Design and implementation of data quality model in spatial databases

Mesued Abdelkadir Mohammedreja

March, 2010

Design and implementation of data quality model in spatial databases

by

Mesued Abdelkadir Mohammedreja

Thesis submitted to the International Institute for Geo-information Science and Earth Observation in partial fulfillment of the requirements for the degree in Master of Science in *GeoInformatics*.

Degree Assessment Board

Thesis advisor	Ms. Dr. I. Ivanova Dr. Ir. R. A. de By
Thesis examiners	Chair: Dr. Ir. R. A. de By External examiner: Dr. S. de Bruin

ITC

INTERNATIONAL INSTITUTE FOR GEO-INFORMATION SCIENCE AND EARTH OBSERVATION

ENSCHEDE, THE NETHERLANDS

Disclaimer

This document describes work undertaken as part of a programme of study at the International Institute for Geo-information Science and Earth Observation (ITC). All views and opinions expressed therein remain the sole responsibility of the author, and do not necessarily represent those of the institute.

Abstract

Spatial data associated with spatial data quality increases the level of confidence for users to use that spatial data for their intended application. The main objective of this thesis project is to integrate spatial data quality with spatial data from design to the implementation of spatial databases. Based on reviews of the existing spatial data quality literature, the characteristics of each quality elements are identified. To solve some of the problems in the literature and fulfill design requirements for storing and evaluating spatial data quality, a modelling guideline of spatial data quality is proposed. The model enables spatial data quality elements to be integrally linked with spatial data for providing access and processing in spatial database, and thereafter stored and documented in the spatial databases structure at attribute, tuple, table, or database levels. Then we develop a prototype that allows users to request the quality information of a dataset from a client-side. With the prototype, spatial data quality elements can be retrieved from the database and report is generated automatically. The prototype has been tested against the dataset described in the use case. The implementation has shown that the prototype can report the quality information at multi-level structure of spatial databases. Therefore, users can get quality information and utilize it in a more practical way. This work develops an alternative approach to evaluate a given dataset's quality and store the quality information in a temporarily table in spatial database, which keep the database performance without storing the quality information permanently. The approaches for evaluating of datasets' quality information presented in this thesis project could support for further research.

Keywords

spatial data quality, design, prototype, database design, ISO, completeness, UML, positional accuracy, transformation, conceptual model, tuple

Acknowledgements

First and for most I would like to gratefully acknowledge to the Government of Afar National Regional State (Ethiopia), for the financial support of my study.

I am deeply indebted to my first supervisor, Ms. Dr. I. Ivánová for her earnest guidance, critical comment; encouragement and timely suggestion that made this research a success. Her intellectual advice, guidance, and regular discussions were very valuable and inspiring in the process of the proposal writing, research undertaking and thesis writing. Ms. Dr. I. Ivánová, your valuable advice, enthusiasm, encouragement and sustained commitment to the ideas of my MSc is greatly appreciated.

My deepest gratitude goes also to my second supervisor Dr. Ir. R. A. de By. His kind support and encouragement gives me strength right from the inception of the topic to the last minute of the research.

Many thanks to my classmates and all my friends, your friendship and encouragements were a real support throughout the research period.

Last but not least, I would like to express my deepest appreciation to my mother, Fatuma Nuru, and my beloved sisters and brothers, for their care and love, advice, and unlimited support in every respect during the course of my study.

Contents

Abstract i

Acknowledgements iii

List of Tables ix

List of Figures xi

1 Introduction 1
- 1.1 Motivation and problem statement . . . 1
- 1.2 Research identification . . . 3
  - 1.2.1 Research objectives . . . 3
  - 1.2.2 Research questions . . . 3
  - 1.2.3 Innovation aimed at . . . 3
- 1.3 Method adopted . . . 4
- 1.4 Thesis Outline . . . 4

2 Spatial data and spatial data quality 7
- 2.1 Introduction . . . 7
- 2.2 Trends of spatial data . . . 7
- 2.3 Concepts of spatial databases . . . 8
- 2.4 Spatial data quality . . . 9
  - 2.4.1 Use of spatial data quality . . . 9
  - 2.4.2 Communicating spatial data quality to users . . . 11
  - 2.4.3 Standards for spatial data quality . . . 12
- 2.5 Terms and definitions . . . 13
- 2.6 Elements of spatial data quality . . . 15
  - 2.6.1 Lineage . . . 15
  - 2.6.2 Purpose . . . 15
  - 2.6.3 Usage . . . 16
  - 2.6.4 Positional accuracy . . . 16
  - 2.6.5 Logical consistency . . . 18
  - 2.6.6 Completeness . . . 20
  - 2.6.7 Thematic accuracy . . . 21
  - 2.6.8 Temporal Accuracy . . . 23
- 2.7 Summary . . . 25

3 **Designing data quality model for a spatial database** **27**
3.1 Introduction . . . 27
3.2 Database design concepts . . . 27
3.2.1 Conceptual design . . . 28
3.2.2 Logical design . . . 29
3.2.3 Physical design . . . 29
3.3 Database models . . . 29
3.4 Current spatial data quality models . . . 31
3.5 Drawbacks of current spatial data quality models . . . 32
3.6 Design requirement for spatial data quality . . . 33
3.7 Modelling guidelines of spatial data quality . . . 34
3.7.1 Integrating the data quality element as table . . . 34
3.7.2 Integrating Data Quality Element (DQE) as functions . 35
3.7.3 Integrating DQE as tuple . . . 35
3.7.4 Integrating DQE as constraints . . . 35
3.8 Summary . . . 35

4 **Use case: Evaluating and storing spatial data quality in spatial database** **37**
4.1 Introduction . . . 37
4.2 Use case description . . . 37
4.2.1 User requirements of spatial data quality . . . 38
4.2.2 Dataset descriptions . . . 40
4.3 Quality evaluation procedures . . . 41
4.4 Spatial data quality requirements . . . 41
4.4.1 Applicable quality elements . . . 43
4.4.2 Data quality evaluation method . . . 43
4.5 Conceptual model for the use case . . . 44
4.6 Summary . . . 45

5 **Implementation of Spatial database integrated with spatial data quality model** **47**
5.1 Introduction . . . 47
5.2 Implementation tools . . . 47
5.3 Transformations from conceptual model to logical model . . . 49
5.4 Transformation from logical to physical model . . . 51
5.5 Functions for the implementation . . . 53
5.5.1 Function for evaluating the positional accuracy at table level . . . 53
5.5.2 Function for evaluating the positional accuracy of a dataset at tuple level . . . 54
5.5.3 Function for evaluating the completeness of a dataset . . 55
5.6 Prototype for evaluating and reporting the quality of spatial data 56
5.7 Summary . . . 57

6 Discussion, Conclusions, and Recommendations 63
6.1 Introduction 63
6.2 Results and discussion 63
6.3 Conclusions 65
6.4 Recommendations 67

Bibliography 69

A Appendix 73
A.1 Physical Database model before change made 73
A.2 The corrected Physical Database model 75
A.3 Function that evaluates the positional accuracy at table level . 77
A.4 Function that evaluate positional accuracy at tuple level 79
A.5 Function that evaluate the error of omission at attribute level . 81

B 83
B.1 Dataset's specification 83
B.1.1 Dataset's specifications 83
B.2 Overview quality elements 85

C 87
C.1 Code for user interface 87
C.1.1 Codes for the open layer 87
C.1.2 Codes for calculating the positional accuracy through the Web 89
C.1.3 Codes for connecting the function positional accuracy and completeness through the Web 92
C.1.4 The codes for final interface that users can interact with and display the output on the from 95

List of Tables

2.1 Summary of quality elements storage 25

4.1 Applicable quality elements and quality element requirements 43

5.1 mapping from conceptual to logical model transformation for the use case 59

List of Figures

2.1 The framework of data quality concepts [17] 10
2.2 Elements and subelements of spatial data quality as defined by ISO 19113 Geographic information quality principles 13

3.1 UML graphical notations [23] 32

4.1 Five steps for quality evaluation adopted from [19] 42
4.2 Data quality evaluation methods adopted from [19] 44
4.3 conceptual schema for use case 46

5.1 System architecture for the prototype user interface 48
5.2 EA standard transformation definition “DDL”, conversion template for class 49
5.3 Enterprise Architect DDL Transformation template for the use case 50
5.4 Logical model for use case 52
5.5 Flow diagram for evaluating positional accuracy at table level . 54
5.6 Flow diagram for evaluating positional accuracy at tuple level . 55
5.7 Flow diagram for evaluating error of omission at attribute level 60
5.8 Steps to retrieve the quality information from the prototype . . 61
5.9 Prototype interface for retrieving the quality information completeness 61
5.10 Prototype interface for retrieving quality information of dataset at tuple level. 62

B.1 Relationship of data product specification to metadata [20] . . . 84

List of Acronyms

ADT	Abstract Data Type
CWA	Closed World Assumption
DBMS	Database Management System
DDL	Data Definition Language
DQE	Data Quality Element
EA	Enterprise Architecture
EA SDK	EA Software Developer Kit
ER	Entity-Relationship
FGDC	Federal Geographic Data Committee
HTML	Hyper Text Markup Language
ISO	International Organization for Standardization
GIS	Geographical Information System
GML	Geographic Markup Language
MDA	Model Driven Architecture
OGC	Open Geospatial Consortium
OMT	Object Modeling Techniques
OWA	Open World Assumption
PHP	Hypertext Preprocessor
QIMM	Quality Information Management Model
RMSE	Root Mean Square Error
SDBMS	Spatial Database Management System
SDI	Spatial Data Infrastructure
TIN	Triangulated Irregular Network

UML	Unified Modeling Language
WFS	Web Feature Service
WMS	Web Map Service
WPS	Web Processing Service
XML	eXtensible Markup Language

Chapter 1

Introduction

1.1 Motivation and problem statement

Data is a representation of real world objects, which is storable in the databases. It is also retrievable from structured databases and can be transferable via a network [6]. Creation and manipulation of the data is achievable by using a matured software system called Database Management System (DBMS). DBMS optimally structures, stores, manipulates, safeguards, and recovers database instances and their metadata. In addition, it provides data integrity functions and handles large volume of data. A spatial database is a special category of databases in which objects and features in space are captured as geometrical point, linear and areal data types, with additional support added to the conventional databases.

When dealing with data and information we are usually trying to represent some part of the real world. Model is a representation of human conceptualization of reality. The view of reality is represented in a simplified manner, which still satisfies the basic information requirement of a user. A data model is thus defined as an abstraction of the real world that incorporates only those properties of the real object considered relevant for the application at hand. A spatial data model is therefore a data model of information with spatial components. Spatially referenced data can be modelled using either field based or object based spatial data modelling approach [41]. Field is a geographic phenomenon for which, for every point in the study area, a value can be determined [32]. Temperature, pressure, and elevation are good examples of field. Object is a geographic phenomenon for which, for every point in the study area, a value can not be determined and such objects are usually well distinguished. They are discrete and are commonly bounded entities [32].

Spatial data handling techniques usually involves different process starting from data acquisition, storage and maintenance, analysis and output. For years, this has been made possible by using analogue data sources, manual processing and the production of paper maps. Hence the introduction of information technologies and computers in all aspects of spatial data handling become highly increase. Geographical Information System (GIS) is the dominant and most widely used software technology in spatial data handling [34].

Today, technology has made it extremely easy for users to find and access spatial data through the Internet. Hence, large authorities that organize spatial databases that are distributed and shared by multiple users at regional and local scales continue to develop, in the context of Spatial Data Infrastructure (SDI) [11]. In line, the need to accumulate, store, and communicate spatial data quality information to such large number of users has become a crucial component of the SDI.

Quality is defined as "the totality of characteristics of a product that bear on its ability to satisfy stated and implied needs" [17]. Spatial data accompanied with quality information is expressed as its appropriateness for an intended application. The term fitness for use is used interchangeably [2]. Proper evaluation and documentation of spatial data quality by data producers allow them to manage effectively data production, storage, updating and reuse. Users can use this information to decide whether the dataset is fit for their application and to minimize the possibility of misuse as well. The spatial data quality elements as identified in International Organization for Standardization (ISO) 19113 (Quality principles), are lineage, usage, purpose, positional accuracy, attribute accuracy, logical consistency, temporal accuracy, and completeness. These elements are reported as part of metadata.

The state of art to communicate quality information of spatial dataset is metadata. It is often provided as a text or stored separately from the data. Using metadata for exchanging quality status of datasets enables users retrieve quality information for overall dataset [31]. However, users who are interested at different hierarchy level of dataset's quality information may not get enough information from the metadata. The metadata communication of spatial data around GIS lack tools that enables easy management of embedded data quality information [48]. Users rarely consult the metadata during GIS operations, which is provided separately from the data, and due to lack of error awareness users usually undermine the outcome of GIS due to expected errors.

Web services are most widely used for accessing and performing geographic computation tasks on spatial datasets. Users can make use of the Web to access spatial data from heterogeneous spatial databases in the form of eXtensible Markup Language (XML) or Geographic Markup Language (GML). Nevertheless, the spatial data which is derived from Web services may not be useful for the user's intended application. This is due to current spatial data quality information being separately mentioned or provided from the actual data. To understand the quality information of the derived data from heterogeneous spatial datasets through the Web services is too complex. This is because we have to look into every spatial dataset's metadata, which in fact is time consuming and complicated while an average user does not even know how to use metadata.

Existing spatial data models, which are the building blocks of the spatial databases, disregard aspects of spatial data quality [8]. However, as data quality is an integral part of the SDI process, its incorporation during the conceptual modelling of the spatial database is essential. Due to different levels of objects in spatial database, users may be interested to examine data quality information at database, table, tuple, or attribute levels. Hence, modelling spatial database with data quality information at multilevel structure is the basis to store data quality information [22]. This enables users to retrieve spatial data associated with spatial data quality information from spatial database at the desired level in hierarchy.

1.2 Research identification

The primary issues addressed in the current research project can be defined through the following research objectives and research questions.

1.2.1 Research objectives

1. To review concepts and mathematical backgrounds of spatial data quality elements with existing spatial database models.

2. To develop a spatial data quality model as an integral part of the spatial database.

3. To create and store a set of functions that evaluate and test the quality of the spatial dataset in a spatial database.

4. To develop a mechanism for sharing the data quality information of spatial dataset for different types of users such as Web service or other spatial databases.

5. To examine the developed spatial data quality model using a selected use case with predefined requirements and achievement criteria.

1.2.2 Research questions

1. What are the theoretical and mathematical concepts that are needed to build a spatial data quality model?

2. How to design spatial data quality model as an integral part of spatial database?

3. How to define and store a set of functions that evaluate and test the quality of spatial dataset in a spatial database?

4. How to explore the stored functions of a quality model in spatial database to be accessed by different types of user?

5. How to test and validate the effectiveness of the developed model using real world data?

1.2.3 Innovation aimed at

The novelty of the research is in developing a spatial data quality model that is integrated with a spatial database model, in a way that different types of users can retrieve the quality information embedded with the spatial data they need.

1.3 Method adopted

The research method consists of four phases Review of literatures and identification of the research problems, design, implementation, and testing. The project begins by reviewing literature and identifying the possible requirements of spatial data quality in the context of SDI. Then, the development of a spatial database with quality model and creation of set of functions proceed. Using the real data pragmatic testing will be done to the developed model to confirm whether the intended objectives are met. The following description explains the activities in detail

1. Review of literatures and identification of the research problems: In this phase a through review of related literature that helps in identifying, first, problems to be addressed, the user requirements, second, the spatial data quality elements and their definitions, and third, the possible mathematical expressions of the spatial data quality elements that help us during the design of spatial database with spatial data quality elements.

2. Design: The development of a spatial database design with spatial data quality to provide a solution to the research problems identified during phase one. This involves developing a spatial database design linking with the relevant spatial data quality elements. This phase discusses how the spatial data quality elements link to the different hierarchy of spatial data in spatial databases during the design process. In this phase we will have a use case that can help us for the design of spatial database.

3. Implementation: To make a transformation of the conceptual design built in phase two into its equivalent logical stage. To be able to make a smooth transformation, a set of functions will be coded using database programming language. The physical design is included in this stage and the possible platform that may consider different types of user will be selected. Finally, the design obtained from the previous step should be optimized according to the needs of user requirements and storage capabilities.

4. Testing: In this phase, a practical test will take place to check whether the created spatial database with spatial data quality model works according to user requirements which were specified in method one of this thesis. The test will be evaluated according the requirements mentioned in phase one the efficiency in storing, querying, and proper data quality evaluation result(fitness for use) for a given dataset.

1.4 Thesis Outline

The thesis is organized in sex chapters as follows:

Chapter 1 describes the motivation to do this research, states the problem, research objectives and questions, and describes the methods adopted in the research.

Chapter 2 reviews the trends of spatial data from GIS to spatial databases. It first addresses the trends of spatial data in storing, manipulating and in communicating with the increasing the advanced technology, the possible differences between the GIS and spatial databases in storage, analysis of spatial data. It discussed the definition of spatial data quality, identifies the spatial data quality elements, the current technology for communicating spatial data quality. The research work associated with standards of spatial data quality is also reviewed. This review provides the background on spatial data quality and identifies the current methods of spatial data quality communication, which are important for this project. Finally, the possible storage, reporting mechanism, data quality measure of each quality elements are identified. Specifically the storage considers the structure of spatial database in order to integrate the quality information with spatial data it describes.

Chapter 3 provides a review of the processes in database design concepts. From the existing database models it identifies the possible database model that can be implemented in the project. It also discussed about the current spatial data quality models and identifies the drawbacks of the existing models. Finally it discussed on the design requirements of spatial data quality in spatial databases and proposes the possible modelling principles of spatial data quality with respect to the structure of spatial databases.

Chapter 4 provides a detail use case description. Accordingly, it identifies the user requirements of spatial data quality to different types of users like human users, Web service users and users of other spatial databases. For the purpose of implementation it describes the dataset specifications. It also discussed the quality evaluation procedure according to ISO 19114 quality evaluation procedures. The applicable quality elements for the dataset are identified. It discussed the quality evaluation methods and finally, according to the described dataset and identified the applicable quality elements a conceptual model is developed and the model integrates spatial data with spatial data quality.

Chapter 5 contains prototype for the implementation and testing part of the project. The concepts of transformational concepts from platform independent to platform dependent are identified. The capability of Enterprise Architecture (EA) and limitations, have discussed. The conceptual model developed in chapter 4 is transformed to its parallel logical model as well as the physical model taken place under this chapter. The physical tuning during transformation from logical to physical model discussed. The possible amendments to fulfill the criteria mentioned in the project under this transformation. Both the creation of functions and testing the prototype is done under this chapter.

Chapter 6 Presents and discussed the results of the project execution. The chapter also presents the conclusions drawn and the recommendations made for the future improvement on topics for further research.

Chapter 2

Spatial data and spatial data quality

2.1 Introduction

This chapter provides review of literature on different aspects of spatial data quality, storage and communication. In Section 2.2, trends of spatial data are described in relation to the improvement of technology in collecting, storing, analyzing, and visualizing of spatial data. Section 2.3, the concept of spatial database is introduced; while Section 2.4 we discussed the definitions of spatial data quality. In addition, the standards for documenting, the current technology for communicating, and use of spatial data quality are presented. The terms and definitions are presented in Section 2.5. The review also shades light on points related to definitions, storage, report, and data quality measure of each quality elements as discussed in Section 2.6. Finally, we conclude the Chapter with a summary of discussion in Section 2.7.

2.2 Trends of spatial data

The introduction of GIS in the 1960s improved the capacities for collecting, storing, analyzing and visualizing of spatial data. Geographic information is subject to quality deficiencies which had traditionally been a subject of concern among the cartographers, land surveyors and geographers [40]. However, with the availability of the new GIS platform, spatial data could be more easily exchanged and manipulated by a larger number of users. Since the 1980s, with the popularization of GIS and the so called “democratization” of spatial data [14], concerns have been raised about the widespread use of quality impaired spatial data. As a result, a wide range of users being able to combine and manipulate information from diverse data sources. Thus generation of a variety of GIS products that were only restricted to experts. As the technology was advansing and the usage of the Internet and other networks became more customary to the public, GIS and spatial data became readily available to a broader user community. These new means of communicating spatial data invited a number of interested users to manipulate spatial data, produce new spatial products and exchange them with other users over the Internet [10].

A spatial database is not the same thing as a GIS, though both systems share a number of characteristics. A spatial database focuses on the functions like concurrency, storage, integrity, and querying, specifically, but not only, spatial data. A GIS, on the other hand, focuses on operating on spatial data. It provides a rich set of analysis functions which allow a user to effect powerful transformations on spatial data. Obviously, a GIS must also store its data, and for this it provided relatively rudimentary facilities. GIS applications use GIS for spatial analysis and a separate spatial database for the data storage. Whenever a GIS needs data, it has to extract it from the spatial database. Therefore, GIS is the principal technology of interest in Spatial Database Management System (SDBMS)[41]. SDBMSs are also designed to handle very large amount of spatial data stored on secondary devices (e.g. magnetic disks, CD-ROM etc.), using specialized indices and query processing techniques. A GIS can be built as the front end of an SDBMS. Before a GIS can carry out any analysis of spatial data, it accesses the data from SDBMS. Thus an efficient SDBMS can greatly increase the efficiency and productivity of a GIS[41].

Moreover, users of spatial information are no longer limited to professionals and researchers. However, it includes people who plan their travel itinerary using an online map service on the Internet, who check the weather conditions and who access geographically referenced information about their communities, the environment, etc. through the World Wide Web. The changes in the collection, management and use of spatial information noted above could probably not have happened; at least not to the extent that is evident, without the power of spatial databases [4]. This trend by which spatial data increased in volume and improved in accessibility, gathered the attention of researchers in the field of GIS;thus prompting action to increase user awareness of spatial data quality.

2.3 Concepts of spatial databases

Spatial database describes the location and shape of geographic features, and their spatial relationship to other features. The information contained in the spatial database is held in form of digital coordinates, which describe the spatial features. These can be points, lines, or polygons.

The SDBMS supports spatial data models, spatial Abstract Data Type (ADT), and a query language from which these ADTs can be used. It supports spatial indexing, and domain specific rules for query optimization. SDBMS are characterized by the large volumes of data they are designed and required to handle. For example spatial database that store remote sensing satellite imagery may require several petabytes (1,000 terabytes, or 1015 bytes) to store the images acquired in a single year. In order to accommodate this huge volume of data, SDBMs store data in arrays of disks, called secondary storage devices. When data are required by a query, they are read into the Central Processing Unit (CPU), which is called the primary storage [4].

The main goal of a spatial database is the effective and efficient handling of spatial data types in two, three or higher dimensional spaces, and the ability to answer queries taking into consideration the spatial data properties [41]. In this research study, we use the following terminologies for the structure of spatial databases:

- Database for the collection of group of objects
- Table for the collection of objects that have the same attributes
- Tuple for an object and information about that object
- Attribute for the characteristics of the objects

2.4 Spatial data quality

Spatial data are a “model of reality”, a simplified representation of the complex geographic reality [35]. Every map or database is therefore a model, produced for a certain purpose in which certain elements are simplified, grouped, or eliminated. This is to make the representation more understandable and thereby encourage the process of information exchange. During the modelling of the spatial data from the real geographic space, a set of factors is added or removed that may lead to the source of errors.

There are a number of definitions for spatial data quality. Korte [5] defines spatial data quality as the degree to which spatial data accurately represents the real world, the suitability of the data for a certain purpose and the degree to which the data meet a specific accuracy standard. According to Jakobsson [3] spatial data quality is the difference between the universe of discourse and the dataset, where the dataset is defined as an identifiable collection of related data and the universe of discourse as the view of the real world that includes everything of interest. The definition of spatial data quality according Worboys and Duckham [29] is “from a client’s perspective a data set may be fit for use even if its quality is low, as long as it suits the client’s purpose”.

