients	Open Layers			Mapnik
Libraries/ toolkits	Proj4js	GEOS GDAL OGR Proj.4	Java Topology Suite (JTS) GeoTools	FWTools
GIS Servers		MapServer	Geoserver Deegree	
Geo-enabled web frameworks	Mapfish			GeoDjango

Figure 16: Development tools [35]

4.2.1 Pure programming approach

The GML is written in XML format, and most of the programming languages have the capabilities to parse XML files, so this gives the possibility to realize the functions by programming. By using a parser, the necessary information can be taken out from the GML files, for example, the feature type, the GML ID, the coordinates, and so forth. This information will be stored in a way that the programming language can directly read. The storing structure can be an array, list, dictionary, etc. This depends on which programming language is used. By doing this, instead of having to parse every time to get the value needed, it only parses once and the program can get needed value right from its own storing structure. It saves much time and computation. After the preparation for necessary data, classes, functions, and exceptions will be designed according to the requirements.

This thesis has used this approach for testing a selection of requirements. The programming language used is Python. The lxml package was installed for supporting the XML and HTML abilities in Python. Three schema languages: DTD, Relax NG and XML Schema are supported by lxml [36]. The lxml.etree API can parse and validate XML. By using the lxml package, the GML file can be parsed and also validated against the GML schema. All the relevant GML schemas should be included, because one schema is not stand-alone, there are dependencies between schemas.

The geographic dataset always contains large number of coordinates, which demands huge computation effort. The functions are mostly working with these coordinates, such as calculating and comparing. One coordinate comparing methodology was proposed in the KVAKK project, which was the early version of SOSI-Control. The dataset should be divided into small divisions according to the levels, for example, if the level is 4*4, then it will be divided into 16 small divisions. The dataset should be divided by the coordinates scale so that different division will not cross. And then compare the coordinates in each small divisions respectively [37]. Because this can largely reduce the computation effort compared with the method without dividing.

Another methodology can be feature based comparison. A big difference between SOSI-Control and the GML validation is that the GML validation framework classifies the requirements complying with ISO standard, and in which the topology rules cover the spatial relation issues. The topology concerns on the relations between geometries. And the features are expressed with geometries and other attributes, we can regard the feature as the instance of geometry. So feature based methodology make it easy for the topology validation, meanwhile also less computation. In addition, in the GML dataset, the coordinates are naturally divided by features. So it is also simple to realize.

The key step is to make the functions according to the requirements. The well-formed, and validation against GML schema are realized by the XML validation, the remaining requirements are the conceptual rules and the topology rules. Functions need to be programmed for each specific requirement. For example, 'all coordinates have to be in boundary defined in the header' requires the function to compare all the coordinates against the boundary coordinates; and 'lines can't self-overlap' requires the function that in each line feature, to compare the coordinates with each other.

This approach is feasible, like normal software developing. But it demands a lot of work on

designing and programming the functions for topology validation.

4.2.2 Combination of programming and library

The difference between the pure programming approach and the combination of programming and library approach is that the library is applied to simplify the topology validation in the later approach.

The library/toolkits is a ready-to-use functions collection. For the topology toolkits, the functions are about the spatial relations between the geometries. So the developer does not need to design and program on how to realize the topology requirements. The library has done the job, the developer only needs to use it in a correct way, and integrate them with other functions. VWFS is an example of developing with GeoTools library. If one chooses this approach, then the programming language chosen must have the library/toolkits. This thesis has partly looked into the JTS [1], so JTS will be used as examples to explain the key content of the library/toolkits.

JTS is one part of the GeoTools library, which is concerned on the geometry. It is written in pure Java, and is open source, under the LGPL license. It conforms to the Simple Feature Specification for SQL published by the OGC. The geometry relationships are: Equals, Disjoint, Intersects, Touches, Crosses, Within, Contains, Overlaps, and DE-9IM Matrix for two Geometries (relate), which have been described in Section 2.1.2. The boolean value will be returned if one tests the above relationships, for example, return geometry.overlaps(geometry2); [38]. If the topology rule is that the lines cannot overlap, then for all the line features, the overlaps relationships should be checked between each other. For the topology rule that define is about non-self overlaps, then isSimple() function can be used, the definition of simple depends on the geometry type, and is specifically defined in Simple Feature specification. JTS only has 2.5 dimension ability while the GML geometry is 3 dimension.

The simple feature spatial relationships and the Egenhofer point-set topological spatial relations are consistent, and they cover the real world's geometry relationships in a complete way. The functions in the library are designed according to these geometry class operations, so the library is able to manage all the topology rules.

So this approach utilizes the available, easy-to-use toolkits to solve the topology validation, while the rest part is more or less the same as the pure programming approach. This approach can save a lot effort for the topology functions design, programming, error handling, and the functions are more systematic and standardized, probably error-free. So this approach is what I recommend to adapt for developing the GML validation system.

4.3 Result handling

After the data validation, the users should get the feedback about the data quality of the dataset. One common way to report the result is to generate a text file with the error or doubtable data information. The information includes the type of the errors, the number of each kind error, where are the errors and so on. The statistic report can give an overview and detail information on the quality problem of the dataset, and will be greatly helpful for doing corrections to the dataset. Another way is to report the result as metadata. The feedback information for report as

[1] Java Topology Suits

metadata will be more general than the statistic report, the percentage or number of errors of one kind of data quality element of the dataset is normal form. The data quality metadata will be extremely important for sharing and reusing of the dataset. So the two ways of report the validation result are based on different purposes and use cases.

Statistic report

Knowing what and where are the errors in the dataset is always the direct and initial idea of using the data validation. The statistics report is to fulfill this kind of demand. Usually the validation systems and software return this kind of result, such as SOSI-Control. The report should include all the error information and warning information one needs to know about. And the error and warning information is accorded with the requirements. The result reporting is usually followed by some implementations to the dataset according to the feedback information. The implementations could be correcting the dataset by the error information, decide if the warning information are errors, or further report it as metadata.

ISO quality evaluation report

A common definition of metadata is 'data about data'. The data itself contains the information of certain subject, for example, the coordinates of all roads in London. The metadata describes the information of the data, for example, what this dataset is used for, who established it, etc. Establishing new geographic data is expensive, the established data should be used by more than the data owner [2]. This decides that most of the people who uses the data are not the one who created the data. Without metadata of the datasets, people face the challenges of searching and using the right datasets for their use. Certain information attached with the datasets will be of great help. Some common information in the metadata can be data resource, usage, purpose, established time, and so forth. And data quality is regarded as an essential aspect in metadata. The ISO/TC 211 standard body defines not only the data quality elements, but also a standard way for measuring and reporting the data quality. ISO/DIS 19113 Geographic information - Quality principles provides an overview of data quality information. And ISO/DIS 19114 is about geographic information - quality evaluation procedures, it addresses the process flow for evaluating data quality, and describe the schema for reporting quality information with a number of examples.

From the figure 17, we see an overview of data quality information. The quality information can be divided into non-quantitative quality information and quantitative quality information. The GML validation is on the quantitative information quality information, the quality information can be reported according to ISO 19114 and ISO 19115. ISO 19114 and ISO 19115 describe schemas for reporting quality information [3]. And the figure 18 shows an example of using the ISO 19114 schema to report the logical consistency quality information. The components of data quality reporting are (DQ is short for Data Quality) DQ_Scope, DQ_Element, Example dataset parameters, and Example quality result meaning, and there are several sub-components to DQ_Element: DQ_Subelement, DQ_Measure, DQ_Date, and DQ_ConformanceLevel. So after getting the statistic report, if the use wants to further report the quality evaluation result in ISO standard way, the information should be filled according to those components. In Chapter 5, the testing result will be reported using the ISO 19114 reporting schema.

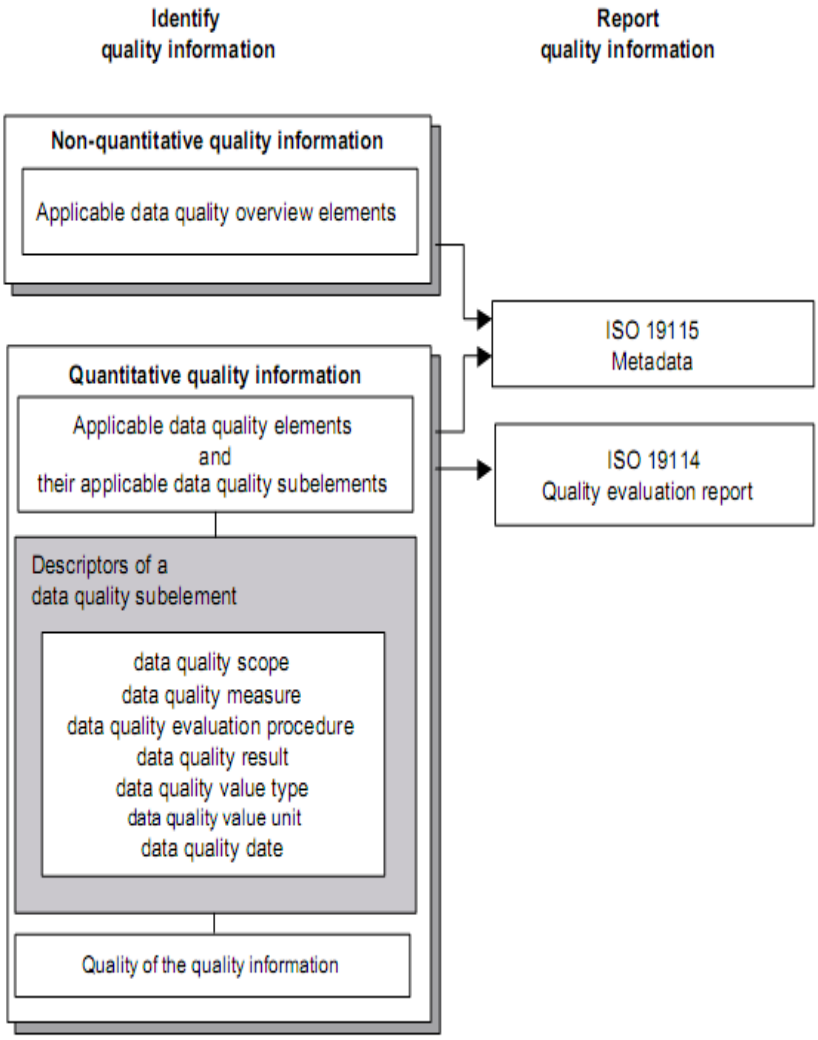


Figure 17: An overview of data quality information [3]

