Automated quality evaluation in the context of spatial data infrastructure

Amin Mobasheri

November, 2010

Automated quality evaluation in the context of spatial data infrastructure

by

Amin Mobasheri

Thesis submitted to the International Institute for Geo-information Science and Earth Observation in partial fulfilment of the requirements for the degree of Master of Science in Geo-information Science and Earth Observation, Specialisation: *Geoinformatics*.

Thesis Assessment Board

Thesis advisor	Dr. Ivana Ivánová Dr. Javier Morales
Assessment Board Chair	Dr. Rolf de By (chair)
Thesis examiners	Dr. Theodor Foerster

UNIVERSITY OF TWENTE.
FACULTY OF 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

Over the past decades, spatial data infrastructures have had a great development all around the world, with almost every country or union which have had a fruitful activity in GIS-related topics, nowadays have constructed their own spatial data infrastructure. The most important outcomes of this technology are the ability to make connection between geo-services, interoperability and harmonization and also to share data in a world wide domain. These shared data are produced and disseminated by big organizations that are responsible for geoinformatics activities or by individual groups that work or research in this field. The level of quality that these datasets conform, plays an important role in their reliability for use in projects. This research aims to provide an automated quality evaluation webservice to evaluate the quality of datasets in Spatial Data Insfrastructure (SDI).

This webservice uses a standard process flow model for spatial data quality evaluation. The process of spatial data quality evaluation is a set of connected activities for producing data quality result, and moreover, to fulfil the quality requirements defined by the costumers. The quality evaluation must be done in a consistent manner in order to determine whether the achieved quality level meets the requirements.

There exist several quality standards for evaluating the quality of datasets. In this research quality evaluation is based on ISO 19100 series of standards for geographic information. ISO 19113 defines quantitative and qualitative data quality elements used in performing quality evaluation. The possibility of the data quality elements for automated quality evaluation are discussed. Based on suitable data quality elements selected for automated quality evaluation, a process flow model for quality evaluation is designed.

Finally, by implementing the designed model of quality evaluation in a web service and testing it, conclusion based on validating the model of quality evaluation and automated quality evaluation webservice is discussed.

Keywords

geographic information, spatial data quality, quality evaluation, web service, spatial data infrastructure, business process modeling

Acknowledgements

This research is in good part the result of the efforts and discussions with my two supervisors: Dr. Ivana Ivánová and Dr. Javier Morales. I express my most sincere appreciation for all the time they dedicated to me during the last six months and for all their advice and orientation when I was doing research.

I would like to express my special thanks to Dr. Ali Mansourian from K.N.Toosi University of technology for helping me with his precious advises.

Also, I am greatly thankful to JKIP coordinators: Dr. Behzad Vosooghi and Mr. Huurneman because of their support during the course.

I extend my heartfelt thanks to my father, my mother, and my three sisters: Bahareh, Azadeh, and Shadab. They have contributed greatly in my career here by giving advises, and bequeathing love and moral support.

Finally my deepest gratitude goes to my love Noushin, who has always inspired me in my life.

Contents

List of Figures

Chapter 1

Introduction

1.1 Motivation and problem statement

1.1.1 Motivation

Over the past decades, Spatial Data Infrastructures have had a great development all around the world, with almost every country or union which have had a fruitful activity in GIS-related topics, nowadays have constructed their own infrastructure. The most important outcomes of this technology are the ability to make connection between geo-services, interoperability and harmonization and also to share data in a world wide domain. These shared data are produced and disseminated by big organizations that are responsible for geoinformatics activities or by individual groups that work or research in this field. These organizations and individuals are so-called spatial data infrastructure (SDI) nodes. SDI nodes are one of the main parts of the SDI Network. [5] defined SDI as "a collaborative network of system and human actors that exploit contributed data and computational resources, many of which are spatially explicit, for one or more targeted objectives, making use of service offerings and consumptions". The above-mentioned definition leads us to several issues such as policies, standards, human resources, data, and services that must be considered in SDI. Quality for all of the mentioned issues is important, but this research will focus on quality of data in SDI network. One might ask about the importance of quality in SDI, the answer is that geographical information is often used for problem solving and decision making. So, the reliability of outcomes which is mandatory for such purpose is based on the fitness for use and quality of the dataset itself as well as on its interoperability with other data sources [7]. Another reason which makes quality important is that most successful technologies are those that give costumers what they want. Satisfied customers are loyal to those suppliers which they feel best understand their requirements.
In addition, several organizations or individuals exist which find a dataset for their project but do not know if it fits for their purpose or not. Also, they can not find any comprehensive and complete software for spatial data quality evaluation. There exist functionality in some softwareś e.g. ESRI or Intergraph products which can be used for this purpose, but they have some disadvantages. For example they are platform-dependent software, and expensive. Apart from

cost, most customers have low level of knowledge in information quality, and its importance. The solution is to prepare a tool for customers so they can evaluate the quality of their data without having software or hardware knowledge. For evaluating quality information in the context of SDI, based on the idea that the data are transmitted via web servers in SDI, and also based on the quality and metadata standards defined by ISO and Open Geospatial Consortium (OGC) web service standards, one solution is that each organization should design a quality information evaluation model for its own, or the other solution is to do the evaluation process by a web service. Web services have several benefits such as being standard-based, interoperable, and available.

1.1.2 Research problem

the aim of this research is to implement a web service for spatial data evaluation to solve the previous mentioned problems and defects. Up to now this aspect has not been sufficiently considered.

1.2 Research Identification

1.2.1 Research hypothesis

An automated quality evaluation web service for spatial data can facilitate the process of spatial data quality evaluation in SDI nodes.

1.2.2 Research objectives

The main objective of this research is to design a prototype quality evaluation web service in SDI. The following are sub-objectives related to the objective:

- To analyze the selected quality elements suitable for automated quality evaluation.
- To Study about available standards of spatial web services.
- To design the process for evaluating spatial data quality in web services.
- To design a web service that automatically evaluates different aspects of quality in spatial data on the internet.
- To validate the designed service.

1.2.3 Research questions

The questions related to research include:

- What quality measures are suitable for evaluating quality of spatial data in web services, and what are the steps for quality evaluation?
- Which standards should be used for web service implementation?

- What different aspects should be considered in implementation of quality evaluation web service?
- What specifications and characteristics does this web service need for being automatic?
- To what extent can the web service satisfy users need for spatial data quality evaluation?
- What are the difficulties and problems for making this quality evaluation web service operational as a SDI node?

1.2.4 Innovation aimed at

It is the first time that a web service is going to be implemented based on ISO standards for spatial data quality evaluation in SDI.

1.3 Method adopted

This research is a technological research and is broken down into four phases. The phases include:

- Literature review:
  In this step the aim is to fully understand the concepts involved in the research topic and to evaluate previous related works for discovering new ideas.
- Design of the automated quality evaluation process:
  In this step, it is assumed that different aspects of spatial data quality has been considered and selected for the evaluation process. Output of this phase is the process which shows appropriate sequence of steps that must be taken to evaluate quality automatically.
- Implementing the automated web service:
  In the third step, the web service is going to be implemented based on the process workflow designed before. The aim is to run the web service as a SDI-node on the World Wide Web.
- Test and prove:
  This phase includes the steps that are going to be taken for testing the automated web service. For this issue, several different kind of spatial data are going to be evaluated via the web service, and the results are going to be discussed.

1.4 Thesis outline

Chapter 2 introduces the main terms and definitions involved in the domain of spatial data quality. Also, main definitions of this research, including quality evaluation procedure, automated quality evaluation, and different types of

users are discussed.
In chapter 3, the process flow model is designed and each sub-process of this model is discussed thoroughly. Also, the communication of quality information to user is covered in this chapter.
Chapter 4 presents the implementation of automated quality evaluation process flow model, and its validation. Finally, in chapter 5, the results of this research is discussed, and conclusions are made. This chapter finishes with the recommendation for future research.

Chapter 2

Spatial data quality and its evaluation

This chapter starts with defining main terms involved in spatial data quality (SDQ), and its evaluation. After this, different users of SDQ, and their use are briefly discussed. The quality evaluation procedure, the methods used in it, and different ways for communication of quality evaluation to specified users takes place in this chapter. Finally, different types of quality evaluation based on level of human interference in performing the evaluation procedure is discussed.

2.1 Spatial Data Quality

This section starts with defining the "quality", and "data quality" terms. After that, definitions of spatial data quality elements and sub-elements are reviewed. Finally, descriptors of data quality elements are mentioned to bring more details about the sub-elements and measures used in quality evaluation.

2.1.1 Data quality definitions

Originally, the term "quality" comes from the Latin "qualis" meaning "of what kind" [7]. It can be rephrased as "what is it?" or "is there?". ISO 9000 defines quality as "Degree to which a set of inherent characteristics fulfills requirements" [2]. Also, American society for quality defines quality as "A subjective term for which each person has his/her own definition [25]. According to them, In technical usage quality can have two meanings:

- The characteristics of a product or service that bear on its ability to satisfy stated or implied needs;
- A product or service free of deficiencies."

Both last definitions refer to requirement as need or expectation. This is the main definition for quality in this research, too. In this research quality is defined as a conditional and fully subjective attribute. Based on different requirements that people have it may be understood differently.

Generally, based on understanding of what quality means, the definition of data quality varies. [24] defines data quality as "the appropriateness and integrity of information collected and used in an assessment or evaluation". While others define it as the concern which data is missing or incorrect [20]. some literatures define data quality as features and characteristics of data that bear on its ability to meet the needs and requirements of the user. [[3], [4]]. The same definition of data quality is understood in this research. Referring to the last definition of data quality, the degree which dataset meets the requirements of its specific user implies the degree of its quality. No matter how many incorrect or missing values might be in the dataset, if it is not against users needs then the dataset might still be considered as an acceptable level of quality. This is the main definition of data quality based on "fitness for use". [23] defines data quality as "a concept that includes a number of attributes that contribute to the usefulness of the data from the perspective of the users". In this research, the data quality evaluation procedure is based on this definition of data quality. Generally, several authors categorize quality into two main groups: internal quality and external quality. Internal quality refers to products that are free from errors [7]. It refers to the degree of similarity which exists between the data produced and the "perfect" data. These perfect data are often called "nominal ground" [7] or "universe of discourse" [14]. In practice, nominal ground is not used for the internal quality evaluation, but a dataset of greater accuracy than the dataset which is called "reference data" is used instead. Reason for this is that nominal ground has no real physical existence. Internal quality evaluation itself includes an external and an internal part. Internal quality can be described by using different criterion. Different criteria has been defined by the main standards in geomatics which will be discussed later in next section. On the other hand, external quality is related to products that meet users needs [7]. The concept of external quality refers to the degree of concordance between a product and user needs, in a given context. This concept implies that quality is not absolute and the same product can have different quality for different users. External quality is often defined as "fitness for use". Its evaluation can imply criteria that describe internal quality. To evaluate whether a dataset meets our needs, we can check to see if the data represent the territory required at an appropriate date include necessary objects and attributes, but also, if the data have sufficient spatial accuracy or completeness, etc.[7].

2.1.2 Data quality elements and sub-elements

Generally, the quality of a dataset can be described using data quality elements and data quality overview elements. Data quality element is a "quantitative component documenting the quality of data set" [14]. According to [14] international standard five quantitative data quality elements exist which are:

- Logical consistency
- Completeness
- Positional accuracy

- Temporal accuracy
- Thematic accuracy

In addition, data quality sub element is a "component of data quality element describing a certain aspect of that data quality element"[14]. For the data quality elements identified above, definition of each element and it data quality sub-elements are followed.

Logical consistency

Logical consistency is defined in [14] as the degree of conformance to logical rules of data structure, attributes and relationships. For this data quality element, four data quality sub-elements are defined to describe the qualitative quality of a dataset which includes: conceptual consistency, domain consistency, format consistency, and topological consistency[14].

- Conceptual consistency:
  [14] defines conceptual consistency as "adherence to rules of conceptual schema". Thus, a dataset is conceptually consistent at the logical level, if it respects the conceptual schema; the structural characteristics of the selected data model[7].

- Domain consistency:
  [14] defines domain consistency as "adherence of values to the value domains".

- Format consistency:
  Format consistency is defined in [14] as "the degree to which data is stored in accordance with the physical structure of the data set".

- Topological consistency:
  Topological consistency is defined in [14] as "correctness of the explicitly encoded topological characteristics of a data set".

Completeness

Completeness is defined in [14] as the availability or non-availability of features, their attributes and relationships. The important goal for measuring completeness of data is to find out that what does and what does not belong to the dataset. In other words, "completeness is an attribute that describes the relationships between objects represented in a dataset and is an abstraction of the same set of objects in the real world"[22]. Its sub-elements include:[14]

- Commission: extra data present in a dataset.
- Omission: data absent from a dataset.

Positional accuracy

Positional accuracy is simply defined in [14] as the accuracy of positions of features within the dataset. It has three sub-element which are:[14]

- Absolute accuracy:
  Absolute positional accuracy is defined as the accuracy of sample coordinate values by considering the reference coordinate values on same coordinate system. [28]

- Relative accuracy:
  relative accuracy is the accuracy of scaled distances between sampled data points on features (for example, building corners), in comparison with the distance measured between the same points on the ground.[28]

- Gridded data position accuracy:
  Grided data position accuracy is defined in [14] as the "closeness of gridded data position values to true values".

In case of three dimensional data, there exist two other kinds of positional accuracy which are vertical (altimetric), and horizontal (planimetric) positional accuracy [7].

Temporal accuracy

Temporal accuracy is the accuracy of the temporal attributes and temporal relationships of features[14]. Its sub-element include:[14]

- Accuracy of a time measurement: correctness of the temporal references of an item.

- Temporal consistency: correctness of ordered events or sequences.

- Temporal validity: validity of data with respect to time.

Thematic accuracy

Thematic accuracy is defined in [14] as "accuracy of quantitative attributes and the correctness of non-quantitative attributes, and of the classification of features and their relationships". Its sub-elements are:[14]

- Classification correctness: comparison of the classes assigned to features or their attributes to a reference dataset.

- Non-quantitative attribute correctness: correctness of non-quantitative attributes.

- Quantitative attribute accuracy: accuracy of quantitative attributes.

By use of data quality elements and sub-elements the degree of which the dataset meets the criteria set in user requirements, could be described. In

addition, data quality overview elements are elements used to describe non-quantitative quality of a dataset. [14] mentions three overview elements which are: purpose, usage, and lineage. Purpose describes the principles for creating a data set and contains information about its application use in the future. On the other hand, usage describes the applications for which a data set has been used. It describes uses of the dataset by the data producer. Note that the dataset's intended use (usage) is not necessarily the same as its actual use (usability), but in some cases it could be the same. Finally, lineage shall describe the history of a data set[14].

2.1.3 Descriptors of a data quality sub-element

[14] lists seven descriptors of a data quality sub-element in order to record information for each applicable data quality sub-element which includes:

- data quality scope
- data quality measure
- data quality evaluation procedure
- data quality result
- data quality value type
- data quality value unit
- data quality date

Data quality scope

A data quality scope can be defined as a suitable portion of a dataset which can fulfill user's requirements. [15] defines data quality scope as "extent or characteristics of the data for which quality information is reported". More specifically we might have three different types of scope which are named: spatial extent, object-based and complex scope. In spatial extent scope, the idea is to consider a smaller set of data inside the whole data-set limited by one or more boundary(ies). This boundary can be a rectangle which is defined by combination of two latitudes and two longitudes. Object based scope is another kind of scope definition which special objects and their attributes are desired data for the user. An example of this scope could be the objects labeled as roads in the dataset. Simply, this means that other data are ignored and the evaluation of quality is performed only for the selected objects. Finally, in complex type, combination of specific objects inside desired spatial extent(s) are defined as the data quality scope. A simple example for this kind of scope definition is the objects labeled as roads within specific boundaries. Thus, only a sub-set of a dataset which have these characteristics will be considered during quality evaluation procedure.

Data quality measure

Generally, data quality measure is the "evaluation of a data quality sub-element"[14]. For each data quality sub-element, several data quality measures can be defined in order to perform the evaluation of that specific data quality sub-element. For example the number of incorrect values and the ratio of incorrect values of an attribute are two data quality measures used for evaluating the quality of data by means of domain consistency check. Each data quality measure has some standard components which are briefly introduced in table2.1.

Line	Component	Description	Obligation/condition
1	Name	Name of the data quality measure applied to the data	M
2	Alias [a]	Another recognised name, an abbreviation or a short name for the same data quality measure	O
3	Data quality element	Name of the data quality element for which quality is reported	M
4	Data quality subelement	Name of the data quality subelement for which quality is reported	M
5	Data quality basic measure	Name of the data quality basic measure from which the data quality measure is derived	C/if derived from basic measure
6	Definition	Definition of the fundamental concept for the data quality measure	M
7	Description	Description of the data quality measure including all formulae and/or illustrations needed to establish the result of applying the measure	C/if the definition is not sufficient for the understanding of the data quality measure concept
8	Parameter [a]	Auxiliary variable used by the data quality measure including its name, definition and optionally its description	C/if required
9	Data quality value type [a]	Value type for reporting a data quality result	M
10	Data quality value structure	Structure for reporting a complex data quality result	O
11	Source reference [a]	Reference to the source of an item that has been adopted from an external source	C/if an external source exists
12	Example [a]	Illustration of the use of a data quality measure	O
13	Identifier	Integer number, uniquely identifying a data quality measure	C/if data quality measures are administered in a register

[a] Multiple entries are allowed. When values for the optional or conditional elements are not present, this should be indicated by assigning the character "—" to the appropriate component.