The definition of Korte [5] and Jakobsson [3] can be classified as a producer’s view of spatial data quality while that of Worboys and Duckham [29] can be classified as a user’s perspective on spatial data quality. According to [17], the definition of spatial data quality is applicable to data producers providing quality information. This is to describe and assess how well a dataset meets its universe of discourse to data users. Thus attempting to determine whether or not specific geographic data is of sufficient quality for their particular application. This spatial data quality concept is shown in figure 2.1. In this thesis, we adopt a definition of spatial data quality that combines the definition from both the producers’ and users’ perspective:

Spatial data quality is “totality of characteristics of a product that bear on its ability to satisfy stated and implied needs” [17].

2.4.1 Use of spatial data quality

Recently, the use of spatial data is becoming important in decision making. This trend has created a high demand for vast amount of spatial data. To meet the

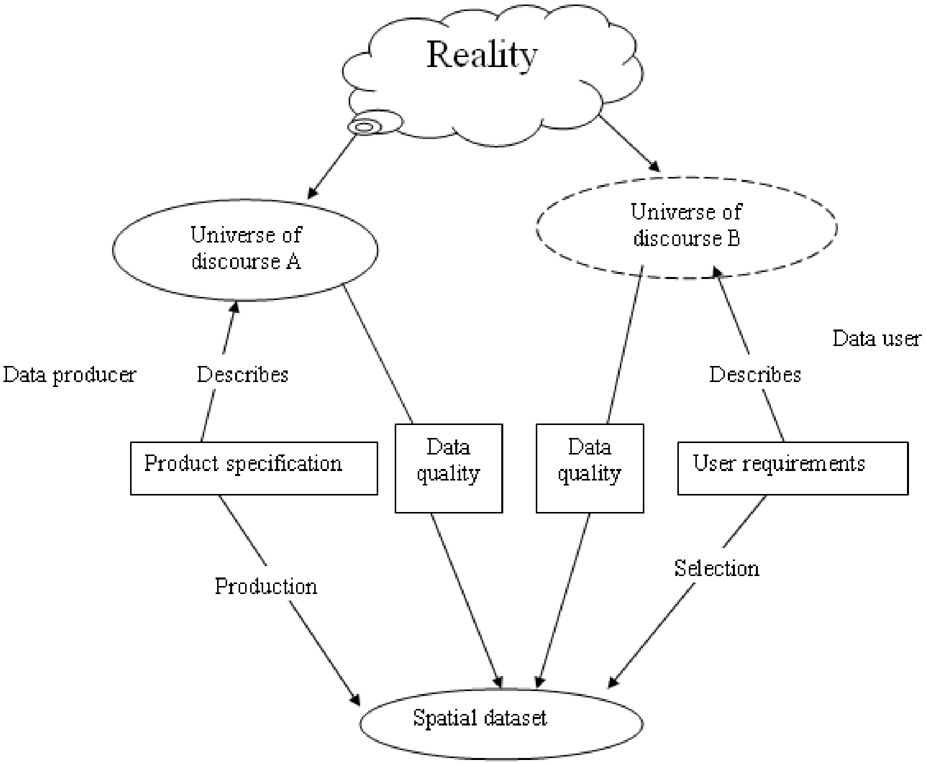


Figure 2.1: The framework of data quality concepts (17)

demands, private companies produce large amount of spatial data, which were, historically, generated by government agencies [49]. There is also the fear of litigation arising from damage caused by the use of data by other parties or damage suffered as a result of decisions made based on low quality data [12]. Consumers have interest in data quality as a means of deciding the fitness of data for their respective use. The increase in user friendly global positioning technologies has enabled non experts to join the data producer's domain. This has created a data-rich society while increasing the risk of misuse of data and use of low quality data. Organizations and individuals are integrating and sharing datasets as a way of improving efficiency and reducing data related costs [40]. Due to the wide spread use of spatial data, the spatial data have played a critical role in various application domains. They have also been employed by a variety of GIS users, from general users using Internet to GIS professionals using highly advanced systems. This increasing demand of spatial data has facilitated spatial data generation at different periods by different government agencies and private companies incorporating different acquisition technologies and softwares. However, the extensive use of spatial data by various applicants and through various data acquisition approaches tremendously increased the risk of data misuse. Such cases of misuse have already occurred and led to the significant social, political and economical consequences, and sometimes raised serious issues with regard to the legal liability [26, 49]. Some of the problems arise due to backside of the current communication of spatial data quality. For effective use of spatial data quality, improvement in the communication of spatial data quality to user is required. This issue is discussed in the following section.

2.4.2 Communicating spatial data quality to users

The information about spatial data quality should allow users to assess how well data can fit their needs. To achieve this goal, different ways of spatial data quality communication have been suggested in literature. The most common way currently used to communicate spatial data quality information to users is metadata. One of the main objectives of metadata is to enable data users to determine the fitness for use of a dataset for a given purpose. However, metadata do not serve this goal for inexperienced users and are also difficult to be understood by many experienced users [37]. This is because the spatial data quality reports are written from the perspective of the data producer. Today's metadata provided with datasets are often stored in a text file which is separated from their data, hence reduces their usefulness for GIS functions [35]. It is important to notice that quality information is useful when it can be explicitly linked with the objects it describe in a database. Approaches which separate quality descriptions from data, risk reducing ease of access. Furthermore, this issue increases the difficulty of updating quality during data manipulation. This is the fact that data quality is usually treated as static by current metadata standards [49].

Another approach for communicating quality information relies on visualization techniques. Visualization provides an effective means for presenting complex information. It has been proposed as an approach for communicating spatial data quality to users [35]. However, the visualizing technique is also the same problem as metadata. This is because the metadata is maintained as text files in only loose association with the data. It is not an effective format for supporting visualization. Therefore, in order for the visualization methodologies to display quality information at various levels of granularity, explicit association of data quality with the dataset at an appropriate level of detail should be modelled.

Web services have become more widely used for accessing and performing geographic computation tasks on spatial datasets [1]. For example, Web Processing Service (WPS) is such a standard developed by the Open Geospatial Consortium (OGC). When using spatial data or generating new data with data processing, e.g. by applying spatial analysis functionality provided by a Web service, it is important to know about the quality of the data. The spatial data quality does not only inform the user about the fitness for use of the data in certain fields of application, but also enables the user to interpret the results from an analysis. However, the spatial data derived from Web services may not be useful for the user's intended application. This is due to current spatial data quality information being separately mentioned from the actual data. To understand the quality information of the derived data from heterogeneous spatial datasets through the Web services is complex. This is because we have to look in every spatial dataset's metadata, which is time consuming while even the average user does not know how to use the metadata.

The state of art of communicating spatial data quality make it difficult for data users to easily access, understand and adapt quality information for a given application. In order to communicate quality information more efficiently to users to help them evaluate the fitness of data, there is a need to provide a more effective approach. This is for presenting quality information in a more informative and understandable way. The approach may include:

- A data quality model that integrates spatial data quality with spatial data is one approach that users can access spatial data with spatial data quality at the possible hierarchy.
- Functions that enable updates of quality of spatial data during spatial data manipulation takes place.

2.4.3 Standards for spatial data quality

Spatial data can be considered as products that are produced with respect to a product specification. This satisfies the needs for a variety of users that intend to use the spatial data for different applications. Since different applications require different quality information of the data, it is not possible to state the fitness for use in a common way. The quality of different datasets to be comparable, it is important to describe in a standardized form. ISO/TC 211 provides a series of standards that deal with various aspects of geographical information which includes:

- ISO 19113 Geographic information, Quality principles
- ISO 19114 Geographic information, Quality evaluation procedures
- ISO 19115 Geographic information, Metadata and
- ISO 19138 Geographic information, Data quality measures

In addition, a variety of established standards around the world had been developed at international, national, and sub national levels [15]. The exact composition of spatial data quality standards and the definition of individual elements within different standards exhibit wide variation. For this thesis, we use the spatial quality elements according to definitions of ISO 19100 series, more specifically in ISO 19113, ISO 19114, ISO 19115, and ISO 19138.

The spatial data quality section in most of these standards is organized into a number of quality elements. The quality of a dataset according to [17] standard is described using components called data quality elements and data quality overview elements. The data quality overview elements provide non quantitative information and it includes lineage, purpose, and usage. The data quality elements consist of five quality elements with quality subelements. These elements are completeness, thematic accuracy, positional accuracy, logical consistency and temporal accuracy. Figure 2.2 provides an overview of each quality elements with their subelements.

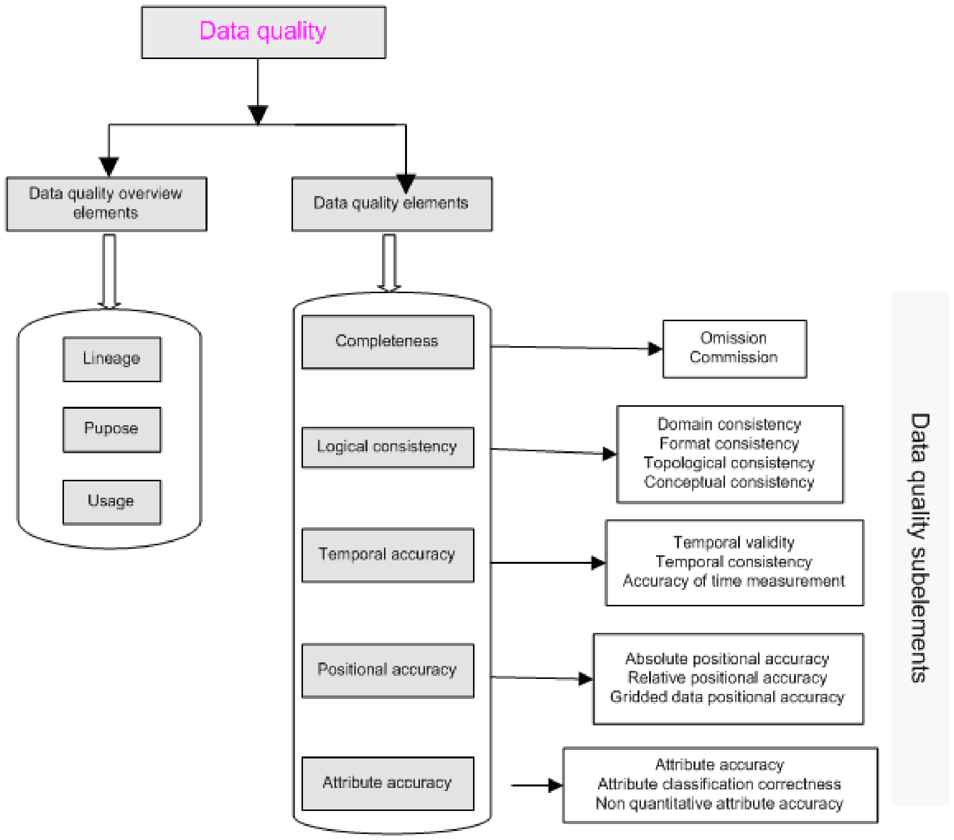


Figure 2.2: Elements and subelements of spatial data quality as defined by ISO 19113 Geographic information quality principles

2.5 Terms and definitions

Before we discusses the definition of each spatial data quality elements, we will discuss some basic terminology that we are going to apply through out the thesis for evaluation and reporting of spatial data quality. These definitions are adopted from ISO 19113,ISO 19114, and ISO 19115.

Data quality scope

Extent or characteristic(s) of the data for which quality information reported is called data quality scope [17]. For example, the data quality scope for a dataset can comprise a dataset series to which the dataset belongs, the dataset itself, or a smaller grouping of data located physically within the dataset sharing common characteristics. This can be an identified feature type, feature attribute, feature relationship, or a specified geographic or temporal extent.

Data quality measure

Data quality measure is "evaluation of data quality subelements" [17]. A data quality measure shall briefly describe and name, where a name exists, the type of test being applied to the data specified by a data quality scope.

Conformance quality level

Threshold value or set of threshold values for data quality results used to determine how well a dataset meets the criteria set forth in its product specification or user requirements to be published [17].

Dataset

Dataset is an identifiable collection of data [18] and may be a smaller grouping of data which, though limited by some constraint such as spatial extent or feature type, is located physically within a larger dataset. For purposes of data quality evaluation, a dataset may be as small as a single feature or feature attribute contained within a larger dataset.

Evaluation method

Evaluation method can be divided in to two direct and indirect evaluation methods. Direct evaluation method is evaluating the quality of a dataset based on inspection of the items within the dataset and indirect evaluation method is evaluating the quality of a dataset based on external knowledge [17].

Reference dataset

Data accepted as representing the universe of discourse, to be used as reference for direct external quality evaluation methods.

Data quality evaluation procedure

One data quality evaluation procedure shall be provided for each data quality measure. A data quality evaluation procedure shall describe the methodology used to apply a data quality measure to the data specified by a data quality scope and shall include the reporting of the methodology [19].

Data quality result

One data quality result shall be provided for each data quality measure. The data quality result shall be either the value or set of values obtained from applying a data quality evaluation procedure to the data specified by a data quality scope.

Data quality value type

One data quality value type shall be provided for each data quality result. For example, the data quality value type for pass-fail is boolean. The other data quality value types are number, ratio, or percentage.

2.6 Elements of spatial data quality

In this section, we discuss the definition, storage, report, and quality measure of each quality elements.

2.6.1 Lineage

Lineage is the history of dataset. By history, we mean the recounting of the life cycle of a dataset, from its collection or acquisition, through the many stages of compilations, corrections, conversions, and transformations to the generation of new interpreted products. It is one of the non quantitative quality information and is usually the first component in a data quality statement. This is probably because all of the other components of data quality are affected by the contents of the lineage and vice versa. Efforts in formalizing lineage information are summarized in metadata standards such as in [18] consisting of items like “Source Information” or “Process Step”.
Storage: The quality element lineage may be stored as a new column in the existing tables in case the lineage information dealing with the currency of the data. Since currency information may be available for each tuple. When the lineage information is described for overall dataset, it should be stored as separate table.
Data quality measure: lineage is evaluated by indirect evaluation method (based on external knowledge).
Report: According to [17] standards lineage contains two unique components, source information and process step. This is useful when determining the suitability of a dataset for a given use.
Example: The dataset consists of facility distribution of the region.

- Source map: Afar National Regional State map.

- Processing involved: Point data for each site collected by GPS with accuracy of 2m. The points were projected from degree to UTM reading. The projected points were exported as text to map source. Finally imported to Arcview as points.

2.6.2 Purpose

Purpose is one of the data quality overview elements that describe the rationale for creating a dataset and contain information about its intended use. The dataset’s intended use may not be necessarily the same as its actual use. Mostly

the actual use of a dataset is described using the data quality overview element usage [17]. This is useful when identifying the possible value of a dataset.
Storage: Purpose can be stored in a separate table. which deals only about the static quality elements. It can not store with the dataset itself, because it deals with the entire dataset. Therefore, to store the overview quality element purpose, a new table quality must be created and linked with the table it describes.
Data quality measure: Purpose is evaluated by indirect evaluation method (based on external knowledge).
Report: It is reported as metadata in conformance with requirements of ISO 19115.
Example: The purpose for the dataset mentioned in the lineage example is: To identify the natural resource distribution per district and use for planning.

2.6.3 Usage

Usage is one of the data quality overview elements that describes the application(s) for which a dataset can be used. It describes uses of the dataset by the data producer or by other, distinct, data users [17]. Usage of a dataset is considered to be important from the point view of new users. If a dataset has been used for applications similar to the one envisaged by the current user, it clearly adds to the potential user's confidence. Also previous combinations of the actual dataset with other datasets can be an indication of the quality of the dataset. It gives an overview of the applications for which the information in the dataset has been used previously and how well the data fitted in these applications. Also the potential use, is an important aspect in this which gives an indication of the possibilities of the data seen from the providers' perspective.
Storage: Usage can be stored as in a separate table. It cannot be stored with the dataset itself, because it deals with the entire dataset. Therefore, to store the overview quality element purpose, a new table can be created and linked with the dataset it describes.
Data quality measure: Usage is evaluated by indirect evaluation method (based on external knowledge).
Report: It is reported as metadata in conformance with requirements of ISO 19115.
Example: The usage for the dataset mentioned in the lineage example is: To analysis the socio economic activity of the region and identify the gap between the districts of the region.

2.6.4 Positional accuracy

Positional accuracy refers to the discrepancy of the location of the database objects in relational to their true positions on the earth. The position of objects in the database is a set of cardinal values that allow the objects to be positioned in an appropriate coordinate system [49]. Descriptions of positional accuracy must consider the quality of the final product after all transformations. According to [17] the positional accuracy has three data quality subelements, namely absolute positional accuracy, relative positional accuracy, and gridded data accuracy.

Absolute positional accuracy

Absolute positional accuracy is the measure of how a spatial objects is accurately positioned on the map with respect to its true position on the ground, within an absolute reference frame. Absolute accuracy is the closeness of the coordinate values in a dataset to values accepted as or being true. This closeness is valued based on a comparison between absolute location of a feature and its location in the universe of discourse.
Data quality measure: The horizontal or vertical accuracy can be derived from statistical test analysis; the results can be described by error indicators such as Root Mean Square Error (RMSE). For example RMSE for positional accuracy of the horizontal accuracy is defined as follows.

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}[(x_i - x_{ref})^2 + (y_i - y_{ref})^2]}, \tag{2.1}$$

when n is the total number of point in the dataset; (x_i, y_i) is the absolute coordinate value of the i^{th} point in the dataset, and (x_{ref}, y_{ref}) is the absolute coordinate value of the point in the universe of discourse. The result is a single value for overall dataset and can be reported with other quality elements. This quality element can also be evaluated for each tuple stored in the table using the following equation

$$RMSE = \sqrt{(x - x_{ref})^2 + (y - y_{ref})^2}, \tag{2.2}$$

When (x, y) is the absolute coordinate value of a point in the dataset, and (x_{ref}, y_{ref}) is the absolute coordinate value of the point in the universe of discourse. The data type of the output is real.
Storage: The absolute positional accuracy can be stored as an attribute with the same table it represent's or in a different quality table and link to the source table. When the positional accuracy is for all the entire dataset it is stored with other quality information in a quality report. The data type of the output is real.
Report: The positional accuracy can be reported either as additional attributes of each spatial object or through a quality report with other quality elements. The report should include the date of the test.

Relative positional accuracy

Relative positional accuracy is the measure of how objects are positioned relative to each other. For example, the distance measurement between two electric poles on the ground and the distance measurement between two poles on the map must be within certain relative accuracy +/-m. The data quality measure, storage, report of this quality subelement is the same as the absolute positional accuracy.

Gridded data positional accuracy

Gridded data positional accuracy is indicating the error between relation position of grid points and expected true position. For example, RMSE of between Triangulated Irregular Network (TIN) grid points elevation value and result of inspection field survey is 1.2m, then the quality of “gridded data positional accuracy” is 1.2m RMSE. The data quality measure, storage, report of this quality subelement is the same as the absolute or relative positional accuracy.

So far we have discussed about the positional accuracy of points. The positional accuracy for lines and polygons are different notions than the positional accuracy of point datasets. The positional accuracy of lines and polygons are more complicated than the positional accuracy of points. The positional accuracy of line can be divided into two components positional point accuracy and shape fidelity [16]. Positional point accuracy can easily be given for well defined points on the line (example, the end points). Shape fidelity is useful to talk about the accuracy of a line as compared to another line. It should indicate to what extent the curvature of two lines are similar. As we have discussed about the point positional accuracy in equation 2.2, we can apply also to find the positional accuracy of end points of the line. End points could be cross roads, dead ends in a road network, end points in a tube network etc... This can be done by identifying the corresponding end points in the reference dataset and the dataset of unknown positional accuracy [16]. The other measures of line positional accuracy are epsilon band error and buffer overlay method. The epsilon band error is based on defining an uncertainty band around a linear element. That is the wider the band the greater the uncertainty in the location of linear element. Epsilon is the value that defines the width of the band. Epsilon can be derived by error propagation or by comparing the linear element to the element collected with higher accuracy [24]. The buffer overlay method is based on defining a buffer around the line of higher accuracy and computing the percentage of the length of the less accurate line within the buffer zone. Then, the width of the buffer is increasing and the percentage is computed again. The process is repeated several times. This allows generating a probability distribution.

2.6.5 Logical consistency

Logical consistency is the degree of adherence to the logical data structure rules and spatial data attribute rules. According to [17] it is classified into conceptual consistency, domain consistency, format consistency, and topological consistency. Each portion of the logical consistency subclass can be described as the following.

Conceptual consistency

Conceptual consistency aims to report any violation of the conceptual schema of features [17].
Data quality measure: The conceptual consistency can be measured by counting the feature numbers and the relationships between features that violate the

dataset conceptual schema, or by dividing the result into the number of features and relationships between dataset features and multiplying the result by 100.
Report: The output of the conceptual consistency may be expressed by ratio or percentage and according to quality evaluation report, if the output is greater than the conformance quality level the dataset fails. The number of features that violates the conceptual schema can also reported in the metadata for that data specified by the data quality scope.
Storage: Conceptual consistency is stored separately from dataset it represent's with other quality elements in ISO 19115 metadata. This quality element is stored as function in spatial database. It is evaluated at run time or during the process. This quality subelement is specified for entire dataset.
Example: Two datasets with one-to-one relation in the conceptual schema are within the data quality scope. One feature relationship exists which is not defined in the conceptual schema then the dataset fails.

Topological consistency

Topological consistency is a primary type of consistency in the spatial domain, which describes the relationships between features in the database and requires conformity to certain topological rules [17].
Data quality measure: For each feature type, the number of items within the dataset scope having topological inconsistency is counted. This number is then compared with the ratio of permissible error in total number of relevant data items. The output can be indicated as percentage, or ratio.
Report: The topological consistency can reported according to the ISO 19114 quality evaluation report if the output is larger than the acceptable conformance quality level, then the dataset is fail. It is not specified for each data item but specified for the entire dataset.
Storage: Topological consistency is stored separately from the source dataset. It can be store with other quality elements in ISO 19115 metadata. This quality element is stored as function in spatial database. It is evaluated at run time or during the process.
Example: “overshoot or undershoot” is a typical topological rule for linear features; “all polygons should be closed” for polygon features. There are also user defined rules for a specific application, such as “school areas should not overlap residential areas”.

Domain consistency

Domain consistency deals with item attributes within a data quality scope that does not lie within a certain domain [17].
Data quality measure: The domain consistency is determined based on the comparison of item attributes within the data as specified by the scope against acceptable attribute domains. The number of items with attributes outside the acceptable domain attribute is counted. This number is then compared with the ratio of permissible error in total number of relevant data items. The output may be expressed by ratio, or percentage.

Report: Domain consistency can be reported according to quality evaluation report. If the output is larger than the acceptable conformance quality level, then the dataset is fail. It is not specified for each data item but specified for the entire dataset.
Storage: Domain consistency can be stored separately from the source dataset. It can be store with other quality elements in ISO 19115 metadata. This quality element is stored as function in spatial database. It is evaluated at run time or during the process.
Example: The attribute defines value from 1 to 10, if the value is out of the range, then there is a domain consistency error.

Format consistency

Format consistency is the degree of adherence to the format of a specified field [17]. This element indicates whether the data that is defined to be coded in a format has a correct format.
Data quality measure: Format consistency is calculated by counting the items having format violations from the specified format. Items from the specified field definitions and structures having format violations are counted and recorded in metadata.
Report: The output of format consistency can be expressed by ratio or percentage and according to quality evaluation report if the output is larger than the acceptable conformance quality level, then the dataset is fail. It is not specified for each data item but specified for the entire dataset.
Storage: The output of format consistency is not specified for each tuple but is specified in the entire dataset. It is stored as function in spatial database and is evaluated at run time or during the process. The output is stored with other quality elements.
Example: If the error in the tag name of data created as XML document is 0%, then it is formatly consistence.

2.6.6 Completeness

Completeness is defined as “presence and absence of features, their attributes and their relationships” [17]. It is one of the data quality elements with two data quality subelements commission and omission. As stated in [6], there are two different assumptions on completeness of data represented in a table. The Closed World Assumption (CWA) states that only the values actually present in a relational table, and no other values represent facts of the real world. In the Open World Assumption (OWA) we can state neither the truth nor the falsity of facts not represented in the tuples of instance. In a model without null values with OWA, in order to characterize completeness we need a reference dataset (external). In the model with null values with CWA, specific definitions for completeness can be provided by considering level of granularity of the model elements, i.e., tuple, attribute, and table. For the granularity of elements in the database the possible explanation will be as follows:

- Tuple completeness, to characterize the completeness of a tuple with respect to the values of all attribute;
- Attribute completeness, to measure the number of null values of a specific attribute in a table;
- Table completeness, to capture the presence of null values in a whole table or the sum of all null values attributes in a table.