Table D.3 *(continued)*

Data quality component	*Example 16*	*Example 17*	*Example 18*
DQ_Scope	All province boundaries in the dataset.	All state boundaries in the United States	All state boundaries in the United States
DQ_Element	2 – Logical consistency	2 – Logical consistency	2 – Logical consistency
DQ_Subelement	4 – topological consistency	4 – topological consistency	4 – topological consistency
DQ_Measure			
DQ_MeasureDesc	Pass-Fail	Number of items with topological inconsistencies	Percentage of items with topological inconsistencies
DQ_MeasureID	20401	20402	20403
DQ_EvalMethod			
DQ_EvalMethodType	1- Internal	1- Internal	1- Internal
DQ_EvalMethodDesc	For each province, check the boundaries to assure closure. Count the number of provinces whose boundaries do not close.	For each state, check the boundaries to assure closure. Count the number of provinces whose boundaries do not close.	For each state, check the boundaries to assure closure. Count the number of provinces whose boundaries do not close. Divide
DQ_QualityResult			
DQ_ValueType	1 – Boolean variable	2 – Number	4 – Percentage
DQ_Value	False	2	2,0%
DQ_ValueUnit	NA	topological inconsistencies	percent of topological inconsistencies
DQ_Date	2000-03-06	2000-03-06	2000-03-06
DQ_ConformanceLevel	Zero items may have topological violations.	Zero items may have topological violations.	Zero percent of the items may have topological violations.
Example dataset parameters	100 items within scope in the dataset. Two of the items have topological inconsistencies.	100 items within scope in the dataset. Two of the items have topological inconsistencies.	100 items within scope in the dataset. Two of the items have topological inconsistencies.
Example quality result meaning	Dataset fails. Topological inconsistencies found.	Dataset fails. Number of topological inconsistencies exceeds conformance quality level.	Dataset fails. Percentage of topological inconsistencies exceeds conformance quality level.

Figure 18: Example of data quality logical consistency measures [20]

5 Bringing it all together

Figure 19 shows an overview of how to evaluate and report the data quality of a dataset. The pre-step is knowing the domain of the dataset, and choosing the corresponding product specification or user requirements. In this thesis, the author also decided the dataset in the road network domain at the first place, and chose the Norwegian road network specification for defining the requirements.

The first step is to identify an applicable data quality element and subelements. The data quality element, data quality subelement, and data quality scope to be tested shall be identified in accordance with the requirements of ISO 19113 [20]. One geographic data file may not include all the elements, which means not all the data quality elements are relevant to one specific file or one specific application domain, for example, one application only demands completeness data quality element and absolute or external accuracy subelement of positional accuracy data quality element. So it is necessary to identify what data quality elements and subelements are relevant to the concrete application. The completeness, logical consistency and positional accuracy data quality elements are significant to the road network domain, and they can be identified as relevant data quality elements.

The following step is identifying a data quality measure. A data quality measure, data quality value type and, if applicable, a data quality value unit shall be identified for each test to be performed [20]. Data quality value type are such as boolean variable, number and percentage, and the value and value unit are corresponding to the value type. This is for the final result reporting. Figure 18 in Section 4.3 was an example of data quality measures. And in this chapter, I will test two requirements, and report the result conforming to the ISO standard.

Step 3 is to select and apply a data quality evaluation method. The classification of the data quality methods, and the discussion of what kind of evaluation method each data quality element requires have been written in Section 4.1. There is hardly a tool that can measure all the data quality elements, which implies that a data quality evaluation method for each identified data quality measure shall be selected [20] and different measurement methods should be combined together to control the different aspects of geographic data quality instead of one single measurement method.

After applying the identified data quality evaluation method, the next step is to determine the data quality result, and if needed, determine conformance.The data quality result will be according to the data quality measure identified earlier, for example, to check the commission subelement of completeness quality element, the measure chosen is the number of commission, then the result will be like 10, 12 such numbers. The conformance quality level, in the other way, is user defined which determine a standard parameter that if the quality result reach the standard, then it is qualified, else, unqualified. So if the conformance level for commission is less than 5, then the dataset with 10 items commission will be reported as fail. If the conformance level is omit or unspecified, only the quantitative result will be reported.

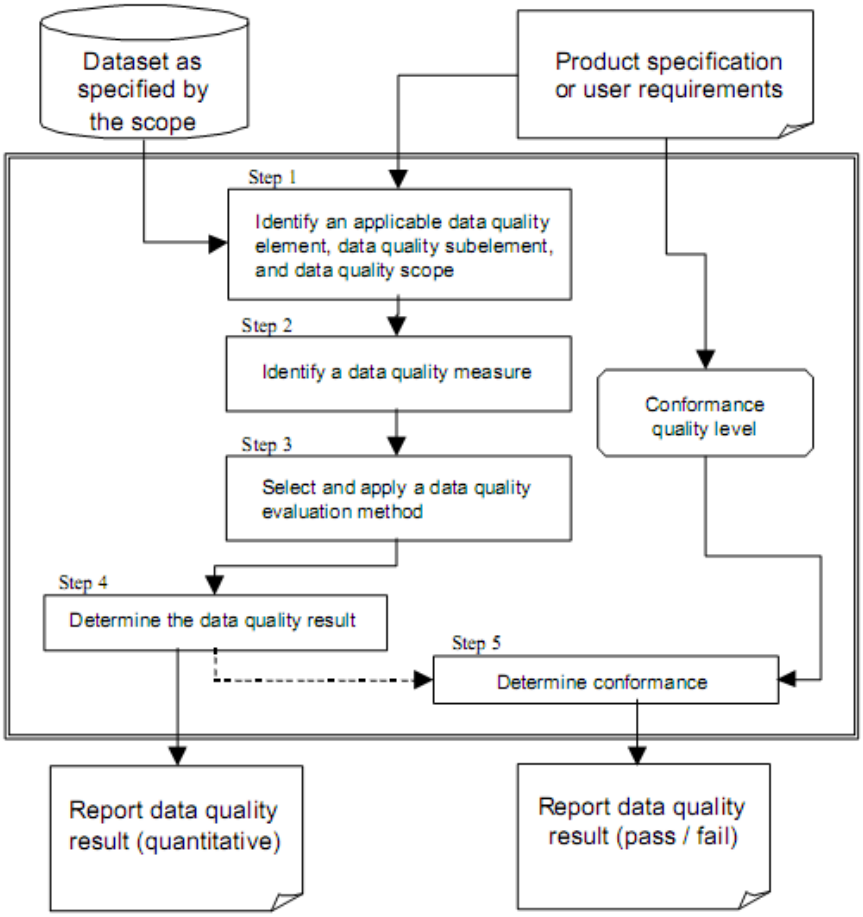


Figure 19: Evaluating and reporting data quality results [20]

In this chapter, the author uses one GML dataset that is transformed from SOSI format by FME software, to test two validation requirements, and report the result in the ISO standard way. The SOSI datasets were downloaded from Norge digitalt, Norwegian spatial data infrastructure, website with help from supervisor Sverre Stikbakke. They are the road network datasets for the whole of Oppland County. The one used for testing was chosen by random. Note that since the datasets were from the national geographic portal, they were through the processing by SOSI-Control. So for testing purpose, the errors appear in the dataset were man-made, instead of from data source or transformation process, but they could be the reasons that cause errors. One more fact is that the GML file is huge, usually thousands of lines, so for demonstration and test purpose, the GML file used was part of the original dataset.

The two validation requirements for testing are 'All coordinates have to be in the boundary defined' and 'Point has to be covered by line'. The first requirement is a conceptual rule which is necessary for conformity to the product specification. 'All coordinates have to be in the boundary defined', in other words, no coordinates outside the boundary can be included in the file. This rule can be useful for checking two things. The feature and object outside the boundary will not appear in the dataset, e.g. the road in Lillehammer should not appear in the Gjøvik road network dataset. Also wrong coordinate values were given to a feature, e.g. only one or two coordinates values are out of boundary, but the rest of the coordinates to the road are not, then this could be the coordinate error caused by bad measurement or typing mistake, instead of exclude the road. And the second requirement 'Point has to be covered by line' is both a topology rule and a conceptual constrain, as discussed in section 4.1, it is sorted to topology consistency. This requirement demands that no stand-alone point occurs in the dataset, all the points or nodes have to be covered by lines. This is because in the definition in the product specification, the node type features that have point coordinates, for example, Roadblock, roadUnderRailway, are always the special points in the LineString type features, which means that the node type features are part of the LineString type features. For instance, roadUnderRailway is the intersection point where one road is under a railway, it is a special point, but still belong to the road. The testing dataset and the source code script can be seen in appendices B and C.

After being parsed, validated by GML schema, and the necessary information extracted from the dataset, apply the two functions. The testing result is that the dataset contains 10 features with lineString geometry type, and 10 features with point geometry type. There is no coordinate which is out of the boundary defined in the header, and there are 9 point features which are not covered by the lineStrings, with their gmlID been reported.

For a normal case, the above result report is sufficient. As described in section 4.3, another alternative is to report the quality result in ISO standard way. According to the Figures 19 and 18, a quality measure needs to be identified, and dependently, a conformance quality level may be identified. It was decided that the measure for the fist requirement, the conceptual consistency, being pass-fail with conformance level of zero coordinate out of boundary. The measure for the second requirement, the topology consistency, being number of violating items without conformance level, also being percentage with conformance level of zero percent violations. Rhe evaluation report is shown in figure 20.

