A UNIFIED DATA MODEL FOR SPATIAL DATA ACCESS

BANCHIALEM MELKE TSADIK
February, 2011

SUPERVISORS:

Dr. J.M. (Javier) Morales
Dr. Ir. R.A. (Rolf) de By

A UNIFIED DATA MODEL FOR SPATIAL DATA ACCESS

BANCHIALEM MELKE TSADIK
Enschede, The Netherlands, February, 2011

Thesis submitted to the Faculty of Geo-information Science and Earth Observation of the University of Twente in partial fulfilment of the requirements for the degree of Master of Science in Geo-information Science and Earth Observation.
Specialization: Geoinformatics

SUPERVISORS:

Dr. J.M. (Javier) Morales
Dr. Ir. R.A. (Rolf) de By

THESIS ASSESSMENT BOARD:

Dr. Ir. R.A. (Rolf) de By (chair)
Dr. T. (Theodor) Foerster

ABSTRACT

Data kept in various organizations sow different levels of heterogeneity due to fragmentation. Various segments in the organizations experience different data heterogeneity such as syntactic heterogeneity, schematic heterogeneity and semantic heterogeneity. This research deals with schematic heterogeneity so that the data sources are categorized into three groups such as structured data , semi-structured data and unstructured data based on the characteristics of their schemas. The research particularly deals with the schematic heterogeneity of structured and semi-structured data so as to facilitate data integration. Spatial Data Infrastructures has a purpose in facilitating spatial data use and sharing, and can be an effective platform to aid in data integration. This research aim at providing an interface that allows a unified way to access and use disparate data sources available at various SDI nodes. Schema mapping and transformation techniques are used to overcome schematic heterogeneity and as a means of facilitating heterogeneous spatial data integration. The schema mapping process is done at the schema level and the schema transformation process is done at the data level. In schema mapping process the the mapping rules from source to target schema are specified based upon the query expression posed on the target schema. This specification is done by using OML(Ontology Mapping Language). The actual data transformation process is based on the mapping rules and undertaken in two steps such as data extraction from the required data sources, execution of transformation functions on the extracted data to transform the data into the target. The result of this research can be used as one functionality within an SDI framework for effective data integration. The result also aim to assist users to follow the same procedure or methodology used for resolving schematic heterogeneity of their data sources and get integrated information for their spatial applications.

Keywords

spatial data heterogeneity,structured data,semi-structured data,Spatial Data Infrastructure,spatial data integration, schema mapping, schema transformation

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ACKNOWLEDGEMENTS

Foremost, I would like to thank GOD for everything he has done in my life. I would like to express my sincere thanks to my first supervisor Dr. J.M. Morales, for his unlimited support and guidance throughout the development of this research. I also want to thank my second supervisor Dr.Ir.R.A. (Rolf) de By, for his support and advice. I wish to express my gratitude for the Netherlands government for giving me the golden chance to pursue my MSc study in Geoinformatics. I want to say thank you to all my classmates and friends for all the moments we share together. I would like to thank my best friends(*'wo Group'*, *yechi agatami beand agenagntan...*) for your amazing friendship and being there for me. My heartfelt thanks go to all my loving family members for their warm affection, support and continuous encouragement.

Chapter 1
Introduction

1.1 MOTIVATION AND PROBLEM STATEMENT

Organizations such as the Bureau of Statistics or a local government planning office commonly keep their data as a highly heterogeneous set of resources that they use in the execution of their functions. These data sources are of different types including structured data such as relational data organized in a database, unstructured data such as images, documents etc, and other semi-structured data such as XML documents [18]. In addition, the heterogeneity of these data sources has three aspects such as syntactic heterogeneity, which is the result of different languages used for modeling the various sources, schematic heterogeneity, which caused by different data structures of source schemas, and semantic heterogeneity, which arises as a result of incompatibility in meanings of data as described in [30]. Furthermore, such data are typically stored in files scattered among multiple file systems (local or network), multiple machines (local desktop, network share, mail server, other kinds of server), and most of all different file formats (xml, shape file,etc.).

Different organizations create and maintain spatial data and deliver it to spatial data decision makers as spatial data plays a significant role in decision making. When more spatial datasets and integrated spatial data for different spatial applications needed, integration of spatial data from different sources will be required. The data sources can be provided by diverse data providers and this leads to data heterogeneity in different aspects based on different standards, policies and organizational arrangements. And, this is a cause of spatial data to have different schema which hinders spatial data integration. Even though the importance of spatial data integration is important for many spatial applications, the fragmentation of the organizations that are in charge of producing and managing different datasets has caused heterogeneities and inconsistencies of spatial data from different aspects.

Integration of heterogeneous data source is crucial for organizations that own a multiple of data sources as data integration facilitates access of highly distributed heterogeneous information sources. In this research we address a problem which is a challenge in integrating heterogeneous and disparate spatial data sources. This problem is the incompatibility of different data source schemas or structures when considering different types of data sources. Thus, the goal the data integration system is to provide users with a uniform interface to these heterogeneous and disparate data sources. The uniform interface is based on a common application schema(target schema) which is based on the users' interest. The user pose queries on the target schema rather than directly on source schemas. There are different approaches for building such a data integration system. Global-as-View (GAV) and Local-as-View (LAV) are the two important ones. In the GAV approach, every object in the global schema is related with a view over the source local schema where as in the Local-as-View (LAV) approach local schemas are defined as views over the global schema [5]. The two approaches are main initiatives to integrate data and answer queries without materializing the global schema . The query reformulation in GAV approach lessens to simple rule unfolding while in LAV the query reformulation has exponential time complexity with respect to query and source schema definitions. Auto Med is another approach to heterogeneous

data integration based on the use of reversible schema transformation sequences, which offers the ability to handle data integration across heterogeneous data sources [2].

Much of the work on data integration has focused on the real time integration of structured data sources such as databases. The focus in integrating structured data ,semi-structured and unstructured data is not as much as integrating structured data though huge amount of spatial data available in different spatial data types. These enormous amount of spatial data sources demands the exploitability of integrating heterogeneous data sources for the provision of integrated information to different spatial applications. Here the challenge is the heterogeneity of the data sources to be integrated. Thus, in order to fully exploit the combination of these various types of data sources the three data heterogeneity such as syntactic,schematic and semantic heterogeneity has to be addressed as of the user requirement. However, overcoming all the three types of data sources heterogeneity is a difficult task with short period of time.

The demand for the integration of heterogeneous and disparity data sources within Spatial data Infrastructures (SDIs) framework is highly increasing because of the availability of data sources with different schemas at the various SDI nodes. This paper studies how to overcome the schematic heterogeneity of the data sources for data integration purpose. Thus, it is important to have a method that exploit data from such kind of data sources through transforming and combining the resources. Schema transformation plays a great role in this part by transforming data from various source schema to the target schema.

The accessibility of consistent and compatible spatial data, from various and disparate data sources has a great role for organizations to achieve their integrated information requirement for their strategic purposes. To this end, spatial data infrastructure (SDI) has a significant impact in facilitating integration of those data stored in different SDI nodes. However, from an SDI point of view the disparate data sources at various SDI nodes pose challenges in integrating and accessing the data [11]. Hence, we need a seamless mechanism that can exploit the heterogeneous data as if they are from a unified source and offer the user an access to multiple information sources in a unified way. Thus, a powerful unified data model is strongly required to represent the disparate and heterogeneous mix of data to provide the user with a unified view of those data and access in a unified way. More importantly, the model will enable the extraction of information that would otherwise be impossible.

The incompatibility and inconsistency of the data because of the high heterogeneity and disparity of the data sources brought a big difficulty to obtain and fuse the required information. A major obstacle in integrating such heterogeneous data sources within the organizations is due to their schematic heterogeneity, which results from different structures of source schemas that hamper exploitability of the data source. The availability of disparate data sources at various Spatial Data Infrastructure (SDI) nodes is also a problem that needs to be solved [11].

1.2 RESEARCH IDENTIFICATION

1.2.1 Research objectives

The objective of this research is to provide a unified way to access and use the disparate data sources available at various SDI nodes. To realize this objective the following sub-objectives are defined:

- To define a classification mechanism for the heterogeneous data in terms of structured, semi-structured and unstructured data to facilitate formalization of schema mappings and transformations of the data sources.

- To define integrated data model to represent and access heterogeneous data in the data sources.
- To specify complete , correct and functional querying mechanism to retrieve information from the available data sources through the integrated model.
- To define a procedure to be followed by the user to get a unified access to the available heterogeneous and disparate data source.

1.2.2 Research questions

To achieve the objectives, the following research questions need to be answered:

1. What classification mechanism can be defined to classify the heterogeneous data source in terms of structured, semi-structured and unstructured data to facilitate formalization of schema mappings and transformations of the data sources?
2. What is the best and suitable approach for schema merging process to derive integrated data model to facilitate data integration in a unified way?
3. What is the best and suitable approach for schema merging process to derive integrated data model to facilitate data access in a unified way?
4. How to specify complete, correct and functional querying mechanism to retrieve information from the available data sources through the integrated model?
5. How to test the quality of the integrated data in the application schema?
6. How to evaluate and test the our schema transformation methodology?

1.2.3 Innovation aimed at

The novelty of the research aimed at defining a methodology that will allow us to represent heterogeneous and disparate data sources available at various SDI nodes in a unified model for better data exploitability.

1.2.4 Related work

Several solutions have been proposed to exploit the availability of heterogeneous data sources using different integration techniques. In one work [3] an integration system for heterogeneous spatial datasets from various sources and different types is proposed. The proposed integration algorithm perform integration of two heterogeneous vector datasets and integration of vector and raster dataset and linked to a federated database. Our alternative implementation differs from that work in such a way that it focuses on providing integrated information from schematically heterogeneous spatial data sources such as some of the datasets have regular and fixed schema and some do not. Moreover, we have used schema transformation method for the integration of schematically heterogeneous spatial data sources.

In [1] a framework that supports data knowledge and data integration at schema and data level is proposed. The data integration at the schema level ensures the availability of a single integrated dataset where as the integration at the data level ensures the data quality. The fact that this work dealing with integrating schematically heterogeneous data sources is the similarity between our work and their work. In the contrary, our work does not maintain the physical integration of data sources. In addition, the quality of the integrated data is not covered in this study.