Table2.1. Components defining a data quality measure (taken from [17])

One of the main component of data quality measure is data quality basic measure. Data quality basic measure is "generic data quality measure used as a basis for the creation of specific data quality measure" [17]. Data quality basic measures are abstract data types and cannot be used directly in data quality report. Each data quality basic measure is described by its name, definition and value type. The main reason of introducing data quality basic measure is to avoid the repetitive definition of the same concept. There exist several data quality measures which have common characteristics. For example, all data quality measures that are dealing with counting the number of errors. There exist two different types of data quality basic measures. The first one deals with counting the number of errors or correct items, while the second kind of basic measures are based on the concept of modeling the uncertainty of measurements with statistical methods, respectively. Based on the discussion which will be made later in this chapter, this research deals with counting re-

lated data quality basic measures. Table2.2 shows the list of data quality basic measures for count related data quality measures defined.

Data quality basic measure name	Data quality basic measure definition	Example	Data quality value type
Error indicator	Indicator that an item is in error	False	Boolean (if the value is true the item is not correct)
Correctness indicator	Indicator that an item is not in error	True	Boolean (if the value is true the item is correct)
Error count	Total number of items that are subject to an error of a specified type	11	Integer
Correct items count	Total number of items that are free of errors of a specified type	571	Integer
Error rate	Number of the erroneous items with respect to the total number of items that should have been present	0.0189 1.89% 11:582	Error rate can either be presented as Real, percentage or as ratio
Correct items rate	Number of the correct items with respect to the total number of items that should have been present	0.9811 98.11% 571:582	Correct items rate can either be presented as Real, percentage or as ratio

Table2.2. Data quality basic measures for count related data quality measures

(taken from [17]) Based on the fact that data quality basic measures are identified by their name, if a data quality measure is using one of the data quality basic measures then the name of the data quality basic measure should be provided otherwise it should be indicated that in this case a data quality basic measure is not applicable.

Data quality evaluation procedure

Data quality evaluation procedure is defined as "Operation(s) used in applying and reporting quality evaluation methods and their results[14]. As the definition mentions, a data quality evaluation procedure might use one or more data quality evaluation methods. Data quality evaluation methods are divided into two main groups: direct and indirect. In direct methods, data is compared with internal and/or external reference information in order to evaluate the data quality. Based on the source of the information required for evaluation, direct methods are subdivided into two classes: internal, and external. The data needed for performing an internal data quality evaluation method is internal to the data set being evaluated[15]. While external direct quality evaluation needs reference data external to the dataset. For example, performing a logical consistency test in means of format consistency of field type check is an internal direct quality evaluation method, because all data needed for such check is in the physical structure of the dataset itself. But a positional accuracy test requires a reference dataset which is an extra dataset. The last example is related to external direct quality evaluation methods. On the other hand, indirect quality evaluation methods are methods which use external knowledge as a basis for quality evaluation[15]. Example of an external knowledge is the dataset lineage, such as production method or source data. These methods are used only if direct evaluation methods cannot be used.

Data quality result

Data quality result refers to value(s) resulting from applying a data quality measure or the outcome of comparing the obtained value against a conformance quality level[14]. Conformance quality level is a threshold value(s) for data quality results, used to determine how well a dataset meets the user requirements[14].

Data quality value type

A data quality value type is always reported for each data quality result. Examples of common value types are Boolean, percentage, and ratio. For example, a data quality result of 75 with a value type of percentage reported for the data quality element and its data quality sub-element "logical consistency, domain consistency" is an example of a value resulting from applying a data quality measure to the data specified a data quality scope.

Data quality value unit

The value unit for reporting a data quality result is called a data quality value unit. It is not always applicable for data quality results, and is an optional property.

Data quality date

Simply, contains the date or series of dates which a data quality measure is applied to the dataset[14].

2.2 Users of Spatial Data Quality

Generally, different types of users might use spatial data quality based on their purpose. The first type of users are those who are expert in geographic information system (GIS) and spatial data quality (SDQ). In this case we expect them to know exactly what data quality element and sub-element are needed to be checked, and what quality conformance level is appropriate for each measure. The other type of users are not expert in GIS and SDQ, but still have acceptable knowledge related to these topics. In such situation they can select the data quality elements and sub-elements needed for performing the evaluation procedure, but they might not be able to determine the quality conformance level value. For solving such problem, as long as they are aware of their application, a default conformance level value can be suggested to them and the evaluation procedure can be performed based on that default conformance level value. The third type of users are called naïve users. Users which have no knowledge or experience in GIS related topics, but still have to use spatial datasets for their projects. In this case, some useful scenario cases can be suggested to them as an example, and the user can see which one is more related to his/her application. Based on that, default appropriate values for each component which is necessary for performing the quality evaluation procedure can be selected. The last

type of users are non-human users. More specifically, they are other services or computer programs which want to use spatial data quality for their application. Later, during chapter 3 and chapter 4 as the model and web service is designed and implemented, the details for interaction with different types of users would be discussed.

2.3 Data quality evaluation procedure and its process flow

In this research quality evaluation procedure is defined as matching the user requirements against the dataset itself, to see if the selected dataset is suitable for the users' needs. Later, we will see that in some cases external reference sources are necessary for handling the quality evaluation procedure. The process given in Figure2.1 represents the sequence of steps that should be taken for obtaining a quality result and reporting it. Furthermore each step of process and its related terms are defined.

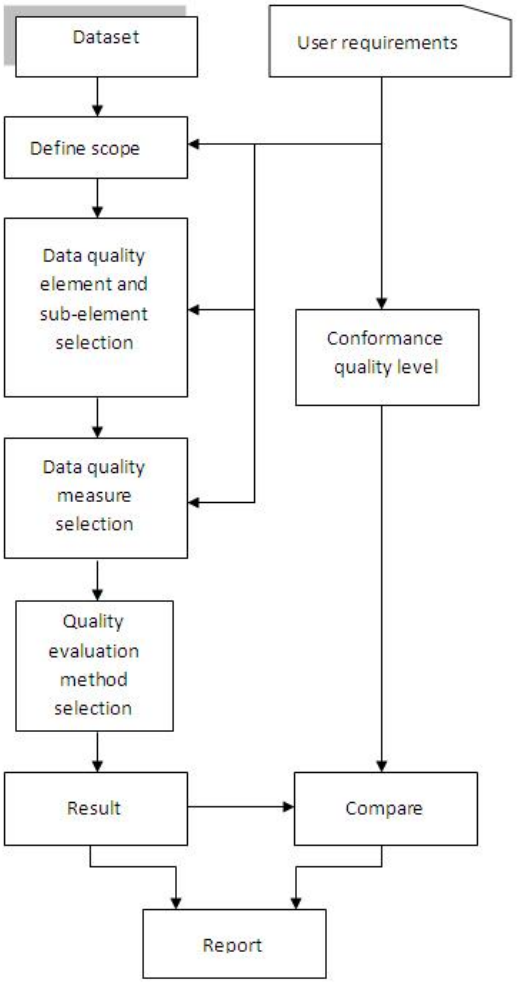

Figure 2.1: Data quality evaluation process flow (adopted from (15))

At first, we will have a brief overview of the process. After that each step of process flow will be explained thoroughly. As figure2.1 shows, the process begins with two main inputs which are the dataset and user requirements. User requirements can be considered as a file produced from the information given by user based on his/her desired needs. Its main properties include data quality scope, data quality element and sub-element, data quality measure, and quality conformance level. By considering the dataset itself and users needs, the process follows by defining each main property of user requirement, one by one. In the next step, for handling each data quality measure, a data quality evaluation method is chosen. After applying the methods, each evaluation will have its own results, and based on users requirements a comparison between the evaluation results and its related quality conformance level would be performed to conclude information about the "fitness for use" of the dataset and report it in an appropriate manner to the user.
More specifically, the process flow has some main components which include:

- Dataset:
  A dataset is defined as an "identifiable collection of data".[14] Later in chapter 4, the selected format of the dataset for this research would be discussed.

- User requirements:
  The user requirements mentioned in the main process flow, includes main component necessary for performing the quality evaluation based on the concept of quality discussed before: "fitness for purpose". These components include:

  - data quality element(s)
  - data quality sub-element(s), and some of its descriptors:
    * data quality scope
    * quality conformance level
    * data quality measure

  Specifically, the user requirements should be considered as a file containing one or more record(s). Each record contains values for beforementioned components, and is applied by a measure. The number of measures necessary for quality evaluation procedure is completely related to the users needs. More details about the user requirements, its format and implementation issues would be discussed in chapter 4.

- Quality evaluation method selection:
  After defining the user requirements, the model uses the values of its components to select an appropriate data quality evaluation method. For each record in user requirements file this task would be performed, and the method would be applied. The data quality result for each method would be passed to the next step of process for making the result. For example, a typical method used for omission check in means of completeness test is to compare street names in the database with another reference file. Another example is to check all records for appropriate range of dates which is a method used for performing temporal consistency check.

- Result analysis:
  Based on the data quality method chosen for each data quality measures, the result should have its own specific properties. Some of these properties were discussed previously as descriptors of a data quality sub-element, which include: data quality result, its value type and value unit. In addition, data quality date is another component which holds the date of the evaluation procedure performed. The result analysis phase takes care of gathering output data of data quality results performed for each data quality measure and using them for further analysis. There exist two possible type of result: pass/fail, and quantitative result. The type of result for each data quality measure completely depends on the components of

Figure 2.2: Result file - XML schema

UR file. Whenever a user defines a quality conformance level, the result of data quality method would be compared to its conformance level, and depending on its result, the result analysis phase can report either pass or fail. In other conditions, which a value for quality conformance level is not available, the result of the evaluation procedure can be reported quantitatively. In case of having more than one quantitative for the evaluation procedure, the result analysis phase uses an aggregation function to determine the final result. By having the values of conformance level for each measure, the final result of evaluation procedure is "pass" as long as result of all measures are "pass", otherwise the final result would be "fail". More specifically, the aggregation function uses the "AND" operator to aggregate the results. In cases which quality conformance level values are undefined, a set of quantitative results are passed to be reported in an appropriate manner. The result file produced in result analysis phase is an XML file. Figure2.2 shows the XML-schema of the result file.

- Report:
  In the final stage of the process flow, the final result of the quality evaluation procedure must be reported to the user in an appropriate way. Generally, the idea of reporting quality information as metadata is a standardized way. By this, the quality information could be used later by any other service. But in some special cases, with having aggregated result, in addition to stating result in metadata, a quality evaluation report should be returned to the user as an output. This report can carry extra detail about the evaluation properties, which is useful for naive users. This report can be designed in a tabular format.

- Sample scenario:
  Suppose that a team is working on a natural resource management project.

They need to handle analytical calculations of values for different attributes. So, they need to have spatial data which covers their region of interest. After searching the Internet, they find some datasets, but they are not sure of which one is more suitable for their project. Also, due to lack of quality information in metadata, and based on the fact that they do not have the ability for checking the quality of the datasets, they pass each dataset to the web service, and make their decision for dataset selection based on the quality information report produced by the web service. The following text is brief information about what is happening inside each step for this sample scenario.

- dataset:
  The dataset is either in XML format or is an output of a Web Feature Service (WFS)[30] in GML. In cases which the dataset in not in an XML-based format, user should use a GIS data conversion software to do the conversion first, and then pass it to the web service.
- User requirement:
  The User Interface (UI) is the main part of the web service interacting with user to receive users requirements. At the first step, the UI asks for the desired data quality element(s) and sub-element(s) which the quality check is going to be performed for them. After that, other mandatory components of user requirement described before would be asked such as scope, and quality conformance level. Suppose that in this example the user wants to find the best forests within a specific country that have only special kind of trees. In their case a domain consistency check in means of logical consistency should be performed for different scopes entered by user, to see data in which scope has less inconsistency. In this example inconsistency occurs whenever a value for a type of tree violates a specific domain of value defined by user. Table2.3 shows a possible value of user requirement for performing a check for quality evaluation. More information about the user requirement is discussed in chapter 3.

User requirement component	Example 1	Example 2
Data quality element	Logical consistency	Logical consistency
Data quality sub-element	Domain consistency	Domain consistency
Scope	All items classified as trees and bounded by longitudes -91.3 -91.4 and latitudes +40.0 +40.2	All items classified as trees and bounded by longitudes -91.5 -91.6 and latitudes +40.0 +40.2
Conformance level	undefined	undefined

Table2.3. The values of user requirement components for the sample scenario.(taken from [15])

In the next step, after data quality method selection, based on user requirements, the method is executed over the dataset, and the result

of it is saved in the result file. For example, a possible data quality method can be comparison of values of tree types to the selected types of tress defined by user. Table2.4 shows the possible values for result file.

Result component	Example 1	Example 2
DQ result value	8	8
DQ value type	Number	Percentage
DQ value unit	Attribute violations	Percent
Example DQ result value meaning	All items within scope in the dataset were checked and eight of the items have attributes that violate the attribute domain defined by user.	All items within scope in the dataset were checked and from each group of 100 items, eight of the items have attributes that violate the attribute domain defined by user.

Table2.4. The values of result file components for the sample scenario.(taken from [15])

After having the results of the quality evaluation method, the result analysis phase makes the final result by using the results and comparing them with quality conformance level values defined in user requirements. The final result would be reported in an appropriate manner. Information regarding communication of model to user for reporting quality information is discussed in chapter 3.

2.4 Automated quality evaluation

By considering the level of human interference in performing the evaluation procedure, three different cases would occur: non-automated, semi-automated, and automated quality evaluation. In case of non-automated evaluation, the procedure of selecting and applying the quality evaluation method is done manually by human. Example for this kind of quality evaluation is to perform thematic accuracy check by defining different classes, and using fieldwork data to make the confusion matrix and evaluate the overall accuracy of dataset. After growth of computer algorithms, the idea of handling the evaluation procedure by computer came in mind. Semi-automated evaluation is the case which the user still has direct interference with the evaluation procedure, and can decide to choose which method for quality evaluation should be used, but with getting help from a computer program controlled by an algorithm. Finally, automated quality evaluation means that the aim is to handle the quality evaluation procedure without direct interfere of human, and by an algorithm which uses a specific process. Generally, the process is defined as the whole steps which starts from a beginning point and leads us to the target point. Automated quality evaluation is most useful for naïve users and also other services that work with spatial data in Internet. Spatial data infrastructures (SDI) contain high quantity of spatial data available for use. An automated quality evaluation web service can act as a node in SDI to receive request from other services, and return the result of quality evaluation without need of human interaction.

For this issue, the available quantitative data quality elements are reviewed and their capability for automated/semi-automated evaluation is discussed. For those elements which are candidate of automation, the model of process flow for quality evaluation is designed and discussed in chapter 3.

2.4.1 Logical consistency

In section 2.2, logical consistency was defined as a quantitative data quality element. For logical consistency, four data quality sub-element should be reviewed which are:

Conceptual consistency

Generally, in database theory, there exist two different schemas: the physical schema, and the conceptual schema. The conceptual schema states how data appears to be organized from the user's point of view [27]. For example, in a relational database, by considering the conceptual schema, information about how data is organized into tables, and what the primary key or foreign key relationships between the tables are can be retrieved. This data quality sub-element is involved with the rules defined in conceptual schema. In spatial datasets, same as non-spatial datasets, the features and their relationships are defined in the conceptual schema of the dataset. For example, the feature class "house" cannot be positioned inside the feature class "lake". This is an example of a rule defined in the conceptual schema of the dataset. Violations against such rules can be counted as data inconsistency. Other examples of conceptual inconsistency can be invalid placement of features within a defined tolerance, duplication of features, and invalid overlap of features [17]. However in practice, not all rules are explicitly defined in the conceptual schema. This is because some rules are completely application dependent (not all overlapping surfaces are necessarily erroneous). In addition, a data model in GIS is mathematical rules for geographic object representations. For example, the vector data model represents geography as collections of points, lines, and polygons [13]. Conceptual consistency checks are completed automatically by software. The integrity constraints defined in the data model ensures that values of feature attribute, geometry and topology, database schema and file formats are valid [7]. So there is no need to perform conceptual consistency test. But, as mentioned before, in some special cases, the user may want to define some rules based on his/her project application. For performing the conceptual consistency check in case of user-defined constraints and logical rules, the model should have the capability to offer a tool for constraint definition to the user. Since, this is not an objective of this research it is suggested for future work.