Data quality component	Requirement 1	Requirement 2	Requirement 2
DQ_Scope	All coordinates in the dataset have to be in boundary defined in the header	Point have to be covered by Line	Point have to be covered by Line
DQ_Element	2-Logical consistency	2-Logical consistency	2-Logical consistency
DQ_Subelement	1-conceptual consistency	4-topological consistency	4-topological consistency
DQ_Measure			
DQ_MeasureDesc	Pass-fail	Number of violating items	Percentage of violating items
DQ_MeasureID	20101	20402	20403
DQ_EvalMethod			
DQ_EvalMethodType	1-Internal	1-Internal	1-Internal
DQ_EvalMethodDesc	Check all the coordinates in the dataset, compare them with the boundary coordinates defined. Count the number of coordinates which are out of the boundary.	Check the point features in the dataset, count the number of point feature which is not covered by line feature.	Check the point features in the dataset, count the number of point feature which is not covered by line feature.
DQ_QualityResult			
DQ_ValueType	1-Boolean variable	2-Number	4-Percentage
DQ_Value	True	9	90%
DQ_ValueUnit	NA	Topological inconsistencies	Percent of topological inconsistencies
DQ_Date	2010-04-20	2010-04-20	2010-04-20
DQ_ConformanceLevel	Zero coordinates out of boundary	Not specified	Zero point is not covered by the lines
Dataset parameters	The coordinates of 10 lineString features were compared with the boundary coordinates, and none of them is out of boundary	Omitted	10 point features in the dataset, and 9 of them are not covered by lines
Quality result meaning	Dataset passes. The number of coordinates out of boundary equals the conformance quality level.	10 point features in the dataset, and 9 of them are not covered by lines	Dataset fails. Percentage of topological inconsistencies exceeds conformance quality level.

Figure 20: Reporting the test result in ISO way

6 Future work and conclusion

Conclusion

Geographic data quality is an important issue in geographical information systems, and it is a macroscopic concept. Quality control can refer to a large range of methods. For geographic data quality problem, as described in section 2.2.2, each quality element has its own features, thus accordingly demands different methods for evaluating them. One method can hardly manage to control all the quality problem. Data validation as a useful quality control tool, is widely adapted in different areas. The thesis intended to develop a validation framework for GML data format based on Norwegian standard. The framework includes three parts, which are validation requirements, approach, and validation result handling.

When determining the validation requirements, it is important to first find out what the data quality problems are, and then analyse how those problems can be managed, and get to know what data quality problems can be controlled by validating the dataset against some rules. Knowing that, the next step is to clarify the rules. The thesis has analyzed the geographic data quality problems, and comes to the conclusion that only logical consistency problem can be handled by validation system. Rules can be mainly divided into three parts: XML validation rules, rules from object product specification definitions, and the topological rules. The XML validation rules and the topological rules are independent on the domains and applications. But product specification is application-dependent. Thus, to develop a validation system based on other standards, or other domains, one needs to re-clarify the constrains from the certain product specification.

The approach to realizing the geographic data validation depends on the geographic data format. GML in this project, has its own properties that it can adapt the existing XML validation for part of the rules checking, and the content inside the file can be accessed by query based languages on the XML format. While the other geographic data format may realize these through other methods. For the other rules, two approaches were proposed: pure programming approach and combination of programming and library approach. They differ at the library has ready-to-use functions for handling the topological checking.

I proposed two ways to handle the result. One way is to return an text file with statistic error and warning information, the other one is to report the result in ISO standard way. The first one is a direct feedback from the validation, tell the users what and where the errors were found, will be helpful for data correction. The second one conforms the ISO relevant standards, and is suitable for transferring and sharing the datasets. The second way is not necessarily adapted for all cases.

It is valuable to develop data quality control system on GML data format, which is the standard exchange geographic data format widely. Validation systems can mainly contribute on logical consistency control that is a vital part of data quality. Other data quality measurement methods that can contribute to the rest aspects of data quality control are expected to use as supplements to the validation system, to form a better geographic information world.

Future work

This thesis has developed a framework of GML validation based on Norwegian standard, the studied case in road network domain. It cleared the way for finding the suitable validation requirements, proposed available approaches to realize the requirements for GML data format, and also the possible ways to handle the validation result. This can be a good resource for:

- Developing the GML validation system based on Norwegian standard for all domains.
- Developing the GML validation system based on other standards.
- Developing validation system of other data formats.
- Correction system after the GML validation process.
- Enhance Web Feature Service

Bibliography

[1] Onstein, E. 2010. email from erling onstein. personal contact.

[2] Kvitle, A. K. 2009. Spatial data infrastructure. presentation notes at GIT for Web Developers course.

[3] ISO. Iso/tc 211 19113 geographic information – quality principles. Technical report, International Organization for Standardization, 2001.

[4] Longley, P. A., Goodchild, M. F., Maguire, D. J., & Rhind, D. W. 2005. *Geographic Information Systems and Science*. WILEY.

[5] Onstein, E. *Inverstigations into Geographical Data Quality*. PhD thesis, Universitetet for miljø og biovitenskap, 2004.

[6] Statkart. Fagområde: Vegnett. Technical report, Statens Kartverk, 2006.

[7] Statkart. Sosi del 3 produktspesifikasjon for fkb - vegnett. Technical report, Statens Kartverk, 2009.

[8] OGC. 2006. Opengis implementation specification for geographic information - simple feature access - part 1: Common architecture. http://www.opengeospatial.org/standards/sfa.

[9] Egenhofer, M. J. & Franzosa, R. D. 1991. Point-set topological spatial relations. *International Journal for Geographical Information Systems*, 5(2), 161–174.

[10] Wikipedia. Geospatial topology. http://wiki.gis.com/wiki/index.php/Geospatial_topology.

[11] ESRI. 2003. Arcgis: Working with geodatabase topology. http://www.esri.com/library/whitepapers/pdfs/geodatabase-topology.pdf.

[12] wikipedia. Geography markup language. http://en.wikipedia.org/wiki/Geography_Markup_Language.

[13] David Arctur, P. E. 2006. Gis standards and interoperability: Understanding and using gml. 2006 ESRI Federal User Conference.

[14] Galdos Systems Inc. 2001. Top 10 benefits of using gml. http://spatialnews.geocomm.com/features/gml/topten.html.

[15] Lu, C.-T., Jr, R. F. D. S., Sripada, L. N., & Kou, Y. 2007. Advances in gml for geospatial applications.

[16] Lake, R., Burggraf, D. S., Trninic, M., & Rae, L. 2004. *Geography Mark-Up Language*. WILEY.

[17] OGC. 2007. Opengis geography markup language (gml) encoding standard. http://www.opengeospatial.org/standards/gml.

[18] Longley, P. A., Goodchild, M. F., Maguire, D. J., & Rhind, D. W., eds. 1999. *Geographical Information Systems*, volume 1. JOHN WILEY and SONS, INC.

[19] ISO. Text for ts 19138 geographic information – data quality measures. Technical report, International Organization for Standardization, 2006.

[20] ISO. Iso/tc 211 19114 geographic information – quality evaluation procedures. Technical report, International Organization for Standardization, 2001.

[21] wikipedia. Extensible markup language. http://en.wikipedia.org/wiki/XML.

[22] wikipedia. Xml validation. http://en.wikipedia.org/wiki/XML_validation.

[23] W3 schools. Xml validation. http://www.w3schools.com/xml/xml_dtd.asp.

[24] User help document of sosi-control.

[25] wikipedia. Esri. http://en.wikipedia.org/wiki/Esri.

[26] Description of arcgis. http://www.esri.com/products/index.html.

[27] About the topology relations in arcgis. http://www.97sky.com/bbs/viewthread.php?tid=37.

[28] ESRI. 2007. Topology rules. http://webhelp.esri.com/arcgisdesktop/9.2/index.cfm?TopicName=Topology_ ArcGIS 9.2 Desktop Help.

[29] Refractions Research Inc. Validating web feature server for opengis architectures. http://cgdi.gc.ca/projects/geoinnovations/2002/REFRACTIONS/Promo V1/index.html.

[30] Geoconnections. Validating web feature server for open gis spatial data infrastructures. http://cgdi.gc.ca/en/aboutGeo/projects/id=259.

[31] Wikipedia. Web feature service. http://en.wikipedia.org/wiki/Web_Feature_Service.

[32] Refractions Research Inc. Spatial validation academic references. http://vwfs.refractions.net/docs/Spatial_Validation_Academic.pdf.

[33] Refractions Research Inc. Validation processor implementation report. http://vwfs.refractions.net/docs/Implementation_Report.pdf.

[34] Refractions Research Inc. Validation web feature server final report. http://vwfs.refractions.net/docs/Final Report.pdf.

[35] Stikbakke, S. develop tools. presentation notes at GIT for Web Developers course.

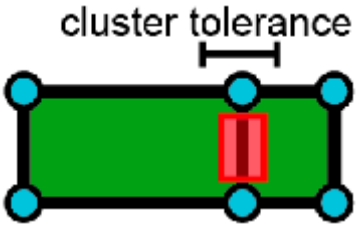
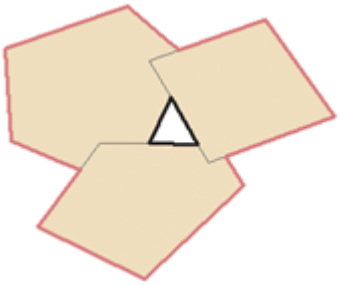
[36] 2010. lxml 2.2.4 documentation. http://codespeak.net/lxml/lxmldoc-2.2.6.pdf.

[37] Tungen, L. & Riise, O. 1994. Further development of kvakk program (in norwegian).

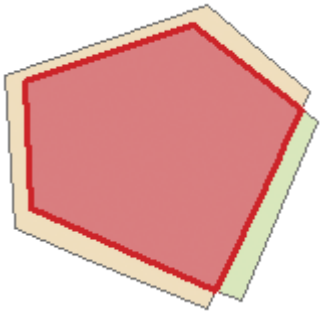
[38] Jts validation suite. http://docs.codehaus.org/display/GEOTDOC/03+JTS+Topology+Suite.