A real time data integration with the aim of building location based service is discussed in [10]. The integration system used to provide harmonized cartographic data from different data sets stored in different databases through schema transformation. But in the case of our research the datasets are from different data sources in which some are in relational database(structured data) and some are in XML(semi-structured data).

An approach for spatial data integration in the SDI frame work is studied in [22]. In this paper both the technical and non-technical challenges the hinder spatial data integration an SDI environment is studied. The technical challenges are those data source heterogeneities such as syntactic, semantic and schematic heterogeneity. The non technical challenges are those problems related with institutional, policy, legal and social issues. Even though our research deals with data integration system in an SDI environment by resolving schematically heterogeneous data sources available at various SDI nodes, the non technical issues is out of our concerning issue.

In one work [19] the integration methodology is presented under an XML global schema; however, it adopts LAV (local as view) approach which is different from ours. In addition, the methodology integrates only relational and tree-structured data sources in particular XML documents. In contrast, our integration approach will be able to handle multiple heterogeneous spatial data sources such as relational database, file and web service using schema mapping and transformation method.

An approach called both as view (BAV), which incorporates data integration techniques such as local-as-view(LAV) and global-as-view(GAV) as presented in [21]. LAV is an approach in which local schemas are described as views over the global schema where as GAV is extracting definitions of the local schemas as views over the global schema. The paper also discusses the ongoing implementation of the BAV approach within the Auto Med project. [2] is mentioned in Auto Med approach in which both as view (BAV) approach incorporates data integration techniques such as global-as-view(GAV), local-as-view(LAV) and global-local-as-view(GLAV) to integrate heterogeneous data source. Global-local-as-view(GLAV) is an approach that combines both LAV and GAV by allowing the definition of the schemas independent of the particular details of the source as defined in [7]. The paper also describes that Auto Med approach handles a wide range of data models in the integration process which opposed our approach that assumes the integration will be performed using a common model.

In [24] a merge algorithm based on the generic role-based metamodel (GeRoMe), a generic metamodel in which each model element is associated with set of role objects that represents specific features of the model element [14], and intentional mappings is given. The merge operator based on GeRoMe can merge models from different Metamodel such as XML Schema and Relational. On the contrary, our merging approach will be based on extensional mappings since they are used to extract data from a source and transform it into a target schema but intentional mappings do not refer explicitly to instances of models and hence they can not be used for data translations.

1.3 PROJECT SET-UP

1.3.1 Method adopted

We can define the following steps for attaining the research objectives.

1. Literature review: Read some literatures to get more understanding of the three types of data sources such as structured, semi-structured and unstructured data; schema mapping and transformations and schema integration process.

2. Data source Classification: Data source classification will be done by searching a suitable classification technique in literature.

3. Schema Analysis: To map the schema of the data sources to the target XML schema it is necessary to derive significant properties from schemas by using the proposed techniques and methodology of conceptual schema analysis in [6].

4. Generic Schema Mapping and Transformations: Analyze the different schema mapping languages such as XSLT(Extensible Stylesheet Language Transformations), OWL(Web Ontology Language), OML(Ontology Mapping Language) and address pertinent schema mapping Languages.

5. Schema Integration: Analysis, design and implementation of the different approaches such as generic role based metamodel GeRoMe to schema merging. Analysis of schema integration tools such as PROMPT and COMA will be done.

6. Specification of querying mechanism: Analyze querying mechanisms based on correctness, completeness and functionality and specify the efficient ones that will be acted through the integrated model.

7. Procedure Definition: Defining the procedure to be followed by the user when s/he want to get integrated information from the available heterogeneous and disparate data sources.

8. Test and evaluate the derived unified data model and the defined procedure: The model and the procedure will be tested by using sample data sources from relational database, file and web service.

Chapter 2

Spatial Data Access and Integration in the context of SDI

In this chapter we will give a general overview of data heterogeneity , data exploitability and integration and the different approaches for spatial data accessibility in the context of SDI(Spatial Data Infrastructure).

The development of SDI as an enabling platform has a purpose to assist people to access, use and integrate spatial data effectively.

Spatial data which forms the basic component of a spatial data infrastructure is used by different users through different mechanisms. Implementation of spatial data infrastructure is important in which data exploitability can be achieved with the help of different functional systems. In an SDI environment making spatial data exploitability easier is an important issue as it has a great role in minimizing cost and effort in data collection, processing and management. Thus, the improvement of spatial data exploitability is crucial in research areas as well as in many other areas. Improved usability of spatial data increases the quality of decisions made using the available spatial data. It is one of the SDI's purpose to facilitate greater access and use of spatial data as the availability and provision of spatial data is high. Eventhough spatial data available at SDI nodes made reachable by the implementation of SDI, exploitation of those spatial data is a problem because of the heterogeneity of the data sources. And thus one of the requirements of SDI implementation is easier exploitability of spatial data which needs great attention to be solved.

Data accessibility has different definitions with different domains and contexts and we define it in a way applicable to our study. In this research access to spatial data is understood as the provision of spatial data through the creation of an interface that retrieve data from heterogeneous and disparate data sources available at various SDI nodes and present it to the user in his or her expected format or application schema. Therefore, to create such an interface we have to deal with the various levels of data heterogeneity and the different mechanisms to handle those heterogeneities.

Organizations store their spatial data according to a certain conceptual view of that part of reality based on the relevance of their functionality. Thus the decision for the schema or structure of their data depends on the purpose of collecting and creating it [13]. Hence, the schema of data sources provided by different organizations which is considered as an SDI node, is different. This in turn is a bottleneck for combining spatial data having different schema.

2.1 DATA HETEROGENEITY

The availability and provision of spatial data from different sources becomes higher. Spatial data is being provided by different organizations with their own format. This leads to a situation that the spatial data may differ semantically, syntactically and schematically as they have been collected and stored differently.

2.1.1 Syntactic heterogeneity

Syntactic heterogeneity is the result of differences in storage formats and software incompatibility that is also a difference in modeling languages. It is various data models used for data storage and access. It is a technical issue which can be addressed by technical means, i.e. syntactic heterogeneity includes issues that are associated to the choices made on how to represent data. Several attempts have been made to tackle this heterogeneity of spatial data by bringing the data in common format. There are different Methods that allow syntactically heterogeneous data sources to communicate such as object-oriented approaches, data warehousing and mediators and ontologies. Even though syntactic heterogeneity poses many technical challenges, and it has been studied in different research areas, there exists an even more difficult and profound problem of heterogeneity for data integration systems, called semantic heterogeneity.

2.1.2 Semantic heterogeneity

Semantic heterogeneity caused by the incompatibility in interpretation of the data. It is the differences in naming conventions and conceptual groupings in different organizations. In general, semantic heterogeneity is a way of naming identical real world concepts differently and it is strongly affected by subjective criteria of the organizations that modeled the concept. Semantic heterogeneity of the data sources is a challenge in integrating data from different sources. Usually organizations create or collect their data independently so that it is quite unnatural to expect that they will use the same terminology for data properties. Organizations use the concepts and terminology specific for their respective field of expertise, and use different parameters and different languages to express their model of a concept. For example roads in one organization can mean quite something different for another organization. These concepts can be interpreted by humans using their common sense or knowledge. But software systems usually do not have any knowledge about the world and have to explicitly be told how to translate one term into another.

Data level semantic heterogeneity can be subdivided into naming and cognitive heterogeneities [1]. Naming heterogeneities caused by naming semantically identical data objects differently and it can be relatively easily resolved with a thesaurus. Different organizations, have different views of the real world so that they describe similar real world objects from their own perspectives. The other subdivision is cognitive heterogeneities which is caused by naming different data objects similarly. For example, a road can be understood as a link in a topological transportation network in traffic management agency where as in the utility industry it can be understood as a surface with different engineering properties, reinstatement issues and access constraints. This kind of heterogeneity can be solved by ontology mapping.

2.1.3 Schematic heterogeneity

Schematic heterogeneity means that different information systems store their data in different structures(e.g. data stored in XML schema, relational database schema). It is resulted from different structures of source schemas which is heterogeneity of conceptual structures of data sources. In addition such heterogeneity can be the result of having different data organizations such as aggregation or generalization hierarchies. Schematic heterogeneity occur when modeling similar application concepts using different data model concepts. Generally, it is the differences in data model between different organizations. Organizations use their own data model(schema) to reflect their own view of functionalities. Thus, different organizations use different structures to refer to similar real world objects. As a result heterogeneities can occur because of the different domain perceptions, business logic and interests of different user groups. And, this affect the way that the data is represented. Addressing schematic heterogeneity is the main concern of this

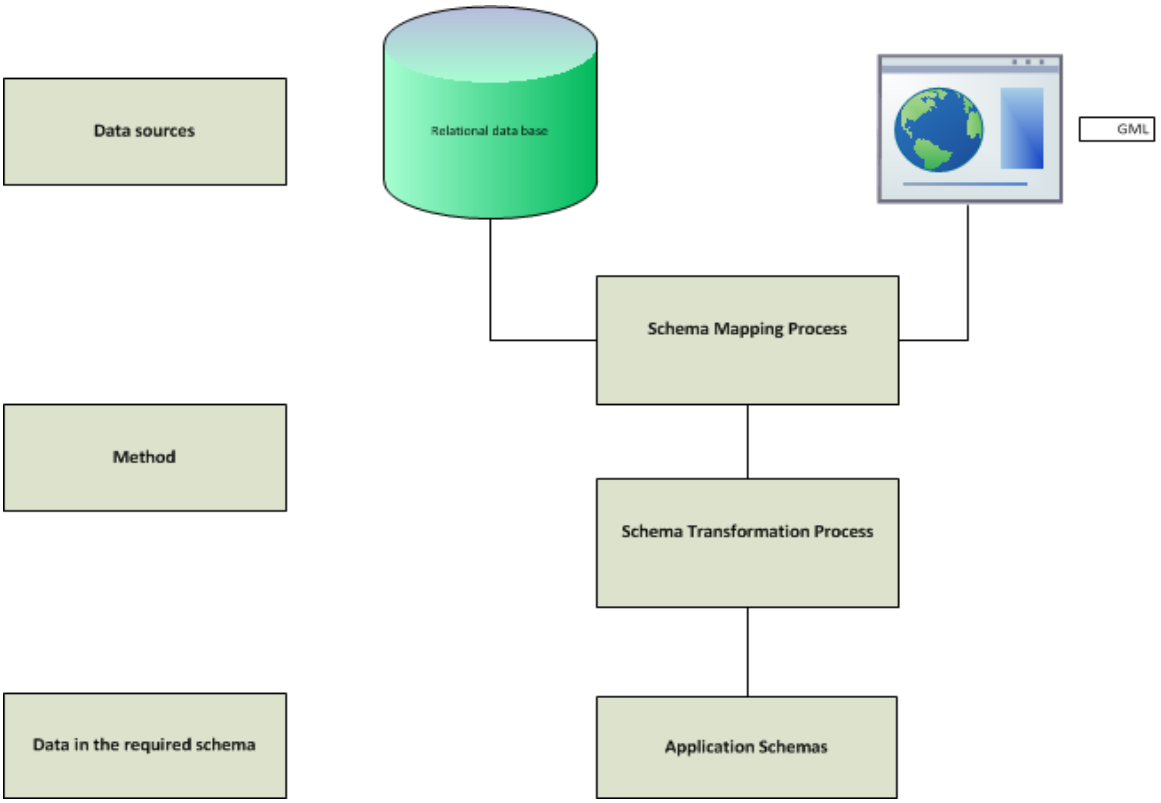

Figure 2.1: Data exploitability

research through the consideration of different data formats such as XML document, relational database.