Therefore according to [6] completeness can be evaluated at different level of granularity at table, attribute, or levels.
Data quality measure: Attribute completeness can be measured by counting the null values an attribute divided by total number of tuples in the specific column. Table completeness is calculated by the sum of the attribute completeness divided by total number of columns in the given table. Hence, completeness can be applied to each attributes and to entire dataset.
Storage: The results of completeness tests can not be stored in the table it represents rather in a separate table.
Report: The output for completeness can be expressed as percentage, or ratio. It is indicated in the quality evaluation report. If the output is larger than the acceptable conformance quality level, then the dataset fails. if the output is less than acceptable conformance level the dataset is pass.
Example: If the acceptable conformance quality level 10% and after the calculated table completeness if the output is 5% then the dataset is pass or it is useful for the intended use. If the output is 15% the dataset is fail or it is not useful for their purpose according to AQL and if the dataset 90% the dataset is pass or it is useful for their purpose.
According to assumption of OWA, we need reference dataset to evaluate completeness. In this case we can classify the completeness into two subelements commission and omission. It is measured by compare count of items in dataset against count of items in universe of discourse.
Example: There are 100 schools on the static book (documents for universe of discourse), and there are 105 schools existing on the spatial data, then the quality subelement is commission 5%. There are 100 parks on the static book, and there are 97 parks exist on the spatial data, and the quality subelement is omission 3%.

2.6.7 Thematic accuracy

Attribute accuracy represents the closeness of the attribute in the dataset in relation to what is considered their true real world values. There are two separate types of attribute accuracy depending on the measure scales of the attribute, which are qualitative and quantitative. The qualitative attributes are usually categorical data used to classify entities in the real world. This type of attribute accuracy refers to uncertainty associated with categorisation or classification. The second type of attribute accuracy refers to uncertainty associated with discrepancies between the assigned value in a dataset and the true value in the real world. It is the accuracy of all attributes other than the positional and temporal attributes of spatial dataset. According to [17] thematic accuracy can be classified into three quality subelements called classification correctness, non quantitative attribute correctness and quantitative attribute accuracy. It can be measured on four measurement scales ratio, interval, ordinal and nominal.

Classification correctness

Classification correctness is classification in correctness occurs when the wrong class is assigned to points or features, or when assigned classes are based on different data capture techniques (or from different observers) is inconsistent.
Data quality measure: Many techniques have been developed to assess image classification accuracy [47]. These techniques involve image sampling locations and comparisons of the class assigned to each location in the reference image. The results are then tabulated in the form of an error (or misclassification) matrix. It can be calculated by the following equation.

$$PCC = \frac{1}{C}\sum_{i=1}^{n}(C_{ij}) \tag{2.3}$$

Where PCC is Proportion Correctly Classified, C_{ij} Sum of the diagonal elements, C total number of sampled pixels for accuracy assessment.
Storage: The output for the classification correctness can be percentage or ratio and can be stored as an attribute with the existing table.
Report: The report for the classification correctness is table confusion matrix and can be indicated in the quality evaluation report.
Example: In land use classification if the conformance level for item correctly classified as water is 80%, if the classification result for water is 60%, then the dataset fails. Percentage of the conformance classification exceeds the classification result.

Qualitative attribute accuracy

Qualitative attribute accuracy is an attribute that is not represented in the form of ordinal data. It may include geographic names, addresses, land use types, and others. Accuracy of the non-quantitative attribute can be assessed by examining weather all (or sample) items within the data set have attribute values differing from those in the universe of discourse.
Data quality measure: The measure of the qualitative accuracy is the number of items with incorrect attribute values, or the percentage of these items in the dataset.
Storage: The qualitative accuracy can be stored in a different quality table and link to the dataset. The data type of the output may as percentage or ratio.
Report: The qualitative accuracy can reported in ISO 19115 metadata.
Example: 100 items with geographic name in the dataset; 5 names are misspelled.

Quantitative accuracy

Quantitative accuracy refers to the level of bias in estimating the values assigned such as estimated values of pH in a soil map. It is described with the

same measures as those used for positional attributes, for example root mean square error (RMSE) [46].
Data quality measure: The quantitative accuracy can be described statistically, such as by using RMSE measure. RMSE can be computed from differences between an attribute value in the dataset and that in the universe of discourse.
Storage: It can be stored as an attribute with the same table it represent's or in a different quality table and link to the dataset. The data type of the output may be indicated as percentage or ratio.
Report: The quantitative accuracy can be reported either as additional attributes of each spatial object or through a quality report with other quality elements.
Example: The conformance level of all temperature data in the dataset is 5%, if measure the difference between temperature reading in the dataset and that in the universe of discourse is 10%, then the dataset fails. Percentage of non conformance temperature data exceeds the conformance quality level.

2.6.8 Temporal Accuracy

Temporal accuracy is "Accuracy of the temporal attributes and temporal relationships of features" [17]. It comprises three subelements, including accuracy of a time measurement, temporal consistency, and time validity.

Accuracy of time measurement

Accuracy of a time measurement deals with the correctness of the reported time measurement. Data within the dataset, as specified by a specific scope, may contain temporal information, such as data occurrence time.
Data quality measure: To assess temporal information accuracy, for example, the difference between each data occurrence time in the dataset and that in the universe of discourse is measured. The RMSE of this time measurement is then computed from the occurrence time differences.
Storage: The accuracy of time measurement can be stored as an attribute with the same table it represent's or in a different quality table and link to the dataset it describes. The data type of the output is percentage.
Report: The accuracy of time measurement can be reported in the metadata.
Example: The conformance level of all traffic accident data in the dataset is 10%, if measure the difference between accident occurrence time in the dataset and that in the universe of discourse is 15%, then the dataset fails. Percentage of non conformance accident data exceeds the conformance quality level.

Temporal consistency

Temporal consistency corresponds to the correctness of ordered events or sequences. It can be detected by checking each historical event to ensure that the event is correctly ordered against the rest of events. The temporal consistency in terms of the number of items, or the percentage of those items wrongly ordered, can then be reported in the metadata.
Data quality measure: The temporal accuracy can be detected by checking

each historical event to ensure that the event is correctly ordered against the rest of events. The temporal consistency in terms of the number of items or the percentage of those items wrongly ordered.
Storage: Depending on the type of phenomenon observed, the management of time related issues will be different. Some data updated at more or less regular intervals for example, aerial photography, others require historical management for example, cadastral maps, and finally some data are between the two types, such as fixed phenomena whose attribute change over time temperature sensor. Therefore, in some cases the temporal accuracy is treated as an attribute separate from the objects and sometimes modelled as a date, an interval, or a temporal range [42].
Report: The temporal measurement can be reported in the metadata. If an acceptable level of the number of items having temporal inconsistency is given, by comparing the number of items ordered wrongly with the acceptable level, a Boolean metadata result is reported.
Example: The acquired building data whose temporal attributes are "start date" and "completion date" contains no data whose completion date is older than start date.

Temporal validity

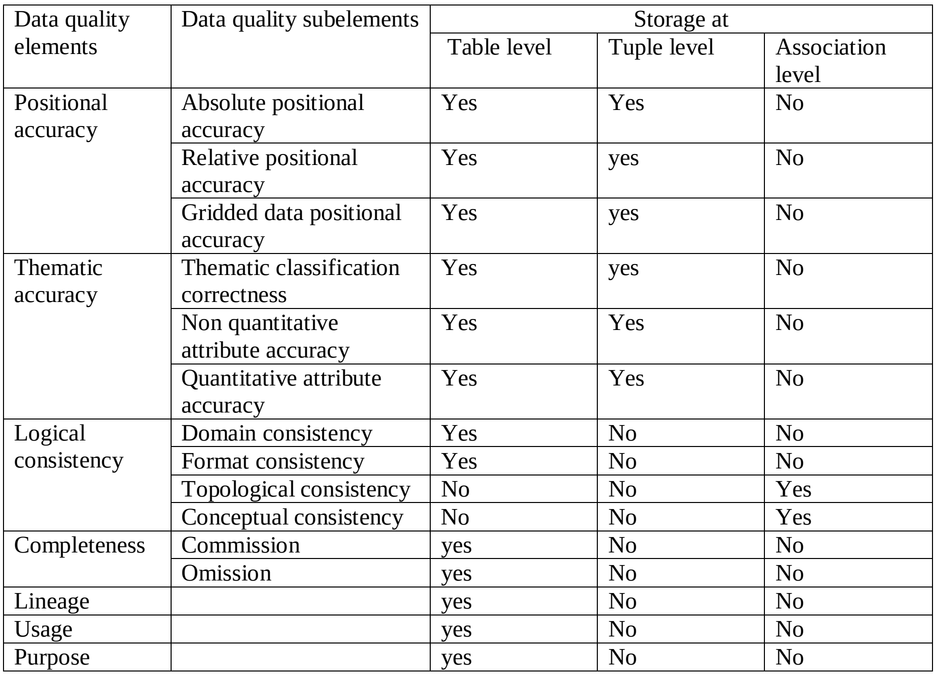
Temporal validity is data validity of with respect to time. It can be determined by checking each data item with the dataset, as specified in the dataset scope, to assure that it was captured on the date as specified in the lineage. The numbers of items failing the check are either then counted and reported in metadata, or the result is divided by the total number of items in the dataset and then multiplied by 100 and presented in metadata in percentage form. In addition, if an acceptable level of the number of items with temporal invalidity is given, a Boolean result will be obtained and reported in metadata.
Data quality measure: Assume that the land price data was surveyed in 2000. Count those that were not surveyed in 2000. Divide the result by the total number of items in data quality scope and multiply by 100.
Storage: The results of this quality sub element is can not be stored in the table it describes rather in a separate table.
Report: It can be reported in the metadata. If an acceptable level of the number of items having temporal invalid is given, by comparing the number of temporal invalid items with the acceptable level, a Boolean metadata result is reported.
Example: 100 items with the collection date of 2000 in the dataset; 95 were actually collected in 2000; 5 were actually collected in 1995. Therefore, in the dataset there are 5 temporal invalid items.

Generally, temporal accuracy is sometimes not defined explicitly for any data quality component, it is usually considered inherent within a data quality element. In this research, temporal accuracy is not a separate quality element and is reported under lineage.

2.7 Summary

The most important motivation for describing spatial data quality is to identify the characteristics of spatial data quality, storage, report, and the possible data quality measure of each quality elements and subelements. Understanding the storage of each quality elements helps in considering storage of quality elements at the multi-level structure: at attribute level, tuple level, table level, or database level. The report of quality elements helps in the understanding of how the data quality is accessed by users. The data quality measure of each quality elements helps in integrating the spatial data with the quality information in spatial databases. All these are important to provide the potential users of a dataset from spatial databases with the necessary quality information to decide on the fitness for use of a dataset for their particular application. In addition, in this chapter we have discussed the definitions of each spatial data quality elements. However, the contents of the definitions of the spatial data quality are almost the same. So for this thesis we have considered the spatial data quality definitions of [17]. Besides, the summary of spatial data quality with spatial quality subelements that can be stored as table, tuple, or association levels are shown in table 2.1.

Table 2.1: Summary of quality elements storage



Data quality elements	Data quality subelements	Storage at		
		Table level	Tuple level	Association level
Positional accuracy	Absolute positional accuracy	Yes	Yes	No
	Relative positional accuracy	Yes	yes	No
	Gridded data positional accuracy	Yes	yes	No
Thematic accuracy	Thematic classification correctness	Yes	yes	No
	Non quantitative attribute accuracy	Yes	Yes	No
	Quantitative attribute accuracy	Yes	Yes	No
Logical consistency	Domain consistency	Yes	No	No
	Format consistency	Yes	No	No
	Topological consistency	No	No	Yes
	Conceptual consistency	No	No	Yes
Completeness	Commission	yes	No	No
	Omission	yes	No	No
Lineage		yes	No	No
Usage		yes	No	No
Purpose		yes	No	No

Chapter 3

Designing data quality model for a spatial database

3.1 Introduction

In chapter 2 we have discussed some of the limitations relating to the current spatial data quality models. These limitations restrict quality information from being easily accessible, understandable, or adopted to the usage context and users' needs. It has been found that metadata usually remain unused in practice by novice GIS users as well as many expert users [49]. In order to overcome these problems, there is a need to design a spatial database with spatial data quality. To have an effective communication of spatial quality requires well structured data quality information. Such a model relates to how the quality statements are stored and integrated with the spatial data, this enables accessing or processing spatial data quality is possible.

The main objective of this chapter is to discuss database design concepts and to find a mechanism incorporating spatial data quality with spatial data during the design of spatial databases. The proposed model is able to solve some of the problems occurred in the current spatial data quality models. In section 3.2 the database design concepts are discussed. In section 3.3 specifies the database model for the project. In section 3.4 discussed the current spatial data quality models and followed by the drawback of current spatial data quality model in section 3.5. Then design requirements and modelling principles are discussed in sections 3.6 and 3.7 respectively. The design of an effective data quality model requires meeting a series of requirements. These basic requirements can serve as the guidelines for developing the modelling guidelines of spatial data quality. Finally, we conclude the chapter with summary of discussion in section 3.8.

3.2 Database design concepts

Database design is an engineering process to develop from a complex, non formalized, dynamic, real world situation that often includes a number of user wishes, to a formal representation implementable on a software platform of choice [41]. Database design is an evolutionary process. It starts with a conceptual database model that represents the real world at a high level of abstraction. This means that the description of the database is entirely conceptual and completely independent of any hardware and software considerations. This phase of data modelling is called conceptual database design. In order to turn such a high level of abstraction into database implementation specifications. It is necessary to take into account the specific software and hardware requirements of a particular DBMS during the subsequent phases of data modelling. A database model that includes software considerations in its description of a data structure is called a logical model. Such a model is developed by translating a conceptual model according to the linguistic syntax and diagrammatic notation of a selected DBMS. A logical model, therefore, is said to be DBMS dependent. The process of translating a logical model to a physical model is commonly referred to as physical database design. Physical modelling is a more complex and technical process than logical modelling because it requires competency in using both the DBMS and the hardware system used to install the database. Since this process is both hardware and DBMS dependent [4].

As data modelling is part of database design, and it is related to the analysis of data objects and their relationships to other data objects. Besides, data modelling implies a transformation process between different abstraction levels. The terms conceptual, logical, and physical are frequently used in data modelling to differentiate levels of abstraction versus detail in the database design [39].

3.2.1 Conceptual design

A data model is conceptual if it enables a direct mapping between the perceived real world and its representation with the requirements of the database design. The representation is handled by a careful analysis of application user requirements [7]. Once the requirements are established, the design of the conceptual model consists of a rewriting of the natural language specifications into the formal description language associated with the data model. A conceptual data model is free from implementation considerations [7]. According to [33], a conceptual design includes:

- The important entities and the relationship types among them
- Attribute specifications
- No primary key specifications

At this abstraction level, the design attempts to point out the highest level relationships among the different entities, A conceptual model may include a few significant attributes to enhance the definition of entities. Besides, it may have some candidate keys but they do not represent entity identifiers, since identifiers are logical choices made with deeper knowledge of the context [7]. Relationships between entities can be generic, such as composition, aggregation, or inheritance. Other relationships apply to particular situations such as “A road connects cities”. Entities and relationships are usually depicted graphically within diagrams such as Entity-Relationship (ER), Unified Modeling Language (UML), or Object Modeling Techniques (OMT) diagrams [33]. Once such a diagram has been designed, one obtains a conceptual design.

3.2.2 Logical design

While the goal of conceptual design is defining a model that reflects user requirements about the database design, the goal of the logical design is to define a corresponding model that conforms to data definition rules of some specific DBMS [7].

- Includes of all entity and relationship types among them
- All attributes for each entity type are specified
- The primary key for each entity type is specified
- Foreign keys are specified

At this abstraction level, the model attempts to describe the data in more detail, without regard of how they will be physically implemented.

3.2.3 Physical design

A physical model is a single logical model instantiated in a specific database management product (e.g. PostgreSQL/PostGIS, Oracle). The physical model specifies implementation details which may be features of a DBMS, as well as configuration choices for that database instance [7]. Feature of the physical model include [33]:

- Specification of tables and columns, indexes, partitioning, and clustering files
- Foreign keys are used to identify relationship types between tables
- Performance considerations or constraints may cause the physical model to be quite different from the logical model
- At this abstraction level, the model shows how the logical model is implemented in a specific DBMS.

3.3 Database models

There are various data models widely used in system design, such as relational data model; however, there are some obvious deficiencies. The relational model is more suitable for dealing with weakly structured data, such as banking accounts and personal records. It fails when it is used for application of data with a complex structure. The object relational databases were developed to overcome the limitations of relational systems to handle complex data required by new database applications scientific simulation and visualization among others [4]. Vendors of database software extended the capabilities of traditional relational systems by introducing many of the concepts of object oriented systems like object storage, user defined data types, inheritance, and encapsulation of methods with data structures. A database that is constructed using a relational database model with object oriented extensions is said to be an object relational database. The model on which the database built is called an object relational database model.

The object relational approach, sometimes called the extended relational model, is a compromise between the concepts of the object oriented and relational models. The object relational model has been adopted for implementation in this research because it utilizes the power and semantics of object orientation and the full functionalities of the underlying relational database system [38]. Nowadays, many database management systems and GIS systems are beginning to incorporate object-oriented features into the existing relational database functionality. For example, ESRI's recent model called the geodatabase fully combines object-oriented concepts with the relational model.

The inheritance relationship of object oriented allows each class of spatial objects in the database to inherit from the parent class. The data quality model enables spatial objects to infer quality information from the table level down to tuple level. The object relational model has the advantages of storage and maintenance of data quality information. Rather than storing data quality for every object, individual spatial objects can associate with quality information where appropriate from their aggregate objects [22]. For example, if all tuples in a table have the same lineage quality information, the lineage information only needs to be stored once at the table level.

Unified modeling language UML

There have been various tools used in designing OO systems over the years. One of commonly used is called UML. The conceptual and logical models have been designed in UML. UML is a language for specifying, visualizing, constructing, and documenting the artifacts of software systems, as well as for business modeling and other non software systems [13]. The UML comprises a number of graphical elements that are combined to form models that provide multiple views of a system. It is important to know that UML models describe what a system is supposed to do and does not show how to implement the system. In UML, different types of diagrams exist, such as use case, class, activity, component, and state chart diagrams.
Classes: As defined in UML reference manual [23], class is the descriptor for a set of objects that share the same attributes, operations, relationships, and behavior. Every objects is an instance of a class. Generally class can be mentioned as follows:

- Class is a template for group of objects
- All objects of the same class must have the same set of operations
- The same set of attributes with different attribute values

- The same set of relationships

Classes are represented in the UML by a solid outline rectangle with three compartments separated by horizontal lines. The top compartment holds the class name and other general properties of the class (e.g. stereotypes). The middle compartment holds a list of attributes. The bottom compartment holds a list of operations. Abstract classes have their names in italics. In cases the class name is formed from two or more words, we capitalize the first letter of each word irrespective of its position. Optional listing of attribute data types and initial values my also be provided, as shown in Figure 3.1. Operations and methods may also be listed (optionally) with their return types and parameters.
Association: Association is represented in the UML as a line connecting two classes with the association name just above the line. This kind of association is called binary association in the UML. Association role is shown at both ends of the line next to the class. Association cardinality or multiplicity is shown just above the association line near the appropriate class. An association may have attributes and operations just like a class. In such case, it is called an association class. This is represented in the UML like an ordinary class with a dotted line connecting it to the association line Figure 3.1.
Generalization: Generalization is a relationship between a superclass and its subclasses. In the UML, generalization is represented by a line that connects the subclass to the superclass, with an open triangle on the end of the line that point to the parent class, as shown in Figure 3.1.
Composition: Composition is a relationship between a composite object and its basic objects. Composition may be shown by a solid filled diamond as an association role adornment. The multiplicity of the composite end may not exceed one (i.e., it is unshared).

3.4 Current spatial data quality models

Managing spatial data quality deals with the updating, storing, querying, and deleting of spatial data quality in spatial database. Researchers applied different approaches of spatial data quality models to manage spatial data quality in spatial databases. These are per-feature, feature-independent, and feature-hybrid approaches. The per-feature approach, in which each geographic feature stored in the database, can be associated with quality information related to that feature. A research that adopted the per-feature model is a model developed by [22]. For storing data quality at different levels within an object-oriented hierarchy of features this model adopted from the concepts of the per-feature quality. The feature-independent approach deals with the storage of spatial data quality in a separate table in database. The feature-hybrid approach is the combination of per-feature and feature-independent approaches. It deals storing sub-feature variation in data quality, while explicitly associating that quality information with the particular feature(s) which it refers. For example, measurement-based GIS, by Goodchild [30] stores quality information at tuple level. Similarly to Qiu and Hunter [22], Duckham [27] takes advantage of the hierarchical structure of object-oriented database to store data qual-

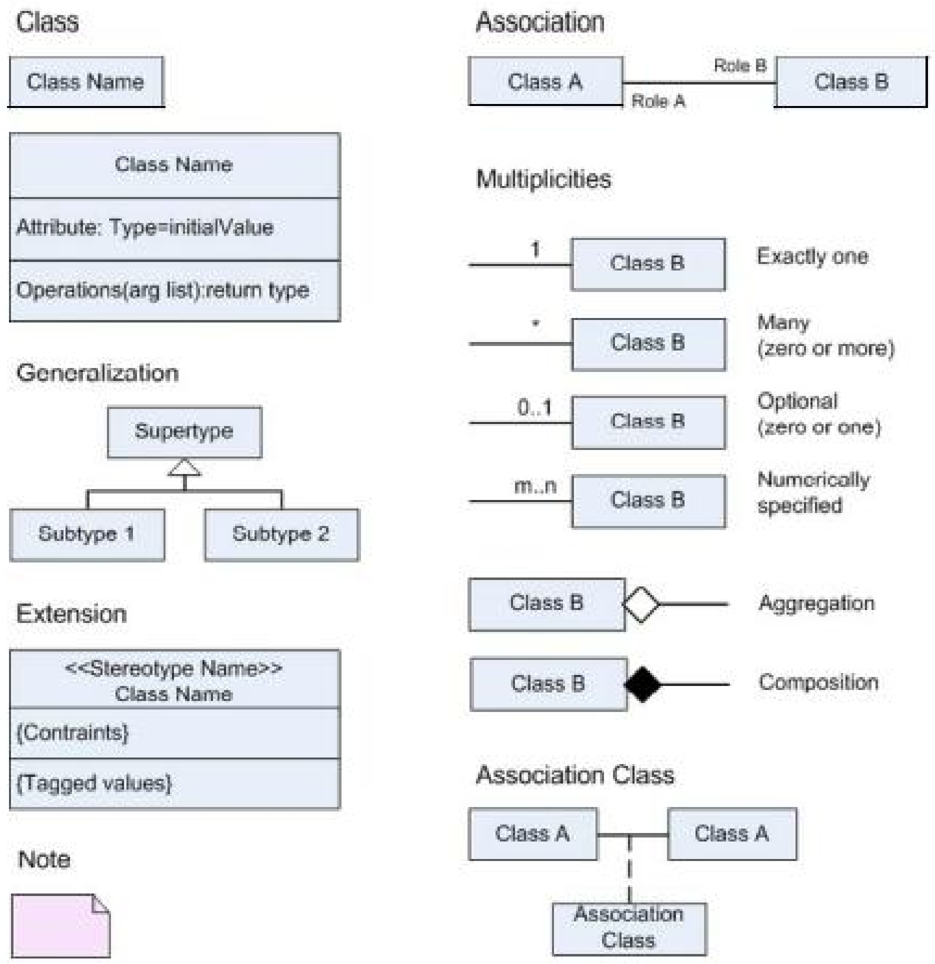


Figure 3.1: UML graphical notations (23)

ity information at multiple levels. Unlike Qiu and Hunter [22], Duckham [27] proposed storing data quality down to sub-feature levels, potentially associating quality information with a feature's component line segments or vertices. Feature hybrid models of data quality do allow sub-feature variation to be represented, at the same time as being linked to individual geographic features in the database.