A Topology Rules

Polygon rules

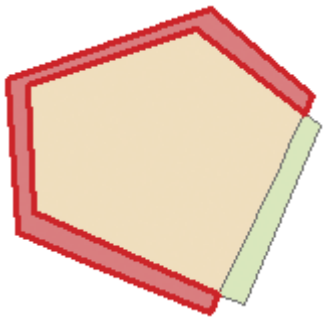
Topology rule	Rule description	Examples
Must Be Larger Than Cluster Tolerance	Requires that a feature does not collapse during a validate process. This rule is mandatory for a topology, and applies to all line and polygon feature classes. In instances where this rule is violated, the original geometry is left unchanged.	 Any polygon feature, such as the one in red, that would collapse when validating the topology is an error.
Must Not Overlap	Requires that the interior of polygons in the feature class not overlap. The polygons can share edges or vertices. This rule is used when an area cannot belong to two or more polygons. It is useful for modeling administrative boundaries, such as ZIP Codes or voting districts, and mutually exclusive area classifications, such as land cover or landform type.	
Must Not Have Gaps	This rule requires that there are no voids within a single polygon or between adjacent polygons. All polygons must form a continuous surface. An error will always exist on the perimeter of the surface. You can either ignore this error or mark it as an exception. Use this rule on data that must completely cover an area. For example, soil polygons cannot include gaps or form voids—they must cover an entire area.	 You can use Create Feature to create a new polygon in the void in the center. You can also use Create Feature or mark the error on the outside boundary as an exception.
Must Not Overlap With	Requires that the interior of polygons in one feature class must not overlap with the interior of polygons in another feature class. Polygons of the two feature classes can share edges or vertices or be completely disjointed. This rule is used when an area cannot belong to two separate feature classes.	

It is useful for combining two mutually exclusive systems of area classification, such as zoning and water body type, where areas defined within the zoning class cannot also be defined in the water body class and vice versa.



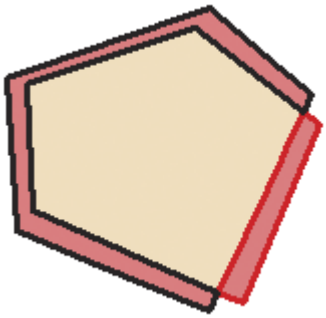
Must Be Covered By Feature Class Of

Requires that a polygon in one feature class must share all of its area with polygons in another feature class. An area in the first feature class that is not covered by polygons from the other feature class is an error. This rule is used when an area of one type, such as a state, should be completely covered by areas of another type, such as counties.



Must Cover Each Other

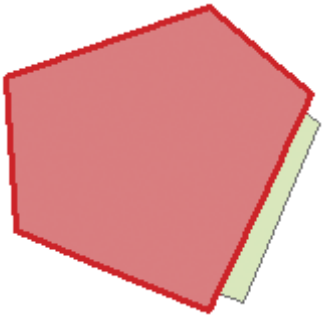
Requires that the polygons of one feature class must share all of their area with the polygons of another feature class. Polygons may share edges or vertices. Any area defined in either feature class that is not shared with the other is an error. This rule is used when two systems of classification are used for the same geographic area, and any given point defined in one system must also be defined in the other. One such case occurs with nested hierarchical datasets, such as census blocks and block groups or small watersheds and large drainage basins. The rule can also be applied to nonhierarchically related polygon feature classes, such as soil type and slope class.



Must Be Covered By

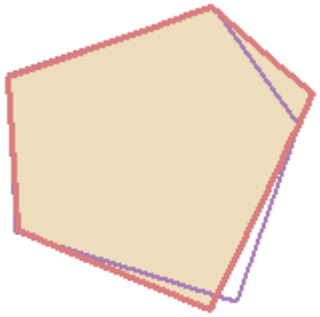
Requires that polygons of one feature class must be contained within polygons of another feature class. Polygons may share edges or vertices. Any area defined in the contained feature class must be covered by an area in the covering feature class. This

rule is used when area features of a given type must be located within features of another type. This rule is useful when modeling areas that are subsets of a larger surrounding area, such as management units within forests or blocks within block groups.



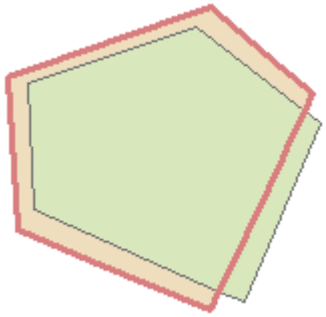
Boundary Must Be Covered By

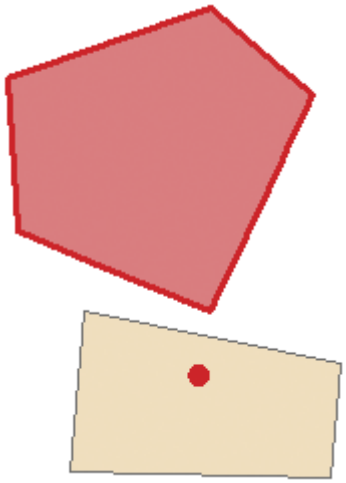
Requires that boundaries of polygon features must be covered by lines in another feature class. This rule is used when area features need to have line features that mark the boundaries of the areas. This is usually when the areas have one set of attributes and their boundaries have other attributes. For example, parcels might be stored in the geodatabase along with their boundaries. Each parcel might be defined by one or more line features that store information about their length or the date surveyed, and every parcel should exactly match its boundaries.



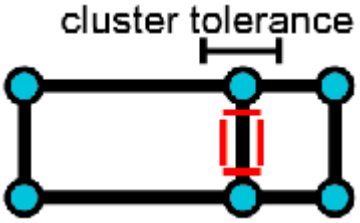
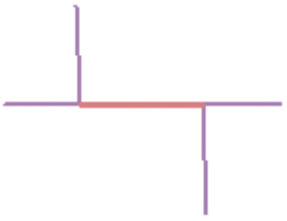
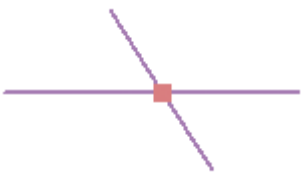
Area Boundary Must Be Covered By Boundary Of

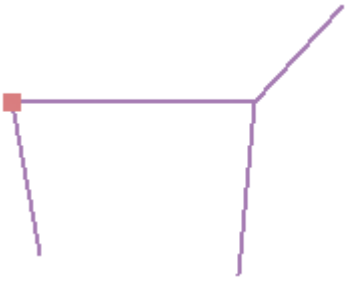
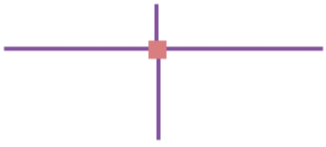
Requires that boundaries of polygon features in one feature class be covered by boundaries of polygon features in another feature class. This is useful when polygon features in one feature class, such as subdivisions, are composed of multiple polygons in another class, such as parcels, and the shared boundaries must be aligned.

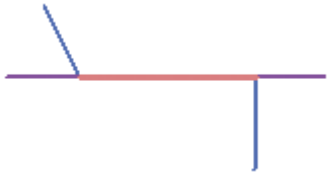
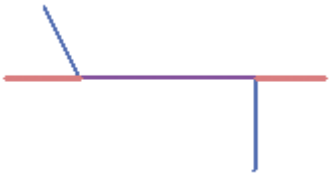
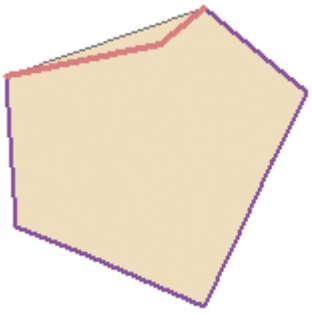
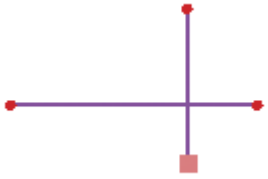


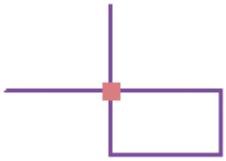
Contains Point	Requires that a polygon in one feature class contain at least one point from another feature class. Points must be within the polygon, not on the boundary. This is useful when every polygon should have at least one associated point, such as when parcels must have an address point.	 The top polygon is an error because it does not contain a point.

Line rules

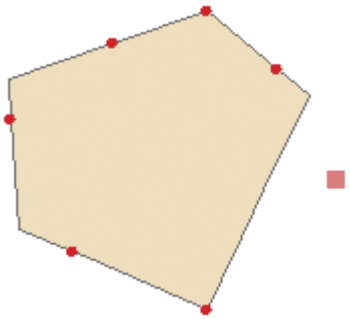
Topology rule	Rule description	Examples
Must Be Larger Than Cluster Tolerance	Requires that a feature does not collapse during a validate process. This rule is mandatory for a topology, and applies to all line and polygon feature classes. In instances where this rule is violated, the original geometry is left unchanged.	 Any line feature, such as these lines in red, that would collapse when validating the topology is an error.
Must Not Overlap	Requires that lines not overlap with lines in the same feature class. This rule is used where line segments should not be duplicated; for example, in a stream feature class. Lines can cross or intersect but cannot share segments.	
Must Not Intersect	Requires that line features from the same feature class not cross or overlap each other. Lines can share endpoints. This rule is used for contour lines that should never cross each other or in cases where the intersection of lines should	

	only occur at endpoints, such as street segments and intersections.		
Must Not Have Dangles	Requires that a line feature must touch lines from the same feature class at both endpoints. An endpoint that is not connected to another line is called a dangle. This rule is used when line features must form closed loops, such as when they are defining the boundaries of polygon features. It may also be used in cases where lines typically connect to other lines, as with streets. In this case, exceptions can be used where the rule is occasionally violated, as with cul-de-sac or dead end street segments.		
Must Not Have Pseudonodes	Requires that a line connect to at least two other lines at each endpoint. Lines that connect to one other line (or to themselves) are said to have pseudonodes. This rule is used where line features must form closed loops, such as when they define the boundaries of polygons or when line features logically must connect to two other line features at each end, as with segments in a stream network, with exceptions being marked for the originating ends of first-order streams.		
Must Not Intersect Or Touch Interior	Requires that a line in one feature class must only touch other lines of the same feature class at endpoints. Any line segment in which features overlap or any intersection not at an endpoint is an error. This rule is useful where lines must only be connected at	Subtract, Split	

	endpoints, such as in the case of lot lines, which must split (only connect to the endpoints of) back lot lines and which cannot overlap each other.	
Must Not Overlap With	Requires that a line from one feature class not overlap with line features in another feature class. This rule is used when line features cannot share the same space. For example, roads must not overlap with railroads or depression subtypes of contour lines cannot overlap with other contour lines.	 Where the purple lines overlap is an error.
Must Be Covered By Feature Class Of	Requires that lines from one feature class must be covered by the lines in another feature class. This is useful for modeling logically different but spatially coincident lines, such as routes and streets. A bus route feature class must not depart from the streets defined in the street feature class.	 Where the purple lines don't overlap is an error.
Must Be Covered By Boundary Of	Requires that lines be covered by the boundaries of area features. This is useful for modeling lines, such as lot lines, that must coincide with the edge of polygon features, such as lots.	
Endpoint Must Be Covered By	Requires that the endpoints of line features must be covered by point features in another feature class. This is useful for modeling cases where a fitting must connect two pipes, or a street intersection must be found at the junction of two streets.	 The square at the bottom indicates an error, because there is no point covering the endpoint of the line.