Domain consistency

For quality evaluation in means of domain consistency test, the attributes of objects within a dataset should be compared against acceptable attribute domain and the values which are outside the domain are determined and counted as inconsistencies. Generally, a domain determines the acceptable attribute values. Whenever a domain is chosen for a field of an attribute, only the values within that domain can be entered into that field. Furthermore, two main properties of a data field should be checked which are field type and domain type [6]. Field type is the type of attribute field which can be set to any of the following:

- Short - short integers
- Long - Long integers
- Float - single-precision floating point numbers
- Double - Double-precision floating point numbers
- Text (coded domain only) - Alphanumeric characters
- Date - Date and time data

Note that field type check has overlap with checking the type of attributes in conceptual consistency, and since the model of conceptual consistency check was not covered in this research, the field type check is covered in domain consistency check. On the other hand, domain types are used for making different kinds of limitation for value choices. There exist two major kinds of domain types which are rang domains and coded domains [1]. A range domain in used for a numeric attribute and specifies a valid range of values that can be entered for the domain. Coded domains can be applied to any type of attribute-text, numeric, date and so on. They specify a valid set of values for an attribute. As mentioned in section 2.3, For data quality evaluation, data quality measures are used. [17] defines the data quality measures for domain consistency which include:

- Value domain non-conformance:
  Indication of if an item is not in conformance with its value domain.
- Value domain conformance:
  Indication of if an item is conforming to its value domain.
- Number of items not in conformance with their value domain:
  Count of all items in the dataset that are not in conformance with their value domain.
- Value domain conformance rate:
  Number of items in the dataset that are in conformance with their value domain in relation to the total number of items in the dataset.

- Value domain non-conformance rate:
  Number of items in the dataset that are not in conformance with their value domain in relation to the total number of items in the dataset.

For means of checking the domain consistency, no matter which data quality measure is selected the method for data quality evaluation is to compare the attribute of items in the dataset against acceptable attribute domain and based on five different data quality measures, five slightly different tasks are performed. For example, in case of value domain non-conformance data quality measure, the item which is not in conformance to the value domain is indicated. While in value domain conformance rate, the number of items in the dataset that are in conformance with their value domain in relation to the total number of items in the dataset is reported. Note that data quality measures are selected based on the information extracted from the user requirements.

Format consistency

The physical structure of the datset can be extracted from the datasets physical schema which is one of two schema's defined in database theory [1]. The physical schema indicates how data is stored in a file. For example, in a relational database, by considering the physical schema, information about what the data types of the field in a specific table are, can be extracted. Format consistency deals with the format and type of the fields that data is stored in. While the conceptual consistency was discussed it was mentioned that data models have constraints defined for the format of the fields inside the dataset. Softwares have the capability to ensure these integrity constraints. In special cases, based on user requirements, the user might want to define a specific structure and check the values inside fields to see whether they obey this structure. For example, Postal codes are defined as string fields in the data model, but except of that, one user might want to check and see if the postal code values obey a specific structure like [1234 AB]. In this example, all items which have a postal code field and do not obey this user-defined field structure are counted as inconsistencies. Two data quality measures for format consistency check is defined which are:[15]

- Physical structure conflicts:
  Counts of all items in the dataset which are stored in conflict with the physical structure of the dataset
- Physical structure conflict rate:
  Number of items in the dataset that are stored in conflict with the physical structure of the dataset divided by the total number of items

Topological consistency

Due to data measurement methods, and map generalization operators such as aggregation, displacement, and simplification, topological inconsistencies occur in spatial datasets. This is because these operators often reduce the shape

and structure of spatial objects [21]. There exist several methods focusing on topological consistency [[9],[10],[8][8]]. The main considerations in topological consistency check are check of polygon boundary closures, check of true connections in linear features (every arc of a network should be connected by a node to another arc), check of the topology and the spatial relationships, and check of polygon overlaps. The first two cases can be checked in means of automated evaluation, since every arc is stored in the database as a straight line connecting a start and end node, and each node has its own identifier. By checking positional values of nodes, the boundary closure check or network connectivity check can be performed. The same procedure can be applied for polygon overlaps.
For checking the topology and spatial relationships of features, topological rules should be defined and used. [19] defined four approaches for establishment of topological relationships between regions with each other, and also line/region relations. Apart from that, several other articles exist which define topological relationships of features [[9],[10],[9]]. Table2.5 shows common relationships between features, divided into scalar relation types and spatial relation types.

Relation Types
Scalar
Equals Relation
Not Equals Relation
Less Relation
Less Equals Relation
Greater Relation
Greater Equals Relation
Begins Relation
Ends Relation
RegExp Relation
Spatial
Spatial Equals Relation
Spatial Disjoint Relation
Spatial Intersects Relation
Spatial Touches Relation
Spatial Overlaps Relation
Spatial Crosses Relation
Spatial Within Relation
Spatial Contains Relation
Spatial Within Distance Relation

Table2.5 list of relation types adopted from [4]

Scalar relation types are relationships between two scalar values of a specific type. While spatial relation types correspond to the ISO/OGC simple feature specification spatial interaction types. Table2.6 lists the predicate types used in this research.

Predicate type
Relational Predicate
Exists Predicate
ForAll Predicate
Conditional Predicate
Referential Predicate
Range Predicate
And Predicate
Or Predicate
Not Predicate

Table2.6 list of predicate types adopted from [4]

It also uses predicate types that are same as a function using two types as input and an operator for check and return results. For example the RelationalPredicate is used to check whether two values have a defined relation. Specifically, it consists of a left value, a right value, and a comparison operator. In addition to predicate types, a list of value types is provided (Table2.7).

Value Type
Static Value
Dynamic Value
Temporary Value
Conditional Value
Aggregate Value
Built-in Function Value
Class Value
Summed Value
Difference Value
Product Value
Quotient Value
Modulus Value
Negated Value

Table 2.7. Value types adopted from [4]

Each value type has its own usage. For example, a StaticValue is a typed constant. Its value can be assigned explicitly within the rule expression, and it can be later used in other comparisons such as relational predicate. For means of automated evaluation, a formal language must be used to express the data consistency rules. Watson used Web Ontology Language (OWL)[29], as the language for expressing consistency rules, providing a simple example for representing a topological consistency test. Due to the complexity involved in defining consistency rules, and as long as it is not an objective of this research, checking topology and spatial relationships of features is excluded from the automated quality evaluation model designed in this research. Data quality measures identified by [17] which are appropriate for use in data quality methods by the model include:

- Number of missing connections due to undershoots:
  Count of items in the dataset, in the parameter tolerance, that are mismatched due to undershoots.
- Number of missing connections due to overshoots:
  Count of items in the dataset, in the parameter tolerance, that are mismatched due to overshoots.

2.4.2 Completeness

As mentioned in section 2.1.2, completeness is defined as errors of omission (measure of the absence of data), and errors of commission (measure of the presence of extra data)[28]. Completeness of a dataset can be suitable for a specific task but not for another. So, when completeness has to be measured the concept of fitness for use comes in mind. Generally, two types of completeness exists which are data completeness and model completeness [7]. Data completeness is the before-mentioned errors of omission and commission. It is measurable and independent of the application. Model completeness is defined as the "comparison between the abstraction of the world corresponding to the dataset and the one corresponding to the application, preferably evaluated in terms of fitness for use" [7]. Furthermore, data completeness contains both formal completeness and object completeness. Formal completeness concerns the data structure, adherence to the standards used, and presence of metadata [7]. Object completeness concerns about attribute and relationships of objects. Completeness monitors both omission and commission in information contained in geographic database by answering the following questions:[7]

- Is the number of objects modeled equal to the number of objects defined in the model?
- Do the modeled objects have the correct number of attributes and are all attribute values present?
- Are all entities represented in the reference data represented in the model?

The data quality measures identified by [17] for completeness check include:

- Excess item:
  Indication that an item is incorrectly present in the data.
- Number of excess items:
  Total number of items in the dataset, which should not have been in the dataset.
- Rate of excess items:
  Number of excess items in the dataset in relation to the total number of items that should have been present.

- Missing item:
  Indication that a specific item is missing in the dataset.
- Number of missing items:
  Count of all items that should have been in the dataset and are missing.
- Rate of missing items:
  Number of missing items in the dataset in relation to the number of items that should have been present.

2.4.3 Positional accuracy

Previously in section 2.1.2, positional accuracy was defined as the accuracy of coordinate values [28]. For performing the positional accuracy checks obtaining true values in the field work is needed. In cases when a field work cannot be performed, a reference dataset of the real world that has an accepted level of quality is used. Reference datasets are produced by spatial data providers. Generally, the values containing the position of objects are stored as a set of cardinal values in the dataset. For example, in field mapping a combination of three (X,Y,Z) values are used to store the position of an object, or in case of GPS position, latitude, longitude, and altitude is recorded [7]. These cardinal values allow the objects to be positioned in three-dimensional cartesian or polar coordinate systems. As long as positional accuracy is strictly related on the acquisition methods and processing of measurements [7], errors related to positional accuracy are most caused in the acquisition phase and data processing phase. There exist some measures for evaluating the positional accuracy, both absolute and relative, such as Root Mean Square Error (RMSE). Also, the nature of evaluating positional accuracy relies on the decision that user makes in sampling, measure selection, etc. Obviously, the only way to measure positional accuracy, both absolute and relative is to compare the dataset with a reference dataset [7]. In this research, since we are relying on "fitness for use" as the meaning of quality, and in case of positional accuracy test, each user might need different accuracy and precision for positional values of object in the dataset. Thus, it is based on users analyze and interaction. In addition, the reference dataset itself, which should be passed by users, is only produced by data providers, and would cost too much for naïve users to afford. Even in exceptional cases which a user has a reference dataset in hand, then he/she can make use of it, and there is no need to evaluate another dataset by it. Based on the mentioned issues, this data quality element and its sub-elements are excluded from the list of candidates for automated quality evaluation.

2.4.4 Temporal accuracy

In section 2.1.2 the concept of quality in this research was defined based on "fitness for use". Due to this issue, the date of data input, or the date of its update becomes an important factor [7]. Some users might want to use date and time information for their applications. Based on the type of feature, the

management of time related issues is different [7]. Some entities are updated at regular time durations such as aerial photographs. While others require historical management, like cadastral maps. This is the reason why the temporal aspect of features are treated in different manners, sometimes as a date, an interval, and sometimes as a temporal range [7]. Another important issue related to this topic is the concept of time. [7] distinguishes three different types of time concepts which are:

- Logical time:
  Indicates the actual date which the phenomenon took place in reality, as stored in the database.
- Time(date):
  The time that the feature was observed.
- Transactional time:
  The date which data was entered to the database.

In practice, the transactional time is often stored in the database, while the logical time is more important for users [7]. Temporal accuracy has three data quality sub-elements which are:[15]

- Accuracy of time measurement: correctness of the time references
- Temporal consistency: correctness of ordered events
- Temporal validity: validity of data with respect to time.

The temporal aspects of features are highly depended on the type of them, and the level of precision in measuring it [7]. This means that the correct interval for confirming the dataset validity is completely based on the features stored in it. Some concepts are applied to the temporal consistency between different features; complex entities require good temporal consistency. An example of this case is topological structure such as road networks. On the other hand, independent features like sign posts do not require it [7]. Evaluation of the accuracy of time measurement, and temporal consistency sub-elements are applicable whenever dealing with different aspects of feature types are possible, and accurate information of temporal references are available. Due to the fact that users do not have access to accurate references, the model designed in this research does not include evaluation of these sub-elements. Instead, the evaluation of temporal validity, which can be handled by internal direct data quality methods, is carried out in this research. The appropriate data quality measures necessary for quality evaluation procedure in means of temporal validity check are the same used for domain consistency check.

2.4.5 Thematic accuracy

Based on the definition of thematic accuracy is section 2.1.2, thematic accuracy can include attributes of feature classification, and change history attributes at

feature level. Usually, the percentage of correct attributes of a given type in a sample of dataset is referred to the attribute accuracy. An accuracy assessment can be used by map users to evaluate the "fitness for use" of the map. By the nature of different features, errors linked to different types of attributes follow different statistics [7]. There exist numerous articles published for this matter, but still the basic structures of a statistical accuracy assessment have not been fully described. [26] describe three basic components for assessing attribute accuracy which are the sampling design, the response design, and the estimation and analysis protocol. They also mention that the accuracy assessment should begin with defining the target population; the area represented by the land-cover map. Based on major decisions made by the sampling protocol, a sample of unit is selected from this population which is necessary for performing the accuracy assessment. Sample of unit is defined as "the link between a spatial location on the map and the corresponding spatial location on earth" [**?**]. This simply means that a rigorous statistical accuracy assessment needs an accurate reference dataset. There are also several image analysis software applications which provide functionalities for thematic classification, and evaluate the accuracy of the classified map mostly by using error matrix. These softwares need a dataset as reference and also need the users opinion for classification method selection, and also entering input for classification method parameters. [11] lists the problems in thematic accuracy assessment. Some of them are related to the accuracy measures used, difficulties related to sampling issues, types of errors and error magnitude, and accuracy of the reference dataset itself. Due to the existence of these problems , and also because of the nature of thematic accuracy check, and the role that the user has in determining the evaluation procedure parameters, thematic accuracy is excluded from the list of data quality elements which are suitable for automated quality evaluation means. Note that in some cases the determination of thematic accuracy is similar to completeness. Comparison of feature names and their descriptors can be considered in both thematic accuracy assessment and completeness. In this research such cases are covered by completeness test.

Chapter 3

Automated quality evaluation model

In this chapter, the model for automated quality evaluation, and all of its elements are discussed. The language of modeling is Business Process Modeling Notation (BPMN). According to BPMN is a standard for business process modeling. It provides a graphical notation for drawing the business process, and is based on a flowcharting technique very similar to activity diagrams in Unified Modeling Language (UML).The BPMN specification also provides a mapping between the graphics and the underlying constructs of execution languages. The execution language used for it is called Business Process Execution Language (BPEL). After defining the model schema, the process flow model for each data quality element and its related sub-elements that are considered as candidates for automated/semi-automated quality evaluation check is designed and discussed. Later in chapter 4, the process flow model designed for automated quality evaluation would be implemented in a web service.

3.1 Automated quality evaluation model schema:

The model designed for automated quality evaluation in BPMN is shown in AppendixA Figure1. The main inputs for the process are the dataset in GML format[12], and a file called "Information about requirements". The later file is not a real file, but it shows the interaction between the user and the web service via the user interface. This interaction is done for completing the User Requirement file. Note that this interaction is not a task which is performed at the first step of process, but instead is combination of tasks within the "Define User Requirement" sub-process. At the first, the process begins by receiving a start message from the user. Before sending this start message, the user interface takes care of receiving the dataset from the user and checking for some special characteristics of the dataset such as its format. When the process begins, the first task is a sub-process which defines the User Requirements, the output of this sub-process is a file called UR file. This UR file is used in other process steps whenever the model needs to use users needs for evaluating the quality of the dataset. Detailed information about the "Define User Requirement"

sub-process comes later. In the next step, "Spatial Data Quality Evaluation" sub-process which is the core sub-process of the model starts to perform. By use of the UR file and performing several task and sub-tasks, the quality of the dataset based on users requirement is evaluated and a result file is produced. Later, in the last sub-process which is called "Result Analysis", both UR file and Result file are used to analyze and report the final result of quality information about the dataset to the user. Finally, the message which shows the end of the process is passed from the web service to user interface, and user interface takes care of showing the result information to the user. Later in section 3.2, communication of quality information to users are discussed.

1. The "Define User Requirements" sub-process:
   Generally, this sub-process takes care of defining the user requirements based on users need by sending and receiving information to the user via the user interface. AppendixA Figure2 demonstrates the expanded version of this sub-process. Note that the notation signs are used to show the relationships of dataset and information about requirement files outside the sub-process with each task within the sub-process. The sub-process begins with receiving a start message and the first task is performed to receive users information about data quality element selection for quality check. Think about this task as combination of questions and possibilities shown to the user via user interface about different choices of data quality elements. The user, based on his/her knowledge about GIS and SDQ, either selects specific data quality element(s), or asks the web service to provide more information about each data quality element with examples, and based on the application of his/her project one or more data quality element(s) and data quality sub-elements would be finally chosen. In the first case if the user selects the data quality element(s) itself, then the next stage is to select data quality sub-elements for quality check. Finally, each data quality element and its related sub-element are entered to the UR file as a record. For each record in UR file, the "Define scope and conformance level" looped sub-task is performed. AppendixA Figure3 and AppendixA Figure4 shows the expanded version of this looped sub-task. After performing this looped sub-task the UR file is designed and the "Define User Requirement" Sub-process is terminated. After that, the main process continues to perform the "Spatial Data quality Evaluation" sub-process. Note that the expanded version of "Define scope and conformance level" first gives some information about the different types of scopes which the user can select for its data quality measure. There exist three different types of scopes. If the user selects the "spatial extent" type, the process leads to the "Receive the spatial extent boundary information and check" looped sub-task. AppendixA Figure5 shows the expanded version of this looped sub-task. In this sub-task, the first step is to receive two ranges of values for latitude and longitude which define the boundaries of spatial extent, from the user via user interface. By considering the dataset, the model checks to be sure the values are available in the dataset and records the value for spatial extent scope in the UR file and terminates. This sub-task is iterative, and can be performed as many

times as user wants to select boundaries for the data quality scope definition. The second type of scope is the object-based scope. If user selects it, the process would proceed to perform the "Receive information and define object-based scope" looped sub-task. As AppendixA Figure6 simply shows, the list of objects within the dataset and their attributes are extracted and showed to the user via user interface. The user selects some of the objects and attributes which want to perform the quality evaluation check on them and the model adds them to the UR file as the defined scope for its specific record of user requirement. Finally, the user has another choice for defining the scope, which it called "complex" scope. In defining the complex scope, as AppendixA Figure7 shows, the aim is to give the user the capability of defining the scope as set of specific objects and their attributes within a specific spatial extent. Indeed, the complex type is a combination of the first two types. The sub-task for defining this type of scope first runs the "Receive the spatial extent boundary information and check" sub-task shown in AppendixA Figure5. After that, all objects and their attributes within the desired spatial extent are extracted from the dataset and listed to the user for selection. The selection of user is then recorded as the scope in the UR file and the sub-task terminates. Note that if the user does not select a specific type of scope the model sets a default value of "Whole dataset" for the data quality scope. Despite the type of scope which user selects the "Define scope and conformance level" looped sub-task shown in Fig8.b continues to receive users information about the conformance level value for each measure in the UR file. After that task, there is a check performed to see whether the user has entered a value for conformance level or not. If yes, the value is entered to the UR file and if No, then a default value for it would be set by the model. In the last step, the "Define scope and conformance level" sub-task is finished. Note that this sub-task is iterative and would be performed for each record of UR file. Finally, after which this looped sub-task is performed the sub-process for defining the User Requirement shown in fig2 terminates, and the process flow continues to the next main sub-process for the model: "Spatial Data Quality Evaluation" Sub-process (AppendixA Figure8).