3.5 Drawbacks of current spatial data quality models

As discussed in chapter 2, today's quality information based on the existing models is typically provided in a text file separated from its data. It is often restricted to present quality information at the dataset level rather than at the detail level. The integration of quality information with the actual data in a spatial database would benefit users who can then directly interact with the quality issues. A number of studies has been carried out and recommended approaches to tackle the problem of spatial data quality. For instance, Duckham and Drummond [28] developed an object oriented data quality model implemented in an object oriented GIS. This model has no capacity for analyzing the mechanism for navigating data quality at feature dataset, feature layer, and feature class. Duckham [27] developed a model of object based variation in spatial data quality, which uses object calculus. It offers the potential to store sub feature variation, however the model is not efficient in storage and query. Hunter and Qiu [22] developed a data quality model using a multilevel structure. The problem of this model does not adequately represent variation in quality within objects and the linkage of quality evaluation with data quality model has not been considered. Devillers et al. [37] proposed a conceptual data model named the Quality Information Management Model (QIMM) using a multidimensional database approach. With this model, data users allow the access of quality information at different levels of detail. This model lacks the capability of linking data quality with data quality evaluation. Ghouse and Duckham [31] have implemented per feature, feature independent, and feature hybrid models. The per-feature is efficient in storage and query, but it does not consider the sub feature variation. Feature independent is simple and efficient to store. It is computationally expensive to query. Feature hybrid is efficient to query. It is more complex with associated database redundancy problems. None of these models are used separately unless the integrated of the three models.

The current standards such as Federal Geographic Data Committee (FGDC) define the data quality content predominantly from a data producer's view. The ignorance of user requirements in implementing spatial data quality standards leads to the user's incapability to judge the fitness-for-use of a specific dataset [50]. Another issue associated with the current models is that there is lack of the flexibility for defining new quality elements. It is often required to define new quality elements since there are various data types for different applications. The major problem associated with current data quality models is that data quality is treated as fundamentally static by current standards. When spatial analysis are performed, there are no corresponding updating mechanisms to automatically produce quality information for derived secondary data sets [50]. Considering the problems of current spatial data quality models mentioned above, the main purpose of this research is to develop a new spatial data quality model, which could provide solutions to some of the problems identified above.

3.6 Design requirement for spatial data quality

A spatial data quality design relates to how spatial data quality components are stored, represented, or linked with the spatial data they describe in a database. It provides the means for the database designer to consolidate user requirements and compare alternatives what the spatial database will be like when it is completed. According to [25] there are five basic requirements to represent data quality information. These are integration, multidimensionality, dynamic state, accessibility, and flexibility.

Integration requires that the quality descriptions should not be separated from the data but integrally linked with them [25]. This means the data model should accommodate measures of data quality and be able to link data quality measures to the associated data fields. This is important in maintaining the consistency as updates are made to the data. Multidimensionality relates the complexity of quality descriptions. This should include information to assess positional, thematic, and temporal accuracy of the data, such as the data quality components described by current metadata standards. Dynamic characteristic describes data changes through various processing, requiring that the quality information should be updated. Accordingly, data quality model should be able to update quality information in conjunction with data updating and processing [25]. Accessibility requires that tools for extracting information from the data should be readily available to users. Data quality reports are typically provided in a different file, and separated from the datasets they describe. It can be time consuming to read through and assess separate quality reports, and thus end users frequently ignore metadata [37]. To facilitate the use of quality information, data quality information should be stored in a spatial database and readily available to users. Also, there is a need to provide data users with a user friendly interface to access and understand quality information easily. Another requirement is flexibility, which requires that the model should be flexible. This can be expressed in different forms, such as text format, reliability diagrams [45]. Data users may not have equal interest in data quality formats. Some users may want very detailed descriptions with statistic reports, while the others will be satisfied with brief summary only. The final requirement is multi-level structure, which users may want to examine data quality at different granularity levels. Some may need a brief report of quality information at a dataset level, while the others may want very detailed quality information for a particular feature dataset level. The multi-level structure for representing quality has the advantage of accessing, reporting and facilitating storage of data quality information [36]. These above requirements provide the important guidelines for the design of the conceptual model of spatial data quality in managing data quality information.

3.7 Modelling guidelines of spatial data quality

From the design requirements mentioned in section 3.6, the integration of spatial data quality with spatial database is the basis for the other requirements. To integrate the quality information, we have to consider the different structure of spatial database. This is because the different DQE can be integrated at attribute, tuple, table, or database levels.

3.7.1 Integrating the data quality element as table

The DQEs lineage, usage, and purpose are the static quality elements that can be stored at table level. These quality elements are stored in a separate table and linked to the table they describe by association.

3.7.2 Integrating DQE as functions

The quality elements completeness, format consistency, domain consistency, and conceptual consistency are not represented as data to associate with attributes, but rather as a process of assessment. These quality elements are integrated as functions with the data they describe. The output from these functions is a single attribute can be stored in a separate table. Due to the dynamic behavior of these quality elements, They are stored in a temporary table.

3.7.3 Integrating DQE as tuple

The quality elements positional accuracy and quantitative attribute accuracy can be integrated with each tuple stored in the table they describe or in a separate table. According to the per-feature and feature independent concepts discussed in section 3.6, these quality elements can be stored as new attribute with the table they describe or in a separate table associated with the table they describe. In our thesis project we stored them in a separate table temporarily and with one to one association with the table they describe.

3.7.4 Integrating DQE as constraints

The quality element topological consistency can be integrated as constraints. The constraints may be at table level (for example, in a land use table, land use polygon geometry should not overlap to each other) or database level (for example, from facility table, facility point geometry should be within district polygon geometry). Therefore, integrating spatial data quality at multilevel structure has the advantage of facilitating storage of data quality information. Objects at lower levels inherit the attributes of their classes. This inheritance permits more efficient in storage and maintenance of data quality information. This is because much of the data quality information for objects at a given level is identical. For example, the lineage details for each tuple can be stored at table level. Accessing and reporting of data quality information also benefits from the multilevel structure. In general, a spatial data quality model at multilevel structure provides the basis for designing a database to store data quality information. The differences in storage, evaluation method, and report of each quality element will lead us to design differently.

3.8 Summary

In this chapter, the main issues concern the modelling guidelines of spatial data quality are addressed. Firstly, this chapter identifies the different abstraction levels of data modelling during database design. It introduced approaches to design spatial data quality with the basic concepts of object relational modelling.

The current spatial data quality models are discussed and followed by the discussion on the drawback of the existing spatial data quality models. The basic requirements for representing data quality are then addressed. The requirements can serve as the guidelines for developing the conceptual data quality model. Finally, we have discussed the modelling guidelines of the spatial data quality. In summary, the data quality elements can be stored as attribute, tuple, or table levels.

Chapter 4

Use case: Evaluating and storing spatial data quality in spatial database

4.1 Introduction

This chapter covers the discussions on use case that will help us for the purpose of our project implementation. The modelling principles which were discussed in chapter 3 are the basis for the conceptual design of the datasets described in the use case. In section 4.2 use case descriptions are discussed. According to ISO 19114 the quality evaluation procedures are discussed in section 4.3. The spatial data quality requirements are presented in section 4.4. Finally, we conclude the chapter with summary of discussion in section 4.5.

4.2 Use case description

Afar National and Regional State is one of the emerging regions in Ethiopia. In the region there are a number of governmental institutions. Some of them are pastoral bureau, water bureau, health bureau, and finance and economic development bureau. Organizing and distribution of up to date regional spatial information is one of the key tasks of finance and economic development bureau. The bureau is mandated by proclamation to prepare statistical and geographic information data that helps planners and decision makers. The sources of information are from different regional sectors of the regional institutions such as water bureau, pastoral bureau, health bureau, etc. Data collected from those organizations are processed in the department of data processing under finance and economic development bureau. The region consists of 5 lower administrative area called provinces. The provinces are further divided into the smaller unit called districts. Each province has 5 or more districts. In total the region has 29 districts. The governmental organization of the provinces and districts have the same structure as the regional structure. The regional government has decided to prepare regional spatial database:

- The spatial data associated with socio economic data.
- Exchange spatial data from spatial databases of the districts to the regional spatial database.
- That evaluates and stores spatial data quality in spatial databases.
- With a number of applications that allow to analyse the current situation and trends of the region
- That allows spatial data processing through the Web processing services.
- Different types of users like human users, Web service users and users of spatial databases should retrieve spatial data associated with quality information.
- While updating of spatial data the quality of the spatial data also update accordingly. So a tool that dynamically update spatial data quality is considered during the preparation of the regional database.

4.2.1 User requirements of spatial data quality

Users' need for spatial data content and the quality of that content are embedded in the purpose for which the data to be used. Spatial data is often information source contributing to decision making. Hence, reliability of outcomes is based on the fitness for purpose of the data source itself as well as on its interoperability with other data sources. Users therefore need to consider whether a data source will provide them with the type and quality of information needed in the context of intended use. For example, users' need to check whether the relevant real world features are located to a sufficient level of accuracy or the required spatial, temporal, and thematic attributes are present and consistent across the area of interest. The different types of users of spatial data quality are human users, Web service users, and users of different spatial databases.

Human users

Nowadays, different users (general user, researchers), use spatial data for decision making. General users may need the spatial data without quality information. For example, car navigation for car driver is very useful but no quality information associated with. Other users like researchers are interested for the further analysis of the spatial data, so they need the quality of the spatial data to evaluate how accurate their analysis. Hence, user community has diverse needs and it is important to have end user queries and reporting tools. So that users can choose the spatial data with spatial data quality as needed. We consider as an example Afar National Regional State (Use case) spatial data according to the user requirements as follows.

- Users should select the dataset and the quality element to get the quality information of the dataset.

- Users should retrieve spatial data with spatial data quality information through the Web browser at click button.
- Users should get quality information at dataset (table) or at attribute level
- During spatial data updating the quality information should also update.
- Users should get quality information according ISO 19115 standard report.
- Acceptable quality element of the dataset should be set by users so that users should retrieve the report according their demand by Web browser with the result (passes or fails).

Web service users

The geospatial Web services are Web services specially designed for providing access to and processing heterogeneous spatial data stored at remote sources. The OGC aims at developing standards for accessing these geospatial Web services. For years, the main focus of OGC was on developing standards for merely accessing spatial data, as for example with the OGC, Web Map Service (WMS), or Web Feature Service (WFS), whereas by now there has been a shift towards providing Web services for the processing of spatial data. The WPS is such a standard developed by the OGC. When using spatial data or generating new data with data processing, e.g. by applying spatial analysis functionality provided by a Web service, it is important to know about the quality of the data. The quality of the data not only informs the user about the fitness for use of the data in certain fields of application, but also enables the user to interpret the results from an analyses. In this scenario we use road data and we will see how Web services generate quality information for the results of spatial analyses (buffer, union). To be able to determine the quality of spatial data the following requirements will be identified

- Road map of one province which represents one street may be collected by different experts and each road segment may have different positional accuracy.
- The quality information for the input data should be associated.
- The back end should be WFS to have both input and output in GML format which is compatible for spatial analysis using WPS
- The different functionality of WPS like buffer, union is required to perform the spatial analysis.
- A Web service interface is required for performing spatial analyses and generating quality information for the output based on the quality of the input data and the spatial analysis method(s) applied. In this case WPS can be used as Web service interface. It provides an interface specification which is appropriate for this scenario.

Users of other spatial database

In the region there is a regional spatial database. The sources of information are from spatial databases' of the lower administrative area districts. The districts have their own boundary and collect the spatial data with in this boundary only. The spatial data collected from the districts are stored in regional spatial database. The duty and responsibility of each districts is to collect facility distribution of the districts according to the boundary. From this we can understand communication of spatial data between heterogeneous spatial databases. The user requirement of this scenario will be as follows

- The districts should collect spatial data as well as non spatial data associated with it.
- The spatial data collected by each district should be the same format and scale.
- Each districts collect spatial data with in their boundary only.
- The spatial data from districts should contain the quality information lineage.
- The spatial database in each districts have the same schema which is prepared by the regional database so that the interaction will be smooth.
- The facilities collected by each districts should be with in district.
- The districts should not overlap to each other.

4.2.2 Dataset descriptions

The dataset definitions for the spatial data prepared to use throughout the thesis project is according to the use case described in section 4.2. The dataset is in a vector format and it is organized as follows:

- Facility is in the context of the region consists of school, water, health, and animal clinics. It is represented as MULTIPOINT geometry in the regional spatial data. The facilities do not overlap to each other and are not within the road network.
- Road is the basic socio economic infrastructure. It is consists of three classes dry-weather roads, asphalt, and railways. Roads are represented as LINESTRING in the regional spatial data. All the districts of the region have at least one road.
- Land use is very important in managing the natural resources. In the region it is monitored by the bureau of agriculture. There are 8 main distinct type of land use classification. These are cultivated land, wood land, grass land, bush land, water body, exposed rock surface, flat sand surface, and salt surface. It is represented as MULTIPOLYGON in the regional spatial data. The land use has reflectance values for each land

use type. The reference dataset for evaluating the land use is stored in a separate table with an attribute of reflectance values for each land use type.

- District is the smallest administrative area. It is represented as POLYGON in the regional spatial data. In the region there are 29 districts. Districts are not overlap to each other. Each district has one or more facility. Province the higher level of administrative area of the region. It is consists of 5 or more districts. The region also consists of 5 provinces.
- Object class (Qu_object) is a temporary class that stores the quality information of positional accuracy or quantitative attribute accuracy at tuple level. In general the quality information that represents for each tuple of the dataset evaluated is stored in this class.
- Attribute class (Qu_attribute) is a temporary class that stores the quality information of completeness, domain consistency, or format consistency. In general the quality information that represents for each attribute of the table evaluated is stored in this class.
- The quality information lineage, usage, and purpose are considered as static quality information. Hence we store them in a separate class called class quality (Qu_class).

4.3 Quality evaluation procedures

Quality evaluation in ISO 19114 quality evaluation procedures describes a process for evaluating and reporting data quality results. There are five procedures to evaluate the quality of a given spatial dataset. First select applicable data quality elements and sub-elements. Second identify the data quality scope (e.g. a certain area in the dataset). Third the data quality measures are defined. Fourth a quality evaluation method is then chosen and applied. Fifth a conformance level is set the evaluator determines conformance comparing the data quality result. The overall quality evaluation procedures can be illustrated as shown in figure 4.1.

4.4 Spatial data quality requirements

As spatial data quality is a relative measure, it is challenging for data users as well as database designers to specify quality requirements and to determine weather data satisfy the requirements [2]. The challenges are first, the applicable quality elements are often not expressed in the conceptual design. Introducing the applicable quality elements in conceptual design is the solution for this challenge. The second challenge is different users need different quality conformance level on the same dataset. A solution is to make it possible for users to evaluate their quality requirements against the data, by including quality information at relevant level of hierarchy. The applicable quality elements of a

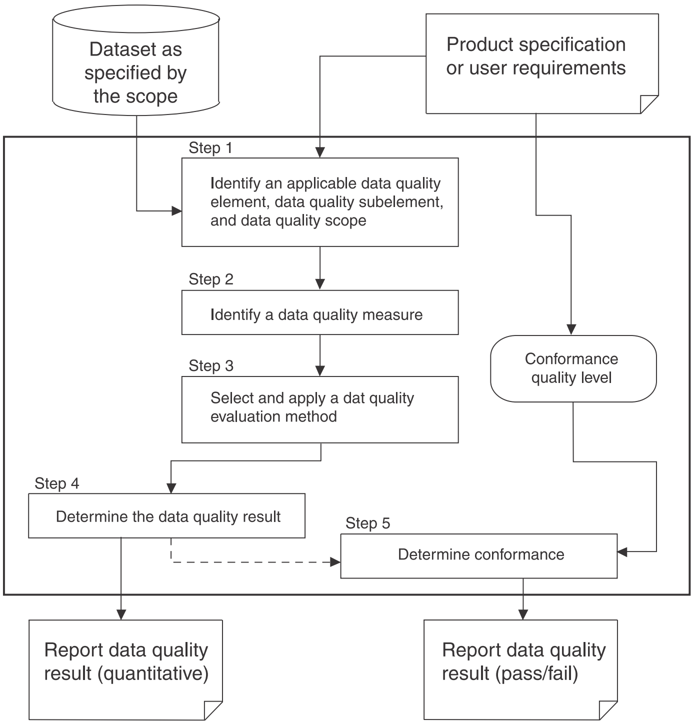

Dataset as specified by the scope
Product specification or user requirements
Step 1
Identify an applicable data quality element, data quality subelement, and data quality scope
Step 2
Identify a data quality measure
Step 3
Select and apply a dat quality evaluation method
Conformance quality level
Step 4
Determine the data quality result
Step 5
Determine conformance
Report data quality result (quantitative)
Report data quality result (pass/fail)

Figure 4.1: Five steps for quality evaluation adopted from (19)

given dataset are prepared by the data producer and the quality conformance levels are set by users according to their application. Hence, the applicable quality elements are considered in the conceptual design.

The quality elements may be varied according to different levels of database structure. It should be possible to specify requirements related to all levels. As an example, a single point in a facility dataset has positional accuracy, which is different from the aggregated positional accuracy of facility dataset in a given table. Both levels may be important to an application. Therefore, when establishing the conformance quality levels for the data product specification, it should be taken into consideration that:

- Different quality evaluation methods may be applied to the given datasets. For example, as we have discussed in chapter 2 the quality evaluation method for positional accuracy at table level and tuple level is different.

- Conformance quality levels can be different for different features in the dataset. For example, the required positional accuracy for features with fuzzy boundaries is usually much lower than for linear and well defined features.

4.4.1 Applicable quality elements

Each dataset has its own quality information. For example, for dataset that consists of only records of students' name and ID, the positional accuracy is nothing to do with this dataset. In general the applicable quality elements for one dataset may not be applicable for the other dataset. Therefore, it is important to identify each dataset's applicable quality element separately. The other issue that we can discuss is that the overview quality elements in each dataset are applicable although the content varies. These quality elements evaluated differently than the other quality elements. They are evaluated by indirect evaluation method based on external knowledge. The overview quality elements are the basis for the other quality elements in the dataset. This is true especially when the overview quality element lineage records quality information like positional accuracy at dataset level. So for the given dataset in the use case the overview quality elements are indicated in Appendix B.2. The applicable quality elements of each dataset and quality element requirements (conformance levels) are as shown in table 4.1.

Table 4.1: Applicable quality elements and quality element requirements

Dataset	Applicable quality elements/subelements	Quality element requirement
Road	Positional accuracy	The sum of line features with positional accuracy less than 10m divide by all features in the same column multiplied by 100 is 80%
	Completeness	80%
	Format consistency	100%
	Domain consistency	100%
District	Completeness	90%
	Format consistency	100%
	Domain consistency	100%
Facility	Positional accuracy	The sum of point features with positional accuracy less than 10m divide by all features in the same column multiplied by 100 is 80%
	Completeness	75%
	Format consistency	90%
	Domain consistency	100%
	Topological consistency	100%
Land use	Completeness	85%
	Attribute accuracy	80%

4.4.2 Data quality evaluation method

Data quality evaluation method is one of the steps of data quality evaluation procedures. The data quality evaluation methods are divided into two main classes, direct and indirect. Direct methods determine data quality through the comparison of the data with internal and/or external reference dataset. Indirect methods infer the data quality using information on the external knowledge or on the datasets' lineage. The direct evaluation methods are further classified by the source of information needed to perform the evaluation. This classification can be shown as in figure 4.2. For the purpose of our project we use the direct evaluation method.

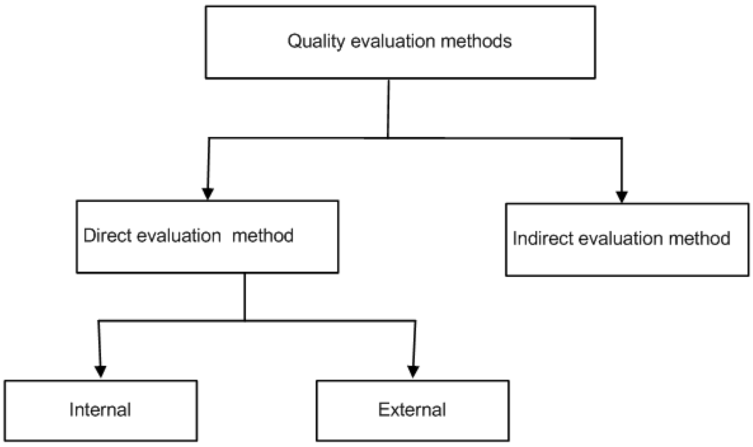


Figure 4.2: Data quality evaluation methods adopted from (19)

4.5 Conceptual model for the use case

For the datasets mentioned in the use case, we have identified the applicable quality elements as indicated in table 4.1. These are the main datasets and quality information that can help us for our modelling the use case. The datasets discussed in section 4.2.2 are represented as classes in our conceptual model. As we have discussed in chapter 3 the quality elements can be stored at different levels at attribute, tuple, table, or database level and this can be reflected in the conceptual model. The quality elements positional accuracy, attribute accuracy, and temporal accuracy are achieved from two datasets, one the reference dataset and the other the dataset going to be evaluated. In order to drive the quality information from a given dataset we need to apply each quality element's data quality measure as discussed in chapter 2. The data quality measure can help us to drive the quality information from reference dataset and the dataset to be evaluated. It is indicated as an operation compartment of the UML class diagram and the output can be stored at tuple level in a separate table called Qu_object. The overview quality elements lineage, usage, and purpose are considered as static quality elements. They are stored at table level in a separate class called Qu_class. The quality elements completeness, format consistency, domain consistency, and conceptual consistency are integrated as functions with the data they describe. The output from these functions is a single attribute can be stored in a separate table called Qu_attribute. The quality information in classes Qu_attribute and Qu_object are store temporarily. Due to the dynamic behavior of these quality elements, we stored them in a temporary table. The overall conceptual model of the spatial data discussed in the use case description is in figure 4.3.

4.6 Summary

In this chapter, the main issue is concerning the design of a conceptual model spatial data quality with spatial data. Firstly, this chapter discussed use case to identify the typical use of spatial data quality for different types of users like human users, Web service users, and users of other spatial databases. Dataset specifications are mentioned that can help us for the preparation of conceptual model of spatial data integrated with spatial data quality. The quality evaluation procedures are discussed. For the dataset mentioned in the use case, the applicable quality elements for each dataset are identified. Then, we developed the conceptual model of spatial data quality with spatial data. The conceptual model as indicated in figure 4.3, we have shown the possible storage, of each quality elements and relationship of each quality elements with the spatial data. The proposed spatial data quality conceptual model treats data quality elements as an object associated with spatial dataset, making it possible to integrate spatial data with its quality description and can be easily updated during spatial data processing. Accordingly, this proposed model may have a potential capability of dealing with dynamic updating of data quality as data changes through various processing. This meets the integration requirement that quality information should be integrally linked with the dataset it describes. The conceptual model is capable of dealing with various data quality elements, such as lineage and positional accuracy. Finally, we have discussed the data quality evaluation procedures that can follow during evaluation of a given dataset. In summary, we conclude that the quality element important for one dataset may not be for the other. Spatial data quality can be integrated with spatial data quality at different hierarchy.