This heterogeneous state of the data within an SDI environment leads to data incompatibility which is a major hindrance in accessing of data. There should be a mechanism to handle this data incompatibility within an SDI environment. And thus, the finding of this research will be used as one of the functionality in an SDI environment.

The enablement of data exploitability requires a method to tackle schematic heterogeneity mentioned above. Thus,our research deals with overcoming the difference in data model so as to facilitate data exploitability to different applications. However overcoming schematic heterogeneity does not necessary mean that the schema of the existing data sources has to be changed rather it is like creating a method to retrieve the required data from the data source without harmonizing its schema,see figure 2.1.

Spatial data with different data models or schemas can be grouped with three categories based on their schema type characteristics in order to indicate the specific type of data source that our research considers. The types are discussed as follows. They are structured data, semi-structured data and unstructured data [9]. Such heterogeneities is known as schematic heterogeneity which is the focus of this research.

2.1.3.1 Structured Data

Structure data is a type of data that is organized in structure so that it is traceable or searchable. It is organized into semantic entities, with similar entities grouped together in relations or classes. The structure is standardized and determined by a data model. If the same data model is used to

gid [PK] serial	tdn_code smallint	type character var	shape_leng numeric	shape_area numeric	the_geom geometry
1	5022	decidious forest	39.2650415412	86.5317762834	0106000020407
2	5023	decidious forest	2005648.67367	19272038.1422	
3	5053	coniferous fores	324845.024511	7574696.41616	
4	5063	mixed forest	873864.399362	20306497.7402	
5	5073	other	116.386088472	703.045947712	0106000020407
6	5083	decidious forest	1515.56321523	32417.4541487	0106000020407
7	5203	arable land	1114955.66055	49694350.7813	
8	5212	meadow	1580.97234733	6967.80598765	0106000020407
9	5213	meadow	3321022.37833	105515668.200	
10	5223	orchard	2081.17886151	27223.9853676	0106000020407
11	5233	other	71586.4967489	1522189.33823	

Figure 2.2: An example of structured data

store and access the data then sharing the data will be possible. There are different technologies that can manage a large amount of structured data and allow data access through querying against predetermined data types and relationships. Example of structured data is given in figure 2.2.

Structured data have the following characteristics as described in [29]:

- Data is organized in semantic entities.
- Similar entities are grouped together(relations or classes).
- Entities in the same group have the same descriptions(attributes).
- Descriptions for all entities in a group(schema)have the same defined format and a predefined length, they are all present and follow the same order.

2.1.3.2 Semi-structured Data

Semi-structured data is a kind of structured data that does not conform with the formal structure of tables and data models associated with relational databases. In semistructured data, the information that is associated with a schema is contained within the data, which is sometimes referred as self-describing. In some forms of semistructured data there is no separate schema, in others it exists but only places loose constraints on the data. Semi-structured data contains tags or other markers to separate semantic elements and hierarchies of records and fields within the data. The usage of semi-structured data is increasing dramatically as it is used for data exchange among ,and integration of, heterogeneous data sources.

The characteristics of semi-structured data are given as follows as described in [4]:

- Organized in semantic entities
- Similar entities are grouped together
- Entities in the same group may not have the same attributes
- Order of attributes not necessarily important
- Not all attributes may be available
- Size of same attributes in a group may defer
- Type of same attributes in a group may defer
- No fixed schema(schema less) and contains structure information in itself (self-describing)
- Structure is implicit and irregular

Example of semi-structured data is given below. The data is a land use data about Enschede in GML format.As we can see from the partial example of semi-structured data, the identical entities(feature objects) such as main roads are not represented with the same attributes.In addition some attributes in the given feature object, are missing in the other one.This implies that it is difficult to organized such data in a database system as it does not conform to the characteristics of structured data.

```
<?xml version="1.0" encoding="UTF-8"?>
<gml:FeatureCollection xmlns:gml="http://www.opengis.net/gml"
xmlns:xlink="http://www.w3.org/1999/xlink"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:fme="http://www.safe.com/gml/fme"
xsi:schemaLocation="http://www.safe.com/gml/fme intgratedgmlfile.xsd">
<gml:boundedBy>
<gml:Envelope srsName="EPSG:28992" srsDimension="2">
<gml:lowerCorner>248532.999990321 464726.310002701</gml:lowerCorner>
<gml:upperCorner>263902.653607171 478450.03926584</gml:upperCorner>
</gml:Envelope>
</gml:boundedBy>

<gml:featureMember>
<fme:e_mainroads gml:id="id210a47ee-2d93-4f27-abcb-df2024336972">
<fme:OBJECTID>2</fme:OBJECTID>
<fme:ENTITY>Polyline</fme:ENTITY>
<fme:LAYER>Main Roads</fme:LAYER>
<fme:ELEVATION>0</fme:ELEVATION>
<fme:THICKNESS>0</fme:THICKNESS>
<fme:Shape_Leng>869.091328498</fme:Shape_Leng>
<gml:curveProperty>
<gml:LineString srsName="EPSG:28992" srsDimension="2">
<gml:posList>261854.380001111 470975.530008751 261943.220004593 470968.530013
</gml:LineString>
</gml:curveProperty>
```

```
</fme:e_mainroads>
</gml:featureMember>

<gml:featureMember>
<fme:e_mainroads gml:id="idb8feb454-42d2-4290-b66f-c60f658cef57">
<fme:OBJECTID>3</fme:OBJECTID>
<fme:LAYER_NAME>Main Roads</fme:LAYER_NAME>
<fme:ELEVATION>0</fme:ELEVATION>
<fme:COLOR>7</fme:COLOR>
<fme:Shape_Leng>370.55676136</fme:Shape_Leng>
<gml:curveProperty>
<gml:LineString srsName="EPSG:28992" srsDimension="2">
<gml:posList>261485.589991304 471011.499998409 261499.089990741 471009.559998535
</gml:LineString>
</gml:curveProperty>
</fme:e_mainroads>
</gml:featureMember>

<fme:e_mainroads gml:id="ide1be028c-5181-4729-a429-717028b1d978">
<fme:OBJECTID>4</fme:OBJECTID>
<fme:NAME>Main Roads</fme:NAME>
<fme:THICKNESS>0</fme:THICKNESS>
<fme:Shape_Leng>673.282815477</fme:Shape_Leng>
<fme:Shape_Le_1>673.28281748</fme:Shape_Le_1>
<gml:curveProperty>
<gml:LineString srsName="EPSG:28992" srsDimension="2">
<gml:posList>262166.280005592 470995.090014527 262066.629988002 471003.559999639
</gml:LineString>
</gml:curveProperty>
</fme:e_mainroads>
</gml:featureMember>
<gml:featureMember>
</gml:FeatureCollection>
```

2.1.3.3 Unstructured Data

Unstructured data has no predefined structure or does not fit well into relational tables , that is data do not easily conform to standard data structures which is well defined schema. Generally unstructured data does not follow any format or sequence. It does not follow any rule. Understanding of such data requires human intervention for interpretation.Unstructured Data can be textual or non textual data such as documents, multimedia content, maps and geographic information, satellite imagery, and web content such as HTML [4]. It is difficult to use data from such category compared to structured data. Example of unstructured data is given in figure 2.3.

2.2 DATA INTEGRATION IN THE CONTEXT OF SDI

The integration of heterogeneous and disparate spatial data is critical to the delivery of the objectives of spatial services which is integrated information. The demand is also growing dramatically