2. The "Spatial Data Quality Evaluation" sub-process: After defining the user requirement file, every input for performing the quality evaluation check is available. The process starts by extracting the list of data quality elements and sub-elements from each record of UR file. The records in the list are passed to the "Quality Evaluation Check" looped sub-task one by one. The sub-task evaluates the quality of selected scope in the dataset, produces the result file, and the sub-process ends (AppendixA Figure8). In the "Quality Evaluation Check" sub-task, there exist a separate sub-task for performing each data quality element and its specific sub-element. Thus, the process starts to check which data quality element and sub-element are selected for quality check. After that, the process is led to its specific sub-task. For example, the "Domain Consistency Check"

sub-task is the specific sub-task that would be performed if the details in the record of UR file mentions that a logical consistency test in means of domain consistency check must be performed. AppendixA Figure9 demonstrates this process flow. After that the desired sub-task for quality check has performed, the result file would be completed and, the end of sub-task and "Spatial Data Quality Evaluation" sup-process would be reached.
As mentioned in the previous chapter, some data quality elements and sub-elements were candidate for automated/semi-automated quality evaluation check. This chapter shows the designed model for these selected data quality element and discusses its process flow.

3. Logical consistency:

   (a) Domain consistency: AppendixA Figure10 demonstrated the process of evaluating the quality of spatial data in means of domain consistency check. The process of this sub-task begins with another inner sub-task called "Preparing data for check". As shown in AppendixA Figure11, this sub-task uses the UR file and a list of available data quality measures for domain consistency to select an appropriate measure and by using the dataset makes the list of attributes for check. After which the data is prepared for check, in the next step, for each item, model figures out the type of check to perform. It consists of two types which are field type or domain type. Based on the type selected, the process proceeds to its appropriate looped sub-task for check. If the type of check is field type check, then the model would use the information in UR file to figure out what kind of check in field type check is needed to perform (AppendixA Figure12). If the user has asked for the information about the field types, then the model extracts all field type of attributes of objects in scope and lists them. The other case is to check the type of each field in the dataset to assure the availability of the type. If available, the value entered for that field could be evaluated in means of checking to assure whether it corresponds to the type of field. The incorrect type of fields or values for specific types are counted as inconsistency, the final result is added to the result file, and the process of this sub-task terminates. On the other hand, if the user has chosen to perform the domain type check, as shown in AppendixA Figure13, the models decides on the kind of domain type check to perform based on user requirements. There are two possibilities, if range domain check is needed, the existing range of values for numeric fields in the dataset are extracted, and listed for the user, then the model receives users desired range of values for each field via user interface, and checks to see whether the values are in defined range. Those values that violate the range are counted as inconsistencies, and the result is entered in the result file (AppendixA Figure13). In the other case, if the coded domain type is selected for check, the model extracts the list of enumerated values for each field in the dataset and lists them to user via user interface. The user decides to select some of them and exclude others from the

list. Based on users defined list of enumeration, the values in each field are checked and those outside the list are counted as inconsistencies. Same as the other case, appropriate result is added to the result file, and the process ends.

(b) Format consistency: AppendixA Figure14 demonstrates the process of "Format consistency Check" sub-task. The process starts by extracting all objects and their attributes within the scope, and lists them to the user, User selects the desired objects, and attributes and by using a tool prepared in user interface, defines the structure and submits it. The model receives the structure, adds it to UR file, and for each record runs the "Format Check" looped sub-task (AppendixA Figure15). In this sub-task, a check is performed to see whether the format of existing value of item conforms to the desired format, and the result is entered in the result file. After performing this sub-task for each item, the process of "Format Consistency Check" sub-task terminates.

(c) Topological consistency:
AppendixA Figure16 shows the process of "Topological Consistency Check" sub-task. It starts with checking the type of check. Based on the type of check, the process proceeds to "Connectivity Check" or "Boundary Overlap Check" sub-tasks. In "Connectivity Check" sub-task, shown in AppendixA Figure17, Model extract objects with positional values and lists them for user. User selects the desired objects from the list and submits them. Model uses those objects to perform the "Arc Connectivity Check" looped sub-task (AppendixA Figure18). The process for checking arc connectivity is simply done by extracting the arcs which make the desired object, and by using the positional values of the start and end nodes of the arcs, their connectivity's are checked. This process is an iterative process and is performed once for each object. Same flow is done for the boundary overlap check. After extracting the arcs which make the objects and comparing the positional values of the start and end nodes of each arc to another arc, their intersection status are determined. Appropriate results are added to the result file, and the process terminates AppendixA Figure19.

(d) Completeness:

AppendixA Figure20 demonstrates the process flow of "Completeness Check". The model uses a reference dataset for comparing the dataset with it, which is called "external file". This dataset is provided by the users, and can be another XML-based dataset which includes the name of the desired objects and attributes, and their value in it. The process flow starts with receiving the reference dataset, and selecting the kind of check which is going to be performed. If the completeness of dataset for objects is going to be checked, the process follows by extracting objects from both datasets, and comparing them

(AppendixA Figure21). AppendixA Figure22 shows the "Completeness Check for Attributes of Objects" sub-task. If the completeness of attributes of objects is the goal for check, the process extracts objects from both datasets, and lists them to the user, the user selects pairs of objects from both, and the process extracts the selected attributes of objects and compares them with together. Finally, if completeness of values of attributes is wanted, then the model first extracts all objects within both datasets and lists them, the user selects pairs of objects, and submits it to the web service. The model receives them, and extracts all attributes within the selected pairs of objects and lists them. Again, the user selects pairs of attributes which the completeness of their values is wanted to be checked. Finally, the model compares their values, and the appropriate result is added to the result file (AppendixA Figure23).

(e) Temporal accuracy:
AppendixA Figure24 demonstrated the process flow of "Temporal Validity Check". The process starts with extracting information about all objects within the dataset which have date-time attributes, and lists them to the user via User Interface. The user selects desired objects, and for each of them defines desired value (or range of values) for the date-time attribute. The process follows with checking the values of date-time attributes to the user defined value (range of values) for each selected object, and adds appropriate result to the result file.

4. The "Result analysis" sub-process: After which quality evaluation methods have been performed and their results have been gathered, in result analysis phase, the results are processed and analyzed to produce final information to report to the user. In simple procedure, when a single measure is applied, the result of that measure is compared to the quality conformance level, and information about the quality status of the dataset, whether it is passed or failed, is reported. In situations which the quality conformance level is not defined by user, the result of the quality evaluation procedure can be reported quantitatively. Beside these two cases, in more real quality evaluation procedures, the numbers of data quality measures are much more than one. Thus, an aggregated function should be used to finalize the result of all measures. The model in this research uses two aggregation functions for this means which are 100% pass/fail, and weighted pass/fail [25]. In 100% pass/fail aggregation function, each result has a Boolean value 'v' of '1' if it is passed, or '0' if it is failed. The function uses the following equation for determining the final result:

DQR=$v_1 * v_2 * v_3 * \ldots * v_n$
Equation1- Equation for 100% pass/fail function. Adopted from [25]
Where 'n' is the number of data quality measures applied. The function

acts like a logical AND operator. If the final value is '1', the dataset is fully conformed to user requirements. Otherwise, the dataset has failed and is non-conformant. In the second aggregation function, each data quality result is given a Boolean value 'v' of '1' if it is passed, or '0' if it is failed. Based on the importance of the measure to the user and to the purpose of application, a weight value 'w' between '0.0' and '1.0' is assigned to each data quality result. The choices of weights are completely based on users' application. The equation used in this function is:

DQR=$v_1*w_1+v_2*w_2+v_3*w_3+...+v_n*w_n$
Equation2- Equation for weighted pass/fail function. Adopted from [25]

Where 'n' is the number of data quality measures applied. Note that, total of all the weights assigned to measures should equal '1.0'. This function provides a magnitude value which indicates the closeness of a dataset to full conformance, but it does not provide quantitative value for indication of where conformance or non-conformance takes place [25]. The "Main Process Flow" shown in fig6, has another main sub-process called "Result Analysis". This sub-process takes care of producing the final result of the quality evaluation procedure, by using the data quality measure results, and data quality conformance level of each measure. Fig28 demonstrates the process flow of this sub-process. It starts with extracting data quality measure results and their conformance level values. If there is only one measure applied (the simplest case), the measure result is compared to the quality conformance level, and the result is added to metadata. If more than one measure is applied, model asks the user about the kind of aggregation function to use. By default, the model uses the 100% pass/-fail aggregation function. It compares data quality measure results with their correspondence quality conformance levels, and by using the 100% pass/fail equation, reports final result quality information as metadata (AppendixA Figure25). In cases which the user wants to use the weighted aggregation function, the list of measures is passed to the user. User defines the values of weights for each data quality measure. The model checks the weights, and makes sure that the sum of them is equal to '1', and the model proceeds to comparing the data quality results for each measure to the correspondent quality conformance level, and after using the weighted aggregation equation the term "aggregated result" is added to metadata. In the next stage, the model produces the quality evaluation report. Every component of the report would have its appropriate value and stored as a seperate file called "Report", which is the output of this sub-process. The components of the final report are discussed in next section. Finally, the sub-process ends, and by end of it, the "Main Process Flow" ends, too. The report file made as output is passed to the user via User Interface.

3.2 Reporting quality information

When the final result of evaluation procedure is determined, the process will report the result. For reporting quality information as mentioned in chapter 2, the standardized way is to add the quality information as metadata. Whether it is a pass/fail result or quantitative result, it can be reported as metadata. In those cases dealing with aggregated result, a sign indicating that the result is an aggregated result should be added to metadata, and the rest of information including all results of measures, and aggregated function used can be reported in a quality evaluation report.
Quality evaluation report component:
The evaluation report produced by the model has additional information about quality results, and can provide useful details about evaluation procedure to users than those recorded in metadata. The report has several components, which some are mandatory and information about it must be available. While others are conditional or optional. Table3.1 shows the quality evaluation report components used by the model in this research. It includes information about the name of the component, its definition, its obligation status, maximum occurrences, data type, and domain. Note that maximum occurrences indicate the maximum times this item can occur within a superior item domain. The component can have different values for data type such as: report section, text, entity, or when not applicable, a '-' is shown. For each component, the domain indicates the value allowed or the ability to use free text [15].

#	Name	Definition/ content	Obligation/ condition	Data type	Domain
1	reportIdentification	Report identification information	M	CharacterString	**Free text**
2	reportScope	Scope of dataset evaluated in this report(ISO 19113)	O	CharacterString	**MD_MetadataScope <<CodeList>>**
3	compQuantDesc	Complementary description of quantitative assessment such as data quality measure values and their reliability limits	M	Report section	**Lines 5-14**
4	dataQualMeasure	Information on definition and value of data quality measure of an object data quality scope	M	Report section	**Lines 6-10**
5	mathDesc	Mathematical description of data quality measure	M	CharacterString	**Free text**
6	compMeasureValue	Values of data quality measure applied	M	CharacterString	**Free text**
7	valType	Unit in which data quality measure value is recorded	M	CharacterString	**Free text**
8	ReliabilityValue	Reliability or confidence limit values of the computer or estimated data quality measure value	O	CharacterString	**Free text**
9	reliabilityValueUnits	Unit in which reliability values are recorded	O	CharacterString	**Free text**
10	conformConfidence	Confidence in conformance	O	Report section	**Lines 12-14**
11	conformConfValue	Confidence in the conformance result.	M	CharacterString	**Free text**
12	dqeMethodTypeInfo	Detailed information about applying the quality evaluation method	M	Report section	**Lines 16-37**
13	dqeMethodType	Quality evaluation method class	M	CharacterString	**Free text**
14	dqeSamplingApplied	Information on inspection strategy applied	M	CharacterString	**Free text**
15	dqeMethodInfo	Information on the data quality evaluation method	M	Report section	**Lines 19-37**
16	dqeParamInfo	Information on parameters used in the data quality evaluation method	O	Report section	**Lines 23-37**
17	dqeParamDefinition	Information on the definition of parameters used	M	CharacterString	**Free text**
18	dqeParamValues	Value of parameter used in the data quality evaluation method	M	CharacterString	**Free text**
19	dqeSampleMethod	Information on sampling method	C	Report section	**Lines 31-37**
20	aggSourceValues	Information on which component datasets are used and what data quality measures are aggregated	C	Report section	**Lines 39-44**
21	aggResult	Description of the value as a quantitative result	M	Report section	**Lines 40-44**

Table3.1. Quality evaluation report components.[15]

Chapter 4

Implementation

In previous chapter the automated evaluation model of spatial data quality was designed and discussed. To see how operational this model can be in practice, a web service prototype is designed based on the concept of the model. The web service includes implementation of three main parts of the model named as the three main sub-process in Fig1 in chapter 3. In this chapter the process of implementing each of these main parts are covered. Based on the capabilities and benefits that .Net framework brings to web applications, ASP.Net 2.0 is selected as the language of web service implementation. Asp.Net is a web application framework developed by Microsoft to allow programmers to build dynamic websites, web applications, and web services. In addition, as mentioned chapter 2, XML-based datasets are appropriate choice for data transmission via web services. The prototype web service implemented in this research accepts XML-based datasets as input and by using the model for domain consistency designed in chapter 3, it evaluates the quality of dataset in means of domain consistency and reports the final result. Asp.Net 2.0 provides "XPath" namespace for navigating through an XML document and reading/writing all necessary elements and values of it. Part of the source code written for parser is covered in AppendixB.