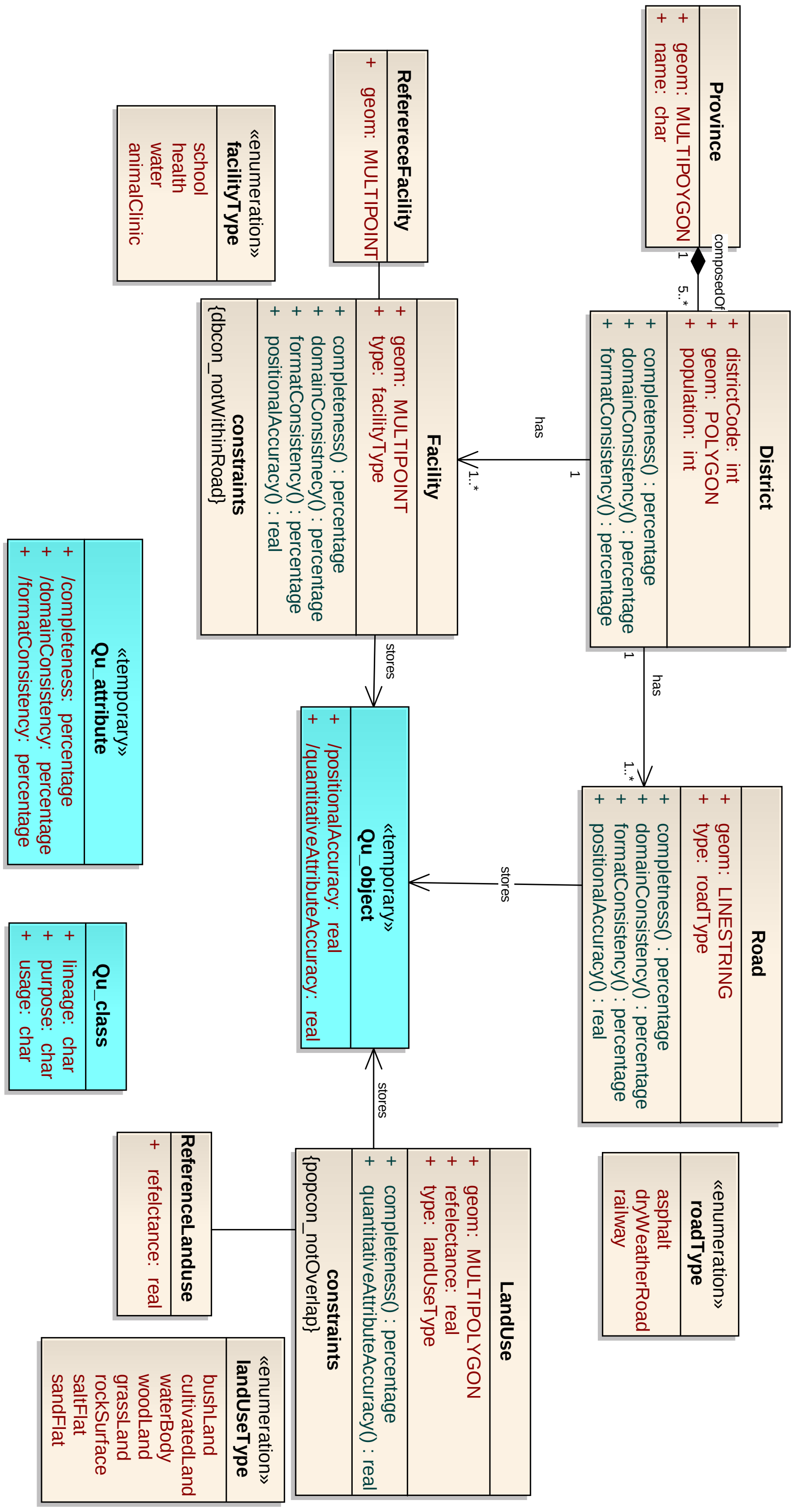


Figure 4.3: conceptual model for the use case

Chapter 5

Implementation of Spatial database integrated with spatial data quality model

5.1 Introduction

This chapter covers the implementation phase. The conceptual model as indicated in Chapter 4 is the basis for the implementation. Implementation tools are discussed in Section 5.2. The conceptual model will be transformed to logical and physical model as discussed in Sections 5.3 and 5.4. After transformation takes place, there could be some functions to be created in order to make the transformation complete then followed by its discussion in Section 5.5. Finally, a Web interface prototype was developed that enables users to retrieve spatial data, spatial data quality being its integral component. This was discussed in Section 5.6. The summary of the chapter will be handled in Section 5.7.

5.2 Implementation tools

The implementation tools applied for the purpose developing the prototype are explained as follows:

1. PostgreSQL/PostGIS is our target DBMS for the implementation of a design database. It has been selected because of the following reasons
   - As we have discussed in Chapter 3 one of the approach to model the spatial data quality model is using the concepts of object relational model.
   - Standards compliant: PostgreSQL/PostGIS follows the OpenGIS simple features specification for SQL [9], the ISO spatial schema specification [44].
   - Extensibility: it can be extended by adding data types, functions, operators and procedural languages.

- High performance: there are many behaviors PostgreSQL/PostGIS offer for increasing performance (such as AUTOVACUUM), advanced indexing options, and its SQL query performance.
- It has full support for foreign keys, joins, views, triggers, and stored procedures.
- Multiple procedural languages: server side code can be written in many languages (example PL/pgSQL,PL/Python,PL/Java).

2. pgAdmin is a free and open source graphical front-end administration tool for PostgreSQL, which is supported on most popular computer platforms. We use this tool to interact with PostgreSQL/PostGIS throughout the implementation and testing phase.

3. OpenLayer is used for spatial data visualization that is needed during the implementation and testing phases.

4. PHP is used to develop the middleware between server-side and client-side interaction.

In general the system architecture for our prototype user interface is as shown in Figure 5.1. As indicated in this figure, the Web server is to run the PHP and WMS, the database server to run the PostgreSQL/PostGIS, the browser is for the openlayers and browser, and the arrows indicated us the internet connection between the services.

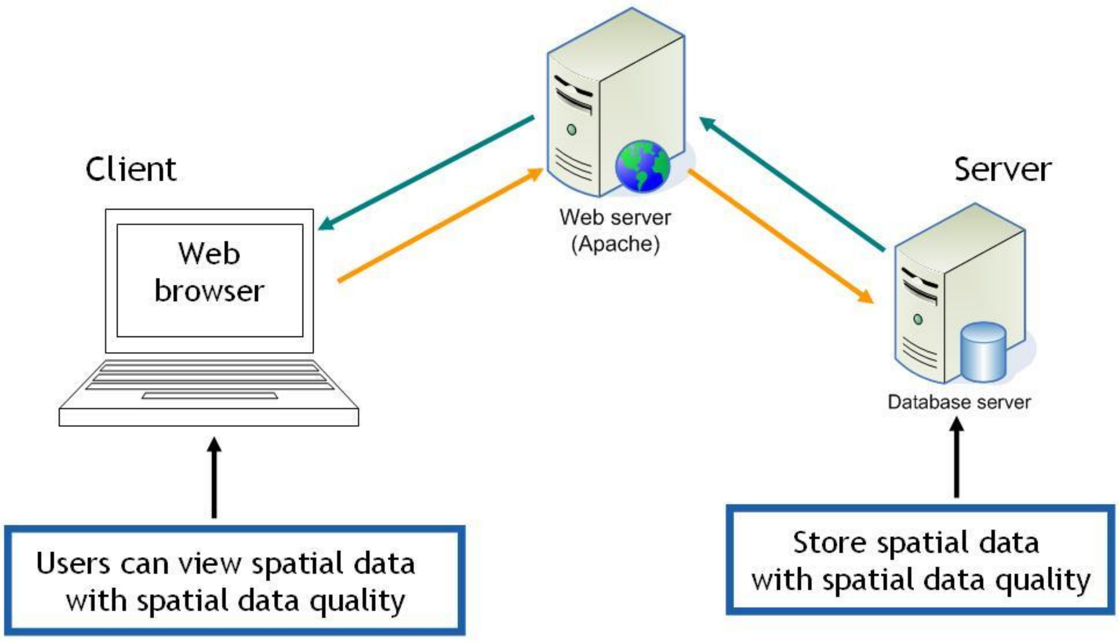

Client
Web browser
Web server (Apache)
Server
Database server
Users can view spatial data with spatial data quality
Store spatial data with spatial data quality

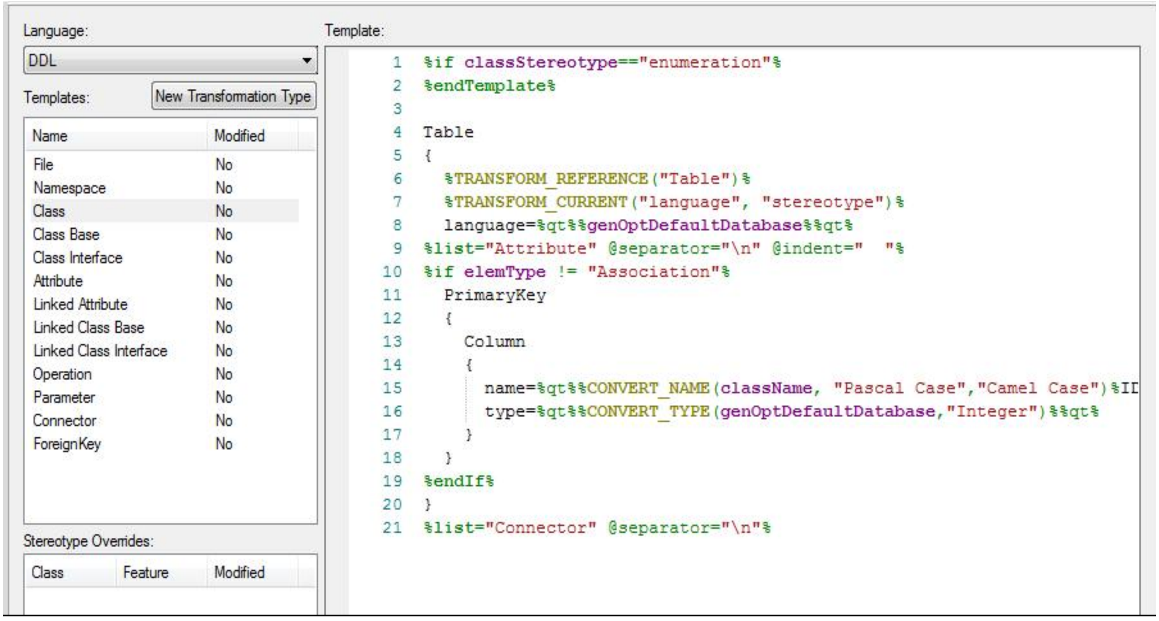
Figure 5.1: System architecture for the prototype user interface

5.3 Transformations from conceptual model to logical model

In EA, transformation from conceptual to logical model is implemented using the available transformation capabilities . EA provides two options for performing model transformation: using transformation definitions and using EA Software Developer Kit (EA SDK) [43].

- The EA SDK is a way of transforming models that allows to add functionality to EA using components developed in an application developed tool and then registered in EA. Application development tools that can be used to develop such components are Microsoft visual basic, C# etc.

- EA transformation definitions enable you to transform a model using transformation templates that specify how specific model elements are transformed. EA comes with a number of standard transformation definitions. An example of such transformation definitions is Data Definition Language (DDL), intended for use in the transformation of database design. This transformation definition is shown in Figure 5.2.



Language: DDL

Templates:

Name	Modified
File	No
Namespace	No
Class	No
Class Base	No
Class Interface	No
Attribute	No
Linked Attribute	No
Linked Class Base	No
Linked Class Interface	No
Operation	No
Parameter	No
Connector	No
ForeignKey	No

Stereotype Overrides:

Class	Feature	Modified

Template:

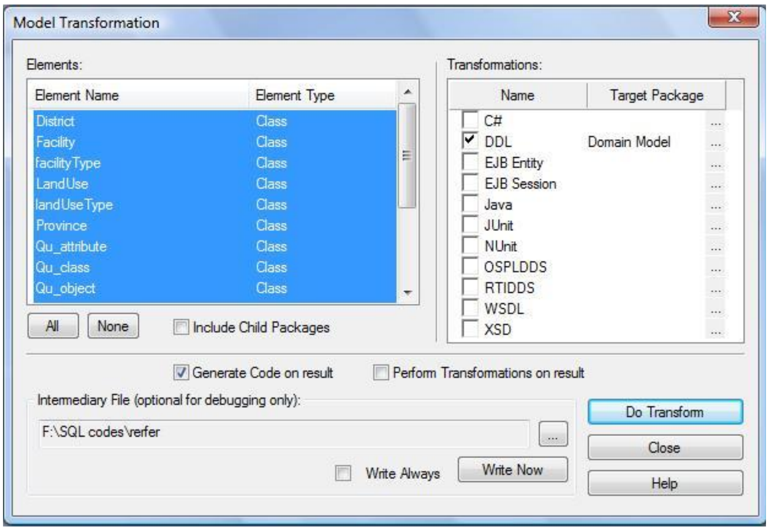
```
%if classStereotype=="enumeration"%
%endTemplate%

Table
{
  %TRANSFORM_REFERENCE("Table")%
  %TRANSFORM_CURRENT("language", "stereotype")%
  language=%qt%%genOptDefaultDatabase%%qt%
%list="Attribute" @separator="\n" @indent="  "%
%if elemType != "Association"%
  PrimaryKey
  {
    Column
    {
      name=%qt%%CONVERT_NAME(className, "Pascal Case","Camel Case")%II
      type=%qt%%CONVERT_TYPE(genOptDefaultDatabase,"Integer")%%qt%
    }
  }
%endIf%
}
%list="Connector" @separator="\n"%
```

Figure 5.2: EA standard transformation definition "DDL", conversion template for class

Time-wise, experience with EA SDK, and available resources in the thesis project, the EA SDK environment, has not been chosen for our transformation. We used transformation definitions without add-ins, and to fine-tune the transformation from conceptual to logical models, which can not be done in the EA transformation definitions has been done manually. The transformation from conceptual to logical model considers the conversion of UML class diagram and relationship types that represent the conceptual model, to a UML class diagram for the logical model. To perform the transformation we must have a UML class diagram as input and the relationship types for the conceptual model compliant with the database specifications [43]. The output is a UML class diagram that represents the logical model. The model transformation using standard DDL transformation shown in Figure 5.3 a logical model yielded from the conceptual model. EA supports and uses an intermediate language for a number of database specification concepts like table, column, and primary key and foreign key where:

- Class : Mapped one-to-one into table
- Attribute : Mapped one-to-one into column
- Primary key created automatically, for each class
- Associations : According to relationship involved in the source and destination class, and generates the primary key and foreign key, respectively.



Model Transformation

Elements:

Element Name	Element Type
District	Class
Facility	Class
facilityType	Class
LandUse	Class
landUseType	Class
Province	Class
Qu_attribute	Class
Qu_class	Class
Qu_object	Class

☐ Include Child Packages

Transformations:

Name	Target Package
☐ C#	
☑ DDL	Domain Model
☐ EJB Entity	
☐ EJB Session	
☐ Java	
☐ JUnit	
☐ NUnit	
☐ OSPLDDS	
☐ RTIDDS	
☐ WSDL	
☐ XSD	

☑ Generate Code on result ☐ Perform Transformations on result

Intermediary File (optional for debugging only):

F:\SQL codes\rerfer

☐ Write Always

Figure 5.3: Enterprise Architect DDL Transformation template for the use case

The result of the transformational design from conceptual to logical model is not only used for creation of the DDL transformation but also used for automatic generation of the DDL statements to run in one of the EA supported DBMS products with some modifications. The classes in our conceptual model indicated in Figure 4.3 are district, province, land use, reference land use, facility, reference facility, road, road segment, Qu_attribute, Qu_object, Qu_class and three enumeration classes. Each class has attributes and operations. Especially, the operation part of the design, one of the main task of our project and almost all the operations represent the spatial data quality elements and sub elements discussed in Chapter 2. The association types in our conceptual model are one-to-one and one-to-many associations indicated in the model. After the transformation from conceptual to logical model, the classes are mapped to tables, the attributes are mapped to columns, and the associations are mapped to primary and foreign keys. This can be summarized in table 5.1.

As indicated in Table 5.1, the transformations do not completely transform all classes, for example, the enumeration class. This is due to the limitation of the EA definition, and could be reserved using the add-in tools in EA, as mentioned in [21]. Hence, the transformation of the enumeration classes were created manually. In general, the transformation from the conceptual schema to logical schema can be shown in UML class diagram as indicated in Figure 5.4.

From the logical model shown in Figure 5.4, it could be observed that all classes in the conceptual model are converted into tables, the attributes are converted into columns, the primary and foreign keys are automatically generated during the transformation. However, the operations which are indicated in the conceptual design are not transformed to the logical design and these operations are the main parts of the project that we dealt with. We created functions in physical modelling. The logical model DDL is converted to SQL using the UML functionality code generation. The other important for our project is that the overview quality elements are represented in the class Qu_class as attributes. These are converted to columns in quality report table. Therefore, the quality elements indicated in the table Qu_class are at table level.

5.4 Transformation from logical to physical model

After logical model are defined, they are transformed into a database specific representation in the form of a physical model. That is, from platform independent to platform dependent model. We can use a wizard to transform the DDL created in logical model to SQL (physical model). However, there is a need for some modification in the logical model and the changes can then propagated to the physical model. For example, we create the enumeration class manually in the physical model the code is as follows.

```
CREATE TYPE facilityType(
school char(50),
health char(50),
water char(50),
animal_clinic char(50)
);
```

The second limitation during transformation is the geometry data type mentioned in the classes are not transformed and we fix them manually, by replacing them with PostGIS-specific geometry types. For example, the code generating for class road segment is:

```
CREATE TABLE RoadSegment (
geom LINESTRING,
positionalAccuracy real,
roadSegmentType roadType,
```

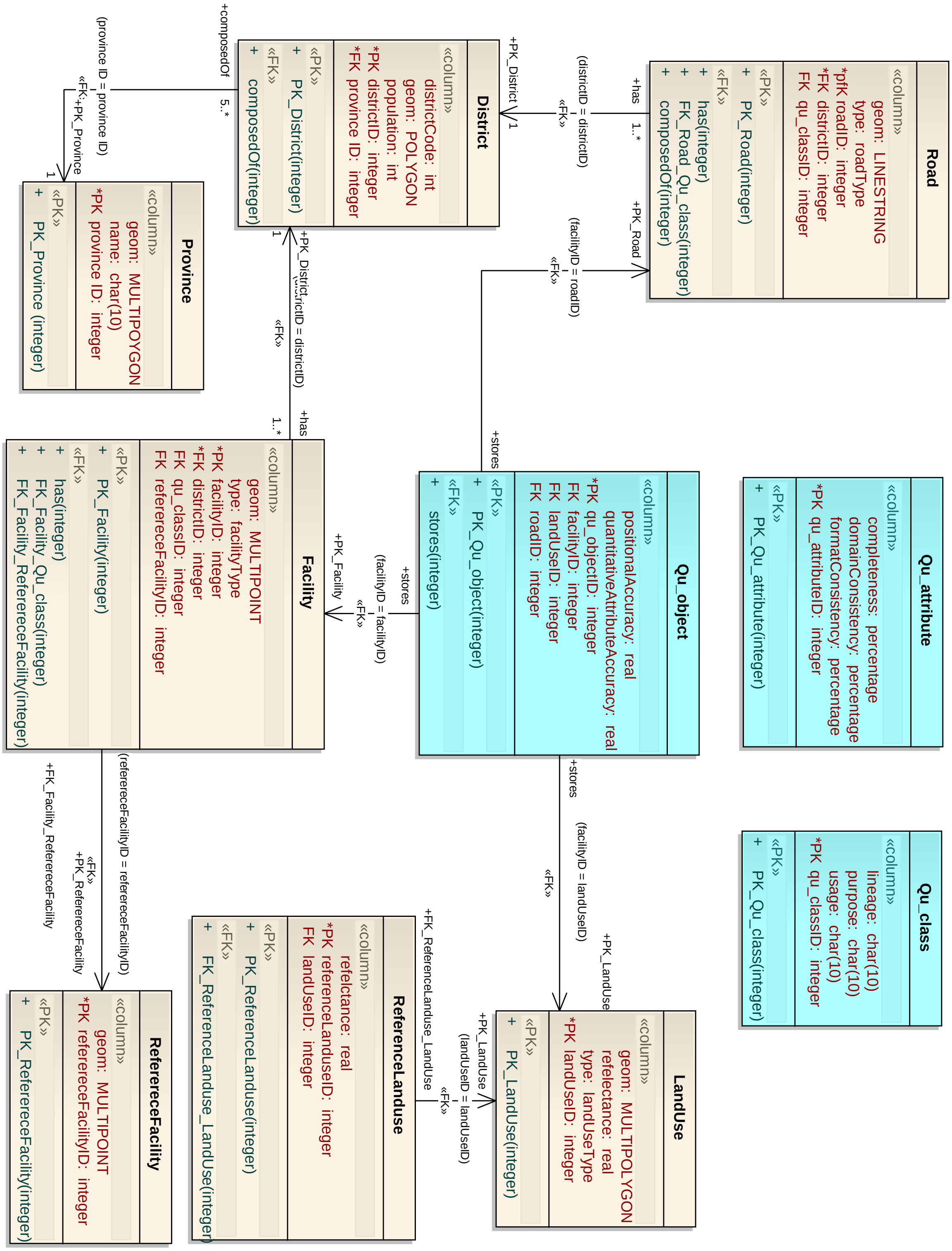


Figure 5.4: Logical model for the use case

```
roadSegmentID integer NOT NULL,
roadID integer NOT NULL
);
```

The geom attribute with data type LINESTRING, is unknown data type. This problem was fixed by deleting the geom attribute and adding the geometry column by the following code:

```
select addgeometrycolumn('public', 'RoadSegment', 'geom', 4326,
'LINESTRING', 2);
```

In addition, the functions which are indicated in the operation compartment in the conceptual model, class diagram are not transformed during transformation to logical model. The physical database model before corrections and after some corrections are indicated in Appendix A.1 and Appendix A.2 respectively. This is the stage where our specific platform will be chosen and in our case the implementation platform is PostgreSQL/PostGIS.

5.5 Functions for the implementation

In Chapter 2 we have discussed the data quality measure of each quality elements and they are the basis for the implementing the functions. As we have shown in Figure 4.3, the spatial data quality elements to be calculated from a given dataset are indicated in an operation compartment of UML class diagrams. During transformation from conceptual to logical model the operations are not transformed. Hence, we create functions in the physical model. These functions are used to evaluate the quality information of a given dataset with unknown quality. The DBMS to implement these functions is PostgreSQL/PostGIS with server-side PL/pgSQL programming language. The functions indicated in the conceptual model are positional accuracy, attribute accuracy, conceptual consistency, format consistency, domain consistency, topological consistency, and completeness. They are the applicable quality elements for the datasets mentioned in the use case. Due to shortage of time, in this project, we have implemented for the quality elements positional accuracy and completeness.

5.5.1 Function for evaluating the positional accuracy at table level

The positional accuracy can be evaluated by the root mean square error as indicated in Equation 2.1. From the equation we can drive function that can evaluate the positional accuracy of a given dataset with reference dataset of the same area. Hence, the output of this function is a single attribute, it can not be stored within the same table it describes, rather in another quality table or a separate table. The codes for function indicated in Appendix A.3. The flow diagram as shown in Figure 5.5 indicated how the function works.