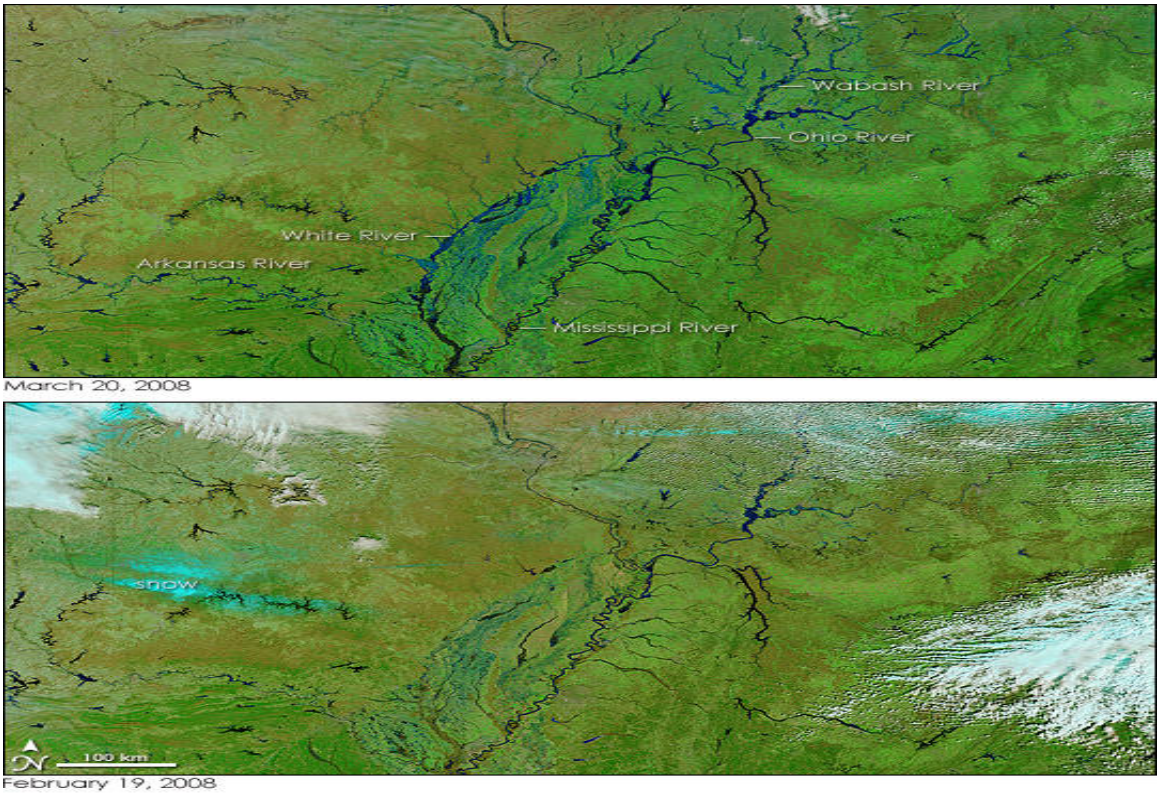

Figure 2.3: An example of unstructured data,MODIS Satellite Flood Image

by different organizations to make effective decisions which are highly dependent on integrated information. Thus, spatial data plays a critical role in informative spatial decision making. However, it is often difficult and sometimes impossible for users to integrate datasets from different sources. This is because of the diversity of data models, semantics and syntax that are used by organizations for their best possible support of the core business processes and requirements which may differ from those of other organizations. Hence, such diversity resulted in inconsistency and heterogeneity of spatial data.

The access and integration of heterogeneous and disparate spatial data is crucial for different spatial applications. However ,integration of heterogeneous and disparate spatial data faces many technical and non-technical problems that hinder effective spatial data integration [22]. Our focus is on the technical perspective and we specifically deal with overcoming the difference in data models used to store spatial data.

Spatial data infrastructure is used to address and overcome the issues and challenges of data integration within its framework with a number of technological components. SDI involves spatial data, people, Internet access and policies and standards to facilitate the integration of heterogeneous and disparate spatial data sources as it is a sharing platform. Thus, integration of spatial data has to be studied and a system has to be developed in the SDI environment.

Generalization, federated database, ontology-based data integration, spatial mediation, spatial ETL(Extract, Transform and Load) and spatial data interoperability can be mentioned as the different approaches for spatial data integration [23]. However, all of them deals with spatial data that can be categorized as structured data according to our definition, see 2.1.3.1. But our approach needs to handle not only structured data but also semi-structured. Therefore, we need to look for another approach which can be used to integrate the two categories of data sources, see 2.1, available at various SDI nodes without altering the schema of the data sources because physical integration of such data sources is impossible. And, the approach has to follow the principle of achieving data harmonization that is a real time data transformation.

2.3 DATA INTEGRATION THROUGH SCHEMA TRANSFORMATIONS

The data transformation process can be discussed at three different levels, i.e. as syntactic, schematic or semantic transformation. In this research we are dealing with data transformation at the schema level. Data transformation at the schema level means modifying the structure and the schema vocabulary of the data model used in the source datasets. To this end, there should be a clear well defined one-to-one, one-to-many or many-to-one mapping from the elements in the source schema to the corresponding elements in the target schema, and it is called schema mapping. This kind of data transformation is well suited to data integration system in an SDI environment where the heterogeneous and disparate data sources are available at various SDI nodes.

Our transformation process is carried out in real time since the process of answering the query posed on the target schema can be done through a web service. The transformations is taken place on spatial data which is a collection of feature objects. The conceptual structure which is used to represent data elements of a given real world object is called a data model where as the description of the model that expressed in some formal modeling language is called the schema of the object [12].

In this research the schema transformation technique refers to a process in which spatial data transformed from one schema into another schema in order to facilitate data integration as shown in figure 2.4 i.e. it is used as a mechanism to enable schematically heterogeneous datasets to be integrated into a common application schema(target schema). we call such process a schematic transformation which is the focus of this research. In this process modification of the structure,

data type and unit system of the source dataset is performed to conform the transformed data to the application schema. The process is undertaken in two steps namely schema mapping and data transformation process, i.e. defining the transformation rules and interpreting the rules. The definition of transformation rules or schema mapping rules is performed at the schema level where as the interpretation of the rules performed at the data instance level. Defining the schema mapping between the source and target schema requires schema description for the source and target schema and the mapping rule. We call this procedure schema mapping process. It is a process of mapping elements of source schema to the corresponding elements of target schema. The schema mapping language used in the research is described in detail in chapter 3. In general, the transformation process involves correspondence determination of source and target schema elements through schema mapping process which can be made with user intervention , semi-automatically, or with total automation and generate schema transformation rules , execution of the rules on source datasets and delivery of the required data sets to the user. The two processes are discussed in chapter 3 and 4 independently.

Data model transformation(schema transformation) is a suitable approach to integrate heterogeneous and disparate spatial data in which data content expressed in one schema transformed into data content expressed in another schema but it does not transform the schema itself [15]. This is done through a process called schema mapping which is a process of determining the correspondence between elements in the source and target schemas before the actual data transformation process taken place. Schema transformation in an SDI environment is important because in an SDI environment spatial data is supposed to be available through the SDI services as a seamless, harmonized spatial content.

The implementation of Spatial data infrastructure has a number of advantages. The provision of integrated spatial information services is one of them that rely on existing spatial data sources. Thus, to provide integrated spatial information services schema transformation plays a great role by addressing lack of data harmonization between various spatial datasets and the difficulty in using such datasets. This can be done by defining a common application schema called target schema in which those data from schematically heterogeneous spatial data sets mapped to this target schema using schema transformation process. To this end, spatial data integration in an SDI environment could be a typical application for schema transformation applications where heterogeneous data sources delivered by different spatial data providers. In schema transformation process within an SDI environment knowledge about source and target schema, underlying standards, used conceptual schema language, basic SDI concepts and much more are required [26].

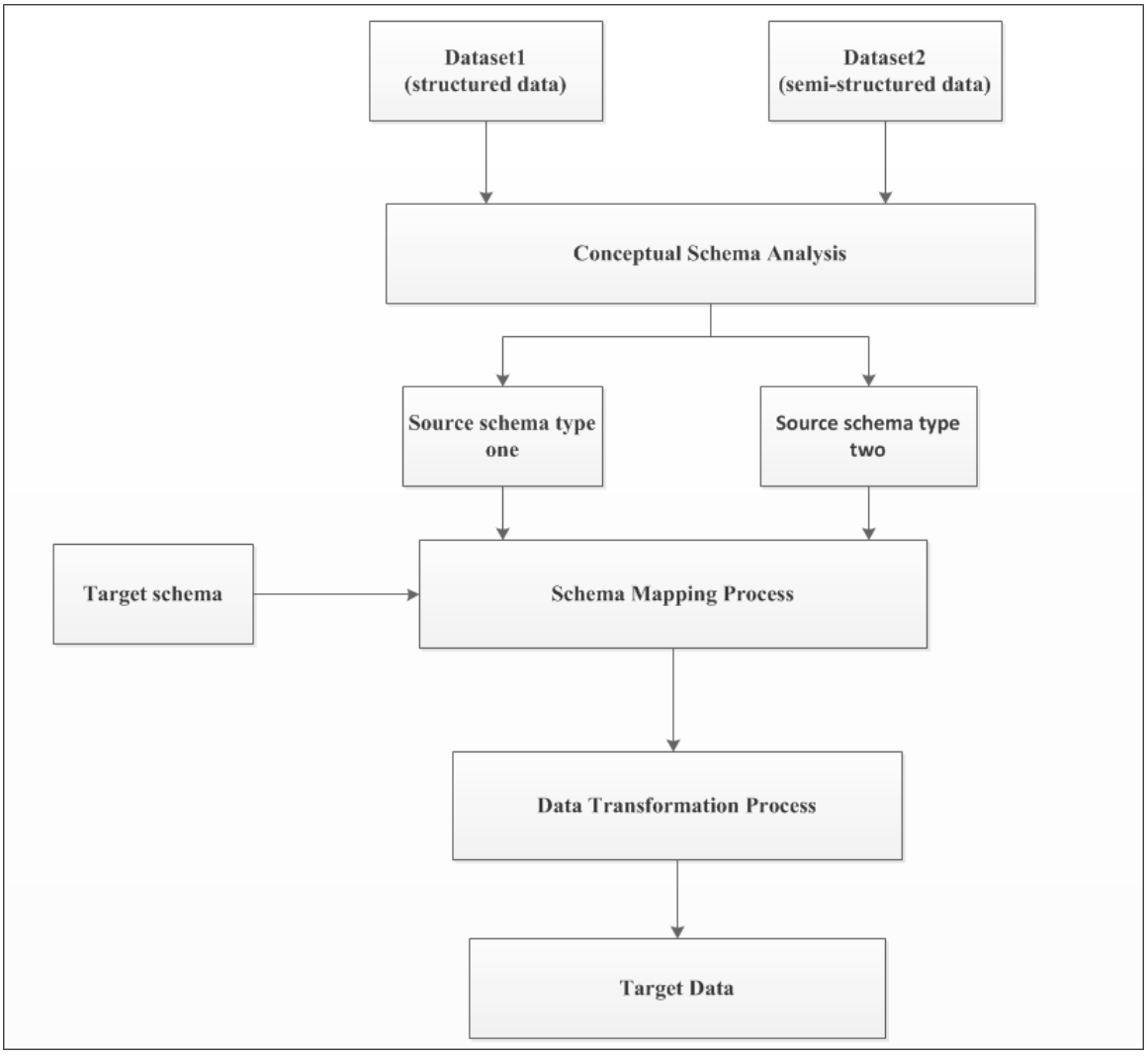

Figure 2.4: schema transformation process

Chapter 3
Schema Mapping Process

This chapter discusses the overall schema mapping process proposed by this research. The chapter also explains in detail on how schema mapping language is used for a schema mapping rule specification.