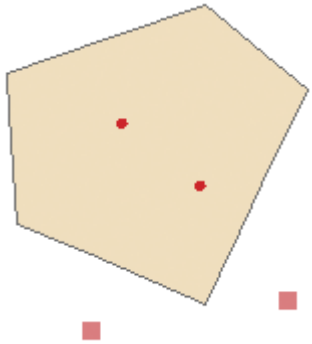
Must Not Self Overlap	Requires that line features not overlap themselves. They can cross or touch themselves, but must not have coincident segments. This rule is useful for features such as streets, where segments might touch in a loop, but where the same street should not follow the same course twice.	 The individual line feature overlaps itself, with the error indicated by the coral line.
Must Not Self Intersect	Requires that line features not cross or overlap themselves. This rule is useful for lines, such as contour lines, that cannot cross themselves.	
Must Be Single Part	Requires that lines have only one part. This rule is useful where line features, such as highways, may not have multiple parts.	 Multipart lines are created from a single sketch.

Point rules

Topology rule	Rule description	Examples
Must Be Covered By Boundary Of	Requires that points fall on the boundaries of area features. This is useful when the point features help support the boundary system, such as boundary markers, which must be found on the edges of certain areas.	 The square on the right indicates an error because it is a point that is not on the boundary of the polygon.

Must Be Properly Inside Polygons

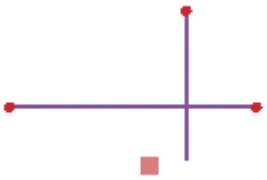
Requires that points fall within area features. This is useful when the point features are related to polygons, such as wells and well pads or address points and parcels.



The squares are errors where there are points that are not inside the polygon.

Must Be Covered By Endpoint Of

Requires that points in one feature class must be covered by the endpoints of lines in another feature class. This rule is similar to the line rule, "Endpoint Must Be Covered By", except that, in cases where the rule is violated, it is the point feature that is marked as an error, rather than the line. Boundary corner markers might be constrained to be covered by the endpoints of boundary lines.



The square indicates an error where the point is not on an endpoint of a line.

Must Be Covered By Line

Requires that points in one feature class be covered by lines in another feature class. It does not constrain the covering portion of the line to be an endpoint. This rule is useful for points that fall along a set of lines, such as highway signs along highways.

The squares are points that are not covered by the line.

B The dataset used for testing

<?xml version="1.0" encoding="UTF-8"?>
<gml:FeatureCollection xmlns:gml="http://www.opengis.net/gml"
xmlns:xlink="http://www.w3.org/1999/xlink"
xmlns:xsi="http://www.w3.org/2001/XMLSchema- instance"
xmlns:fme="http://www.safe.com/gml/fme"
xsi:schemaLocation="http://www.safe.com/gml/fme 0515VBase.xsd">
<gml:boundedBy>
<gml:Envelope srsName="EPSG:32632" srsDimension="3">
<gml:lowerCorner>489589.6 6809355.4 -9999</gml:lowerCorner>
<gml:upperCorner>521877.4 6875598.3 1615.9</gml:upperCorner>
</gml:Envelope>
</gml:boundedBy>
<gml:featureMember>
<fme:VegSenterlinje gml:id="id21d7b75a-5d46-48f7-a918-ee20186b8654">
<fme:SOSI_id>1</fme:SOSI_id>
<fme:HOVEDPARSELL>1</fme:HOVEDPARSELL>
<fme:METER_TIL>3313</fme:METER_TIL>
<fme:MALEMETODE>51</fme:MALEMETODE>
<fme:VEGSTATUS>V</fme:VEGSTATUS>
<fme:OBJTYPE>VegSenterlinje</fme:OBJTYPE>
<fme:VEGKATEGORI>P</fme:VEGKATEGORI>
<fme:VKJORFLT>1#2</fme:VKJORFLT>
<fme:DATAFANGSTDATO>19980204</fme:DATAFANGSTDATO>
<fme:KOMM>0515</fme:KOMM>
<fme:NOYAKTIGHET>1500</fme:NOYAKTIGHET>
<fme:VEGNUMMER>98824</fme:VEGNUMMER>
<fme:VFRADATO>19500101</fme:VFRADATO>
<fme:METER_FRA>0</fme:METER_FRA>
<gml:curveProperty>
<gml:LineString srsName="EPSG:32632" srsDimension="3">
<gml:posList>507558.4 6815042.5 994 507477.3 6814980.6 -9999 507450.6
6814960.2 -9999 507430.1 6814945.1 -9999 507411 6814931.2 -9999 507392
6814914.6 -9999 507375.4 6814894.4 -9999 507358.8 6814879.1 -9999 507338.4
6814867.7 -9999 507307.9 6814861.5 -9999 507287.6 6814860.2 -9999 507255.7
6814854 -9999 507232.8 6814843.8 -9999 507203.6 6814829.9 -9999 507176.9
6814813.5 -9999 507137.3 6814788.2 -9999 507103 6814769.3 -9999 507070 6814749
-9999 507035.5 6814726.1 -9999 506999.9 6814698.2 -9999 506970.6 6814678 -9999
506938.8 6814656.4 -9999 506913.4 6814641.2 -9999 506884.1 6814618.4 -9999
506856 6814590.5 -9999 506834.4 6814563.9 -9999 506814 6814541.1 -9999
506787.1 6814509.4 -9999 506771.8 6814485.2 -9999 506755.2 6814458.6 -9999
506734.8 6814429.4 -9999 506709.4 6814402.7 - 9999 506684 6814383.8 -9999
506658.4 6814369.8 -9999 506631.7 6814352.1 -9999 506597.3 6814330.5 -9999
506569.3 6814311.5 -9999 506542.7 6814292.5 -9999 506520.9 6814273.6 - 9999
506501.8 6814257.1 -9999 506472.5 6814233 -9999 506448.3 6814212.7 -9999
506425.4 6814188.6 -9999 506398.7 6814174.6 -9999 506369.4 6814165.8 -9999
506340.1 6814154.5 - 9999 506319.8 6814146.9 -9999 506295.5 6814136.8 -9999
506271.5 6814121.5 -9999 506247.3 6814100 -9999 506229.4 6814081 -9999 506209
6814058.2 -9999 506191.1 6814032.8 -9999 506178.4 6814008.6 -9999 506169.4
6813983.2 -9999 506163 6813961.7 -9999 506149 6813930 -9999 506134.9
6813908.4 -9999 506118.4 6813885.5 -9999 506097.9 6813866.5 -9999 506080.1
6813846.2 -9999 506061.1 6813824.7 -9999 506035.5 6813795.4 -9999 506012.6
6813772.6 - 9999 505987.1 6813747.2 -9999 505961.7 6813724.5 -9999 505937.4
6813701.6 -9999 505920.8 6813687.6 -9999 505900.4 6813670 -9999 505869.9
6813640.8 -9999 505852 6813626.8 -9999 505832.9 6813609 -9999 505816.4
6813592.6 -9999 505806.2 6813569.7 -9999 505801 6813546.9 -9999 505797.1
6813513.8 -9999 505791.9 6813476.9 -9999 505788 6813441.4 -9999 505777.8
6813396.9 -9999 505771.4 6813352.4 -9999 505759.7 6813300.3 -9999 505753.2

```
6813259.7 - 9999 505748.2 6813226.7 -9999 505744.3 6813194.9 -9999 505735.3
6813156.9 -9999 505721.1  6813123.8 -9999 505703.3 6813084.4 -9999 505688
6813047.6 -9999 505680.3 6813017.1 -9999  505680.2 6812991.7 -9999 505686.5
6812968.7 -9999 505699.2 6812928 -9999 505707.9  6812887.4 -9999 505713
6812865.7 -9999 505721.8 6812830.1 -9999 505730.8 6812784.3 -9999  505738
6812747.5 -9999 505741.8 6812718.2 -9999 505748.2 6812694 -9999 505749.3
6812675  -9999 505746.8 6812662.3 -9999 505744.2 6812650.8 -9999 505739.2
6812640.6 -9999 505736.6  6812634.3 -9999 505734 6812630.5 -9999 505732.7
6812627.9 -9999</gml:posList>
</gml:LineString>
</gml:curveProperty>
</fme:VegSenterlinje>
</gml:featureMember>
<gml:featureMember>
<fme:VegSenterlinje gml:id="id4e202968-3594-4174-abb7-62a417b129c1">
<fme:SOSI_id>2</fme:SOSI_id>
<fme:HOVEDPARSELL>1</fme:HOVEDPARSELL>
<fme:METER_TIL>36</fme:METER_TIL>
<fme:MALEMETODE>20</fme:MALEMETODE>
<fme:VEGSTATUS>V</fme:VEGSTATUS>
<fme:OBJTYPE>VegSenterlinje</fme:OBJTYPE>
<fme:VEGKATEGORI>K</fme:VEGKATEGORI>
<fme:VKJORFLT>1#2</fme:VKJORFLT>
<fme:DATAFANGSTDATO>19920701</fme:DATAFANGSTDATO>
<fme:KOMM>0515</fme:KOMM>
<fme:NOYAKTIGHET>200</fme:NOYAKTIGHET>
<fme:VEGNUMMER>501</fme:VEGNUMMER>
<fme:VFRADATO>19500101</fme:VFRADATO>
<fme:METER_FRA>0</fme:METER_FRA>
<gml:curveProperty>
<gml:LineString srsName="EPSG:32632" srsDimension="3">
<gml:posList>505788.6 6859269.8 398.5 505791.4 6859267.7 398.7 505794.1
6859266.6 398.9  505799.3 6859265.4 399.1 505802.9 6859264.9 400.1 505811.2
6859266.6 402.1 505821.1  6859269.3 404.1 505822.3 6859269.9
404.3</gml:posList>
</gml:LineString>
</gml:curveProperty>
</fme:VegSenterlinje>
</gml:featureMember>
<gml:featureMember>
<fme:VegSenterlinje gml:id="id6815b3ee-e7d1-4bdb-8dab-5aac3512c3d7">
<fme:SOSI_id>3</fme:SOSI_id>
<fme:HOVEDPARSELL>1</fme:HOVEDPARSELL>
<fme:METER_TIL>376</fme:METER_TIL>
<fme:MALEMETODE>20</fme:MALEMETODE>
<fme:VEGSTATUS>V</fme:VEGSTATUS>
<fme:OBJTYPE>VegSenterlinje</fme:OBJTYPE>
<fme:VEGKATEGORI>K</fme:VEGKATEGORI>
<fme:VKJORFLT>1#2</fme:VKJORFLT>
<fme:DATAFANGSTDATO>19920701</fme:DATAFANGSTDATO>
<fme:KOMM>0515</fme:KOMM>
<fme:NOYAKTIGHET>200</fme:NOYAKTIGHET>
<fme:VEGNUMMER>501</fme:VEGNUMMER>
<fme:VFRADATO>19500101</fme:VFRADATO>
<fme:METER_FRA>36</fme:METER_FRA>
<gml:curveProperty>
<gml:LineString srsName="EPSG:32632" srsDimension="3">
```