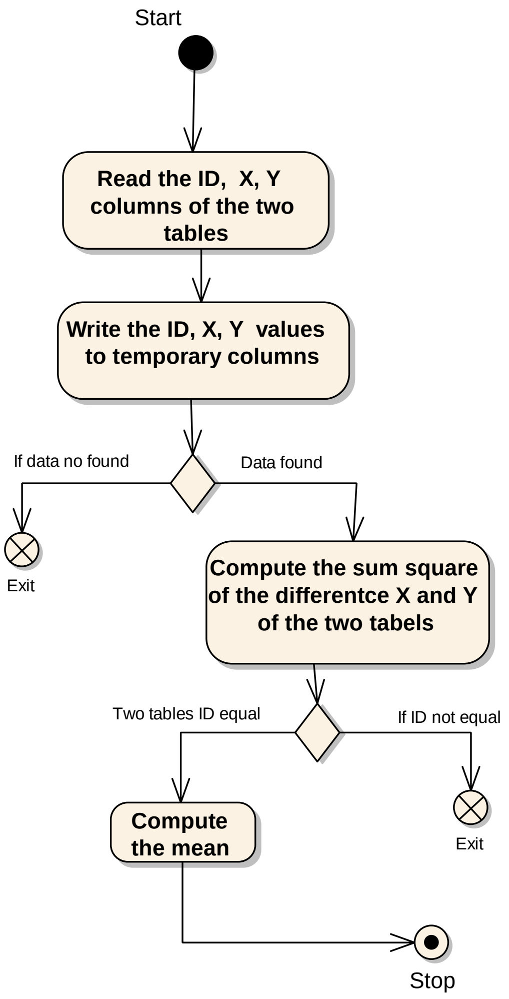

Start
Read the ID, X, Y columns of the two tables
Write the ID, X, Y values to temporary columns
If data no found
Data found
Exit
Compute the sum square of the differentce X and Y of the two tabels
Two tables ID equal
If ID not equal
Compute the mean
Exit
Stop

Figure 5.5: Flow diagram for evaluating positional accuracy at table level

5.5.2 Function for evaluating the positional accuracy of a dataset at tuple level

The positional accuracy can be evaluated at tuple level. Hence, we use different data quality measure as indicated in Equation 2.2. Due to the difference in data quality measure, we build another function for positional accuracy that can evaluate the quality at tuple level. The output for this function can be stored as a new column with dataset it describes or in a new table. According to the conceptual model in Figure 4.3, the output should be stored as an attribute of the table it describes. However, to use the function as generic function, we created a new temporary table that can store the positional accuracy of each tuple and this is also the concept from per-feature model as discussed in Chapter 3. Therefore, when this function evaluates the dataset, the output gets stored in the temporary table and users can see the output from the temporary table.

The codes for this function is indicated in Appendix A.4. Also the flow diagram as shown in Figure 5.6 indicated how the function works.

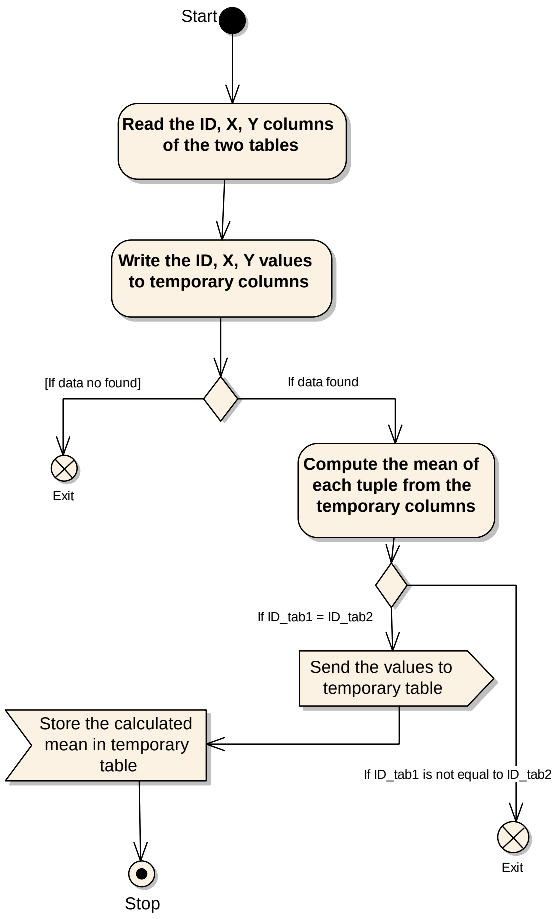

Start
Read the ID, X, Y columns of the two tables
Write the ID, X, Y values to temporary columns
[If data no found]
If data found
Exit
Compute the mean of each tuple from the temporary columns
If ID_tab1 = ID_tab2
Send the values to temporary table
Store the calculated mean in temporary table
If ID_tab1 is not equal to ID_tab2
Exit
Stop

Figure 5.6: Flow diagram for evaluating positional accuracy at tuple level

5.5.3 Function for evaluating the completeness of a dataset

As we have discussed in Chapter 2, completeness can be calculated under two different assumptions. The first assumption is the CWA, the values assumed as actually present in a relational table. In OWA, we can state neither the truth nor the falsity of facts not represented in the tuples of a given instance. However, we developed the completeness function under CWA. The flow diagram as shown in Figure 5.7 is for the completeness function that evaluate the error of omission. It takes one input parameter table and it calculates the error of omission for each columns in the table. The output is at attribute level and stored as a temporary table. The code of this function is shown in Appendix A.5. The dataset to be evaluated by this function should be already stored in PostgreSQL/PostGIS DBMS. When the dataset got updated, the quality information also updates accordingly.

5.6 Prototype for evaluating and reporting the quality of spatial data

As we have seen above, some of the functions are built on the server-side in PostgreSQL DBMS, so that general users can not understand what the functions do. It is important to develop a prototype user interface so that general users can understand it easily. The prototype is implemented in such a way that it allows the user to provide input data to the system and get back the output. This can be indicated by UML sequence diagram as shown in Figure 5.8.

Since the main functionality of our prototype is to compute the quality information of a given dataset and the Web interface allows the user to specify the table for which the quality information is to be computed. It consists of a form that allows the user to choose the table and the applicable quality elements for which s/he would like to compute the quality information of the table and submit the input to the system as illustrated in Figure 5.9. After submitting the form, the user retrieves the resulting quality information which displays in a table format. The user interface allows the user to visualize the output as a summary of the quality information. The map indicated in the left side is the dataset and the result shown in the right is the quality information of the dataset evaluated according to users request. The code of the user interface was written in Java Scripting language, Hypertext Preprocessor (PHP), and Hyper Text Markup Language (HTML) as indicated in Appendix C.

As the quality report indicated in Figure 5.9, the quality information is calculated for each column of a given table. Therefore, the output is stored in a temporary table at an attribute level. The last record of the report gives the total quality information at table level. This result is not included in the temporary table which is stored in the database. We developed business process with PHP programming on the Web server (Apache) to calculate the overall quality of the given table on the fly and then reported at the same time with the quality information report. Finally, if the overall quality is less than the acceptance value, the dataset fails. And, if the overall quality is greater than or equal to acceptance value the dataset is said to be passed. This business process is also done on the Web server using PHP programming. From this we can understand that a lot of work can be done on the Web server using PHP programming language to fulfill the quality evaluation procedures. The positional accuracy of a given dataset can be calculated on the fly, like what we did for completeness. However, to evaluate the positional accuracy we need a reference dataset, so users should select the dataset to be evaluated together with the reference dataset of the area. The output can be display as quality report through the Web interface as shown in Figure 5.10. This output is stored in a temporary table at tuple level. Hence, user can get quality information at tuple level.

In general, so far the implementation is according to the user requirement mentioned in Section 4.2.1. More specifically for human users. The human users requirements include:

- Users can select the dataset and the applicable quality element to get the quality information of the dataset.
- Users can retrieve spatial data with spatial data quality information through the Web browser at click button.
- Users can get quality information at tuple level as shown in Figure 5.9 or at attribute level as shown in Figure 5.8
- During spatial data updating, the quality information should also be updated and the quality information is dynamically updated.

The output of the quality information is stored in a temporary table. When users request the dataset's quality information, it is automatically generated on the fly and consequently the report is updated. This dynamic characteristic applies to all the functions discussed above. The concept can be extended to develop a XML schema that consists of quality information according ISO 19115 metadata on client-side, then the calculated quality information could be stored to the schema on the fly. For our purpose, the temporary table are stored in database and at the same time we develop a Web application using PHP programing on the Web server, to display the temporary table on the client-side. So that, clients get quality information according to schema in the database. The limitation of our prototype from the requirements mentioned in the user requirement is that:

- The acceptable quality element should be set by user and
- The report format should be according to the ISO 19115. These are the requirements that are not implemented and requires to develop business process in Web server. These can be proposed for future work.

The implementation for the user requirements like Web service users and users of other spatial databases are not implemented. Especially, to implement the WPS the back-end should be WFS. Hence, to work on the Web services it needs a detail understanding the specification according to OGC Web services, which are the main important issues to implement the Web services. For users of other spatial databases can be done by putting in place different sources of spatial databases. These are requirements mentioned in Section 4.2.1 due to time constraint, the implementation is recommended for future work.

5.7 Summary

In summary, the implementation steps for the transformation were discussed in this Section. The transformation implementation was separated in two parts: the transformation from conceptual to logical model and from logical to physical model. In the transformation from conceptual to logical model, EA was used as a modelling and transformation tool. The UML class diagram that represents the conceptual model transformed to its logical representation by using EA transformation definition called DDL. The SQL statement for postgreSQL/PostGIS obtained during the transformation from logical to physical model were verified by executing the obtained SQL statement. Some of the limitations during transformation from conceptual to logical model were solved manually in the physical model. For example, the operations in conceptual were changed to functions in DBMS manually. In addition, we proposed the prototype for specific user, human users.

The main objective of the prototype was to show that it is possible to evaluate the quality information of a given dataset on the fly and send the quality information to users in the form of report. The report can provide data quality information in a more meaningful and useful way for data users. So, they can assess the fitness of certain data for an intended use to a given area. It should be noted that the actual operation of the prototype is limited to only completeness and positional accuracy for vector data type, although the data quality model and the prototype system allow for different data quality parameters to be selected. The other issue we noticed in this chapter was that the quality evaluation procedure can be done by developing business process using PHP programming on Web server. For example, the output of the function that calculates the quality information of completeness was store in a temporary table in the database. When we report to users through the Web, we made a business process on the Web server using PHP programming to calculate the overall completeness stored in the temporary table. Thus, the quality evaluation procedures can be done on the Web server. The functions developed in this project can update the quality information when the dataset update.

Table 5.1: mapping from conceptual to logical model transformation for the use case

Transformation of classes	
Classes in conceptual model	Mapped to table in logical model
District	District
facility	Facility
Land use	Land use
Road	road
Road segment	Road segment
province	Province
Qu_class	Qu_class
Qu_object	Qu_object
Qu_attribute	Qu_attribute

Transformation of enumeration classes	
Enumeration classes in conceptual model	The enumeration classes are not transformed to logical model then we treated them manually in physical modelling. By creating tables with user define data type.
landUseType	
roadType	
facilityType	

Transformation of associations	
Associations in conceptual model	Mapped to in logical model
One-to-one associations	The associations are transformed to primary or foreign key in logical modelling automatically.
One-to-many associations	
Composition associations	

Transformation of operations	
Operations in conceptual model	The operations are not transformed to logical model then we treated them manually in physical modelling by creating functions.
Completeness	
Positional accuracy	
Format consistency	
Domain consistency	
Quantitative attribute accuracy	

Transformation of constraints	
Constraints in conceptual model	The constraints are not transformed to logical model then we treated them manually in physical modelling by creating triggers
Database constraints (between tables):- facilities should not be within road	
Population constraints (with in table) :- land use type not overlap to each other	

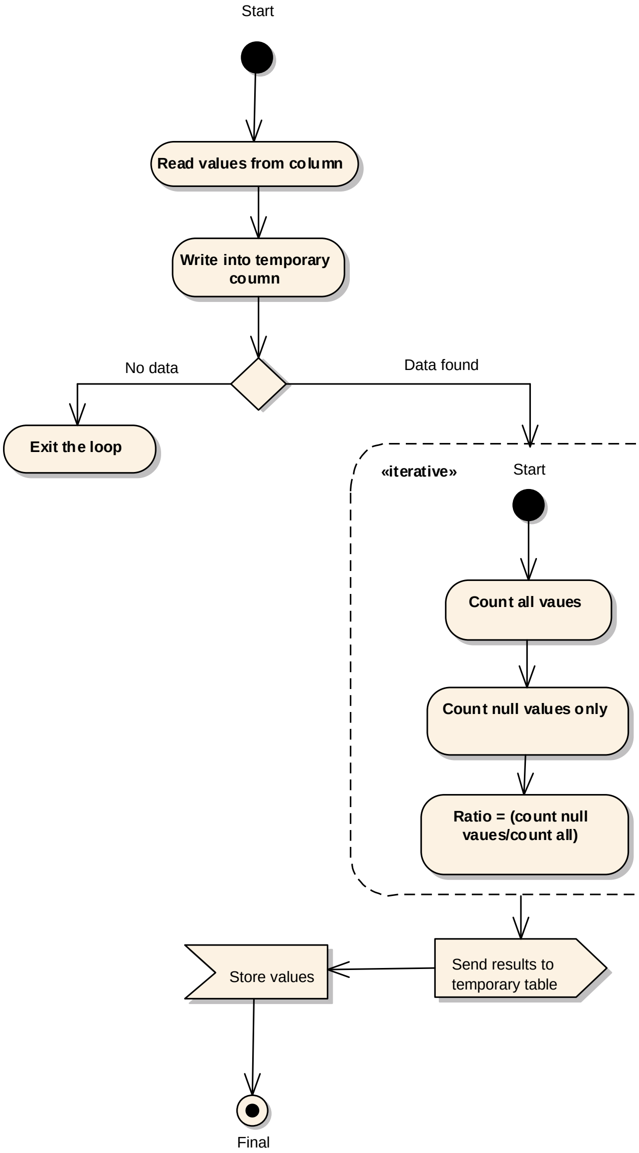


Figure 5.7: Flow diagram for evaluating error of omission at attribute level

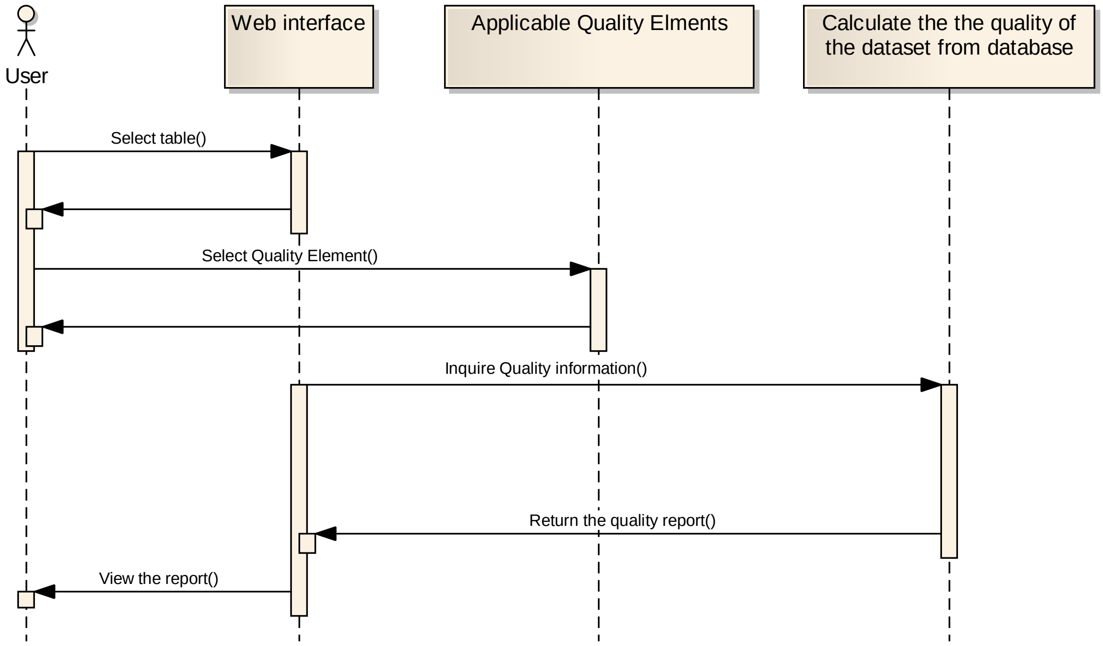

User
Web interface
Applicable Quality Elments
Calculate the the quality of the dataset from database
Select table()
Select Quality Element()
Inquire Quality information()
Return the quality report()
View the report()

Figure 5.8: Steps to retrieve the quality information from the prototype

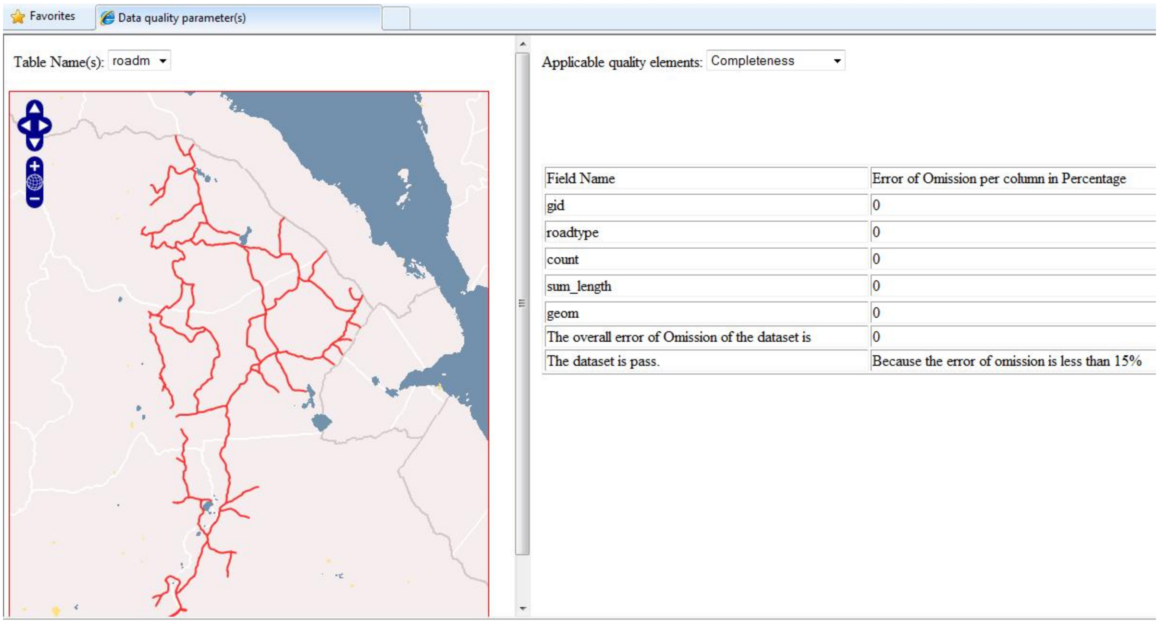


Table Name(s): roadm

Applicable quality elements: Completeness

Field Name	Error of Omission per column in Percentage
gid	0
roadtype	0
count	0
sum_length	0
geom	0
The overall error of Omission of the dataset is	0
The dataset is pass.	Because the error of omission is less than 15%

Figure 5.9: Prototype interface for retrieving the quality information completeness

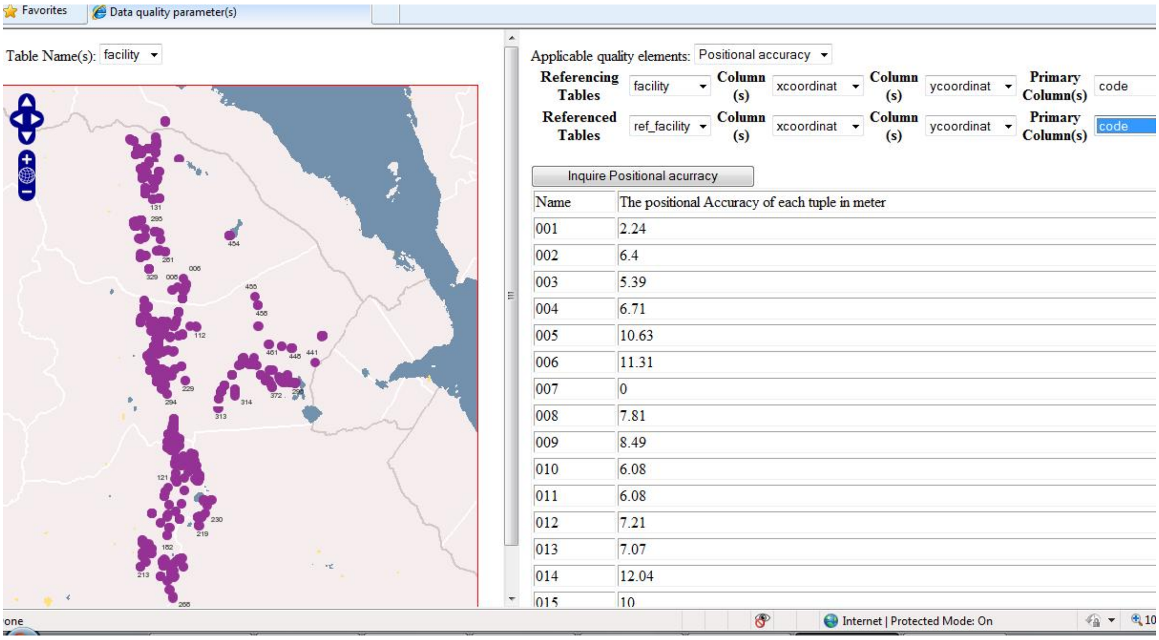


Table Name(s): facility

Applicable quality elements: Positional accuracy

Referencing Tables	facility	Column (s)	xcoordinat	Column (s)	ycoordinat	Primary Column(s)	code
Referenced Tables	ref_facility	Column (s)	xcoordinat	Column (s)	ycoordinat	Primary Column(s)	code

Inquire Positional acurracy

Name	The positional Accuracy of each tuple in meter
001	2.24
002	6.4
003	5.39
004	6.71
005	10.63
006	11.31
007	0
008	7.81
009	8.49
010	6.08
011	6.08
012	7.21
013	7.07
014	12.04
015	10

Figure 5.10: Prototype interface for retrieving quality information of dataset at tuple level.

Chapter 6

Discussion, Conclusions, and Recommendations

6.1 Introduction

This chapter presents a summary of the results of the research project, discusses those results, and presents the conclusions drawn from them and the recommendations made for future improvement. Section 6.1 contains the results and their discussion. Section 6.2 contains the conclusions and finally section 6.3 presents the recommendations.

6.2 Results and discussion

The results are presented and discussed with respect to individual research questions specified in section 1.2.2.

The theoretical and mathematical concepts of spatial data quality

Chapter 2 presents related work to spatial data quality research. One of the characteristics identified from the review is the data quality measure of each quality elements. These are the main characteristics of spatial data quality that could help us to integrate spatial data quality with spatial database. The mathematical background of each quality elements helps us to formulate the functions. From the developed functions, we infer the output of each quality element to consider the hierarchy level storage of spatial data quality in spatial database. For example, if the output is a single value after the evaluation, it is stored in a separate table. If the output is for each tuples evaluated in the dataset, the storage may be within the table it describes. All this information is the result of the data quality measure of each quality elements. Therefore, our observation shows that the data quality measure of each quality elements is the basis for effective design and implementation spatial data quality in spatial database with the possible hierarchy of spatial database structure.

Design of spatial data quality model as an integral part of spatial database

Chapter 3 highlighted the design requirements for spatial data quality. These requirements are the basis for design of spatial data quality with spatial data. To meet these requirements, the object relational data model is considered as a better solution to design the spatial data quality model. For the purpose of design of spatial data quality with spatial data we describe use case in chapter 4. Finally, according modelling guidelines of data quality discussed in chapter 3, we develop the conceptual model that associates spatial data quality with spatial data. It is possible that the spatial data quality can be designed at different hierarchy of spatial database. As indicated in the conceptual model in figure 4.3, some of the quality elements are integrated at tuple level like positional accuracy and attribute accuracy as shown in class Qu_object. The quality elements like lineage, usage, and purpose are integrated at class level as shown in class Qu_class and and the quality elements like completeness, format consistency, and domain consistency are represent at attribute level as shown in class Qu_attribute. Topological consistency is one of the data quality subelement can be shown as an association in the design. Therefore, quality elements can be linked with spatial data that they describe at different hierarchies. The quality elements that need to be processed are indicated at an operation compartment of the UML class diagram. The integration of spatial data with its quality description and can be easily updated during spatial data processing. Moreover, the proposed model may have a potential capability of dealing with dynamic updating of data quality as data changes through various processing. This meets the integration requirement that quality information should be integrally linked with the dataset it describes.

Define and store a set of functions that evaluate and test the quality of spatial dataset in a spatial database

Chapter 5 discussed about the implementation of the conceptual model developed in chapter 4. Transformation from conceptual to logical model was performed. While doing transformation, due to the limitation of the current modelling EA tool the functions designed in an operation compartment of UML class diagram is not transformed. Hence, we have to create functions manually. As we have discussed in chapter 5, in this project we create functions that can evaluate the completeness and positional accuracy of a given dataset. The DBMS to implement these functions is PostgreSQL/PostGIS with server-side PL/pgSQL programming language. The function completeness takes one input parameter; it calculates the completeness of each column existing in the input table. Hence, this function evaluates the given dataset at attribute level and store in a separate table. The other function that calculates the positional accuracy of a given dataset needs reference and only restricted to point at geographic data.