In order to achieve real time integration of heterogeneous data in an SDI (Spatial Data Infrastructure) environment, a schema mapping needs to be created between a source and a target schema. A schema mapping is an important means to enable spatial data accessibility from a bunch of data sources through a common data model. It can be used as a key technique for the accessibility of data for different applications that depend on data collected from different formats and models as it is used to overcome the challenges caused by the availability of applications with heterogeneous datasets in different schemes, which hamper accessibility of data according to user's requirements. We call such schema differences schematic heterogeneity which is particularly an obtrusive problem. Having different schemas of data sources and being disparate, put a requirement for an involvement of data model mapping in order to achieve the exploitability of the data. For that, schema mapping is one of the solutions to be considered to facilitate data transformation from those disparate datasets with heterogeneous schemas without the physical materialization of the data sources. Therefore, schema mapping is pertinent in overcoming the data's heterogeneity at the level of data models.

Schema mapping is a process of determining the correspondence between the data items in the source and target schemas before the actual data extraction can be performed [15]. That is, the schema mapping process used to construct the mapping rule which are applied at the conceptual level on the source schemas in order to provide the data set in the target schema which is user's application schema. The mapping rules are set of correspondences which are interpreted by a transformation function to transform source data into target data or a filter that selects elements of the source schema. The mapping rules are interpreted as a data transformation using different transformation functions or a filter that selects elements of the source schema in order to materialize the data at the defined target schema. In this research an example of a target schema defined based on a specific spatial application such as Flood Risk Assessment. We have chosen this use case because it is an easy approach that would help us to prove our approach. This target schema requires a bunch of data sets to be populated with.

The schema mapping process proposed in our research make use of the query expression posed on the target schema. The query expression may contain attribute names, the associated attribute name source, the operator and some more. These informations given in the query expression will be used to identify the feature and its attributes, the data source,the transformation function. This process is depicted in figure 3.2.

A schema mapping rules can be expressed in different languages such as XSLT (Extensible Stylesheet Language Transformation) [16], OML(Ontology Mapping Language) [27]. These languages are used to state that data for the target element can be obtained by fetching the data stored at the corresponding source elements based on the specified transformation functions. We have used OML to express our schema mapping rules and we will discuss more about the language in

section 3.2.1.

Schema mapping has a wide range of usages in information integration or data exchange process. It is a key operation for some data processing operations such as data integration, ontology matching, and semantic query processing. It is an important approach to solve the problem of data exchange and data integration as schema mappings constitute the essential building blocks in such data interoperability tasks[8]. schema mapping is one of the mechanisms to solve the problems of data transformation among different sources through schema mapping rule specification between source and target schema[17]. To this end, the definition of schema mapping rule is crucial in such processes. In this research the purpose of schema mapping is to map elements of source schemas to target schema with the goal of transforming the corresponding spatial sets. Thus, our focus is on the technique of conceptual schema mapping to describe the relationship between the two inputs such as source and a target schemas.

3.1 SCHEMATIC HETEROGENEITY

A schema is an exact description of a data model ,where a data model is a conceptual structure to represent data associated to a given real world object, expressed using some formal modeling language [20]. The explosion of different data modeling languages to specify the schema is the one to be mentioned as one of the causes for the diversity aspects of the data source schemas. Different data modeling languages can be the reason for the creation of different schemas to represent data which in turn bring a big difference in the structure of the data. Having differences in the schema of the datasets is an obstacle to get an integrated information in an SDI environment. Thus, we need to find a method through which this problem can be solved potentially.

There are different schema types in information system such as XML schema, database schema. Because of the increasing number and diversity of the applications making use of spatial data, it is necessary to support various data representations. Hence, in this research both types of schemas are considered such that the data to be available for access can be represented by xml schema or database schema , as we aim to handle schematic heterogeneity of the data source. Schema difference is not only occur because of the existence of different types of schemas. The difference can also happen even using the same type of schema. For example if we consider xml schema a different data or even the same data can be represented using the same schema in various conceptual structure i.e. the way that similar datasets structured and encoded can vary significantly. This leads to schematic heterogeneity which is an aspect of structural heterogeneity.

Schematic heterogeneity is the result of disparities in the schema of different data sources used to model similar application concepts. It can be considered as a schematic inconsistency which is the main cause of data access limitation when the availability of the data source becomes huge. Thus, it is a crucial problem in data integration scenario [1]. Integrating data with disparate schema is a big problem which has to be solved so that it will be possible to get integrated information from different data sources. Thus, overcoming these heterogeneities is a major focus of this research through schema mapping and transformation technique. The technique is used to address schematic heterogeneity by translating data from schematically heterogeneous data sources to a unified target schema. It is one of the mechanisms to resolve the schematic inconsistency among schematically disparate source schemas.

The data to be made accessible is represented with different schemas. Different schemas are described in different modeling languages(metamodel)that fit certain requirement of the component such as representation power of tractability. For example a database may use an object oriented modeling language where as an XML document may be enriched with semantics by employing an ontology of the domain. If we consider the schema of a relational database, a schema S is a

collection of relations R(T1,T2,...,Tn)where R is the name of the relation and T1,T2,...,Tn are its attributes. Each attribute is associated with values. And, an XML schema used to define XML vocabularies and contains rich set of modeling elements for constraining both the structure and the data types available for use in a valid document. Thus, the schema mapping process will handle in connecting all the different types of schemas in a way that data represented in one schema related with data represented in the target schema.

As we are considering heterogeneous data source for the data integration system, some of the data sources may be represented with relational schemas and some with XML schema. We grouped these data sources as structure data, semi-structure data and unstructured data respectively. We will discuss how we can handle the three data categories with our schema mapping and data transformation process in Chapter 4.

3.2 SCHEMA MAPPING LANGUAGE

A language to specify a schema mapping rules between source and target schema based on some requirements is needed. The schema mapping language is used for conceptual schema mapping. The requirements for the language is explained below. The ability of the language to be declarative is an important character since any procedural language will make the specification too complicated. The language to be used for conceptual schema mapping would be preferred to have the following characteristics:

- Generic: It should not be bound to specific implementation as we consider schematically heterogeneous data sources.
- Declarative: a declarative style of mapping specification is a favour over a procedural style as a procedural style is less amenable to different implementations. The language do not specify a particular method for performing a certain function, but it suppose to specify more what is required.
- Expressive enough: The language should allow the mapping specification to contain enough information to facilitate instance transformations. It must support renaming of classes and attributes, restructuring, reclassification, and also a number of general (non-spatial), geometric and topological functions to transform geographic data.
- Complete and unambiguous: provide the necessary information required at run time to do the actual data transformation.
- Instantiable: The actual mapping code for different implementations can be derived from it.
- Standards based: Preferably it builds on existing standards or initiatives.

There are different languages for the description of a schema mapping rule between data models such as OML (Ontology Mapping Language), XSLT (Extensible Style sheet Language Transformations) , OWL (Web ontology Language). However the degree of expressiveness of the languages in constructing the mapping rule is different. Hence, there is a need to make an assessment of the schema mapping languages. XSLT is one of the schema mapping languages which is a high level language for performing XML-encoded data transformation. It is intended to translate data from one XML document into another XML document. It has functionalities such as filtering elements, selection to read values, condition for restriction, loop over elements, create new elements

and attributes. However,the language lacks some important operations such as Splitting and combining features/properties and geometric operations[16]. Furthermore, XSLT cannot be used in case of non-XML input. This implies that the expressiveness and declaratively of the language is poor to use it for our study.

The other language is OWL which is used to represent the meaning of terms in vocabularies and the relationships between those terms. That is,the language is intended to be used for describing ontologies and creating semantic annotations. Web Ontology Language(OWL) extends the expressiveness of RDFs(Resource Description Framework) **??**. It consists of three increasingly expressive sub-languages such as OWL Lite , OWL DL and OWL Full. Even though it has high expressiveness it is not extensible which is an important characteristics for the schema mapping language to be used for the purpose of this research. The third candidate is OML(Ontology Mapping Language) which is selected as most suitable candidate for it satisfy some of the requirements mentioned above such as expressiveness ,declarative and extensibility so that it can expressively define the relation between the source schemas and target schema without having the details of transformation execution in it. The language is good enough for instance transformation. Another advantage of using OML is that it can be used with different schemas. That is , the language has the ability to complex correspondences independently from the language in which the ontologies are modeled, and thus to represent any kind of schema mapping. In addition, OML is capable of being extended to cover schema transformation on spatial data. Thus, we have found that OML is most suitable language to be used to satisfy the purpose of this research.

As our approach is to separate the mapping specifications from the actual data transformation, the selected language should have the ability to specify the mapping rule that will further be used by the transformation functions for the actual data transformation during the transformation execution process and OML has the capability for deriving the actual data transformation process as it has high capability to be parse able. The derivation of the actual data transformation is done by two main processes. One is the interpretation of the specified mapping rule to get all the required information about the corresponding source and target items using a parser and the second one is applying those transformation functions discussed in chapter 4.

The other requirement to be satisfied by the schema mapping language is that it has to be built on existing standards or initiatives to facilitate spatial data transformation. Thus, OML needs to have an extensively characteristics so that it will have geographic characteristics which enable the language to support geometric or topological functions for spatial data transformation. And so, the Humboldt project extended this language to enable spatial data handling. We have adopt the geographic profile of OML from HUMBOLDT project with some modification to make it suitable so that it will be easily readable by our transformation functions.

3.2.1 OML(Ontology Mapping Language)

OML is one of the schema mapping languages for expressing conceptual schema mapping which is specified to provide a means to describe both declaration(specification) and transformations and designed by Ontology Management Working Group(OMWG) [28]. The basic functionality of the language is to give the user the possibility of expressing mappings between ontologies. It is being used to express conceptual-level transformations and is used for the exchange and storage of those. OML has the ability to provide the following capabilities as described in [25]:

- Expressiveness: The ability to express even complex conceptual mappings and attributive transformations so that some mismatches are thus avoided.
- Loose coupling: Mappings are not bound to a specific Schema Language (UML, OWL, GML Application SchemasĚ) and can thus be used with different schema formalisms.

- Usage in semantic web technologies :possible to validate semantic correctness of created mapping (and thus of translated data sets).
- Parse ability :Possibility to parse the OML documents and derive data transformation operations from the mappings.