4.1 Defining user requirements

As it was covered in chapter 2, the main process flow for spatial data quality evaluation includes an important input file called "User Requirements". The same file exists in the "Defining User Requirements" sub-process model covered in chapter 3 specifically named "UR_File". As mentioned previously, this file contains main components necessary for performing the quality evaluation procedure(see chapter 2). For the same reason that we decided to use xml-based datasets as the input of the model, the UR_File designed in the automated quality evaluation model is implemented in a xml file format. This makes it extensible and easily transferable via web services in Internet. Figure4.1 shows the User Requirement file (UR_File) XML schema. This file contains some mandatory components such as data quality element, data quality sub-element, and data quality measure. While other components such as scope, and quality con-

formance level are optional. In cases which user does not provide information about the optional components, the model sets default values and proceeds to next step.

```
<?xml version="1.0"?>
<User_Requirements>
  <measure id='1'>
    <DQ_Element>Logical Consistency</DQ_Element>
    <DQ_SubElement>Domain Consistency</DQ_SubElement>
    <DQ_Scope type='1'>
      <Scope id='1'>
        <Boundary>Lat1 Long1 Lat2 Long2 Lat3 Long3 Lat4 Long4</Boundary>
      </Scope>
      <Scope id='2'>
        <Boundary>Lat1 Long1 Lat2 Long2 Lat3 Long3 Lat4 Long4</Boundary>
      </Scope>
    </DQ_Scope>
    <DQ_ConformanceLevel>
      <CL_Value>5</CL_Value>
      <CL_Type>percent</CL_Type>
    </DQ_ConformanceLevel>
  </measure>
  <measure id='2'>
    <DQ_Element>Logical Consistency</DQ_Element>
    <DQ_SubElement>Domain Consistency</DQ_SubElement>
    <DQ_Scope type='2'>
      <Scope id='1'>
        <object id='1'>
          <Obj_Name>road</Obj_Name>
          <attribute>length</attribute>
        </object>
      </Scope>
    </DQ_Scope>
    <DQ_ConformanceLevel>
      <CL_Value>5</CL_Value>
      <CL_Type>number</CL_Type>
    </DQ_ConformanceLevel>
  </measure>
</User_Requirements>
```

Figure4.1. A UR_File template in xml format

The first duty of webservice is to create this UR_File by facilitating interaction between the user and the model. This is done via User Interface. User Interfaces are combination of tools which make interaction between client users and server scripts possible. The language for design of UI in this web service is HTML and Javascript. Apart from the graphical presentation of UI, there is a code behind environment for each task that controls the behaviour of UI and manages the storing and retrieving of information. For this, C#.Net 2.0 is selected for implementation of the webservice. Figure4.2 shows the HTML scripts used for UI design which handles the interaction of user and algorithm for defining the UR_File. Figure4.3 shows part of source code of algorithm for handling such action.

```
<asp:Panel ID="Pnl_Step2" runat="server" BackColor="Thistle" BorderStyle="Solid"
    BorderWidth="1px" Height="16px" Width="800px" Visible="False">
    <table width="780">
        <tr>
            <td rowspan="2" style="width: 92px; height: 33px">
                <asp:Label ID="Label7" runat="server" Font-Bold="True" Font-Names="Verdana" ForeColor=
                    Text="Step 2"></asp:Label></td>
            <td style="width: 471px; height: 33px">
                <asp:Label ID="Label8" runat="server" Text="Select the spatial data quality element:">
                <asp:DropDownList ID="Combo_SDQE" runat="server" AutoPostBack="True" ForeColor="Blue"
                    OnSelectedIndexChanged="DropDownList1_SelectedIndexChanged" Width="220px">
                    <asp:ListItem Selected="True" Value="0">Select...</asp:ListItem>
                    <asp:ListItem Value="1">Logical consistency</asp:ListItem>
                    <asp:ListItem Value="2">Completeness</asp:ListItem>
                    <asp:ListItem Value="3">Temporal accuracy</asp:ListItem>
                </asp:DropDownList></td>
            <td style="height: 33px">
            </td>
        </tr>
    </table>
</asp:Panel><asp:Panel ID="Pnl_Step3" runat="server" BackColor="Wheat" BorderStyle="Solid"
```

Figure4.2. HTML and ASP source code for part of UI design

```
protected void DropDownList1_SelectedIndexChanged(object sender, EventArgs e)
{
    if (Combo_SDQE.SelectedIndex == 0)
    {
        Pnl_Step3.Visible = false;
    }
    else
    {
        Combo_SDQSE.Items.Clear();
        switch (Combo_SDQE.SelectedIndex)
        {
            case 1:
                {
                    Combo_SDQSE.Items.Add("Select...");
                    Combo_SDQSE.Items.Add("Conceptual consistency");
                    Combo_SDQSE.Items.Add("Topological consistency");
                    Combo_SDQSE.Items.Add("Domain consistency");
                    Combo_SDQSE.Items.Add("Format consistency");
                    break;
                }
            case 2:
                {
                    Combo_SDQSE.Items.Add("Select...");
                    Combo_SDQSE.Items.Add("Commission");
                    Combo_SDQSE.Items.Add("Omission");
                    break;
                }
            case 3:
                {
                    Combo_SDQSE.Items.Add("Select...");
                    Combo_SDQSE.Items.Add("Temporal consistency");
                    break;
                }
            default:
                break;
        }
        Pnl_Step3.Visible = true;
    }
```

Figure4.3. C#.Net 2.0 source code for part of UI design

4.2 Spatial data quality evaluation

For extracting information from dataset, a parser is coded. The parser is a program which reads the dataset for quality evaluation check. The parser is designed based on the format and characteristics of the dataset. For implementing this web service, CityGML datasets were used which are a special application of GML datasets. The heart of the model is contained of a collection of tasks for evaluating the quality of spatia data. Each task has the responsibility of performing quality check for specific spatial data quality element. For this webservice protoype, the model of logical consistency check in means of domain consistency test is implemented. According to section 2.1.2, for domain consistency two types of checks can be performed which are field type and domain type check. Domain type check is implemented in the prototype. Figure4.4 shows part of the source code of functions which control and handles the coded domain check.

```
protected void CodedDomain_Check()
{
    string[] Inconsistency_array = new string[ListBox4.Items.Count];
    int inconsistencies = 0;
    int Total_Value = 0;
    string element = ListBox1.Items[ListBox1.SelectedIndex].Value.ToString();
    string atribute = ListBox2.Items[ListBox2.SelectedIndex].Value.ToString();
    XmlTextReader reader = new XmlTextReader(Server.MapPath(@"~\Files\Dataset\stations.xml"));
    while (reader.Read())
    {
        while (reader.MoveToNextAttribute())
        {
            if (reader.Name.ToString() == atribute)
            {
                Total_Value += 1;
                for (int k2 = 0; k2 <= Inconsistency_array.Length - 1; k2++)
                {
                    if (reader.Value.ToString() == Inconsistency_array[k2])
                    {
                        inconsistencies += 1;
                        break;
                    }
                }
            }
        }
    }
}
```

Figure4.4. C#.Net 2.0 source code for part of the coded domain check

The CityGML dataset are created based on predefined and known model schemas. The model schemas are used by the parser in order to extract data from the datasets based on the elements defined in the model schemas. Obviously, if the dataset does not refer to a valid CityGML or gml model schema, the web service does not accept it as a dataset. The web service was designed to support both automated and semi-automated domain consistency check. In semi-automated domain consistency check the user enters user requirement values and also defines the coded domain values that violates his/her expected coded domain values. After that, the web service evaluates the dataset, and reports the final results back to the user. On the other hand, if user cannot define the user requirements (or the user is another web service), the web service uses default values for each user requirement component and counts those attributes of items which have "Null" as their value, as inconsistency, and reports the result back to user.

4.3 Result analysis and report

Based on the characteristics requested by the user for evaluation, the result of quality evaluation can be a single or aggregated result. The model for result analysis creates a result file including all information about the evaluation characteristics and its final result. This file was called "Result_file" in chapter 3. Figure4.5 shows the xml schema of this file.

```
<?xml version="1.0"?>
<Result_File>
  <Result_Measure id='1'>
    <DQ_Measure>
      <DQ_Measure_Description>Percentage of inconsistencies</DQ_Measure_Description>
      <DQ_Measure_ID>1023</DQ_Measure_ID>
      <DQ_Evaluation_Method>
        <DQ_EM_Type>Internal</DQ_EM_Type>
        <DQ_EM_Description>Devide count of inconsistent items in dataset by count of items in scope of
        universe of discourse</DQ_EM_Description>
      </DQ_Evaluation_Method>
      <DQ_QualityResult>
        <DQ_ValueType>percentage</DQ_ValueType>
        <DQ_Value>10</DQ_Value>
        <DQ_ValueUnit>percent</DQ_ValueUnit>
      </DQ_QualityResult>
      <DQ_Date>2010-08-01</DQ_Date>
      <DQ_ConformanceLevel>
        <CL_ValueType>percentage</CL_ValueType>
        <CL_Value>5</CL_Value>
        <CL_ValueUnit>percent</CL_ValueUnit>
      </DQ_ConformanceLevel>
      <DQ_Measure_Result>
        <Final_Result>Fail</Final_Result>
      </DQ_Measure_Result>
    </DQ_Measure>
  </Result_Measure>
</Result_File>
```

Figure4.5. A Result_File template in xml format

In the final stage of the process flow, the final result of the quality evaluation procedure must be reported to the user in an appropriate way. Generally, the idea of reporting quality information as metadata is a standardized way. By this, the quality information could be used later by any other service. But in some special cases, with having aggregated result, in addition to stating result in metadata, a quality evaluation report should be returned to the user as an output. This report can carry extra detail about the evaluation properties, which is useful for naive users. This report can be designed in a tabular format. The prototype implemented for domain consistency check supports evaluating one measure at a time, and does not evaluate aggregated results. But it still creates quality evauation report in addition to metadata for helping users to gain more information about the evaluation procedure and its related results. Figure4.6 shows a sample output of ealuation report in metadata. Note that based on the specific characteristics set for the evaluation procedure some elements of report might not be available. Figure4.7 shows the table of quality evaluation report created by the same test.

```
<?xml version="1.0"?>
<!--This is the Metadata file created and used by the automated quality evaluation web service-->
<Metadata>
  <Data>
    <Dataset_Name>stations.xml</Dataset_Name>
    <Last_Update>10-09-2010</Last_Update>
  </Data>
  <DataQuality>
    <dqScope>
      <scpType>1</scpType>
      <scpExtent>Whole dataset</scpExtent>
      <exDesc>The whole features and their attributes are evaluated</exDesc>
    </dqScope>
    <dqReport>
      <eleTypeCode>002</eleTypeCode>
      <subEleCode>002</subEleCode>
      <addSubEle>Domain Consistency</addSubEle>
      <addDesc>Domain consistency of the dataset</addDesc>
      <dqResult>
        <measName>Number of errors</measName>
        <dateTime>09-09-2010</dateTime>
        <Last_Update>10-09-2010</Last_Update>
        <measResult>
          <Result>
            <resTitle>The conformane of Dataset for XYZ Project</resTitle>
            <conExpl>Conformance to user requirements</conExpl>
            <pass>0</pass>
          </Result>
          <quanResult>
            <quanValDomain>Number</quanValDomain>
            <quanRes>353</quanRes>
          </quanResult>
        </measResult>
      </dqResult>
    </dqReport>
  </DataQuality>
</Metadata>
```

figure4.6. A sample of metadata file produced by web service as an output carrying quality information result
(based on [18])

Report Identification	Quality evaluation report for the forest dataset
Report scope	Scope defined in metadata (See: dqScope)
Data Quality Measure	
Measure ID	
Measure Description	Number of domain inconsistencies
CompMeasure Value	353
Value type	Number
Conform reliability	
Conform reliability values	Coformance level=10
Conform reliability domain	Number
Data quality evaluation method type info	
Data quality evaluation method type	2 (direct internal)
Data quality evaluation sampling applied	0 (no sampling)
Data quality evaluation method info	
Data quality evaluation method description	compare attributes of items whithin scope against acceptable attribute domain (acceptable values) and determine if any are outside the domain
Data quality evaluation parameter information	
Data quality evaluation parameter definition	Acceptable domain values
Data quality evaluation parameter values	
Data quality evaluation sample method	
Data quality evaluation sample method description	All items whithin scope in dataset
Data quality evaluation result	
Data quality evaluation value type	Number
Data quality evaluation value	353
Data quality evaluation value unit	Attribute domain violations
Data quality evaluation date	09-09-2010

figure4.7. A sample of quality evaluation report created by web service carrying extra quality information

4.4 Test and validation

By using the prototype web service designed for performing logical consistency test in means of domain consistency check, different CityGML datasets were downloaded and passed to the web service. Both scenarios of entering UR and performing a semi-automated quality evaluation, and performing an automated quality evaluation were tested. The result of metadata, and quality evaluation report table produced by the web service are given in fig and fig , repectively. Note that some problems of this web service were experienced. Firstly, the webservice does not accept files larger than 4-5 Mb. This could be related to settings of the server that runs the web service. Secondly, the time duration for quality evaluation is directly related to the size of the dataset. As the size of the dataset increases the quality evaluation procedure becomes more time-consuming.

Chapter 5

Discussion, conclusion and recommendation

In this chapter the process for analysing the problem definition, strategies used for designing the model of automated quality evaluation, and web service implementation is discussed. After that, the conclusion based on the result of this research is mentioned. Finally, some suggestion for future work is given.

5.1 Discussion, and conclusion

The main objective of this research was to design and implement a prototype web service that has the ability of evaluating spatial data quality. This web service can act as a node in spatial data infrastructure (SDI) to serve this functionality to other web services. For this issue, it was mentioned that the term quality in this research is understood as the appropriateness of spatial data for users use. The web service should obey a process flow for receiving dataset and user requirements from user, performing the quality evaluation procedure, and reporting the results back to the user. Since the definition of quality is based on user requirements, the model of process flow contains a part for defining user requirements. For this issue, ISO 19100 series of standards for geographic information was used for determining the necessary components of user requirements. In addition, based on the definition that was discussed for the meaning of "automatic" and "semi-automatic" in this research, all data quality elements and sub-elements defined in ISO19113 were reviewed and their capability for automated/semi-automated implementation were studied. After study of these quantitative data quality elements, it was concluded that some of the elements could not be evaluated automatically/semi-automatically. Examples of these elements are positional and thematic accuracy. Meanwhile, other data quality elements were founded suitable for automated/semi-automated quality evaluation check. After study and analysis, based on the fact that we have different types of users for this web service,the model was designed to support two different scenarios. First case happens for naive users or other web services that want to use this web service. They can not set values for parameters and components of user requirement and because of this they just pass the dataset

and want to check it automatically. For this issue, the model uses some data quality elements and subelements for evaluation which can be performed automatically, or default values can be set for user requirement components. On the other hand, users that have geographic information system (GIS), and spatial data quality (SDQ) knowledge can define their needs and because of this, the model can provide more functionalities for them.
In addition, available standards for geospatial web services were reviewed. The output of this study was that XML-based datasets (e.g. GML) were selected for the dataset format that web service accepts as an input and can perform the quality evaluation procedure on it. The important issue about these datasets are that they should obey spatial data model schemas, and the reference to that schema must be available. The web service makes use of the schemas to extract information from the dataset and prepare it for the quality evaluation procedure. Examples for this datasets are CityGML datasets. For test and verifying the web service prototype implemented in this research CityGML datasets were used. If the user passes an xml-based dataset which does not contain reference to data model schemas, the web service would deny the acception of it. Although it might use a specific data model schema for storing information, but the reference to that schema must be defined and declared. Examples of this kind of datasets are those datasets converted from another format to an XML-based format which the tool for conversion does not convert and insert correct schema declaration for the final dataset.
Furthermore, there exists standards for geographic data handling in Internet such as WFS, and WCS. These standards can be used by other web services and can produce datasets in GML format. Since the automated quality evaluation web service acts as a node in SDI, the input of this service can be passed by a web feature service (WFS), the evaluation procedure is performed, and the results of quality information can be returned back to the origin for further use.
Due to above-mentioned issue, for implementing the communication of quality information to the users were categorized into two ways. First case is to store quality information as metadata. The metadata schema defined by [16] was studied and the appropriate elements for this information report were selected. Extensible mark-up language (XML) is used in the quality evaluation model to create the metadata file. The metadata file that the web service returns can be simply read and used by other web services. Also, users can save this metadata with the dataset itself for future use of dataset. On the other hand, for naive users which can not read and understand quality information in metadata a quality evaluation report is designed and produced by the webservice which carries additional information about the evaluation procedure, the criterion used in the evaluation procedure, and the final results. This report can carried along with the dataset which brings more information to other users of the dataset in future.
Generally, datasets are read by use of a parser. In this research, the XPath namespace in visual studio.Net 2.0 and its predefined classes and methods were used to code the parser of the webservice. This parser was coded in such a way to extract data from CityGML datasets by using the predefined CityGML model schemas. Furthermore, after running the web service with different datasets,

and analysing the results it can be concluded that standards on spatial data quality and its evaluation are difficult to implement. Due to this issue, the benefits that they could bring to costumers are not always obtained. The quality evaluation procedure must me handled in a standardized and consistent manner in order to determine whether achieved quality level meets the requirement. This research presented a process flow framework to enhance the evaluation and implementation of the geographic information quality standards. The use of BPMN allows us to combine workflow models from the functional and behavioral perspectives which bring benefits such as:

- It makes the understanding of the evaluation process easier for the different process participants and verifies that the evaluation is done in a consistent and standardized manner.
- It is clear that the process modeling presents significant challenges throughout organizations. BPMN enables to obtain a standard representation of the processes, and it is also sharable and reusable.

Finally, the designed model can be used in implementation and can work fine in practice. Although it has some limitations which include:

- The model has some limitations with providing quality information about some data quality elements and sub-elements.
- Receiving and evaluating large datasets are time-consuming and sometimes impossible.

5.2 Recommendation

For making the most use of the standardized model designed in BPMN it is highly recommended to translate it to BPEL, and implement the webservice by making use of standard web service technologies such as SOAP, XMI, and WSDL. Due to lack of time, in this research this process was done by using the model as a concept for designing an algorithm to handle quality evaluation check. Also, predefined .Net 2.0 namespaces and classes were used to manage the communication between user and server that runs the service, instead of using SOAP, XMI, and WSDL. By using the before-mentioned technologies the machine to machine communication can be possible; the scenario of what we defined it in user types, specifically users that are web services. The strengths and weaknesses of using capabilities that BPEL provides to web services should be observed in future researches. Another important factor for such webservices is the dataset that they accept as an input. In this research XML-based datasets were chosen, while in practice different datasets do exist, and quality evaluation models for handling them is required. In this research we suggested to make use of tools that convert other dataset types to XML-based dataset types, but in practice conversion tools are not easily available, and we cannot expect naive users to be familiar of working with data conversion tools. So, the model

and webservice can be completed by accepting different dataset types.
As explained before, the parser used for reading the dataset file was coded by our own algorithm which works specifically for CityGML datasets. An improvement of this parser is recommended, in such a way to support every XML-based datasets. If the model is enhanced to accept other datasets except of XML-based, then the parser should be improved to read those datasets, too.
For designing the automated quality evaluation model, some data quality elements were excluded. More research should be carried out in order to enhance the model by providing some functionalities for supporting such data quality elements.
More specifically, for evaluating positional accuracy and thematic accuracy a suggestion could be to enable the model to accept metadata and extract quality information from metadata regarding these data quality elements.