<gml:posList>505822.3 6859269.9 404.3 505828.5 6859271.5 405.2 505841.4
6859276.1 406.7 505852.8 6859281 409.1 505866.8 6859289.5 411 505876.4
6859296.7 412.8 505883.5 6859303.1 414.2 505895 6859313.6 416.2 505905
6859322.3 418.3 505917.3 6859331 420.4 505928.6 6859337 422.1 505944.8
6859343.6 424.4 505960 6859350.3 425.5 505974.5 6859355.8 426.4 505988.8
6859360.5 427.5 505999.9 6859363.7 428.8 506012.8 6859368.4 430.9 506027.8
6859372.3 431.7 506041.3 6859375.5 433.5 506053.3 6859377 435 506064.3
6859377.7 436.3 506073.2 6859378.1 437.4 506084.9 6859378.9 438.6 506100.4
6859380.8 440.3 506113.1 6859382.6 442.1 506128.1 6859385.8 444.8 506131.4
6859386.4 445.2</gml:posList>
</gml:LineString>
</gml:curveProperty>
</fme:VegSenterlinje>
</gml:featureMember>
<gml:featureMember>
<fme:VegSenterlinje gml:id="id5f6ce135-4d0c-48b6-b17f-932cd31d169c">
<fme:SOSI_id>4</fme:SOSI_id>
<fme:HOVEDPARSELL>1</fme:HOVEDPARSELL>
<fme:METER_TIL>606</fme:METER_TIL>
<fme:MALEMETODE>20</fme:MALEMETODE>
<fme:VEGSTATUS>V</fme:VEGSTATUS>
<fme:OBJTYPE>VegSenterlinje</fme:OBJTYPE>
<fme:VEGKATEGORI>K</fme:VEGKATEGORI>
<fme:VKJORFLT>1#2</fme:VKJORFLT>
<fme:DATAFANGSTDATO>19920701</fme:DATAFANGSTDATO>
<fme:KOMM>0515</fme:KOMM>
<fme:NOYAKTIGHET>200</fme:NOYAKTIGHET>
<fme:VEGNUMMER>501</fme:VEGNUMMER>
<fme:VFRADATO>19500101</fme:VFRADATO>
<fme:METER_FRA>376</fme:METER_FRA>
<gml:curveProperty>
<gml:LineString srsName="EPSG:32632" srsDimension="3">
<gml:posList>506131.4 6859386.4 445.2 506142.4 6859388.2 446.6 506156.8
6859390.1 449.2 506171.2 6859392.7 452.1 506186.6 6859394.7 454.5 506197.6
6859396.1 456.4 506207.1 6859398.2 457.9 506221.7 6859399.5 458.8 506235.2
6859399.1 461.2 506247.2 6859399.2 463.7 506261.2 6859399.8 465 506276
6859400.4 467 506290.5 6859400.9 469.1 506306.7 6859401.7 471.2 506323.3
6859402.3 472.7 506338.1 6859402.4 473.9 506350.2 6859402.4 474.7 506355
6859402.4 475.6 506357.5 6859402.3 475.9</gml:posList>
</gml:LineString>
</gml:curveProperty>
</fme:VegSenterlinje>
</gml:featureMember>
<gml:featureMember>
<fme:VegSenterlinje gml:id="id37181cf5-1a1a-4924-9347-28170c19faf5">
<fme:SOSI_id>5</fme:SOSI_id>
<fme:HOVEDPARSELL>1</fme:HOVEDPARSELL>
<fme:METER_TIL>741</fme:METER_TIL>
<fme:MALEMETODE>20</fme:MALEMETODE>
<fme:VEGSTATUS>V</fme:VEGSTATUS>
<fme:OBJTYPE>VegSenterlinje</fme:OBJTYPE>
<fme:VEGKATEGORI>K</fme:VEGKATEGORI>
<fme:VKJORFLT>1#2</fme:VKJORFLT>
<fme:DATAFANGSTDATO>19920701</fme:DATAFANGSTDATO>
<fme:KOMM>0515</fme:KOMM>
<fme:NOYAKTIGHET>200</fme:NOYAKTIGHET>
<fme:VEGNUMMER>501</fme:VEGNUMMER>

```
<fme:VFRADATO>19500101</fme:VFRADATO>
<fme:METER_FRA>606</fme:METER_FRA>
<gml:curveProperty>
<gml:LineString srsName="EPSG:32632" srsDimension="3">
<gml:posList>506357.5 6859402.3 475.9 506367.6 6859401.9 477.2 506383.6
6859400.6 479.2  506398.8 6859398.6 481 506415.1 6859397.3 482.2 506431.8
6859395.8 484.8 506444.4  6859394.7 486.4 506461.3 6859394.6 487.7 506476.8
6859395.3 489.5 506490.8 6859396.7  491.8</gml:posList>
</gml:LineString>
</gml:curveProperty>
</fme:VegSenterlinje>
</gml:featureMember>
<gml:featureMember>
<fme:VegSenterlinje gml:id="id49bf0709-a539-410a-bb1d-f6e0b1c5bb83">
<fme:SOSI_id>6</fme:SOSI_id>
<fme:HOVEDPARSELL>1</fme:HOVEDPARSELL>
<fme:METER_TIL>820</fme:METER_TIL>
<fme:MALEMETODE>20</fme:MALEMETODE>
<fme:VEGSTATUS>V</fme:VEGSTATUS>
<fme:OBJTYPE>VegSenterlinje</fme:OBJTYPE>
<fme:VEGKATEGORI>K</fme:VEGKATEGORI>
<fme:VKJORFLT>1#2</fme:VKJORFLT>
<fme:DATAFANGSTDATO>19920701</fme:DATAFANGSTDATO>
<fme:KOMM>0515</fme:KOMM>
<fme:NOYAKTIGHET>200</fme:NOYAKTIGHET>
<fme:VEGNUMMER>501</fme:VEGNUMMER>
<fme:VFRADATO>19500101</fme:VFRADATO>
<fme:METER_FRA>741</fme:METER_FRA>
<gml:curveProperty>
<gml:LineString srsName="EPSG:32632" srsDimension="3">
<gml:posList>506490.8 6859396.7 491.8 506504.4 6859397.9 493.5 506522
6859398.2 495  506537.4 6859397.7 497.3 506548.1 6859396.9 499.2 506565.3
6859397.3 500.7 506568.8  6859397.3 501</gml:posList>
</gml:LineString>
</gml:curveProperty>
</fme:VegSenterlinje>
</gml:featureMember>
<gml:featureMember>
<fme:VegSenterlinje gml:id="id86d9f755-070e-415a-800f-87ab14d476b6">
<fme:SOSI_id>7</fme:SOSI_id>
<fme:HOVEDPARSELL>1</fme:HOVEDPARSELL>
<fme:METER_TIL>1407</fme:METER_TIL>
<fme:MALEMETODE>20</fme:MALEMETODE>
<fme:VEGSTATUS>V</fme:VEGSTATUS>
<fme:OBJTYPE>VegSenterlinje</fme:OBJTYPE>
<fme:VEGKATEGORI>K</fme:VEGKATEGORI>
<fme:VKJORFLT>1#2</fme:VKJORFLT>
<fme:DATAFANGSTDATO>19920701</fme:DATAFANGSTDATO>
<fme:KOMM>0515</fme:KOMM>
<fme:NOYAKTIGHET>200</fme:NOYAKTIGHET>
<fme:VEGNUMMER>501</fme:VEGNUMMER>
<fme:VFRADATO>19500101</fme:VFRADATO>
<fme:METER_FRA>820</fme:METER_FRA>
<gml:curveProperty>
<gml:LineString srsName="EPSG:32632" srsDimension="3">
<gml:posList>506568.8 6859397.3 501 506581.3 6859397.7 502.1 506598.9
6859398.4 504.5  506615 6859400.4 506.7 506631 6859402.2 508.5 506646.5
```

```
6859404.3 510.6 506660.2 6859406  512.5 506673.3 6859407.7 514.3 506685.8
6859408.4 515.8 506697.4 6859408.9 516.6 506709.1  6859409.4 517.8 506723.9
6859410 519.7 506736.7 6859410.8 522.1 506749 6859411.9 523.8  506761.1
6859412.5 525.9 506771.9 6859413 527.9 506781.1 6859413.5 529 506789.2
6859413.9  530.1 506794.8 6859414.2 530.7 506796.4 6859414.1 531.2 506798.8
6859412.6 532.1 506799.1  6859411.5 532.7 506799.1 6859407.9 534.1 506798.9
6859405.3 535.2 506795.7 6859399.6  535.9 506791.9 6859393.2 536.3 506787.1
6859386.7 536.6 506783.1 6859382.5 536.8 506776.4  6859377.3 537 506771.2
6859374.5 537 506766.4 6859373 537 506759.4 6859369.3 537.8  506754.5
6859366.4 538 506746.2 6859359.3 538.3 506735 6859345.6 539.6 506725.6
6859331.5  540.8 506717.5 6859315.9 542.8 506707.8 6859300.5 544.9 506700.6
6859289.7 546.5 506689.5  6859274.1 548.5 506679.3 6859260.2 550.4 506669.9
6859245.5 552.2 506659.7 6859231.4  554.1 506649.3 6859217.4 555.3 506643
6859207.8 556 506636.5 6859193.8 556.6 506630.7  6859180.6 556.7 506624.5
6859167.1 556.8 506618.6 6859155.9 557.1 506611.2 6859144 559.3  506605.1
6859136.1 560.6 506600.3 6859130.7 561.8</gml:posList>
</gml:LineString>
</gml:curveProperty>
</fme:VegSenterlinje>
</gml:featureMember>
<gml:featureMember>
<fme:VegSenterlinje gml:id="idaf303bfd-ed5f-498d-9184-869786c9a099">
<fme:SOSI_id>8</fme:SOSI_id>
<fme:HOVEDPARSELL>1</fme:HOVEDPARSELL>
<fme:METER_TIL>1583</fme:METER_TIL>
<fme:MALEMETODE>20</fme:MALEMETODE>
<fme:VEGSTATUS>V</fme:VEGSTATUS>
<fme:OBJTYPE>VegSenterlinje</fme:OBJTYPE>
<fme:VEGKATEGORI>K</fme:VEGKATEGORI>
<fme:VKJORFLT>1#2</fme:VKJORFLT>
<fme:DATAFANGSTDATO>19920701</fme:DATAFANGSTDATO>
<fme:KOMM>0515</fme:KOMM>
<fme:NOYAKTIGHET>200</fme:NOYAKTIGHET>
<fme:VEGNUMMER>501</fme:VEGNUMMER>
<fme:VFRADATO>19500101</fme:VFRADATO>
<fme:METER_FRA>1407</fme:METER_FRA>
<gml:curveProperty>
<gml:LineString srsName="EPSG:32632" srsDimension="3">
<gml:posList>506600.3 6859130.7 561.8 506597 6859127.3 562.7 506587.6
6859118.7 564.2  506575.7 6859107.8 565.4 506565.5 6859097.2 567.1 506555.3
6859086.5 568.6 506544.1  6859076.1 569.7 506533.3 6859066.9 571 506522
6859057.5 572.8 506513.7 6859049.4 574.4  506503.9 6859040 575.7 506493.8
6859030.7 577 506484.7 6859022.7 578.3 506474.8 6859013.3  580.3 506473.2
6859010.7 580.8</gml:posList>
</gml:LineString>
</gml:curveProperty>
</fme:VegSenterlinje>
</gml:featureMember>
<gml:featureMember>
<fme:VegSenterlinje gml:id="idd613f5c3-f2f1-44b3-a194-0ae44ef8f6b1">
<fme:SOSI_id>9</fme:SOSI_id>
<fme:HOVEDPARSELL>1</fme:HOVEDPARSELL>
<fme:METER_TIL>1937</fme:METER_TIL>
<fme:MALEMETODE>20</fme:MALEMETODE>
<fme:VEGSTATUS>V</fme:VEGSTATUS>
<fme:OBJTYPE>VegSenterlinje</fme:OBJTYPE>
<fme:VEGKATEGORI>K</fme:VEGKATEGORI>
```