Exploring the stored functions of a spatial data quality model in spatial database to be accessed by different types of user

As we have discussed in chapter 4, there are different users of spatial data quality that include: human users, Web service users, or users of other spatial databases. The functions created on the server-side (PostgreSQL/PostGIS), can not be accessed by general users. Therefore, we develop a prototype user interface using PHP as middleware that calls the function from back-end and calculates the quality of a given dataset on the fly client-side. For example, to call the function completeness to the Web browser from the database we develop program by programming language PHP and stored in a Web server. The prototype is implemented in such a way that it allows the user to provide input data to the system and get back the output. Since the main functionality of our user interface is to compute the quality information of a given dataset and the interface allows the user to specify the table for which the quality information is to be computed. Hence, from this we can understand that the functions stored in DBMS can be accessible to users.

Validating the effectiveness of the developed model using real world data

- According to the developed prototype we have test with the data described in the use case. The prototype can give us quality report at attribute, tuple, or table level as indicated in Section 5.7 and 5.8. Therefore, the developed prototype can report the quality information at multi-level structure. The design requirements multi-level structure for spatial data quality is fulfills.
- According the proposed modelling guidelines discussed in Chapter 3, we develop a conceptual model using the spatial data from use case described in Chapter 4. This shows us that spatial data quality can be integrated with spatial data in spatial databases. This fulfills one of the design requirements for spatial data quality requirements that are mentioned in section 3.6.
- The quality information could be updated when the dataset updated. This is an indication that the possibility of getting quality information dynamically.

Therefore, from the developed prototype we can observe that quality information can be reported at different hierarchy, integrated with spatial data, accessible to user through quality report, and the dynamic updating of spatial data quality information are the witness for the effectiveness of the prototype.

6.3 Conclusions

The research project began by investigating how spatial quality is managed in the existing spatial data quality models. Limitations were identified with regard to storage and communication of spatial data quality in spatial databases.

To overcome these limitations, we projected to design a model of spatial data quality that has been incorporated into the model of the spatial data they describe.

In this MSc project, we proposed the creation of a methodology that integrates spatial data quality with spatial data. With spatial data quality being integrated into spatial database model, the users would have access to data quality information usable to evaluate whether a given dataset is appropriate and fit for their respective applications.

First, the identification of the characteristics of each data quality elements that forms the basis to integrate spatial data quality elements into spatial databases was required. These quality characteristics include: data quality measure, storage, and report. For instance, from the storage of each quality element we can be able to identify which quality element is stored at different level in the database structure. Thereafter, we developed modelling guideline of spatial data quality. This directs us to understand how spatial data quality models integrate with spatial database structure.

We later implemented the modelling guidelines of data quality using a working use case as described in Chapter 4. Then, we identified the applicable data quality elements that can be stored together with spatial data. For instance, if data quality element is of type attribute, it can be entered into the respective attribute records in the database.

According to our modelling guidelines of data quality elements, we developed a conceptual model of spatial data. The conceptual model was transformed to the logical model. While doing the conceptual model transformation, some data quality elements were not transformed, that is, the quality elements were defined as the operation in the conceptual design. The quality elements that failed to be transformed were created manually. We later developed functions used to evaluate the quality the given datasets. To make the quality elements available to the users, we developed a Web interface prototype that enabled users to trigger the stored database functions and use them to evaluate the data quality of a given dataset.

By adopting the modelling guidelines of spatial data quality, we were able to introduce more usefulness to the stored data by accompanying them with their quality information, which have previously been stored and accessible in a separate metadata repository. Hence, this is believed to enrich the usage of stored spatial data in meeting a wide range of user needs.

Limitations

- The developed prototype is only limited to report form of quality information.
- The function for the positional accuracy can not be interpolated the quality information for the unknown quality without reference dataset.
- The prototype is limited to quality elements completeness and positional accuracy
- The developed prototype is specific for human users only.

- Using the developed prototype users can not update the spatial data and quality information.

6.4 Recommendations

Based on the results of this thesis project, the following recommendations are made for future improvements.

- Expanding the actual functionality of the prototype to different data quality elements while the prototype system allows for different data quality parameters to be selected. Currently the actual operation of the prototype is limited to completeness and positional accuracy. It is suggested that more research is required in investigating other data quality elements more specifically for the positional accuracy of lines and polygons.
- The prototype should be enhanced by making it interactive. This means data users need to set the acceptable quality element according to their need to make the judgment on the acceptable quality information for their particular application.
- Extending the prototype to provide functionalities of tracking spatial data quality through data processing and manipulating to inform users during analysis of spatial data. For example, spatial data processing during the Web processing service.
- The quality report generated from the prototype could be linked to the catalogue services. So that the catalogue service could serve us with updated quality information dynamically.
- In the use case description we discussed about the different types of users human users, Web service users, and users of other spatial databases. Due to shortage of time, the Web service users and users of other spatial databases were not addressed in this thesis project. However, the approach that we followed in this thesis project could be applicable to implement these users.
- We have developed a conceptual model that integrated spatial data quality with spatial data. The model could be enhanced to respect the principles of Model Driven Architecture (MDA)

Bibliography

[1] Donaubauer A., Kutzner T., and Straub F. Towards a Quality Aware Web Processing Service. 2008.

[2] Friis-Christensen A., Christensen J.V., and Jensen C.S. A Framework for Conceptual Modeling of Geographic Data Quality. pages 605–616, 2005.

[3] Jakobsson A. Data quality and quality management–examples of quality evaluation procedures and quality management in European national mapping agencies. *Spatial Data Quality*, pages 216–229, 2002.

[4] Yeung A.K.W. and Hall G.B. *Spatial database systems: design, implementation and project management*. Springer Verlag, 2007.

[5] Korte G. B. *The GIS book*. Onword Press, 1997.

[6] Batini C. and Scannapieco M. *Data quality: Concepts, methodologies and techniques*. Springer-Verlag New York Inc, 2006.

[7] Parent C., Spaccapietra S., and Zimányi E. *Conceptual modeling for traditional and spatio-temporal applications: the MADS approach*. Springer-Verlag, 2006.

[8] Penninga F. and Van Oosterom P. A Compact Topological DBMS Data Structure For 3D Topography In: SI Fabrikant and M. Wachowicz (Eds.) The European Information Society-Leading the Way with Geo-Information. *Lecture Notes in Geoinformation and Cartography, Springer*, pages 455–471, 2007.

[9] Open GIS. Consortium, OpenGIS Simple Feature Specification for OLE/COM. *OpenGIS Implementation Specifications, Revision*, 1, 1999.

[10] Hunter G.J., Bregt A.K., Heuvelink G.B.M., Bruin S., and Virrantaus K. Spatial Data Quality: Problems and Prospects. In *Research Trends in Geographic Information Science*, pages 101–121. 2009.

[11] Hunter G.J., Bregt A.K., Heuvelink G.B.M., De S., Wageningen N., and Espoo F. Spatial Data Quality: Problems and Prospects.

[12] Hunter G.J. and Bruin S. de. A Case Study in the Use of Risk Management to Assess Decision Quality. In *Research Trends in Geographic Information Science*, pages 271–282. 2006.

[13] Object Management Group. OMG unified modelling language specification. http://www.omg.org/technology/documents/formal/uml.htm, September 2001.

[14] Couclelis H. Two Perspectives on Data Quality for quality monitoring. *NCGIA Technical Technical Report 92-12*, page 21pp, 1992.

[15] Moellering H. and Hogan R.L. *Spatial database transfer standards 2: Characteristics for assessing standards and full descriptions of the national and international standards in the world*. Pergamon, 1997.

[16] Tveite H. and Langaas S. Accuracy assessments of geographical line data sets,the case of the digital chart of the world. http://statisk.umb.no/ikf/gis/dcw/pap-scan.pdf Access Date 20-11-2009, 2001.

[17] ISO. *Text of 19113 Geographic information - Quality principles, as sent to the ISO Central Secretariat for registration as FDIS*. ISO/TC 211 Secretariat, 2002.

[18] ISO. 19115, Geographic information–Metadata. *International Standard*, 2003.

[19] ISO. *Text of 19114 Geographic information - Quality evaluation procedures, as sent to the ISO Central Secretariat for publication*. ISO/TC 211 Secretariat, 2003.

[20] ISO. *Text for ISO 19131 Geographic information - Data product specification, as sent to the ISO Central Secretariat for issuing as FDIS*. ISO/TC 211 Secretariat, 2006.

[21] Hespanha J., van Bennekom-Minnema J., Van Oosterom PJM, and Lemmen C. The Model Driven Architecture approach applied to the Land Administration Domain Model version 1.1-with focus on constraints specified in the Object Constraint Language. In *FIG Working Week*, pages 14–19, 2008.

[22] Qiu J. and Hunter G. J. A GIS with the capacity for managing data quality information. In Shi W., Fisher P.F., and Goodchild M.F., editors, *Spatial Data Quality*, pages 230–250. Taylor and Francis, 2002.

[23] Rumbaugh J., Jacobson I., and Booch G. *Unified Modeling Language Reference Manual, The*. Pearson Higher Education, 2004.

[24] Ramirez J.R. and Ali T. Progress in metrics development to measure positional accuracy of spatial data. page 10. Document Transformation Technologies, 2003.

[25] Beard K. Representations of data quality. *Geographic Information Research: Bridging the Atlantic*, pages 280–294, 1997.

[26] Fassnacht K.S., Cohen W.B., and Spies T.A. Key issues in making and using satellite-based maps in ecology: A primer. *Forest Ecology and Management*, 222(1-3):167–181, 2006.

[27] Duckham M. Object Calculus and the Object-Oriented Analysis and Design of an Error-Sensitive GIS. *GeoInformatica*, 5(1):261–289, 2001.

[28] Duckham M. and Drummond J. Implementing an Object-Oriented Approach to Data Quality. *Innovations in GIS*, pages 53–64, 1999.

[29] Worboys M. and Duckham M. GIS: A Computing Perspective. `http://books.google.co.uk/books`, 2004.

[30] Goodchild M.F. Measurement-based GIS. *Spatial Data Quality*, pages 5–17, 2002.

[31] Sadiq M.Z. and Duckham M. Integrated Storage and Querying of Spatially Varying Data Quality Information in a Relational Spatial Database. *Transactions in GIS*, 13(1):30–42, 2009.

[32] Huisman O. and de By R.A. *Principles of geographic information systems : an introductory textbook*. ITC, 2009.

[33] Rigaux P., Scholl M.O., and Voisard A. *Spatial databases: with application to GIS*. Morgan Kaufmann, 2002.

[34] Raju P.L.N. Satellite remote sensing and gis applications in agricultural meteorology. `http://www.wamis.org/agm/pubs/agm8/Paper-6.pdf` Access Date 08-11-2009, September 2001.

[35] Devillers R. and Jeansoulin R. *Fundamentals of spatial data quality*. ISTE, 2006.

[36] Devillers R., Bédard Y., and Jeansoulin R. Multidimensional management of geospatial data quality information for its dynamic use within GIS. *Photogrammetric Engineering & Remote Sensing*, 71(2):205–215, 2005.

[37] Devillers R., Bédard Y., Jeansoulin R., and Moulin B. Towards spatial data quality information analysis tools for experts assessing the fitness for use of spatial data. *International Journal of Geographical Information Science*, 21(3):261–282, 2007.

[38] Elsmari R. and Navathe S. Fundamentals of database systems, 2000.

[39] France R. and Rumpe B. Model-driven development of complex software: A research roadmap. In *2007 Future of Software Engineering*, pages 37–54. IEEE Computer Society, 2007.

[40] Wang R.Y. and Strong D.M. Beyond accuracy: What data quality means to data consumers. *Journal of management information systems*, 12(4):33, 1996.

[41] Shekhar S. and Chawla S. *Spatial databases : a tour*. Prentice Hall, 2003.

[42] Guptill S.C. and Morrison J.L. *Spatial data quality*. Elsevier on behalf of the International Cartographic Association (ICA), 1995.

[43] Sparx systems. Enterprise architect. http://www.sparxsystems.com.au/products/ea/ Access Date 08-01-2009.

[44] ISO TC. 211/WG 2, 1999a,Geographic information-Spatial schema. Technical report, Technical Report second draft of ISO 19107 (15046-7), International Organization for Standardization, 1999.

[45] National Mapping Division United States Geological Survey. *Spatial Data Transfer Standard (SDTS)*. American National Standards Institute, Inc., 1998.

[46] van Oort P. *Spatial data quality : from description to application*. PhD thesis, Nederlandse Commissie voor Geodesie (NCG), 2005.

[47] Shi W. *Principles of modeling uncertainties in spatial data and spatial analysis*. CRC, 2008.

[48] Shi W., Goodchild M.F., and Fisher P. *Spatial data quality*. CRC, 2002.

[49] He Y. *Spatial data quality management*. PhD thesis, Faculty of Engineering, UNSW, 2009.

[50] Ting Y. *Visualisation of spatial data quality for distributed GIS*. PhD thesis, School of Surveying and Spatial Information Systems, UNSW, 2007.

Appendix A

Appendix

A.1 Physical Database model before change made

```
ALTER TABLE Facility DROP CONSTRAINT not within;
ALTER TABLE QualityReport DROP CONSTRAINT FK_QualityReport_District;
ALTER TABLE QualityReport DROP CONSTRAINT FK_QualityReport_Facility;
ALTER TABLE QualityReport DROP CONSTRAINT FK_QualityReport_LandUse;
ALTER TABLE QualityReport DROP CONSTRAINT FK_QualityReport_Road;
ALTER TABLE Road DROP CONSTRAINT FK_Road_District;
ALTER TABLE RoadSegment DROP CONSTRAINT composedOf;
DROP TABLE District;
DROP TABLE Facility;
DROP TABLE LandUse;
DROP TABLE QualityReport;
DROP TABLE Road;
DROP TABLE RoadSegment;
CREATE TABLE District (
geom polygon,
name char(10),
districtID integer NOT NULL
);
CREATE TABLE Facility (
geom point,
positionalAccurcy real,
type facilityType,
facilityID integer NOT NULL,
roadID integer
);
CREATE TABLE LandUse (
attributeAccuracy real,
geom multipolygon,
name char(10),
reflectance real,
type landUseType,
landUseID integer NOT NULL
);
CREATE TABLE QualityReport (
attributeAccuracy percentage,
completeness percentage,
lineage char(10),
logicalConsistency percentage,
positionalAccuracy percentage,
purpose char(10),
usage char(10),
qualityReportID integer NOT NULL,
districtID integer NOT NULL,
facilityID integer NOT NULL,
landUseID integer NOT NULL,
roadID integer NOT NULL
);
COMMENT ON TABLE QualityReport
    IS 'Report of applicable quality elements for a given dataset at
    table level';
CREATE TABLE Road (
geam polyline,
name char(10),
positionalAccuracy real,
roadID integer NOT NULL,
districtID integer NOT NULL
);
CREATE TABLE RoadSegment (
geom line,
positionalAccuracy real,
type roadType,
roadSegmentID integer NOT NULL,
roadID integer NOT NULL
);
ALTER TABLE District ADD CONSTRAINT PK_District
PRIMARY KEY (districtID);
ALTER TABLE Facility ADD CONSTRAINT PK_Facility
PRIMARY KEY (facilityID);
ALTER TABLE LandUse ADD CONSTRAINT PK_LandUse
PRIMARY KEY (landUseID);
ALTER TABLE QualityReport ADD CONSTRAINT PK_QualityReport
PRIMARY KEY (qualityReportID);
ALTER TABLE Road ADD CONSTRAINT PK_Road
PRIMARY KEY (roadID);
ALTER TABLE RoadSegment ADD CONSTRAINT PK_RoadSegment
PRIMARY KEY (roadSegmentID);
ALTER TABLE Facility ADD CONSTRAINT not within
FOREIGN KEY (roadID) REFERENCES Road (roadID);
ALTER TABLE QualityReport ADD CONSTRAINT FK_QualityReport_District
```

```
FOREIGN KEY (districtID) REFERENCES District (districtID);
ALTER TABLE QualityReport ADD CONSTRAINT FK_QualityReport_Facility
FOREIGN KEY (facilityID) REFERENCES Facility (facilityID);
ALTER TABLE QualityReport ADD CONSTRAINT FK_QualityReport_LandUse
FOREIGN KEY (landUseID) REFERENCES LandUse (landUseID);
ALTER TABLE QualityReport ADD CONSTRAINT FK_QualityReport_Road
FOREIGN KEY (roadID) REFERENCES Road (roadID);
ALTER TABLE Road ADD CONSTRAINT FK_Road_District
FOREIGN KEY (districtID) REFERENCES District (districtID);
ALTER TABLE RoadSegment ADD CONSTRAINT composedOf
FOREIGN KEY (roadID) REFERENCES Road (roadID);
```

A.2 The corrected Physical Database model

```
ALTER TABLE Facility DROP CONSTRAINT not within;
ALTER TABLE QualityReport DROP CONSTRAINT FK_QualityReport_District;
ALTER TABLE QualityReport DROP CONSTRAINT FK_QualityReport_Facility;
ALTER TABLE QualityReport DROP CONSTRAINT FK_QualityReport_LandUse;
ALTER TABLE QualityReport DROP CONSTRAINT FK_QualityReport_Road;
ALTER TABLE Road DROP CONSTRAINT FK_Road_District;
ALTER TABLE RoadSegment DROP CONSTRAINT composedOf;
DROP TABLE District;
DROP TABLE Facility;
DROP TABLE LandUse;
DROP TABLE QualityReport;
DROP TABLE Road;
DROP TABLE RoadSegment;
CREATE TABLE District (
geom polygon,
districtName char(50),
districtID integer NOT NULL
);
CREATE TABLE Facility (
geom point,
positionalAccurcy real,
serviceType facilityType,
facilityID integer NOT NULL,
roadID integer
);
CREATE TYPE facilityType (
school char(50),
health char(50),
water char(50),
animal_clinic char(50)
);
```

```
CREATE TABLE LandUse (
attributeAccuracy real,
geom multipolygon,
landUsename char(50),
reflectance real,
classType landUseType,
landUseID integer NOT NULL
);
CREATE TYPE landUseType (
bushLand char(50),
cultivatedLand char(50),
waterbody char(50),
woodLand char(50),
grassLand char(50),
rockSurface char(50),
saltFlat char(50),
sandFlat char(50)
);
CREATE TABLE QualityReport (
attributeAccuracy percentage,
completeness percentage,
lineage char(10),
logicalConsistency percentage,
positionalAccuracy percentage,
purpose char(10),
usage char(10),
qualityReportID integer NOT NULL,
districtID integer NOT NULL,
facilityID integer NOT NULL,
landUseID integer NOT NULL,
roadID integer NOT NULL
);
COMMENT ON TABLE QualityReport
    IS 'Report of applicable quality elements for a given dataset at
    table level';
CREATE TABLE Road (
geam polyline,
roadName char(50),
positionalAccuracy real,
roadID integer NOT NULL,
districtID integer NOT NULL
);
CREATE TABLE RoadSegment (
geom line,
positionalAccuracy real,
roadSegmentType roadType,
roadSegmentID integer NOT NULL,
roadID integer NOT NULL
);
CREATE TYPE landUseType (
asphalt char(50),
railway char(50),
dry_weather_road char(50)
);
ALTER TABLE District ADD CONSTRAINT PK_District
PRIMARY KEY (districtID);
ALTER TABLE Facility ADD CONSTRAINT PK_Facility
PRIMARY KEY (facilityID);
ALTER TABLE LandUse ADD CONSTRAINT PK_LandUse
PRIMARY KEY (landUseID);
ALTER TABLE QualityReport ADD CONSTRAINT PK_QualityReport
PRIMARY KEY (qualityReportID);
ALTER TABLE Road ADD CONSTRAINT PK_Road
PRIMARY KEY (roadID);
ALTER TABLE RoadSegment ADD CONSTRAINT PK_RoadSegment
PRIMARY KEY (roadSegmentID);
ALTER TABLE Facility ADD CONSTRAINT not within
FOREIGN KEY (roadID) REFERENCES Road (roadID);
ALTER TABLE QualityReport ADD CONSTRAINT FK_QualityReport_District
FOREIGN KEY (districtID) REFERENCES District (districtID);
ALTER TABLE QualityReport ADD CONSTRAINT FK_QualityReport_Facility
FOREIGN KEY (facilityID) REFERENCES Facility (facilityID);
ALTER TABLE QualityReport ADD CONSTRAINT FK_QualityReport_LandUse
FOREIGN KEY (landUseID) REFERENCES LandUse (landUseID);
ALTER TABLE QualityReport ADD CONSTRAINT FK_QualityReport_Road
FOREIGN KEY (roadID) REFERENCES Road (roadID);
ALTER TABLE Road ADD CONSTRAINT FK_Road_District
FOREIGN KEY (districtID) REFERENCES District (districtID);
ALTER TABLE RoadSegment ADD CONSTRAINT composedOf
FOREIGN KEY (roadID) REFERENCES Road (roadID);
```

A.3 Function that evaluates the positional accuracy at table level

```
-- Function: total_positional_accuracy(character varying, character varying,
character varying, character varying, character varying, character varying,
character varying, character varying)

-- DROP FUNCTION total_positional_accuracy(character varying, character
varying, character varying, character varying, character varying,
character varying, character varying, character varying);

CREATE OR REPLACE FUNCTION total_positional_accuracy(vartable1
character varying, varcol11 character varying, varcol12 character
varying, varcolref1 character varying, vartable2 character varying,
varcol21 character varying, varcol22 character varying, varcolref2
character varying) RETURNS real AS
$BODY$
DECLARE
    curs1 REFCURSOR;
    curs2 REFCURSOR;
    curs3 REFCURSOR;
    curs4 REFCURSOR;
    curs5 REFCURSOR;
    curs6 REFCURSOR;
    x1Coordinate double precision;
    y1Coordinate double precision;
    x2Coordinate double precision;
    y2Coordinate double precision;
    p_tname character varying;
    p_refname character varying;
    sum1 real := 0;
    sum2 real := 0;
    counter1 real := 0;
BEGIN
    OPEN curs1 FOR EXECUTE ’SELECT ’ || quote_ident(varCol11) || ’ FROM ’
    || quote_ident(varTable1);
    OPEN curs2 FOR EXECUTE ’SELECT ’ || quote_ident(varCol12) || ’ FROM ’
    || quote_ident(varTable1);
    OPEN curs3 FOR EXECUTE ’SELECT ’ || quote_ident(varCol21) || ’ FROM ’
    || quote_ident(varTable2);
    OPEN curs4 FOR EXECUTE ’SELECT ’ || quote_ident(varCol22) || ’ FROM ’
    || quote_ident(varTable2);
  OPEN curs5 FOR EXECUTE ’SELECT ’ || quote_ident(varcolref1) || ’ FROM ’
    || quote_ident(varTable1);
 OPEN curs6 FOR EXECUTE ’SELECT ’ || quote_ident(varcolref2) || ’ FROM ’
    || quote_ident(varTable2);
LOOP
   -- some computations
FETCH curs1 INTO x1Coordinate;
FETCH curs2 INTO y1Coordinate;
FETCH curs3 INTO x2Coordinate;
FETCH curs4 INTO y2Coordinate;
FETCH curs5 INTO p_tname;
FETCH curs6 INTO p_refname;
    counter1 := counter1 + 1;
    IF  p_tname = p_refname THEN
    sum1:= sum1 + power(abs(x1Coordinate-x2Coordinate),2);
    sum2:= sum2 + power(abs(y1Coordinate-y2Coordinate),2);
    exit;
```

```
     END IF;
  END LOOP;
    CLOSE curs1;
    CLOSE curs2;
    CLOSE curs3;
    CLOSE curs4;
    CLOSE curs5;
    CLOSE curs6;
    RETURN sqrt((sum1 + sum2)/counter1);

END;
$BODY$
  LANGUAGE 'plpgsql' VOLATILE
  COST 100;
ALTER FUNCTION total_positional_accuracy(character varying, character
varying, character varying, character varying, character varying,
character varying,character varying, character varying) OWNER TO mesued;
```