Appendix A

The designed automated quality evaluation process flow model

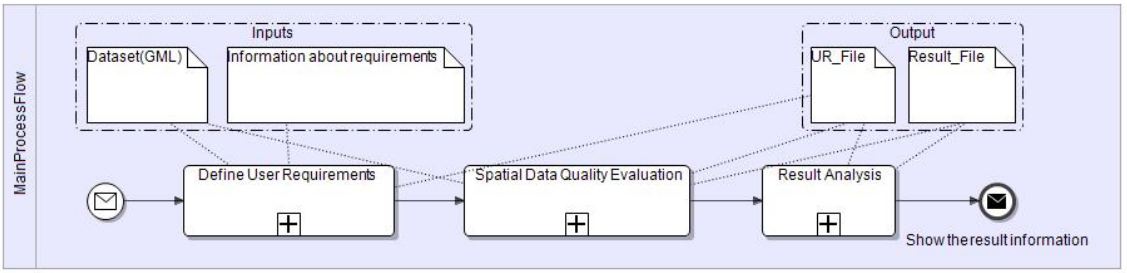

Figure A.1: Main process flow designed in Business Process Modeling Notation.

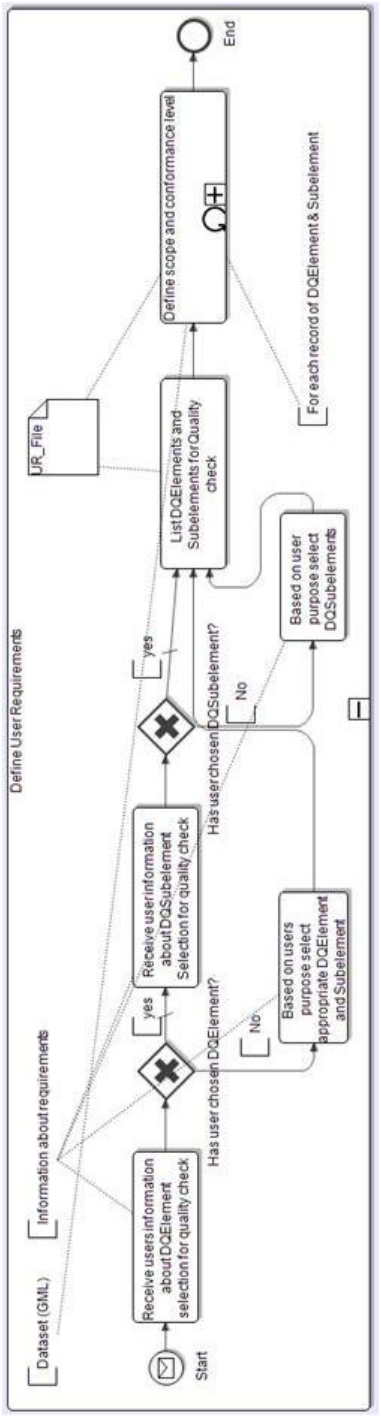

Figure A.2: Expanded version of "Design User Requirement" sub-process in BPMN.

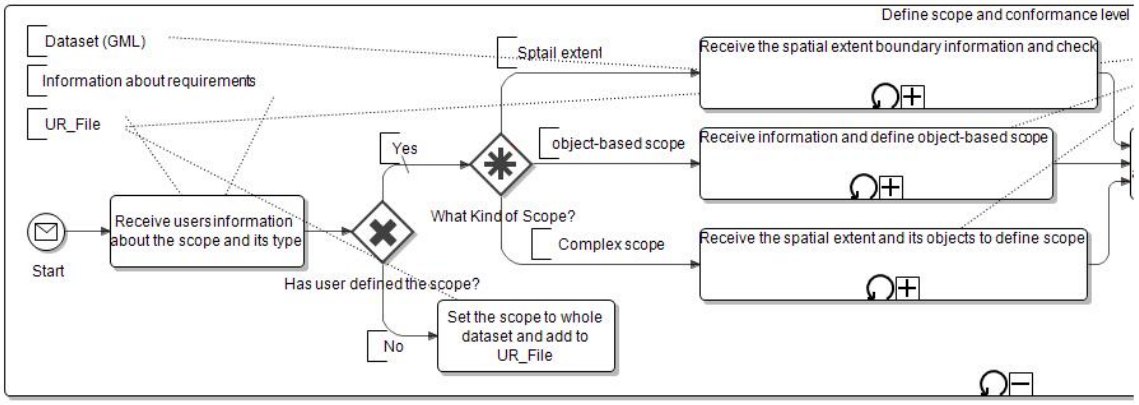

Figure A.3: Expanded version of "Define scope and conformance level" looped sub-task(PartA)

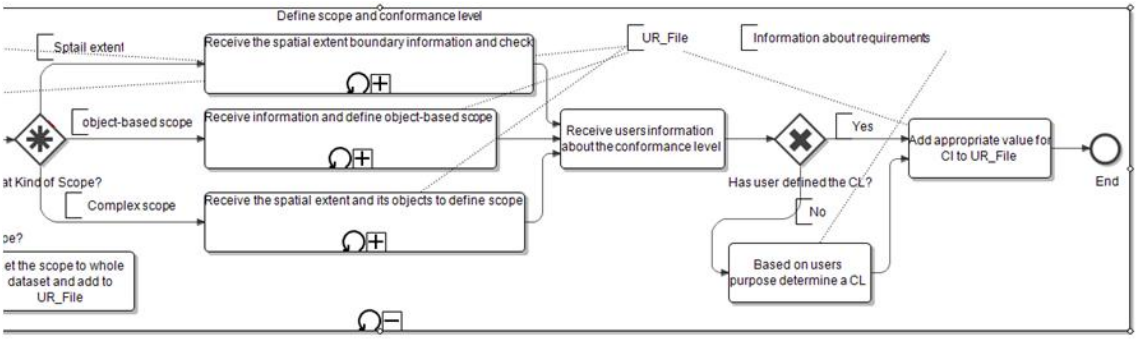

Figure A.4: Expanded version of "Define scope and conformance level" looped sub-task(PartB)

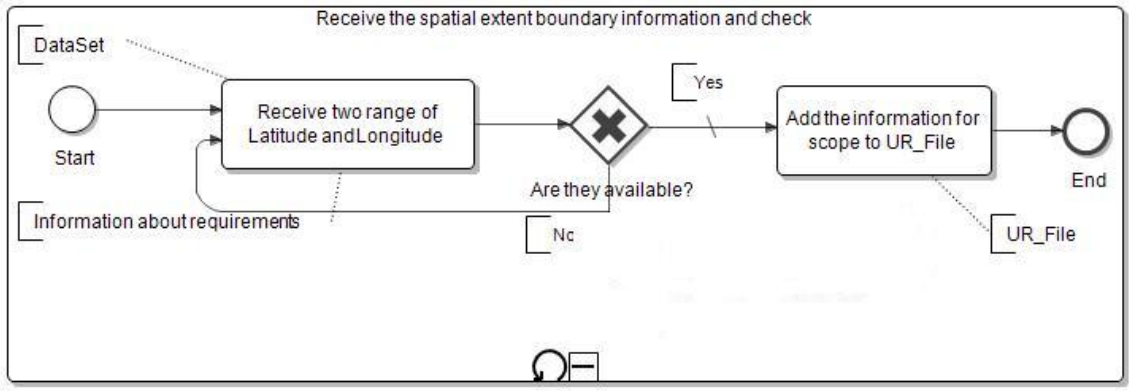

Figure A.5: Expanded version of "Receive the spatial extent boundary information and check" looped sub-task.

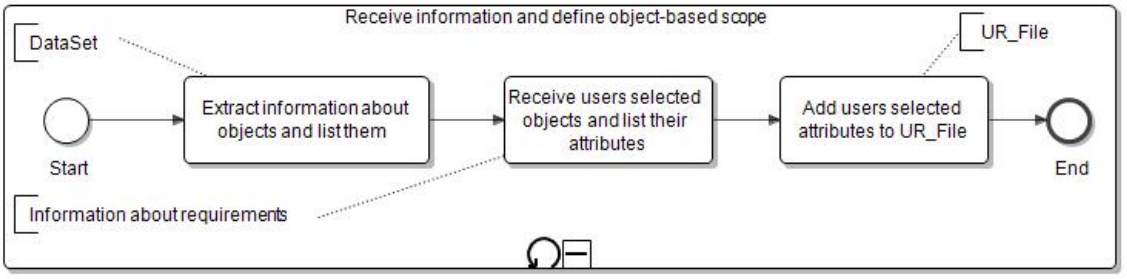

Figure A.6: Expanded version of "Receive information and define object-based scope" looped sub-task.

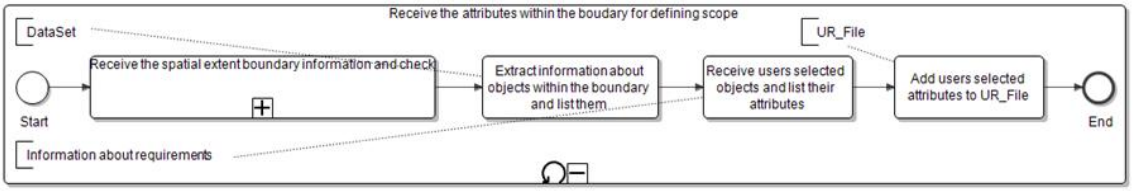

Figure A.7: Expanded version of "Expanded version of "Receive the attributes within the boundary for defining scope" looped sub-task.

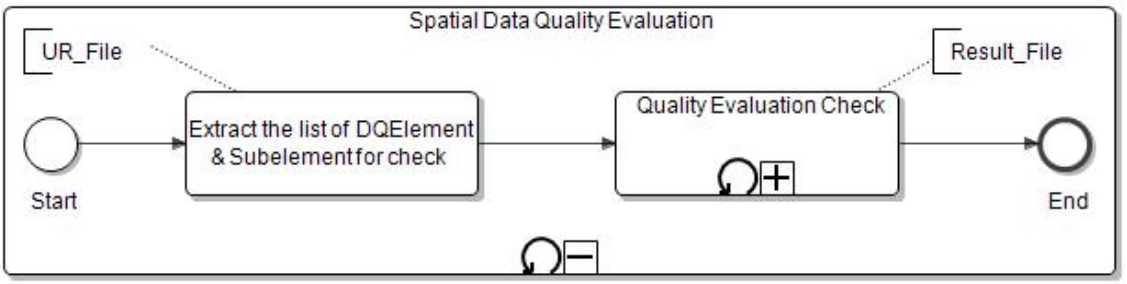

Figure A.8: Expanded version of "Spatial Data Quality Evaluation" sub-process.

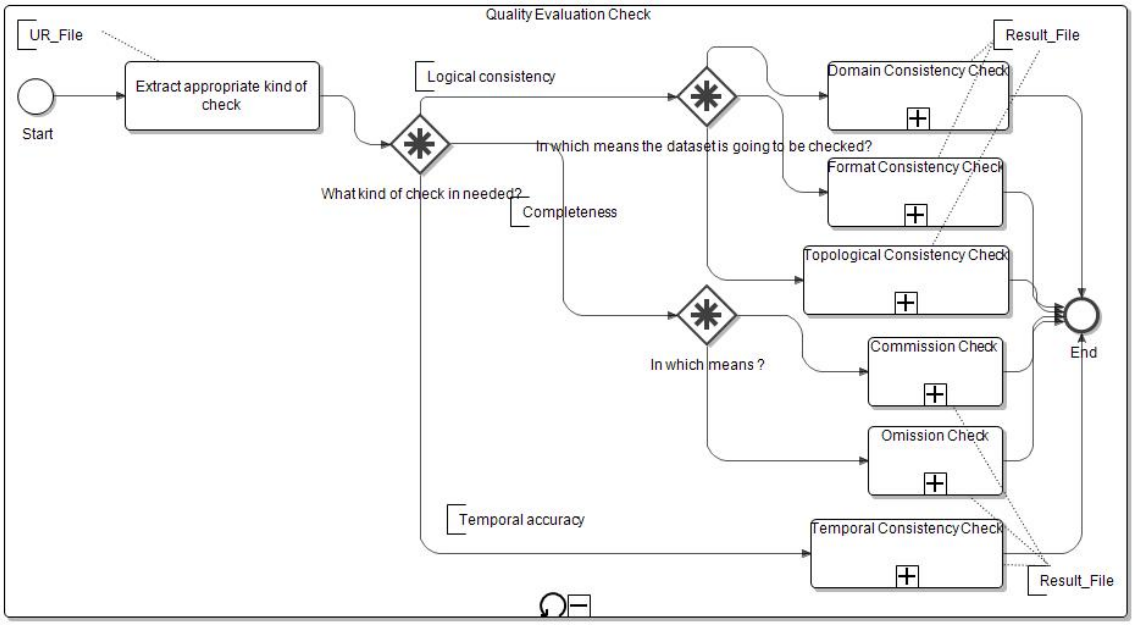

Figure A.9: Expanded version of "Quality Evaluation Check" looped sub-task.

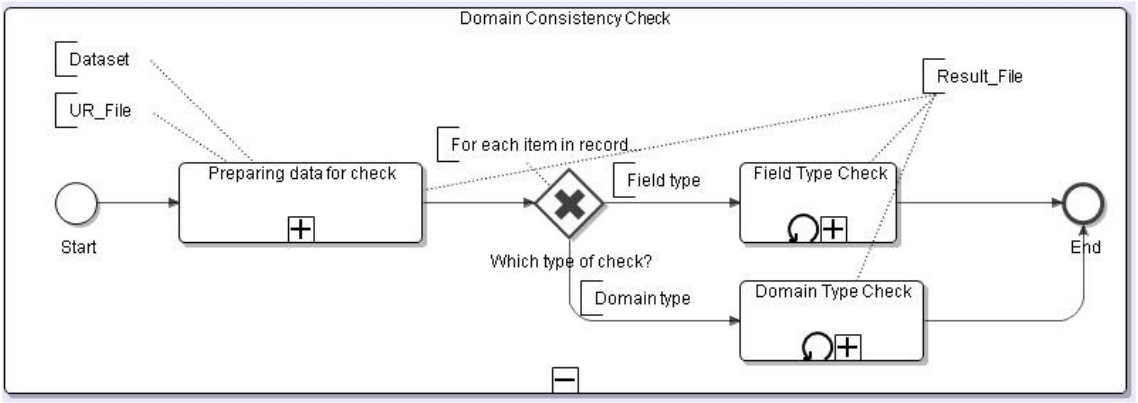

Figure A.10: Expanded version of "Domain Consistency Check" sub-task.

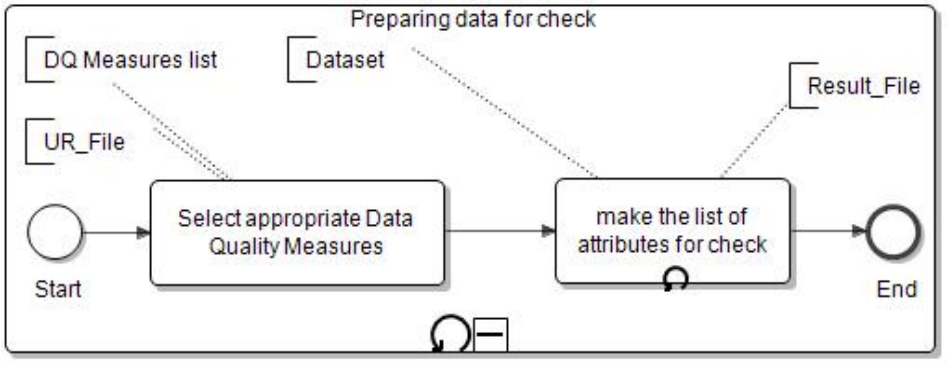

Figure A.11: Expanded version of "Preparing data for check" looped sub-task.

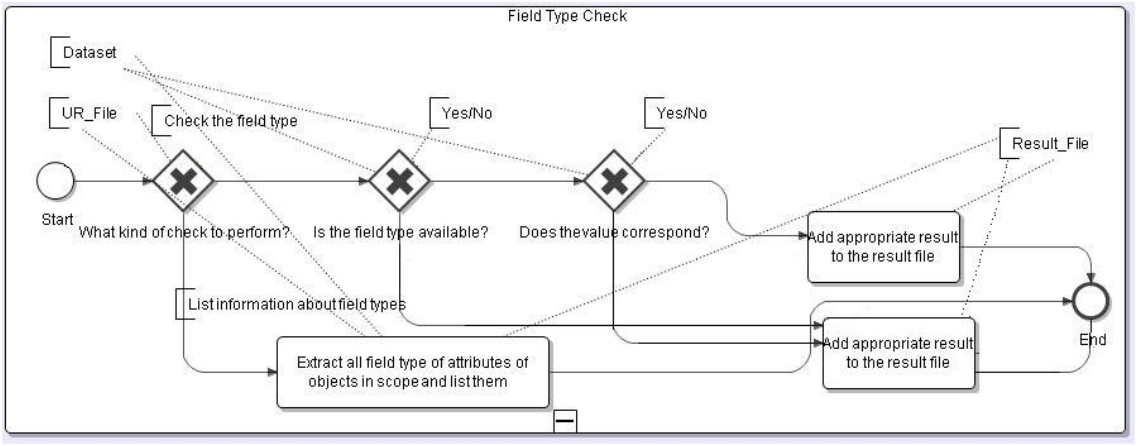

Figure A.12: Expanded version of "Field Type Check" sub-task.