<fme:VKJORFLT>1#2</fme:VKJORFLT>
<fme:DATAFANGSTDATO>19920701</fme:DATAFANGSTDATO>
<fme:KOMM>0515</fme:KOMM>
<fme:NOYAKTIGHET>200</fme:NOYAKTIGHET>
<fme:VEGNUMMER>501</fme:VEGNUMMER>
<fme:VFRADATO>19500101</fme:VFRADATO>
<fme:METER_FRA>1583</fme:METER_FRA>
<gml:curveProperty>
<gml:LineString srsName="EPSG:32632" srsDimension="3">
<gml:posList>506473.2 6859010.7 580.8 506471.3 6859008.9 580.8 506466.2
6859004.5 581.1 506460.3 6858999 581.6 506452.8 6858991.7 582.1 506444.8
6858983.7 582.1 506436.7 6858975.7 582 506430.6 6858969.9 581.9 506419.9
6858961.7 581.7 506409 6858953.7 581.2 506400.6 6858947.8 581.2 506388.7
6858940.8 581.1 506376.7 6858933.9 581.1 506371.1 6858929.6 581.1 506363.5
6858924.3 581 506351.6 6858917.3 580.4 506340.2 6858910.2 579.6 506329
6858902.5 578.9 506316.9 6858895.2 578.6 506304.6 6858888.1 578.6 506294.4
6858881.6 578.6 506282.6 6858874.3 578.4 506270.1 6858867 578 506259.3
6858860.5 577.4 506250.2 6858856.5 576.7 506237.2 6858852.8 575.9 506223.8
6858849.6 575.2 506215.8 6858847.6 574.6 506200.2 6858845.7 573.7 506187
6858844.8 573 506171.7 6858844.3 572.3 506170.6 6858844.4 572.2</gml:posList>
</gml:LineString>
</gml:curveProperty>
</fme:VegSenterlinje>
</gml:featureMember>

<gml:featureMember>
<fme:Kommunedele gml:id="id38ed7517-ca5d-4722-9a67-68e0b28dd4e6">
<fme:SOSI_id>2814</fme:SOSI_id>
<fme:MALEMETODE>60</fme:MALEMETODE>
<fme:DATAFANGSTDATO>20071102</fme:DATAFANGSTDATO>
<fme:KOMM>0515</fme:KOMM>
<fme:NOYAKTIGHET>200</fme:NOYAKTIGHET>
<fme:SOSI_guid>a7aef614-5cf8-0e47-8766-0a4462f62942</fme:SOSI_guid>
<fme:OBJTYPE>Kommunedele</fme:OBJTYPE>
<gml:pointProperty>
<gml:Point srsName="EPSG:32632" srsDimension="3">
<gml:pos>507558.4 6815042.5 994</gml:pos>
</gml:Point>
</gml:pointProperty>
</fme:Kommunedele>
</gml:featureMember>
<gml:featureMember>
<fme:Kommunedele gml:id="idcab97fd1-663a-45c0-aa84-41691199cca9">
<fme:SOSI_id>2815</fme:SOSI_id>
<fme:MALEMETODE>60</fme:MALEMETODE>
<fme:DATAFANGSTDATO>20071102</fme:DATAFANGSTDATO>
<fme:KOMM>0515</fme:KOMM>
<fme:NOYAKTIGHET>200</fme:NOYAKTIGHET>
<fme:SOSI_guid>0bbbdc6e-673a-634c-b5fe-9b33629b07c9</fme:SOSI_guid>
<fme:OBJTYPE>Kommunedele</fme:OBJTYPE>
<gml:pointProperty>
<gml:Point srsName="EPSG:32632" srsDimension="3">
<gml:pos>514379.6 6863333.2 1487.2</gml:pos>
</gml:Point>
</gml:pointProperty>
</fme:Kommunedele>
</gml:featureMember>

<gml:featureMember>
<fme:Kommunedele gml:id="id1b0f4dfd-756f-430c-9211-fc01ad473561">
<fme:SOSI_id>2816</fme:SOSI_id>
<fme:MALEMETODE>60</fme:MALEMETODE>
<fme:DATAFANGSTDATO>20071102</fme:DATAFANGSTDATO>
<fme:KOMM>0515</fme:KOMM>
<fme:NOYAKTIGHET>200</fme:NOYAKTIGHET>
<fme:SOSI_guid>74ddfca5-a038-1745-a60f-252c4d47ca1d</fme:SOSI_guid>
<fme:OBJTYPE>Kommunedele</fme:OBJTYPE>
<gml:pointProperty>
<gml:Point srsName="EPSG:32632" srsDimension="3">
<gml:pos>514943.6 6862790.7 1614.7</gml:pos>
</gml:Point>
</gml:pointProperty>
</fme:Kommunedele>
</gml:featureMember>
<gml:featureMember>
<fme:Kommunedele gml:id="id640da5a9-fd63-4c7a-a093-94d0230a1d1d">
<fme:SOSI_id>2817</fme:SOSI_id>
<fme:MALEMETODE>60</fme:MALEMETODE>
<fme:DATAFANGSTDATO>20071102</fme:DATAFANGSTDATO>
<fme:KOMM>0515</fme:KOMM>
<fme:NOYAKTIGHET>200</fme:NOYAKTIGHET>
<fme:SOSI_guid>1b33e451-0fc3-d742-9c01-1989f5d12010</fme:SOSI_guid>
<fme:OBJTYPE>Kommunedele</fme:OBJTYPE>
<gml:pointProperty>
<gml:Point srsName="EPSG:32632" srsDimension="3">
<gml:pos>497846.2 6875598.3 1118.8</gml:pos>
</gml:Point>
</gml:pointProperty>
</fme:Kommunedele>
</gml:featureMember>
<gml:featureMember>
<fme:Vegsperring gml:id="id171f940a-ac3b-4bcb-b0db-c753a060a570">
<fme:SOSI_id>2818</fme:SOSI_id>
<fme:VEGSPERRINGTYPE>LÃ¥st bom</fme:VEGSPERRINGTYPE>
<fme:METER_TIL>369</fme:METER_TIL>
<fme:MALEMETODE>95</fme:MALEMETODE>
<fme:VEGSTATUS>V</fme:VEGSTATUS>
<fme:OBJTYPE>Vegsperring</fme:OBJTYPE>
<fme:VEGKATEGORI>K</fme:VEGKATEGORI>
<fme:DATAFANGSTDATO>20020812</fme:DATAFANGSTDATO>
<fme:KOMM>0515</fme:KOMM>
<fme:NOYAKTIGHET>200</fme:NOYAKTIGHET>
<fme:VEGNUMMER>1023</fme:VEGNUMMER>
<fme:VFRADATO>19500101</fme:VFRADATO>
<fme:VKJORFLT/>
<fme:HOVEDPARSELL>1</fme:HOVEDPARSELL>
<fme:METER_FRA>369</fme:METER_FRA>
<gml:pointProperty>
<gml:Point srsName="EPSG:32632" srsDimension="3">
<gml:pos>504853.8 6859892.3 370.1</gml:pos>
</gml:Point>
</gml:pointProperty>
</fme:Vegsperring>
</gml:featureMember>
<fme:Vegsperring gml:id="idff254d08-1d51-4497-8634-2f9530b8dc4f">