We have defined an OML schema used to express the schema mapping rule as depicted in 3.1. This schema is defined to specify the mapping rule not only between one source and a target schema rather among a number of source schemas and target schema. The schema contains the main constructs of OML such as Alignment, location of the data source, cell and entity. The definition of these constructs are given below.

- Entity represents elements of a schema that can be mapped. The entities are identified by using the local name. Entities have a transformation function assigned which are defined on the source features.
- A Cell contains a mapping between two Entities where Entity represent Feature or Attribute objects from one or more sources schemas and one target schema. The basic unit of conceptual schema mapping belongs to a cell. It is possible to make a cell conditional based on a certain restriction on attribute or features by including a local name to express the restriction. An instance split and merge conditions can also be defined in a cell. An instance split is a case where from one entity represented in the source schema, multiple entities in the target schema are created where as in the case of instance merge multiple source features from one or more source schemas are used to create a single target feature [25].
- Alignment represents a complete set of mappings defined among one or more conceptual schemas.

The OML construct Entity represents different element types of source and target schema and they are defined below:

- Feature: It is a feature object of a schema. Spatial Object Type and Feature Type can be mentioned as equivalent terms as they are termed in INSPIRE and OGC respectively.
- FeatureCollection: used to express statement such that the union of two types in a schema or more than one schema is equal to one type in another schema.
- Attribute: is representative of a single property of a Feature; an attribute can be of primitive type (i.e. contain a literal value, such as a String or a number), or it can be a complex attribute.
- AttributeCollection: used to represent when a value in a target schema requires more than a single attribute value from one source or more than one source.

3.3 SCHEMA MAPPING SPECIFICATION

Schema mapping rule specification is a prerequisite for data transformation. That is the schema mapping rules specify how data transformation between source and target schemas should be performed. The schema mapping specification is done on conceptual schema level using Ontology mapping language to express a schema mapping rule. Since we deal with different source schemas and target schema, we have constructed the mapping rule specification from all the available source schemas to the target schema including one-to-one,one-to-many and many-to-one, in

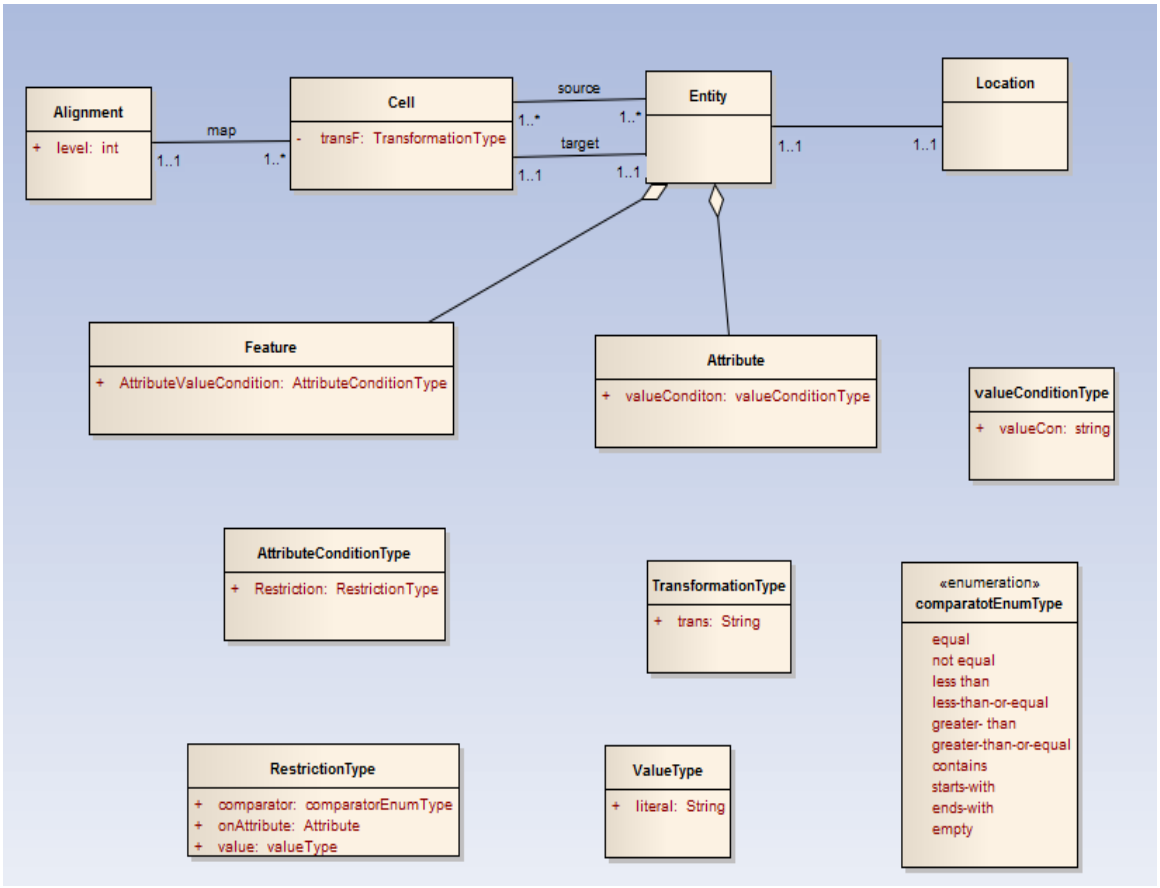

Figure 3.1: OML Schema used in this research

one Alignment which means in one complete set of mapping having a number of cell containing a mapping between two entities from one or more source schemas to a target schema.

Our schema mapping specification can be expressed mathematically as follows.

$Let\ SS = \{s1, s2, ...sn\},\ where\ s\ belongs\ to\ a\ source\ schema$
$and\ n\ is\ an\ element\ of\ a\ Natural\ number$
$TS = \{t\}, t\ is\ a\ target\ schema$
$M = f(S, T),\ M\ is\ a\ function\ of\ a\ mapping\ between\ S\ and\ T$
$where\ S\ is\ a\ subset\ of\ SS\ or\ an\ intersection\ set\ of\ si\ and\ sj\ where\ i\ and\ j\ are\ natural\ numbers$
$and\ T\ is\ an\ element\ of\ TS$

The schema mapping specification is used to express the relationship between two entities of different schemas. The schema mapping specification requires at least two inputs such as source schemas and a target schema to produce the correspondence between the elements of the schemas given as inputs. For instance, a schema mapping between two schemas can be formalized as a triple m=(S,T,£) consisting of a source schema S, a target schema T, and a set of mapping £,specifying the relationship between the source schema and the target schema. That is, for a source instance S and a schema mapping £, a target instance T is a solution for S if S and T together satisfy a certain conditions. As an example, consider there is a need to have an integrated data about utilities of a country where there are several different data providers who are responsible for delivering utilities data, each using a different data model or schema belonging S and utility application schema which is the target schema belonging T.

After the identification of the corresponding entities in the source and target schema based on the query expression posed on the target schema(as shown in figure 3.2)through the information that which source datasets are needed for the required data at the target schema where the target schema is an application schema, a relation needs to be specified between the entities of both schemas. The correspondence between the entities of source and target schemas is specified by using a language called OML as mentioned earlier. The query can be expressed in any language in accordance with the structure of the target schema as long as it is expressive enough to give the required information for the specification of the schema mapping rule. For example,if the target schema described in relational schema then the query can be expressed in SQL(sequential Query Language). The query expression can be used to get information about which data sources from the available ones required, which transformation function that is the relation between the source schema and target schema has to be used and the condition specified in the query used for the restriction in the filtering function which is one of the transformation function see 4.2. Here,ontology_based semantic description model for the retrieval of such information is needed.

In the OML schema defined for schema mapping specification a mapping among more than one schemas is called an Alignment where a number of cells can be found. A cell can contain more than one source entity in the case of entity merging from different sources, and a single target entity. Such entities can be features or attributes, as well as Featurecollections or Attributecollections. For each entity a transformation function can be defined and called to transfer source entities data to the target entity.

The profile for the OML construct defined in this study is given below:

```
<?xml version="1.0" encoding="UTF-8"?>
<!-- edited with XMLSpy v2006 rel. 3 sp2 (http://www.altova.com) by ?ITC (ITC) -->
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema" elementFormDefault="qualified"
attributeFormDefault="unqualified">
<xs:element name="Alignment">
<xs:annotation>
<xs:documentation>Comment describing your root element</xs:documentation>
```