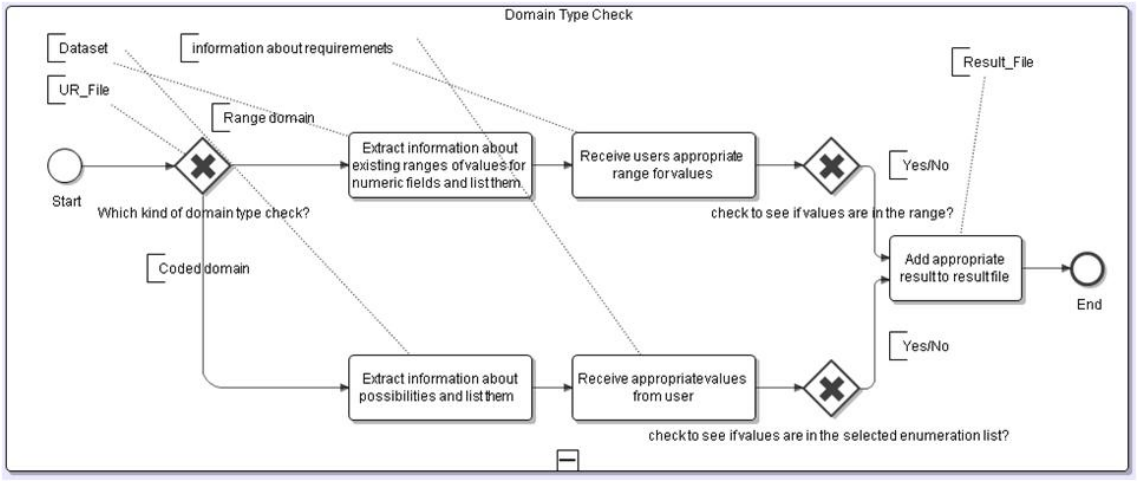

Figure A.13: Expanded version of "Domain Type Check" sub-task.

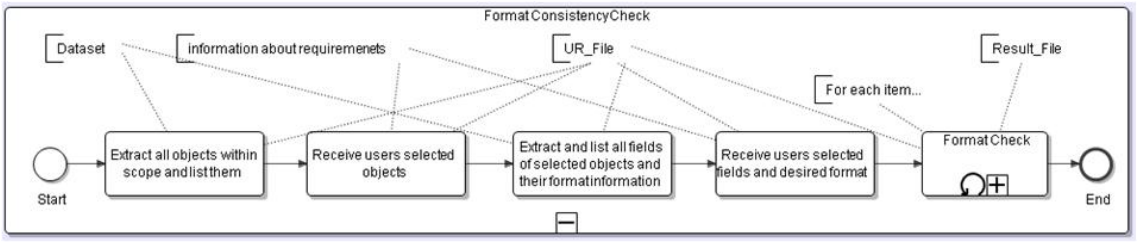

Figure A.14: Expanded version of "Format Consistency Check" sub-task.

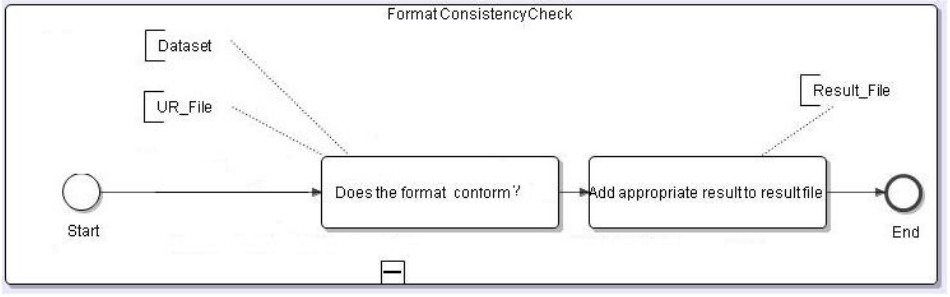

Figure A.15: Expanded version of "Format Check" looped sub-task.

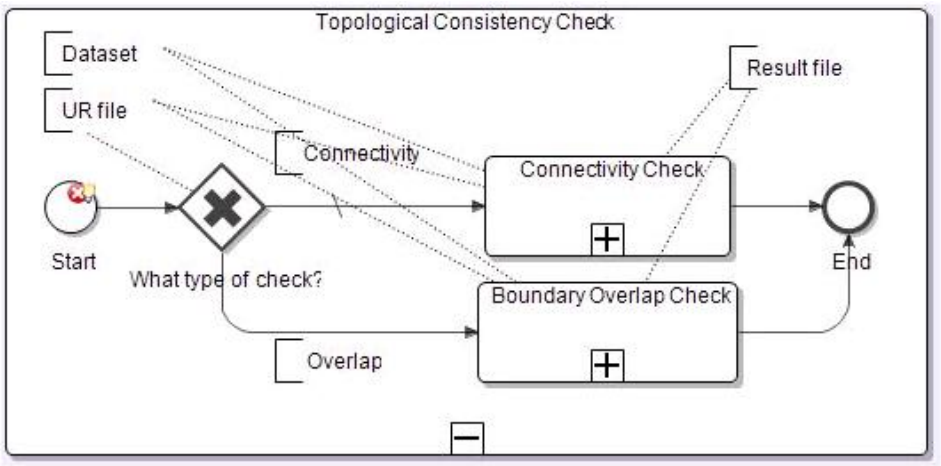

Figure A.16: Expanded version of "Topological Consistency Check" sub-task.

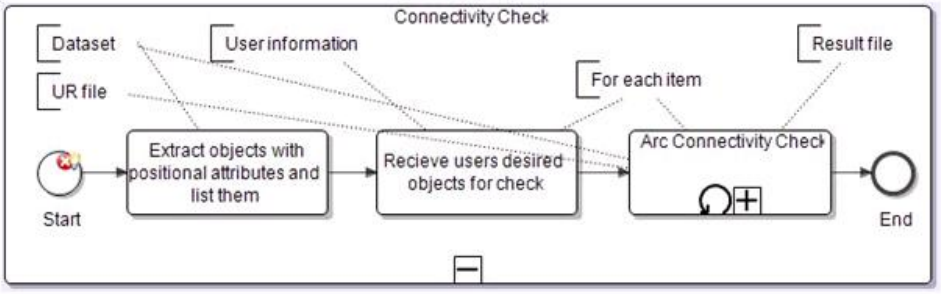

Figure A.17: Expanded version of "Connectivity Check" sub-task.

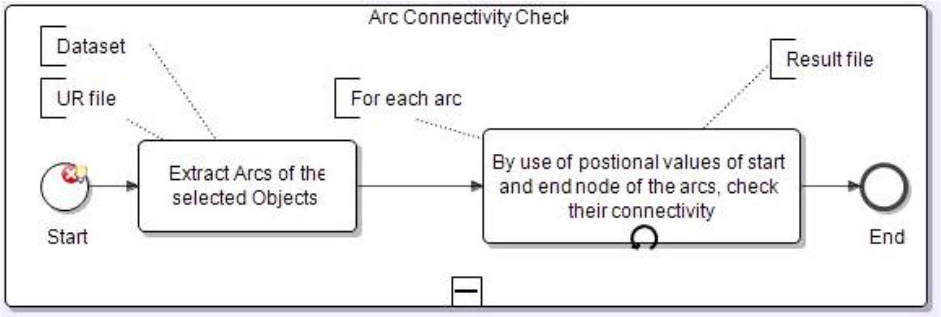

Figure A.18: Expanded version of "Arc Connectivity Check" looped sub-task.

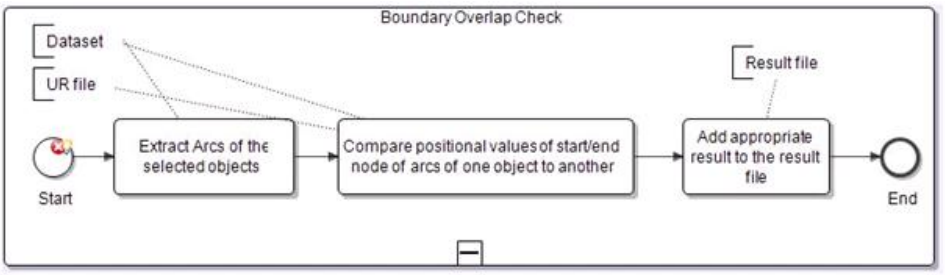

Figure A.19: Expanded version of "Boundary Overlap Check" sub-task.

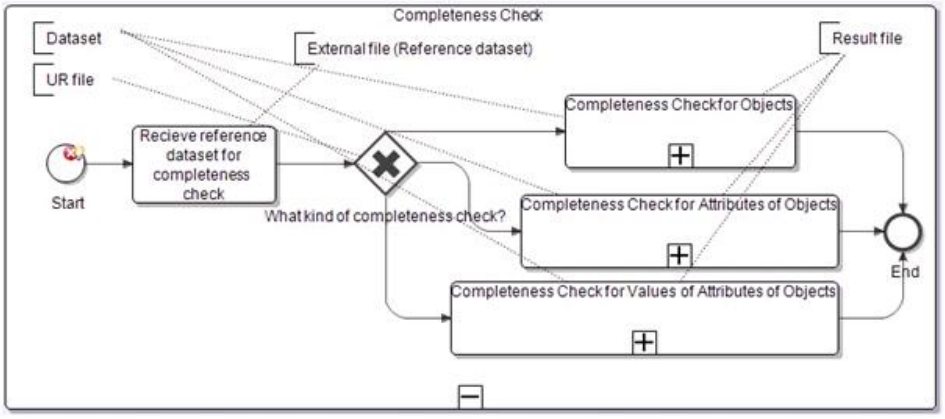

Figure A.20: Expanded version of "Completeness Check" sub-task.

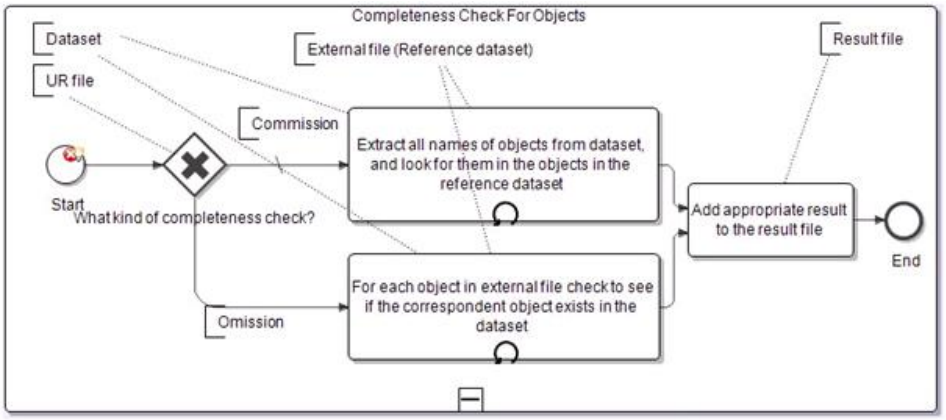

Figure A.21: Expanded version of "Completeness Check for Objects" sub-task.

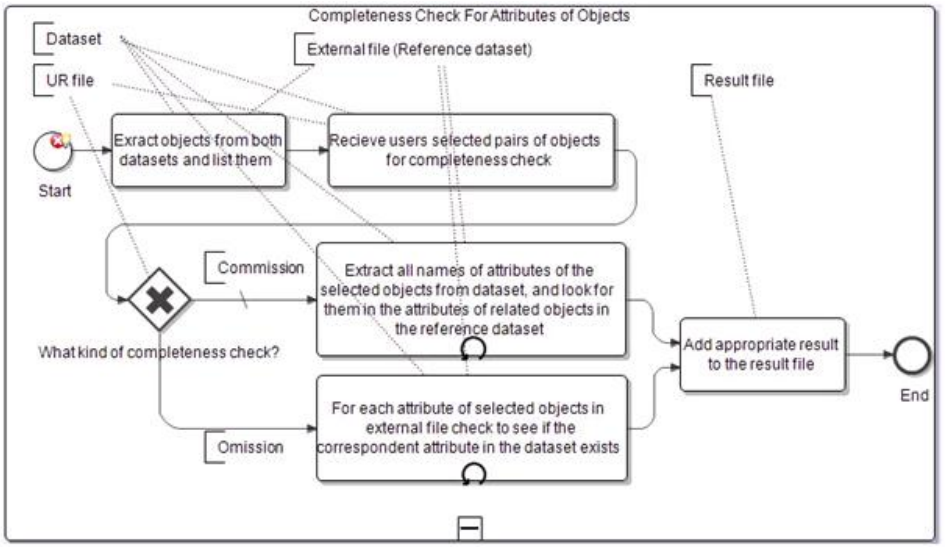

Figure A.22: Expanded version of "Completeness Check for Attributes of Objects" sub-task.

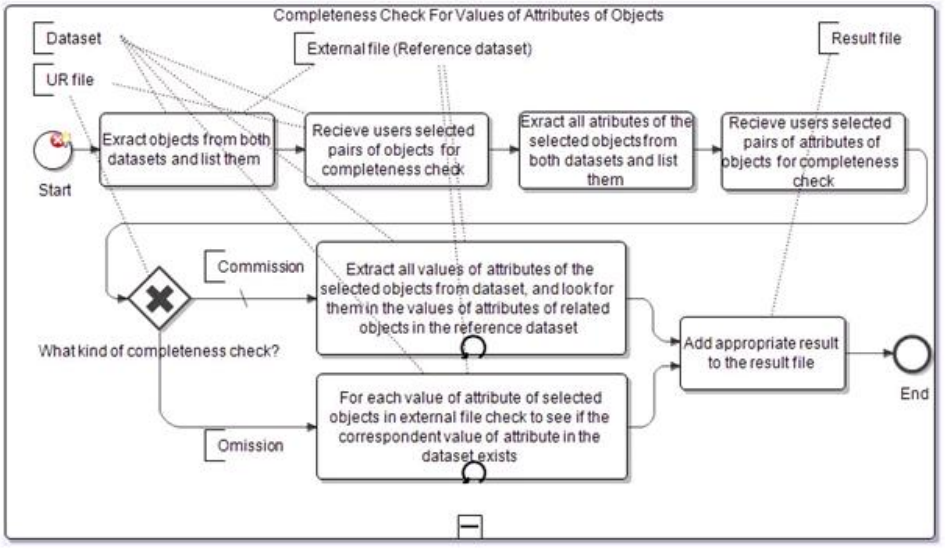

Figure A.23: Expanded version of "Completeness Check for Values of Attributes of Objects" sub-task.

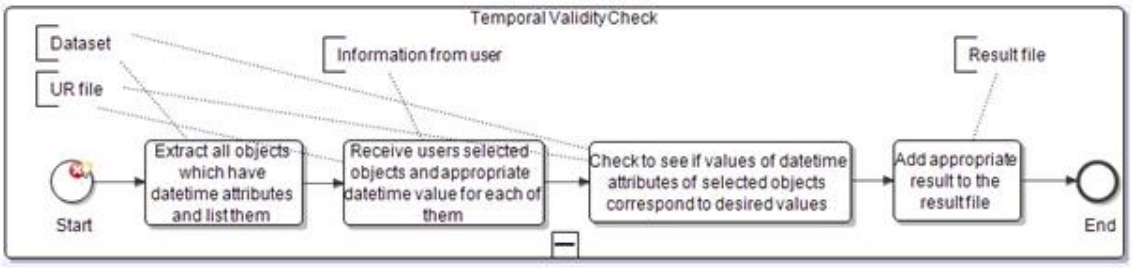

Figure A.24: Expanded version of "Temporal Validity Check" sub-task.