<fme:SOSI_id>2856</fme:SOSI_id>
<fme:VEGSPERRINGTYPE>LÃ¥st bom</fme:VEGSPERRINGTYPE>
<fme:METER_TIL>37</fme:METER_TIL>
<fme:MALEMETODE>24</fme:MALEMETODE>
<fme:VEGSTATUS>V</fme:VEGSTATUS>
<fme:OBJTYPE>Vegsperring</fme:OBJTYPE>
<fme:VEGKATEGORI>P</fme:VEGKATEGORI>
<fme:DATAFANGSTDATO>20040812</fme:DATAFANGSTDATO>
<fme:KOMM>0515</fme:KOMM>
<fme:NOYAKTIGHET>200</fme:NOYAKTIGHET>
<fme:VEGNUMMER>99033</fme:VEGNUMMER>
<fme:VFRADATO>19500101</fme:VFRADATO>
<fme:VKJORFLT/>
<fme:HOVEDPARSELL>1</fme:HOVEDPARSELL>
<fme:METER_FRA>37</fme:METER_FRA>
<gml:pointProperty>
<gml:Point srsName="EPSG:32632" srsDimension="3">
<gml:pos>491711.7 6817390.7 966.9</gml:pos>
</gml:Point>
</gml:pointProperty>
</fme:Vegsperring>
</gml:featureMember>
<fme:Kommunedele gml:id="id4f024eea-e635-4f33-8beb-e665ce1a840c">
<fme:SOSI_id>2858</fme:SOSI_id>
<fme:MALEMETODE>60</fme:MALEMETODE>
<fme:DATAFANGSTDATO>20071102</fme:DATAFANGSTDATO>
<fme:KOMM>0515</fme:KOMM>
<fme:NOYAKTIGHET>200</fme:NOYAKTIGHET>
<fme:SOSI_guid>d3d91eec-a69b-e942-9f0c-c71bafdc35f7</fme:SOSI_guid>
<fme:OBJTYPE>Kommunedele</fme:OBJTYPE>
<gml:pointProperty>
<gml:Point srsName="EPSG:32632" srsDimension="3">
<gml:pos>520463.3 6854874.3 640.4</gml:pos>
</gml:Point>
</gml:pointProperty>
</fme:Kommunedele>
</gml:featureMember>
<gml:featureMember>
<fme:Kommunedele gml:id="idf35dec9f-3b55-4626-ad34-81d84bfaf367">
<fme:SOSI_id>2859</fme:SOSI_id>
<fme:MALEMETODE>60</fme:MALEMETODE>
<fme:DATAFANGSTDATO>20071102</fme:DATAFANGSTDATO>
<fme:KOMM>0515</fme:KOMM>
<fme:NOYAKTIGHET>200</fme:NOYAKTIGHET>
<fme:SOSI_guid>8f7edb78-e3ac-9a41-93ce-16e6038bd6df</fme:SOSI_guid>
<fme:OBJTYPE>Kommunedele</fme:OBJTYPE>
<gml:pointProperty>
<gml:Point srsName="EPSG:32632" srsDimension="3">
<gml:pos>510988.9 6825390.3 935.2</gml:pos>
</gml:Point>
</gml:pointProperty>
</fme:Kommunedele>
</gml:featureMember>
<gml:featureMember>
<fme:Vegsperring gml:id="id02878309-7c73-4dd3-869c-ab44e335c98b">
<fme:SOSI_id>2860</fme:SOSI_id>
<fme:VEGSPERRINGTYPE>LÃ¥st bom</fme:VEGSPERRINGTYPE>

<fme:METER_TIL>255</fme:METER_TIL>
<fme:MALEMETODE>20</fme:MALEMETODE>
<fme:VEGSTATUS>V</fme:VEGSTATUS>
<fme:MEDIUM>L</fme:MEDIUM>
<fme:OBJTYPE>Vegsperring</fme:OBJTYPE>
<fme:VEGKATEGORI>S</fme:VEGKATEGORI>
<fme:DATAFANGSTDATO>19920701</fme:DATAFANGSTDATO>
<fme:KOMM>0515</fme:KOMM>
<fme:NOYAKTIGHET>200</fme:NOYAKTIGHET>
<fme:VEGNUMMER>6</fme:VEGNUMMER>
<fme:VFRADATO>19500101</fme:VFRADATO>
<fme:VKJORFLT/>
<fme:HOVEDPARSELL>1</fme:HOVEDPARSELL>
<fme:METER_FRA>255</fme:METER_FRA>
<gml:pointProperty>
<gml:Point srsName="EPSG:32632" srsDimension="3">
<gml:pos>498049.8 6854929.5 679.8</gml:pos>
</gml:Point>
</gml:pointProperty>
</fme:Vegsperring>
</gml:featureMember>
<gml:featureMember>
<fme:Vegsperring gml:id="idb75b53a5-fb30-4bb8-9107-5c1e4ed0bff9">
<fme:SOSI_id>2862</fme:SOSI_id>
<fme:VEGSPERRINGTYPE>LÃ¥st bom</fme:VEGSPERRINGTYPE>
<fme:METER_TIL>193</fme:METER_TIL>
<fme:MALEMETODE>24</fme:MALEMETODE>
<fme:VEGSTATUS>V</fme:VEGSTATUS>
<fme:OBJTYPE>Vegsperring</fme:OBJTYPE>
<fme:VEGKATEGORI>S</fme:VEGKATEGORI>
<fme:DATAFANGSTDATO>20040812</fme:DATAFANGSTDATO>
<fme:KOMM>0515</fme:KOMM>
<fme:NOYAKTIGHET>200</fme:NOYAKTIGHET>
<fme:VEGNUMMER>57</fme:VEGNUMMER>
<fme:VFRADATO>19500101</fme:VFRADATO>
<fme:VKJORFLT/>
<fme:HOVEDPARSELL>1</fme:HOVEDPARSELL>
<fme:METER_FRA>193</fme:METER_FRA>
<gml:pointProperty>
<gml:Point srsName="EPSG:32632" srsDimension="3">
<gml:pos>501476.1 6842829.4 778.8</gml:pos>
</gml:Point>
</gml:pointProperty>
</fme:Vegsperring>
</gml:featureMember>
</gml:FeatureCollection>

C Source code script

```
# main
from lxml import etree
import wellForm
import validate
import strToFloat
import allCoor
import pointCoverByLine
import boundary

#Description of the functions:
#wellForm function uses the XML parser capability from lxml package to parse the
#XML/GML file.
#validate function uses the XML validation capability from lxml package to validate the
#XML/GML file against its schema.
#strToFloat function transform the string value to float. This is because the coordinates
#extracted from the GML file are of string format, and the comparison and implementation
#must based on float format. In addition, the string contains many coordinates with space
#for separating, so direct applying float(string) won't work.
#allCoor function does the job of taking out the necessary information for further usage by
#other functions. It includes taking out all the lineString coordinates, all the point
#coordinates, the IDs, and the dimensions of the coordinates.
#pointCoverByLine function compare the point coordinates from each point feature with all
#the line coordinates, to check if the point coordinates appear in the line coordinates. If not,
#then the point is not covered by lines in the dataset, and its gml:ID will be reported
#boundary function compare all the line coordinates with the boundary coordinates. The
#boundary coordinates defined in the header; they are the coordinates for the low-left-
#corner, and the up-right-corner. So the coordinates in the dataset must bigger than the
#'low-left-corner-coordinates' and smaller than the 'up-right-corner-coordinates'. The
#feature's gml:ID will be reported if the coordinates inside that feature are out of boundary.

#Upload the GML file for validation, and the corresponding schema
filePath = raw_input('the path of the gml file: ')
schemaPath = raw_input('the path of the gml schema: ')

# Parse the files, be well-formed, get the query-able GML file
gmlDoc = wellForm.parse(filePath)
schemaDoc = wellForm.parse(schemaPath)

# Validate the GML file against its schema
validate.valid(schemaDoc,gmlDoc)

# Take out necessary information, and store them in the structure which program can direct
#read from. The information includes coordinate, dimension, gml:ID
```

```
linecoordinates=allCoor.getAllLineCoor(gmlDoc)
floatAllLineCoor=allCoor.floatAllCoor(linecoordinates)
lineDimension=allCoor.dimension(linecoordinates)
lineID=allCoor.gmlID(linecoordinates)

pointcoordinates=allCoor.getAllPointCoor(gmlDoc)
floatAllPointCoor=allCoor.floatAllCoor(pointcoordinates)
pointDimension=allCoor.dimension(pointcoordinates)
pointID=allCoor.gmlID(pointcoordinates)

# check if all the points covered by line
pointCoverByLine.coverBy(floatAllPointCoor,floatAllLineCoor)

# Check if all coordinates in the dataset are inside the boundary
# lowerConer and upperCorner are the boundary coordanates
# Since all the point coordinates shoule be covered by line coordinates
# It is sufficient to only check if all the line coordinates are in boundary
findL = etree.ETXPath("//{http://www.opengis.net/gml}lowerCorner")
lowerCorner = findL(gmlDoc)[0].text
findU = etree.ETXPath("//{http://www.opengis.net/gml}upperCorner")
upperCorner = findU(gmlDoc)[0].text

floatLowerCorner = strToFloat.convert(lowerCorner)
floatUpperCorner = strToFloat.convert(upperCorner)

boundary.inBoundary(floatAllLineCoor)
```