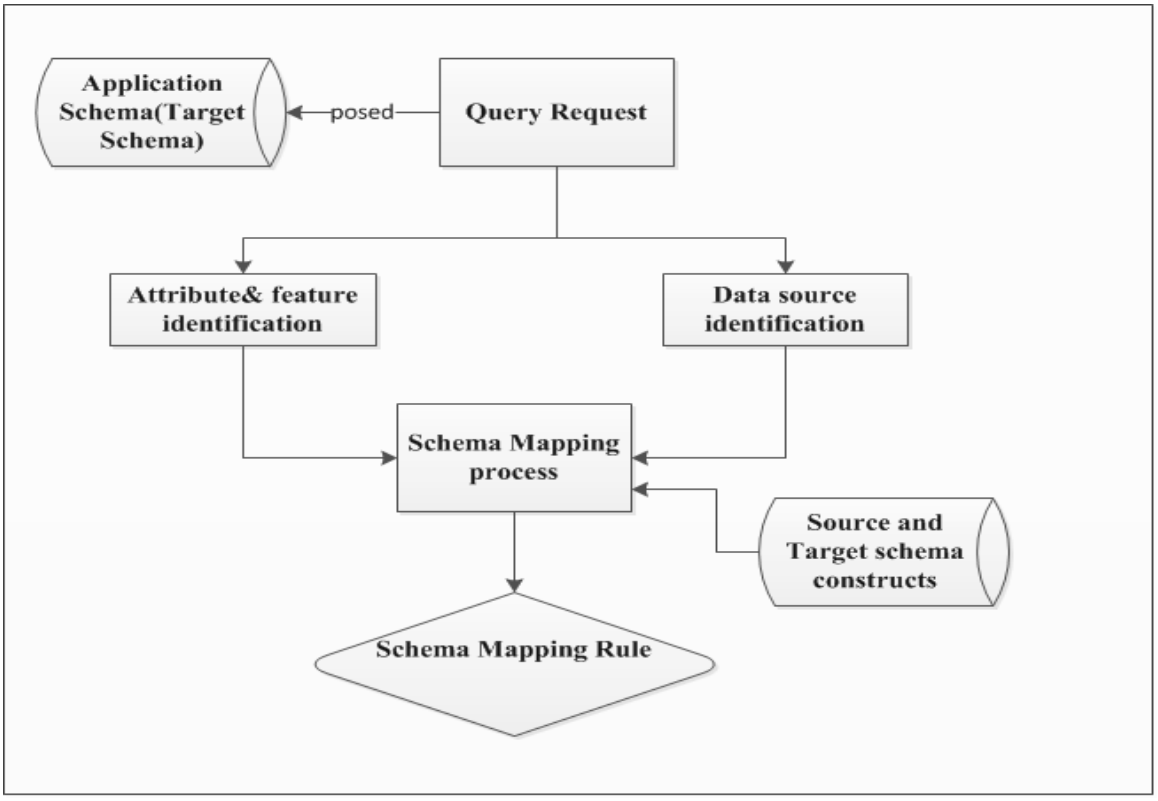

Figure 3.2: Schema mapping rule construction process

```
</xs:annotation>
<xs:complexType>
<xs:sequence>
<xs:element name="map" maxOccurs="unbounded">
<xs:complexType>
<xs:sequence>
<xs:element name="cell" maxOccurs="unbounded">
<xs:complexType>
<xs:sequence>
<xs:element name="source" maxOccurs="unbounded">
<xs:complexType>
<xs:sequence>
<xs:element name="feature_type" type="xs:string"/>
<xs:element name="feature_name"/>
<xs:element name="attribute" type="xs:string"/>
<xs:element name="location" type="xs:string"/>
<xs:element name="param">
<xs:complexType>
<xs:sequence>
<xs:element name="name" type="xs:string"/>
<xs:element name="value" type="xs:string"/>
</xs:sequence>
</xs:complexType>
</xs:element>
<xs:element name="transf" type="xs:string"/>
<xs:element name="restriction"/>
</xs:sequence>
</xs:complexType>
</xs:element>
<xs:element name="target">
<xs:complexType>
<xs:sequence>
<xs:element name="feature_type" type="xs:string"/>
<xs:element name="feature_name"/>
<xs:element name="attribute" type="xs:string"/>
<xs:element name="location"/>
</xs:sequence>
</xs:complexType>
</xs:element>
</xs:sequence>
</xs:complexType>
</xs:element>
</xs:sequence>
</xs:complexType>
</xs:element>
```

```
</xs:schema>
```

Chapter 4
Schema Transformation Process

This chapter discusses our approach to the interpretation of the schema mapping rule specified at the schema level to transform the data according to the rule. The chapter also discusses how the data transformation process handle the two categories of data sources(structured and semi-structured data) including the definition of each transformation function that perform the actual data transformation.

Our approach to schema transformation at the data level is preceded by schema mapping process which is used to specify the transformation rules for transforming data from source to target schema ,the schema mapping process is discussed in chapter 3. As mentioned earlier the transformation rule is constructed by using OML(Ontology Mapping Language) in the schema mapping process, which will be interpreted to transform the data from source to target schema. The overall data transformation process proposed in this study is described in figure 4.2.

4.1 SCHEMA MAPPING RULE INTERPRETATION

After the schema mapping rule specification, performing the the schema mapping rule interpretation will be the next to be done to retrieve information about the corresponding entities of the source and target schemas. We have defined a function that parse through the mapping rule file and get the necessary informations about the corresponding source and target schema to extract the required data. This function take the schema mapping rule in an XML format and use an XML parser to parse through the file and get the corresponding source and target elements such as feature type, feature name, attribute name, filtering condition and the location of source and target schema. These informations will be used to extract the required data from the one or more sources. As we have mentioned earlier, we consider heterogeneous data sources so that the data can be in relational database schema or in XML schema. If the source at hand is in relational schema, there could be two options to extract the required data. One is transforming the data in the database into GML format or specifying the necessary information in the mapping rule to connect to the database and get access. If the data sources are in GML or XML format, the function use the XML parser to parse through the dataset and return the selected data. Figure 4.1 shows how the schema mapping rule is interpreted.

4.2 TRANSFORMATION FUNCTIONS

The transformation functions used for data transformation process from the underlying data sources to the required application schema (target schema). We can see the transformation functions into two categories such as those of them that are applied on alphanumeric attributes and those that are applied on geometric attributes of source datasets. This would help to do some pre-processing data manipulation on geometry of source feature before the necessary transformation function applied on. This means that the geometrical representation of source feature geometry is different based on the format of the dataset for example the geometry of a feature in GML format

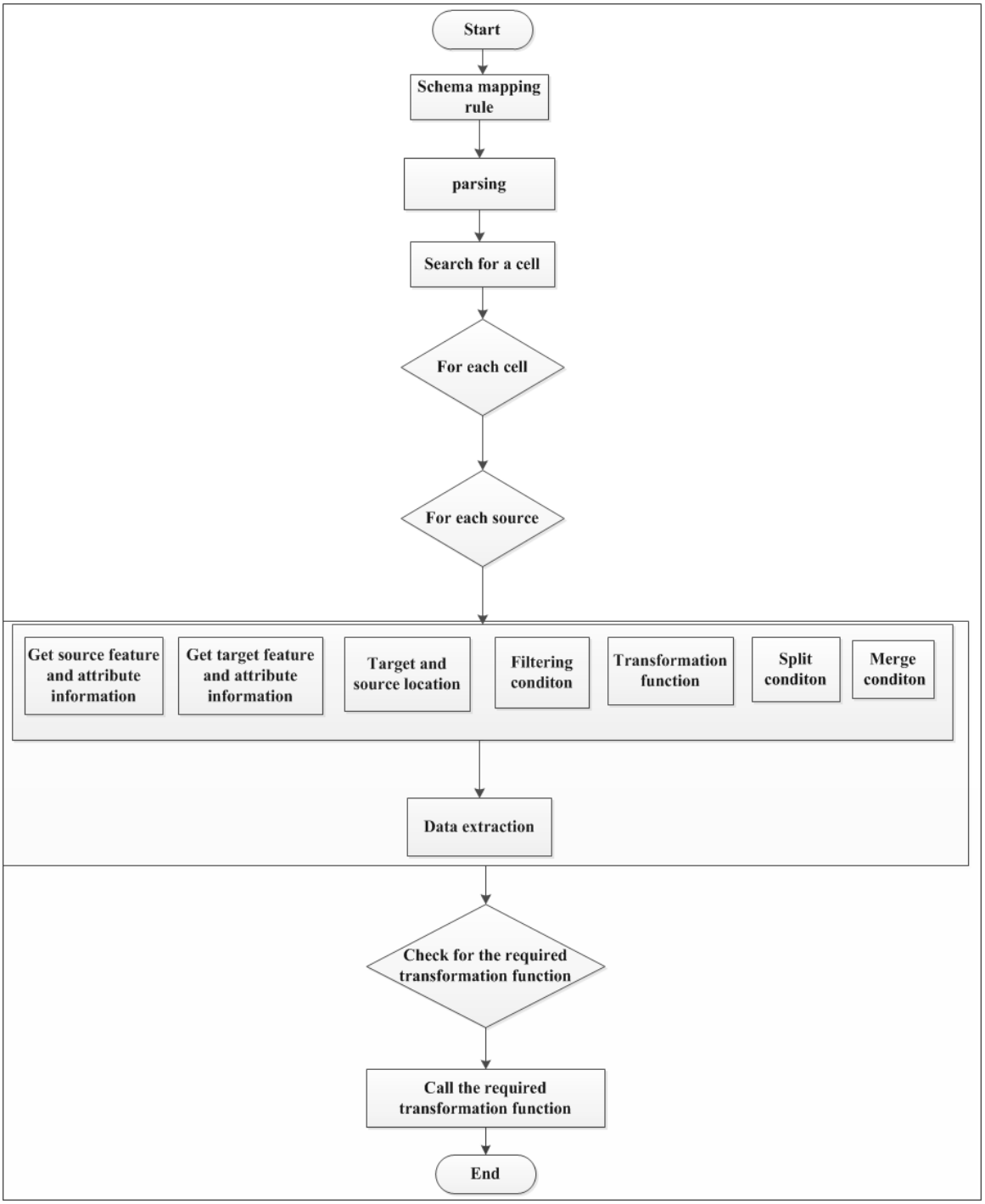

Figure 4.1: Schema mapping rule interpretation

extracted as list of points in textual format so that before transforming the data into the target schema, these list of points exposed to some manipulation. The functions are grouped as follows according to a classification given by [15].

- Filtering of Features and their Attributes: Based on values of Properties or Feature Properties that is based on conditional statements only selected features or attributes are mapped to the target schema.
- Renaming of Features and Attribute values: e.g. a translation between two natural languages.
- Reclassification of Features and Attribute values: Based on an attribute value of feature from one or more source schemas are translated to a single feature in the target schema. In the case of reclassification of attribute values a coarser classification system in the target attribute value domain can be performed.
- Merging or splitting Features and Attributes:
  - Merging Features: Features in two or more source datasets are transformed to a single target FeatureCollection.
  - Splitting Features Collections: FeatureCollection in a single source dataset are transformed to Feature in two or more target Feature.
  - Merging Attributes: Two or more Attribute values in the source Feature are transformed to a single Property value in the target Feature.
  - Splitting Composed Attributes: A single AttributeCollection value in the source Feature is transformed to two or more Property values in the target Feature.
- Reordering of attributes: Change of Properties order inside a Feature and change the order of component Properties inside a AttributeCollection.
- Value conversions:
  - Spatial conversion: Simplification(e.g. polygon to line)
  - Unit of measure conversions of Property values(e.g. converting miles to kilometers)
- Augmentation: Deriving values for target schema attributes missing on source schema based on the values of other attributes existing in source schema and filling in default attribute values in target schema.

4.3 ALGORITHMS FOR TRANSFORMATION FUNCTIONS

Description for the transformation functions are given above. In this section we will give an algorithm for each of them on how the functions perform the data transformation task.

4.3.1 Filtering Function

This function extract data from the source dataset based on a condition and map the selected attribute values to the target schema.