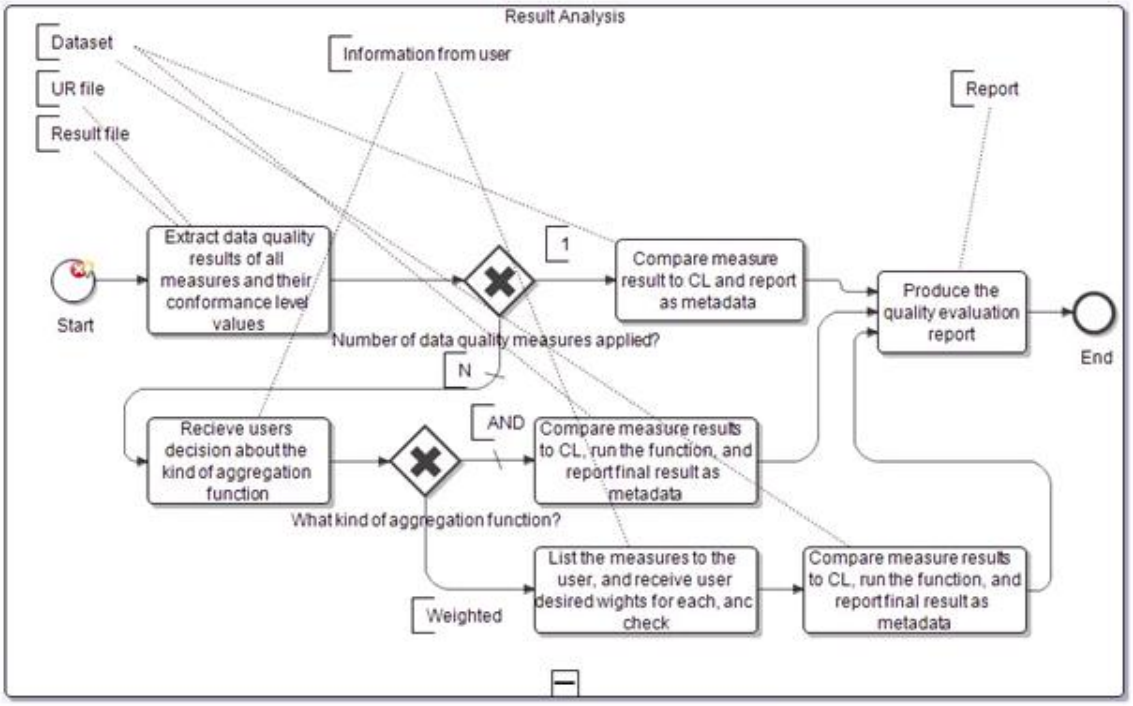

Figure A.25: Expanded version of "Result Analysis" sub-process.

Appendix B

Source codes of automated quality evaluation web service

```
using System;
using System.Data;
using System.Configuration;
using System.Collections;
using System.Web;
using System.Xml;
using System.Xml.XPath;
using System.Web.Security;
using System.Web.UI;
using System.Web.UI.WebControls;
using System.Web.UI.WebControls.WebParts;
using System.Web.UI.HtmlControls;

public partial class AQEWS : System.Web.UI.Page
{
string Result;
public int[] Measures = new int[5];
protected void DropDownList1_SelectedIndexChanged(object sender, EventArgs e)
{
if (Combo_SDQE.SelectedIndex == 0)
{
Pnl_Step3.Visible = false;
}
else
{
Combo_SDQSE.Items.Clear();
switch (Combo_SDQE.SelectedIndex)
{
case 1:
{
Combo_SDQSE.Items.Add("Select...");
```

```
Combo_SDQSE.Items.Add("Conceptual consistency");
Combo_SDQSE.Items.Add("Topological consistency");
Combo_SDQSE.Items.Add("Domain consistency");
Combo_SDQSE.Items.Add("Format consistency");
break;
}
case 2:
{
Combo_SDQSE.Items.Add("Select...");
Combo_SDQSE.Items.Add("Commission");
Combo_SDQSE.Items.Add("Omission");
break;
}
case 3:
{
Combo_SDQSE.Items.Add("Select...");
Combo_SDQSE.Items.Add("Temporal consistency");
break;
}
default:
break;
}
Pnl_Step3.Visible = true;
}
}
protected void Button1_Click(object sender, EventArgs e)
{
if (FileUpload1.HasFile)
{
FileUpload1.SaveAs(Server.MapPath(@"~\Files\Dataset\" + FileUpload1.FileNa
Button1.Enabled = false;
Pnl_Step2.Visible = true;
XmlTextWriter textWriter = new XmlTextWriter(Server.MapPath(@"~\Files\" +
textWriter.WriteStartDocument();
textWriter.WriteComment("This is a temporary file created and used by the
textWriter.WriteStartElement("Root");
textWriter.WriteStartElement("DatasetName");
textWriter.WriteString(FileUpload1.FileName);
textWriter.WriteEndElement();
textWriter.WriteEndElement();
textWriter.WriteEndDocument();
textWriter.Close();
}
else
{
// please select a dataset and Submit it...
Button1.Enabled = false;
```

```
Pnl_Step2.Visible = true;
XmlTextWriter textWriter = new XmlTextWriter(Server.MapPath(@"~\Files\" + "Tem
textWriter.WriteStartDocument();
textWriter.WriteComment("This is a temporary file created and used by the auto
textWriter.WriteStartElement("Root");
textWriter.WriteStartElement("DatasetName");
textWriter.WriteString("stations.xml");
textWriter.WriteEndElement();
textWriter.WriteEndElement();
textWriter.WriteEndDocument();
textWriter.Close();
}
}
protected void Combo_Scope_SelectedIndexChanged(object sender, EventArgs e)
{
Pnl_Step5.Visible = false;
Pnl_Note.Visible = true;
switch (Combo_SDQE.SelectedIndex)
{
case 1:
{
switch (Combo_SDQSE.SelectedIndex)
{
case 1:
{
TextBox1.Text = "The webservice does not support conceptual consistency check
break;
}
case 2:
{
TextBox1.Text = "The webservice does not support topological consistency check
break;
}
case 3:
{
break;
}
case 4:
{
TextBox1.Text = "The webservice does not support format consistency check at t
break;
}
}
break;
}
case 2:
{
```

```
TextBox1.Text = "The webservice does not support completeness check at the
break;
}
case 3:
{
TextBox1.Text = "The webservice does not support temporal accuracy check a
break;
}
}
switch (Combo_Scope.SelectedIndex)
{
case 1:
{
if (Combo_SDQE.SelectedIndex == 1 && Combo_SDQSE.SelectedIndex == 3)
{
TextBox1.Text = "You have chosen to perform the logical consistency check
Pnl_Step5.Visible = true;
}
break;
}
case 2:
{
TextBox1.Text += " The webservice does not support spatial extent scope at
break;
}
case 3:
{
TextBox1.Text += " The webservice does not support object-based scope at t
break;
}
case 4:
{
TextBox1.Text += " The webservice does not support complex scope at the mo
break;
}
}
}

protected void Combo_CheckType_SelectedIndexChanged(object sender, EventAr
{
Pnl_FieldType.Visible = false;
Pnl_InformationAboutFieldTypes.Visible = false;
Pnl_RangeDomain.Visible = false;
Pnl_CodedDomain.Visible = false;
Pnl_Result.Visible = false;
Combo_DomainType.Items.Clear();
switch (Combo_CheckType.SelectedIndex)
```

```
{
case 1:
{
Combo_DomainType.Items.Add("Select...");
Combo_DomainType.Items.Add("Information about the field types");
Combo_DomainType.Items.Add("Field type check");
break;
}
case 2:
{
Combo_DomainType.Items.Add("Select...");
Combo_DomainType.Items.Add("Range domain");
Combo_DomainType.Items.Add("Coded domain");
break;
}
}
Combo_DomainType.Visible = true;
}
protected void Quality_Evaluation_Check()
{
switch (Combo_SDQE.SelectedIndex)
{
case 1:
{
switch (Combo_SDQSE.SelectedIndex)
{
case 3:
{
//do domain consistency check...
Domain_Consistency_Check();
break;
}
default:
{
break;
}
}
break;
}
default:
{
break;
}
}
}
protected void Domain_Consistency_Check()
{
```

```
switch (Combo_CheckType.SelectedIndex)
{
case 1:
{
Field_Type_Check();
break;
}
case 2:
{
Domain_Type_Check();
break;
}
default:
break;
}
}
protected void Domain_Type_Check()
{
switch (Combo_DomainType.SelectedIndex)
{
case 1://range domain check
{
Prepare_Coded_Domain_Panel();
Pnl_RangeDomain.Visible = true;
Pnl_CodedDomain.Visible = false;
Pnl_InformationAboutFieldTypes.Visible = false;
break;
}
case 2://coded domain check
{
Prepare_Coded_Domain_Panel();
Pnl_CodedDomain.Visible = true;
Pnl_RangeDomain.Visible = false;
Pnl_InformationAboutFieldTypes.Visible = false;
break;
}
default:
break;
}
}
protected void Prepare_Coded_Domain_Panel()
{
XmlDocument xmlDocument = new XmlDocument();
xmlDocument.Load(Server.MapPath(@"~\Files\Temp_FileName.xml"));
XPathNavigator nav = xmlDocument.CreateNavigator();
string datasetname = nav.SelectSingleNode("Root/DatasetName").Value.ToStri
int status = 0;
```

```
string root = string.Empty;
XmlTextReader reader = new XmlTextReader(Server.MapPath(@"~\Files\Dataset\"+da
reader.Read();
reader.MoveToElement();
root = reader.Name.ToString();
while (reader.Read())
{
switch (reader.NodeType)
{
case XmlNodeType.Element:
{
for (int i = 0; i <= ListBox1.Items.Count - 1; i++)
{
if (ListBox1.Items[i].Value.ToString() == reader.Name.ToString())
{
status = 1;
}
}
if (status == 0 && root != reader.Name.ToString())
{
ListBox1.Items.Add(reader.Name.ToString());
ListBox5.Items.Add(reader.Name.ToString());
ListBox8.Items.Add(reader.Name.ToString());
ListBox10.Items.Add(reader.Name.ToString());
}
break;
}
}
}
}

protected void Coded_Domain_Check()
{
string[] Inconsistency_array = new string[ListBox4.Items.Count];
for (int i = 0; i <= ListBox4.Items.Count - 1; i++)
{
if (ListBox4.Items[i].Value.ToString() == "Null")
{
Inconsistency_array[i] = " ";
}
else

Inconsistency_array[i] = ListBox4.Items[i].Value.ToString();
}
XmlDocument xmlDocument = new XmlDocument();
xmlDocument.Load(Server.MapPath(@"~\Files\Temp_FileName.xml"));
XPathNavigator nav = xmlDocument.CreateNavigator();
```

```
string datasetname = nav.SelectSingleNode("Root/DatasetName").Value.ToStri
int inconsistencies = 0;
int Total_Value = 0;
string element = ListBox1.Items[ListBox1.SelectedIndex].Value.ToString();
string atribute = ListBox2.Items[ListBox2.SelectedIndex].Value.ToString();
XmlTextReader reader = new XmlTextReader(Server.MapPath(@"~\Files\Dataset\
while (reader.Read())
{
switch (reader.NodeType)
{
case XmlNodeType.Element:
{
if (reader.Name.ToString() == element)
{
reader.MoveToFirstAttribute();
if (reader.Name.ToString() == atribute)
{
Total_Value += 1;
for (int k1 = 0; k1 <= Inconsistency_array.Length - 1; k1++)
{
if (reader.Value.ToString() == Inconsistency_array[k1])
{
inconsistencies += 1;
break;
}
}
}
else
{
while (reader.MoveToNextAttribute())
{
if (reader.Name.ToString() == atribute)
{
Total_Value += 1;
for (int k2 = 0; k2 <= Inconsistency_array.Length - 1; k2++)
{
if (reader.Value.ToString() == Inconsistency_array[k2])
{
inconsistencies += 1;
break;
}
}
}
}
}
}
}
```

```
break;
}
}
int cl = int.Parse(Txt_Cl.Text);
switch (Combo_CL.SelectedIndex)
{
case 0:
{
if (inconsistencies > cl)
{
Lbl_Result.Text = "Dataset Failed! ";
Result = "Fail";
}
else
{
Lbl_Result.Text = "Dataset Passed! ";
Result = "Pass";
}
break;
}
case 1:
{
if ((inconsistencies * 100 / Total_Value) > cl)
{
Lbl_Result.Text = "Dataset Failed! ";
Result = "Fail";
}
else
{
Lbl_Result.Text = "Dataset Passed! ";
Result = "Pass";
}
break;
}
}
Lbl_Result.Text += "count of inconsistent items: " + inconsistencies.ToString(
Pnl_Result.Visible = true;
XmlTextWriter textWriter = new XmlTextWriter(Server.MapPath(@"~\Files\" + "Tem
textWriter.WriteStartDocument();
textWriter.WriteComment("This is a temporary file created and used by the auto
textWriter.WriteStartElement("Root");
textWriter.WriteStartElement("CL_Type");
textWriter.WriteString(Combo_CL.SelectedItem.ToString());
textWriter.WriteEndElement();
textWriter.WriteStartElement("CL_Value");
textWriter.WriteString(Txt_Cl.Text);
textWriter.WriteEndElement();
```

```
textWriter.WriteStartElement("Incon");
textWriter.WriteString(inconsistencies.ToString());
textWriter.WriteEndElement();
textWriter.WriteStartElement("Result");
textWriter.WriteString(Result);
textWriter.WriteEndElement();
textWriter.WriteEndElement();
textWriter.WriteEndDocument();
textWriter.Close();
}
```

Bibliography

[1] ISO/TS 19103. *Geographic Information-Conceptual schema language*. International Organization for Standardization(ISO), Oslo, Norway, 2005.

[2] ISO 9000. *Quality management systems – fundamentals and vocabulary*. International Organization for Standardization, Oslo, Norway, 2005.

[3] Life Cycle Assessment. Glossary-entry:data quality, Retrieved 20-10-2010, from http://www.lcacenter.org/LCA/LCA-definitions.html.

[4] The Brownsfield and Land Revitalization Technology Support Center. Glossary-entry:data quality, Retrieved 20-10-2010, from http://www.brownfieldstsc.org/glossary.cfm.

[5] Lemmens R. Morales J. de By, R. *A skeleton design theory for spatial data infrastructure.* Earth Science Informatics, 2009.

[6] ArcGIS 9.2 desktop help. An overview of attribute domains, Retrieved 10-10-2010, from http://webhelp.esri.com/arcgisdesktop/9.2/index.cfm?topicName=An_overview_of_attribute_doma

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

[8] Clementini E. Di Felice P. Egenhofer, M.J. *Topological relations between regions with holes*. International Journal of Geographical Information Systems 8(2)(1994) 129-144, 1994.

[9] Sharma J. Egenhofer, M.J. *Topological consistency*. Fifth International Symposium on Spatial Data Handling, 1992, pp.335-343, 1992.

[10] Sharma J. Egenhofer, M.J. *Assessing the consistency of complete and incomplete topological information*. Geographical Systems 1(1) (1993) 47-68, 1993.

[11] G.M. Foody. Status of land cover classification accuracy assessment. 80:185–201, 2002.

[12] GML. *OpenGIS Geography Markup Language (GML) Implementation Standard*. Open Geospatial Consortium, 2007.

[13] de By R.A. Huisman, O. *Principles of geographic information systems : an introductory textbook (Fourth edition)*. Enschede: ITC, 2009.

[14] 19113 ISO/TC 211. *Geographic Information-Quality Principles*. International Organization for Standardization(ISO), Oslo, Norway, 2002.

[15] 19114 ISO/TC 211. *Geographic Information-Quality evaluation procedures*. International Organization for Standardization(ISO), Oslo, Norway, 2003.

[16] 19115 ISO/TC 211. *Geographic Information-Metadata*. International Organization for Standardization(ISO), Oslo, Norway, 2003.

[17] 19138 ISO/TS. *Geographic Information-Data quality measures*. International Organization for Standardization(ISO), Oslo, Norway, 2006.

[18] 19139 ISO/TS. *Geographic Information - Metadata - XML schema implementation*. International Organization for Standardization(ISO), Oslo, Norway, 2007.

[19] Kim T.W. Li K.J. Kang, H.K. *Topological consistency for collapse operation in multi-scale databases, in S.Wang et al. (Eds.)*. 23rd International Conference on Conceptual Modeling, Lecture notes in Computer Science 3289, 2004, pp. 91-102, 2004.

[20] Information Management and SourceMedia. Glossary-entry:data quality, Retrieved 20-10-2010, from http://www.information-management.com/glossary/d.html.

[21] Shea K.S. McMaster, R.B. *Generalization in digital cartography*. Association of American Geographers, 1992.

[22] H. Moellering. *The Proposed Standard for Digital Cartographic Data*. The American Cartographer 15(1), 1998.

[23] Accounts Chamber of the Russian Federation. Key national indicators:draft guide to terms and concepts, Retrieved 20-10-2010, from http://www.ach.gov.ru/userfiles/tree/OECD-GAO.doc.

[24] Instructional Assessment Resources. Glossary-entry:data quality, Retrieved 20-10-2010, from http://www.utexas.edu/academic/ctl/assessment/iar/glossary.php.

[25] American society for quality. Glossary-entry:quality, Retrieved 20-10-2010, from http://www.asq.org/glossary/q.html.

[26] Csaplewski R.L. Stehman, S.V. *Design and analysis for thematic map accuracy assessment: fundamental principles*. Remote Sensing of Environment 64 (1998) 331-344.

[27] Stylus studio. Syntactic and conceptual schemas, Retrieved 10-10-2010, from http://www.stylusstudio.com/xmldev/199908/post20200.html.

[28] P. van Oort. *Spatial data quality : from description to application*. Wageningen University, Wageningen, 2006.

[29] W3C. Web ontology language (owl), Retrieved 20-10-2010, from http://www.w3.org/TR/owl-features/.

[30] WFS. *Web Feature Service Implementation Specification*. Open Geospatial Consortium, 2009.