A.4 Function that evaluate positional accuracy at tuple level

```
-- Function: root_mean_square_error_individual(character varying,
character varying, character varying, character varying, character
varying, character varying, character varying, character varying)

-- DROP FUNCTION root_mean_square_error_individual(character varying,
character varying, character varying, character varying, character
varying, character varying, character varying, character varying);

CREATE OR REPLACE FUNCTION root_mean_square_error_individual(var_table
character varying, var_tname character varying, var_xcol character
varying, var_ycol character varying, var_reftable character varying,
var_refname character varying,var_refxcol character varying, var_
refycol character varying) RETURNS real AS
$BODY$
DECLARE
    curs1 refcursor;
    curs2 refcursor;
    curs3 refcursor;
    curs4 refcursor;
    curs5 refcursor;
    curs6 refcursor;
    xcoord real;
    xcoordRef real;
    xresidual real;
    ycoord real;
    ycoordRef real;
    yresidual real;
    p_tname character varying;
    p_refname character varying;

BEGIN
    delete from each_point_rmse_tmp; -- temporary table
    open curs1 FOR EXECUTE ’SELECT ’ || quote_ident(var_xcol) || ’ FROM ’
    || quote_ident(var_table);
    open curs2 FOR EXECUTE ’SELECT ’ || quote_ident(var_ycol) || ’ FROM ’
    || quote_ident(var_table);
 open curs3 FOR EXECUTE ’SELECT ’ || quote_ident(var_refxcol) || ’ FROM ’
    || quote_ident(var_reftable);
 open curs4 FOR EXECUTE ’SELECT ’ || quote_ident(var_refycol) || ’ FROM ’
    || quote_ident(var_reftable);
 open curs5 FOR EXECUTE ’SELECT ’ || quote_ident(var_tname) || ’ FROM ’
    || quote_ident(var_table);
 open curs6 FOR EXECUTE ’SELECT ’ || quote_ident(var_refname) || ’ FROM ’
    || quote_ident(var_reftable);

LOOP
   -- some computations
FETCH curs1 INTO xcoord;
FETCH curs2 INTO ycoord;
FETCH curs3 INTO xcoordRef;
FETCH curs4 INTO ycoordRef;
FETCH curs5 INTO p_tname;
FETCH curs6 INTO p_refname;
    IF  NOT FOUND THEN
        return 0;  -- exit loop
    END IF;
    xresidual:= pow(xcoordRef - xcoord,2);
    yresidual:= pow(ycoordRef - ycoord,2);
   if  p_tname = p_refname then
    insert into each_point_rmse_tmp values(p_tname, sqrt(xresidual +
    yresidual)); -- new result into the temporary table
    end if;
END LOOP;

    CLOSE curs1;
    CLOSE curs2;
    CLOSE curs3;
    CLOSE curs4;
    CLOSE curs5;
    CLOSE curs6;
    RETURN 1;
```

```
END;
$BODY$
  LANGUAGE 'plpgsql' VOLATILE
  COST 100;
ALTER FUNCTION root_mean_square_error_individual(character varying,
character varying, character varying, character varying, character
varying, character varying, character varying, character varying)
OWNER TO mesued;
```

A.5 Function that evaluate the error of omission at attribute level

```
-- Function: completenessbytable(character varying)

-- DROP FUNCTION completenessbytable(character varying);

CREATE OR REPLACE FUNCTION completenessbytable(vartable character
varying) RETURNS double precision AS
$BODY$
DECLARE
    curs1 REFCURSOR;
    colName character varying;
    resultPer double precision;
BEGIN
delete from tmp;
OPEN curs1 FOR EXECUTE 'select column_name from information_schema.
columns where table_schema = ''public'' and table_name ='
|| quote_literal(varTable);
LOOP
FETCH curs1 INTO colName;
IF  NOT FOUND THEN
EXIT;  -- exit loop
END IF;
select completenessByCol(varTable, colName) into resultPer;
insert into tmp values(colName, resultPer);
END LOOP;
CLOSE curs1;
return 0;
END;
$BODY$
  LANGUAGE 'plpgsql' VOLATILE
  COST 100;
ALTER FUNCTION completenessbytable(character varying) OWNER TO mesued;
```

Appendix B

B.1 Dataset's specification

B.1.1 Dataset's specifications

A data product specification defines the requirements for a data product. It forms the basis for producing data. It may also help potential users to evaluate the data product to determine its fitness for use [20]. The information contained in a data product specification is different from that contained in metadata, which provides information about a particular physical dataset. Information from the data product specification may be used in the creation of metadata for a particular dataset that is created in conformance with that data product specification. Thus metadata describes how a dataset actually is, whilst a data product specification describes how it should be. The relationship between a data product specification and metadata can be shown as in Figure 3.2.
According to [20] the specification of a data product shall include a description of its scope, which may be restricted in terms of spatial or temporal extent, feature types, and properties included, spatial representation, or position within a product hierarchy. In addition the data quality of the product specification shall identify the data quality requirements for the data product in accordance with ISO 19113. This shall include a statement on acceptable conformance quality levels and corresponding data quality measures as defined in ISO/TS 19138. Preparation of detail specification of spatial data is beyond the scope of this project. In this project more emphasis is given for the data quality part of product specification. General information about the creation of the dataset specification for the project is mentioned in "Appendix A: Dataset specification ". **Title**: Afar National Regional State Spatial dataset
Reference date: January 7, 2010-01-09
Responsible party: Afar National Regional State, Finance and Economic Development Bureau, Afar Region (Ethiopia)
Language: English
Format: Vector data
Distribution format: PDF
Purpose: The main purpose of this spatial data is to provoke different research ideas by showing the distribution of infrastructure in different part of the region. The research result, in turn, helps planners and policy makers to tackle different development issues.

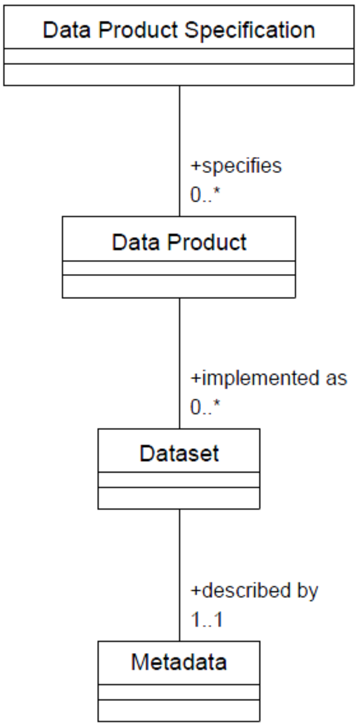


Figure B.1: Relationship of data product specification to metadata (20)

Spatial extent:
Upper left X coordinate = 483370m
Upper left Y coordinate = 1610033m
Lower right X coordinate = 886601m
Lower right Y coordinate = 988473m
Temporal extent: 2002 – present; update yearly.
Data definitions for the spatial data prepared to use through out the thesis project is as follows:

- Regional database: This is an abstract class an aggregation of the regional road network, districts, and facility datasets. The abstract class consists of the overview quality elements lineage, usage and purpose as an attribute. The quality elements positional accuracy and completeness as an operation for the abstract class. The quality element information in this class is at dataset level but store as an attribute in this abstract class.

- Road is the basic socio economic infrastructure. It is the means to the country for international linkage. It is represented as MULTILINESTRING geometry in the regional spatial data.

- Facility is in the context of the region consists of school, water, health, and animal clinics. It is represented as POINT geometry in the regional spatial data. All the facility points in the database are not within the road network.

- District is the smallest administrative area. It is represented as POLYGON geometry in the regional spatial data. In the region there are 29 districts. Districts are not overlap to each other, 1 or more facilities should lie within the districts and each district has road.

B.2 Overview quality elements

Road

Lineage: This map depicted the road density of the region
Source map: Central Statistical Authority (CSA), 1:2,000,000
Processing involved: Digitizing (polygon boundaries and database),Polygon formation (topology formation),Creating shaded relief from the Digital Terrain Model (DTM) and preparing the layout for printing.
Usage: Road network, in the context of the region, is the basic socio economic infrastructure. It is the means to the country for international linkage, i.e. for import and export.
Purpose: for transportation feasibility study.

Facility

Lineage: The dataset consists of facility distribution of the region.
Source map: Afar National Regional State Finance and Economic Development Bureau.
Processing involved: Point data for each site collected by GPS. Then transformed to map source software as UTM reading from GPS. Then exported as text. Finally imported to Arcview as points.
Usage: To analysis the socio economic activity of the region and identify the gap between the districts of t he region.
Purpose: To use for planning purpose

District

Lineage: The dataset consists of 29 districts of the region.
Source map: Central Statistical Authority (CSA), 1:1000000
Processing involved: Digitizing (polygon boundaries and database),Polygon formation (topology formation).
Usage: To identify the natural resource distribution per district and use for planning.
Purpose: The dataset was developed to record information necessary for the administration of the area for the purpose of population.

Appendix C

C.1 Code for user interface

The code for the interface is consists of HTML, PHP, and javascript.

C.1.1 Codes for the open layer

```
<html><head>
<script src="http://openlayers.org/api/OpenLayers.js"></script>
<script type="text/javascript">
var Afar = new OpenLayers.Layer.WMS(
              "quality_Afar",
              "http://itcnt07/cgi-bin/mapserv.exe?map=//itcnt03/gfm/
              abdelkadir21031/www/quality/config.map&",{layers: "Afar",
              transparent: "true", format: "image/png"});
var facility = new OpenLayers.Layer.WMS(
              "quality_facility",
              "http://itcnt07/cgi-bin/mapserv.exe?map=//itcnt03/gfm/
              abdelkadir21031/www/quality/config.map&",{layers: "facility",
              transparent: "true", format: "image/png"});
var roadm = new OpenLayers.Layer.WMS(
              "quality_roadm",
              "http://itcnt07/cgi-bin/mapserv.exe?map=//itcnt03/gfm/
              abdelkadir21031/www/quality/config.map&",{layers: "roadm",
              transparent: "true", format: "image/png"});
var baseMap = new OpenLayers.Layer.WMS(
"OpenLayers_WMS",
"http://labs.metacarta.com/wms-c/Basic.py?",
{layers: "basic"}
);
function init(){
map = new OpenLayers.Map("myMapHolder");
map.addLayers([baseMap,Afar,roadm,facility]);
var bounds = new OpenLayers.Bounds(98,6,104,20);
map.addControl(new OpenLayers.Control.MousePosition());
map.addControl(new OpenLayers.Control.ScaleLine());
var bounds = new OpenLayers.Bounds(39.36,8.726,42.581,14.586)
map.zoomToExtent(bounds);
map.getLayersByName("quality_Afar")[0].setVisibility(false);
map.getLayersByName("quality_roadm")[0].setVisibility(false);
map.getLayersByName("quality_facility")[0].setVisibility(false);
}
function chooseMyLayer(varOpts) {
if (varOpts != null) {
queryLayername = varOpts.getAttribute("value");
// set the visiblity of the layers
map.getLayersByName("quality_Afar")[0].setVisibility(queryLayername.
toUpperCase() == "Afar".toUpperCase());
map.getLayersByName("quality_roadm")[0].setVisibility(queryLayername
== "roadm");
map.getLayersByName("quality_facility")[0].setVisibility(queryLayername
== "facility");
//set selected table name to the hidden text in another frame
parent.details.document.forms[0].selectedTblName.value = queryLayername;
}
}
</script>
<head>
<body onload="init()">
<form>
<?php
$selectedTable = $_POST['tableNames'];
$conn = pg_Connect("host=itcnt07.itc.nl dbname=data_quality user=mesued
password=moham2009");
$result = pg_query($conn, "select table_name from information_schema.
tables where table_schema='public' and table_type = 'BASE TABLE' and
(table_name in ('afar','facility','roadm'))");
if ($result) {
    echo "Table Name(s):&nbsp;";
echo "<select name='tableNames' onChange='chooseMyLayer(this);'>";
echo "<option value=''/>";
while($myrow = pg_fetch_assoc($result)) {
$v_value = $myrow['table_name'];
if ($v_value==$selectedTable) {
$isSelected = "selected";
}
echo "<option value='$v_value' " . $isSelected . " >$v_value</option>";
$isSelected = "";
}
}
echo "</select>";
?>
    <div id="myMapHolder"
  style="border: 1px solid red; position: absolute; left: 5px; top:58px;
```

```
    width: 500px; height: 600px; overflow: hidden;">
    </div>
    </form>
</body>
</html>
```

C.1.2 Codes for calculating the positional accuracy through the Web

```
<form action="positionalPanel.php" mothod="post">
<table>
<tr>
<th>Referencing Tables</th>
<td>
<?php
$selectedTable1 = $_REQUEST['tableNames1'];
$selectedColumn1 = $_REQUEST['colNames1'];
$selectedColumn2 = $_REQUEST['colNames2'];
$selectedColumPrim1 = $_REQUEST['colNamesPrim1'];
$conn = pg_Connect("host=itcnt07.itc.nl dbname=data_quality user=mesued
password=moham2009");
$result = pg_query($conn, "select table_name from information_schema.
tables where table_schema='public' and table_type = 'BASE TABLE' and
(table_name in ('tmap', 'tcheck'))");
if ($result) {
echo "<select name='tableNames1' onChange='document.forms[0].submit();'>";
echo "<option value=''/>";
while($myrow = pg_fetch_assoc($result)) {
$v_value = $myrow['table_name'];
if ($v_value==$selectedTable1) {
$isSelected = "selected";
}
echo "<option value='$v_value' " . $isSelected . " >$v_value</option>";
$isSelected = "";
}
}
echo "</select>";
if ($selectedTable1) {
echo "</td><th>Column(s)</th><td>";
$conn = pg_Connect("host=itcnt07.itc.nl dbname=data_quality user=
mesued password=moham2009");
$result = pg_query($conn, "select column_name from information_schema.columns
where table_name='" . $selectedTable1 . "'");
echo "<select name='colNames1'>";
while($myrow = pg_fetch_assoc($result)) {
$v_value = $myrow['column_name'];
if ($v_value == $selectedColumn1) {
```

```
$isSelected = "selected";
}
echo "<option value=’$v_value’ " . $isSelected . " >$v_value</option>";
}
echo "</select>";
echo "</td><th>Column(s)</th><td>";
$result = pg_query($conn, "select column_name from information_schema.
columns where table_name=’" . $selectedTable1 . "’");
echo "<select name=’colNames2’>";
while($myrow = pg_fetch_assoc($result)) {
$v_value = $myrow[’column_name’];
if ($v_value==$selectedColumn2) {
$isSelected = "selected";
}
echo "<option value=’$v_value’ " . $isSelected . " >$v_value</option>";
}
echo "</select>";
echo "</td><th>Primary Column(s)</th><td>";
$result = pg_query($conn, "select column_name from information_schema.
columns where table_name=’" . $selectedTable1 . "’");
echo "<select name=’colNamesPrim1’>";
while($myrow = pg_fetch_assoc($result)) {
$v_value = $myrow[’column_name’];
if ($v_value==$selectedColumPrim1) {
$isSelected = "selected";
}
echo "<option value=’$v_value’ " . $isSelected . " >$v_value</option>";
}
}

?>
</td>
</tr>
<tr>
<th>Referenced Tables</th>
<td>
<?php
$selectedTable2 = $_REQUEST[’tableNames2’];
$selectedColumn3 = $_REQUEST[’colNames3’];
$selectedColumn4 = $_REQUEST[’colNames4’];
$selectedColumPrim2 = $_REQUEST[’colNamesPrim2’];
$conn = pg_Connect("host=itcnt07.itc.nl dbname=data_quality user=
mesued password=moham2009");
$result = pg_query($conn, "select table_name from information_schema.
tables where table_schema=’public’ and table_type = ’BASE TABLE’ and
(table_name in (’tmap’, ’tcheck’))");
if ($result) {
```

```
echo "<select name=’tableNames2’ onChange=’document.forms[0].submit();’>";
echo "<option value=’’/>";
while($myrow = pg_fetch_assoc($result)) {
$v_value = $myrow[’table_name’];
if ($v_value==$selectedTable2) {
$isSelected = "selected";
}
echo "<option value=’$v_value’ " . $isSelected . " >$v_value</option>";
$isSelected = "";
}
}
echo "</select>";
if ($selectedTable2) {
echo "</td><th>Column(s)</th><td>";
$conn = pg_Connect("host=itcnt07.itc.nl dbname=data_quality user=mesued
password=moham2009");
$result = pg_query($conn, "select column_name from information_schema.columns
where table_name=’" . $selectedTable2 . "’");
echo "<select name=’colNames3’>";
while($myrow = pg_fetch_assoc($result)) {
$v_value = $myrow[’column_name’];
if ($v_value==$selectedColumn3) {
$isSelected = "selected";
}
echo "<option value=’$v_value’ " . $isSelected . " >$v_value</option>";
}
echo "</select>";
echo "</td><th>Column(s)</th><td>";
$result = pg_query($conn, "select column_name from information_schema.
columns where table_name=’" . $selectedTable2 . "’");
echo "<select name=’colNames4’>";
while($myrow = pg_fetch_assoc($result)) {
$v_value = $myrow[’column_name’];
if ($v_value==$selectedColumn4) {
$isSelected = "selected";
}
echo "<option value=’$v_value’ " . $isSelected . " >$v_value</option>";
}
echo "</td><th>Primary Column(s)</th><td>";
$result = pg_query($conn, "select column_name from information_schema.columns
where table_name=’" . $selectedTable1 . "’");
echo "<select name=’colNamesPrim2’>";
while($myrow = pg_fetch_assoc($result)) {
$v_value = $myrow[’column_name’];
if ($v_value==$selectedColumPrim2) {
$isSelected = "selected";
}
echo "<option value='$v_value' " . $isSelected . " >$v_value</option>";
}
}
?>
</td>
</tr>
</table>
</form>
```

C.1.3 Codes for connecting the function positional accuracy and completeness through the Web

```
form method="post" action="details.php">
<input type="hidden" name="selectedTblName"/>
<script language='javascript'>
frm = parent.mapImage.document.forms[0];
if (frm != null) {
document.forms[0].selectedTblName.value = parent.mapImage.document.
forms[0]. tableNames.getAttribute("value");
}
objSelDq = document.forms[0].dataQu;
if (objSelDq != null) {
selDq = document.forms[0].dataQu.getAttribute("value");
if (selDq == 0) {
document.all.pos_panel.style.visibility = 'visible';
}
}
function doSubmit() {
selTableName = parent.mapImage.document.forms[0].tableNames.
getAttribute("value");
selDq = document.forms[0].dataQu.getAttribute("value");
if (selTableName != "") {
if (selDq != 0) {
parent.frames[1].document.forms[0].submit();
} else {
document.all.pos_panel.style.visibility = 'visible';
}
}
}
function doPositionalSubmit() {
selRefTblName = document.forms[0].tableNames1.getAttribute("value");
selTblRefName = document.forms[0].tableNames2.getAttribute("value");
if (selRefTblName != "" && selTblRefName != "") {
selRefColx = document.forms[0].colNames1.getAttribute("value");
selRefColy = document.forms[0].colNames2.getAttribute("value");
selRefColPrim = document.forms[0].colNamesPrim1.getAttribute("value");
selColRefx = document.forms[0].colNames3.getAttribute("value");
```

```
selColRefy = document.forms[0].colNames4.getAttribute("value");
selColRefPrim = document.forms[0].colNamesPrim2.getAttribute("value");
if (selRefTblName != "" && selTblRefName != "" && selRefColx != "" &&
selRefColy != "" && selColRefx != "" && selColRefy != "") {
parent.details.document.forms[0].submit();
}
}
}
</script>
<?php
$selectedTable = $_POST['selectedTblName'];
$selectedDataQu = $_REQUEST['dataQu'];
$conn = pg_Connect("host=itcnt07.itc.nl dbname=data_quality user=mesued
password=moham2009");
$result = pg_query($conn, "select id, descriptions from mam.data_qu");
if ($result) {
    echo "Applicable quality elements:&nbsp;";
echo "<select name='dataQu' onChange='doSubmit();'>";
echo "<option value='999'/>None</option>";
while($myrow = pg_fetch_assoc($result)) {
$v_id = $myrow['id'];
$v_desc = $myrow['descriptions'];
if ($v_id == $selectedDataQu) {
$isSelected = "selected";
}
echo "<option value='$v_id'" . $isSelected . ">$v_desc</option>";
$isSelected = "";
}
}
echo "</select>";
if ($selectedDataQu != "" and $selectedDataQu == 0) {
echo "<div id='pos_panel'>";
} else {
echo "<div id='pos_panel' style='visibility:hidden'>";
}
?>
<?php include("positionalPanel.php");?>
<input type="submit" value="Inquire Positional acurracy" onClick=
"doPositionalSubmit();"/>
</div>
<?php
if ($selectedTable != "" and $selectedDataQu != "") {
if ($selectedDataQu == 4) {
pg_query($conn, "select completenessbytable('" . $selectedTable . "')");
$result = pg_query($conn, "select * from tmp");
echo "<table width='100%' border='1'>";
$v_sum = 0;
```

```
echo "<tr><td>Field Name</td><td>Error of Omission per column in
Percentage</td></tr>";
$count =  pg_num_rows($result);
while($myrow = pg_fetch_assoc($result)) {
$v_field = $myrow['fieldname'];
$v_percentage = round($myrow['percentage'],2);
$v_sum = $v_sum + $v_percentage;
echo "<tr>";
echo "<tr><td>$v_field</td><td> $v_percentage</td></tr>";
echo "</tr>";
}
$mean = round($v_sum/$count,2);
echo "<tr><td>The overall error of Omission of the dataset is </td><td>$
mean</td></tr>";
if ($mean <= 15) {
echo "<tr><td>The dataset is pass.</td><td>Because the error of omission
is less than 15%</td></tr>";
} else {
echo "<tr><td>The dataset is failed.</td><td>Because the error of
omission is greater than 15%</td></tr>";
}
echo "</table>";
}
if ($selectedDataQu == 0) {
$refTblName = $_REQUEST['tableNames1'];
$refColx = $_REQUEST['colNames1'];
$refColy = $_REQUEST['colNames2'];
$refPrim = $_REQUEST['colNamesPrim1'];
$tblRefName = $_REQUEST['tableNames2'];
$colRefx = $_REQUEST['colNames3'];
$colRefy = $_REQUEST['colNames4'];
$primRef = $_REQUEST['colNamesPrim2'];
if ($refTblName != "" and $refColx != "" and $refColy != "" and
$tblRefName != "" and $colRefx != "" and $colRefy != "") {
$sqlQuery = "select mean_residual7('" . $refTblName . "', '" . $refPrim .
"', '" . $refColx . "', '" . $refColy . "', '" . $tblRefName. "', '" .
$primRef . "', '" . $colRefx . "', '" . $colRefy . "')";
pg_query($conn, $sqlQuery);
$result = pg_query($conn, "select * from each_point_rmse_tmp");
echo "<table width='100%' border='1'>";
$v_sum = 0;
echo "<tr><td>Name</td><td>The positional Accuracy of each tuple
in meter</td></tr>";
$count =  pg_num_rows($result);
while($myrow = pg_fetch_assoc($result)) {
$v_commonn = $myrow['commonn'];
$v_rmse = round($myrow['rmse'],2);
echo "<tr>";
echo "<tr><td>$v_commonn</td><td> $v_rmse</td></tr>";
echo "</tr>";
}
echo "</table>";
}
}
}
?>
</form>
```

C.1.4 The codes for final interface that users can interact with and display the output on the from

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Frameset//EN"
   "http://www.w3.org/TR/html4/frameset.dtd">
<HTML>
<HEAD>
<TITLE>Data quality parameter(s)</TITLE>
</HEAD>
<FRAMESET cols="43%, 57%" frameborder="0">
    <FRAME src="mapImage.php" name="mapImage" noresize="noresize"/>
    <FRAME src="details.php" name="details"/>
</FRAMESET>
</HTML>
```