$ApplyFilter()$
$BEGIN$

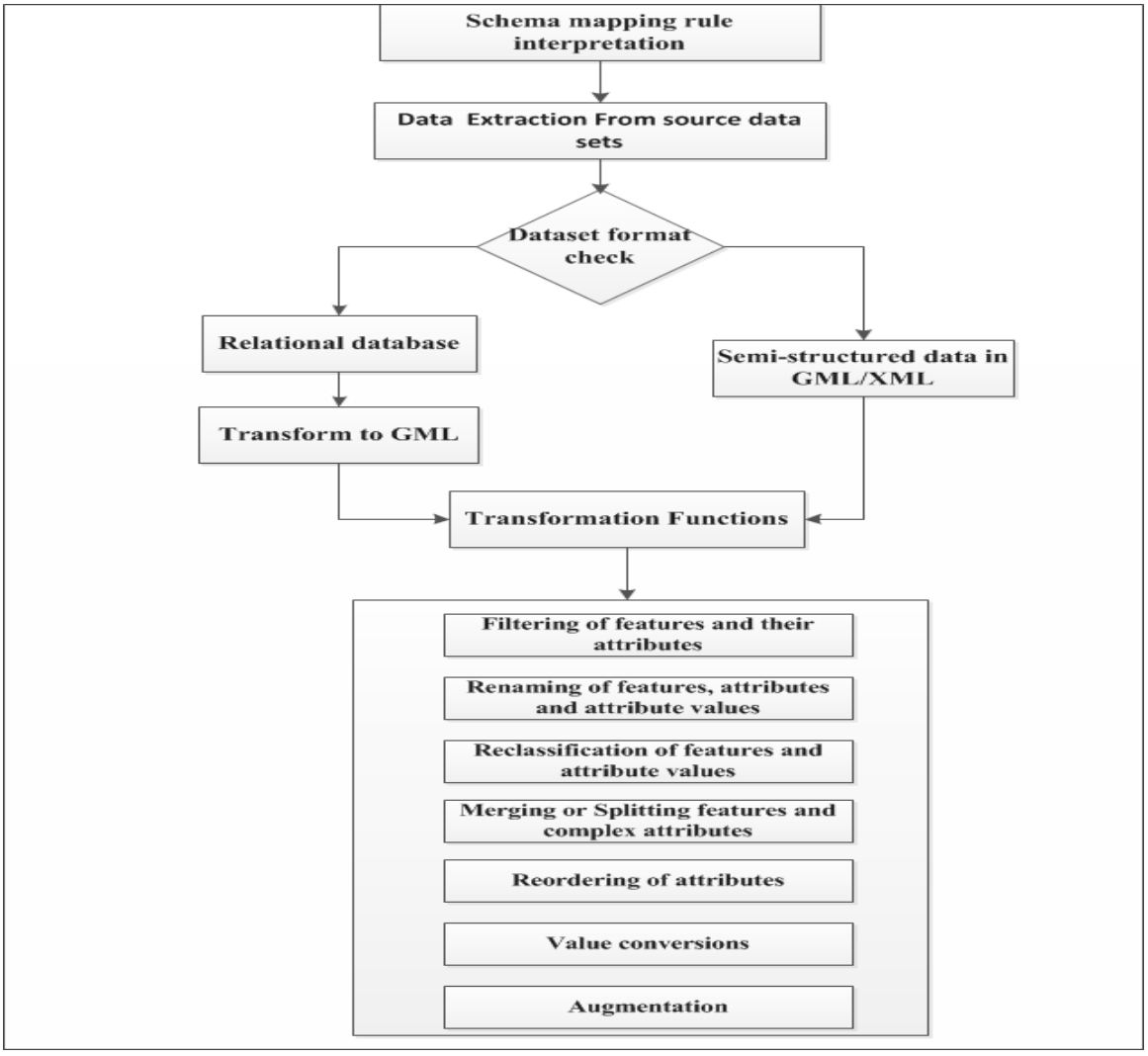

Figure 4.2: Data transformation process

Input : *Mapping rule in the XML format*
Getsourcefeaturename
Getsourcefeatureattributename
Get Filter condition
while *Filtering condition* **do**
Get the attribute value in the dataset
store selected attribute value in a storage
end while
Insert the filtered attribute values into target
END

4.3.2 Feature Rename Function

This function creates a new feature in the target schema by copying the source geometry into the target schema or by trivial conversions if the source geometry and target geometry are compatible.

FeatureRenameFunction()
*Input*1 : *Get the extracted feature geometry instance from the underlying source dataset*
*Input*2 : *Get target information*
Get the geometric type of source feature
Get the geometric type of target features
CASE 1 : *ONE* − *TO* − *ONE*
if *Source Feature GeometryType is equal* → *Target Feature Geometry Type* **then**
Copy source geometry instance into the target
else
CallSpatial Type Conversion
TargetFeatureGeometryType)
end if
CASE 2 : *ONE* − *TO* − *MANY*(*INSTANCE SPLIT*)
Call Split Feature Function
CASE 3 : *Many* − *TO* − *ONE*(*INSTANCE MERGE*)
Call Merge Feature Function

4.3.3 Feature Split

This function perform the data transformation by applying geometry split technique on the source feature geometry based on the specified split condition and make the sub geometry compatible with the target schema.

*Input*1 : *Get the extracted feature geometry instance from the underlying source dataset*
*Input*2 : *Get target information*
Get the geometric type of source feature
Get the geometric type of target features
Get split condition
Extract sub geometry based on split condition
Check for the geometric types ofthe extracted sub geometry
if *POINT* **then**
Make new point feature
if *target feature geometry is Point* **then**
Copy the new feature in the target schema

else
Call Spatial Type Conversion function
end if
else if *POLYGON* **then**
Make new polygon feature
if *target feature geometry is Polygon* **then**
Copy the new feature in the target schema
else
Call Spatial Type Conversion function
end if
else if *LINESTRING* **then**
Make new linestring feature
if *target feature geometry is LineString* **then**
Copy the new feature in the target schema
else
Call Spatial Type Conversion function
end if
else
Error message
end if

4.3.4 Feature Merge

The feature merging function perform the data transformation by merging the source feature geometry

*Input*1 : *Get the extracted feature geometry instance from the underlying source datasets*
*Input*2 : *Get target information*
Get the geometric type of feature of the underlying sources
Get the geometric type of target feature
Check for geometric type compatibility
if *Source feature Geometry types are compatible* **then**
Check for the geometric types
if *POINT* **then**
make new MultiPoint feature
if *target feature geometry is MultiPoint* **then**
Copy the new feature in the target schema
else
Call Spatial Type Conversion function
end if
else if *LINESTRING* **then**
make new MultiLineString feature
if *target feature geometry is MultiPoint* **then**
Copy the new feature in the target schema
else
Call Spatial Type Conversion function
end if
else if *POLYGON* **then**
make new MultiPolygon feature
if *target feature geometry is MultiPoint* **then**

Copy the new feature in the target schema
else
Call Spatial Type Conversion function
end if
else
Error message
end if
end if

4.3.5 Alphanumeric Attribute Rename Function from a single source

This function perform transformation of alphanumeric attributive values from a data source in which the mapping rules are from a source to target schema.

*Input*1 : *Get the extracted data*
*Input*2 : *Get target information*
if *The extracted source data conform to the target schema* **then**
Copy source instance into target
else
Convert source instance into string
if (*target attribute type* = *Integer*) **then**
do string to integer conversion
else if (*target attribute type* = *Long*) **then**
do string to Long conversion
else if (*target attribute type* = *Double*) **then**
do string to Double conversion
else if (*target attribute type* = *Float*) **then**
do string to Float conversion
else
Error message
end if
Copy source instance into target using target information
end if

4.3.6 Alphanumeric Attribute Rename Function from Multiple Source

This function perform transformation of alphanumeric attributive values in which the mapping rules are specified to perform the data transformation from one or more source datasets.

*Input*1 : *Get the extracted data from the underlying datasets*
*Input*2 : *Get target information*
doConcatenation
if *The concatenated source data conform to the target schema* **then**
Copy source instance into target
else
Convert concatenated source instance into string
if (*target attribute type* = *Integer*) **then**
do string to integer conversion
else if (*target attribute type* = *Long*) **then**
do string to Long conversion

else if $(target\ attribute\ type\ =\ Double)$ **then**
$do\ string\ to\ Double\ conversion$
else if $(target\ attribute\ type\ =\ Float)$ **then**
$do\ string\ to\ Float\ conversion$
else
$Error\ Message$
end if
$Copy\ source\ instance\ into\ target\ using\ target\ information$
end if

4.3.7 Spatial Type Conversion

Spatial type conversion function is one of the transformational functions that is used to convert the source feature type into target feature type to conform source datasets to target schema.

$Input1 : source\ feature\ type$
$Input2 : target\ feature\ type$
$Input3 : Get\ target\ information$
if $(Input1\ =\ Point)\ OR\ (Input2\ =\ MultiPoint)\ OR)$ **then**
$do\ Point\ to\ MultiPoint\ conversion$
if $((Input1\ =\ LineString)\ OR\ ((Input2\ =\ MultiPoint))$ **then**
$do\ LineString\ to\ Multipoint\ conversion$
else if $((Input1\ =\ Polygon)\ OR\ (Input2\ =\ MultiPoint))$ **then**
$do\ Polygon\ to\ MultiPoint\ conversion$
else if $((Input1\ =\ MultiPolygon)\ OR\ (Input2\ =\ MultiPoint))$ **then**
$do\ MultiPolygon\ to\ MultiPoint\ conversion$
else if $((Input1\ =\ MultiLinestring)\ OR\ (Input2\ =\ MultiPoint))$ **then**
$do\ MultiLinestring\ to\ MultiPoint\ conversion$
else if $(Input1\ =\ Polygon)\ OR\ (Input2\ =\ MultiLinestring))$ **then**
$do\ Polygon\ to\ MultiLinestring\ conversion$
else if $(Input1\ =\ Polgon)\ OR\ (Input2\ =\ LineString))$ **then**
$do\ Polgon\ to\ LineString\ conversion$
else if $(Input1\ =\ MultiPolygon)\ OR\ (Input2\ =\ MultiLineString))$ **then**
$do\ MultiPolygon\ to\ MultiLineString\ conversion$
else if $(Input1\ =\ Multipolygon)\ OR\ (Input2\ =\ LineString))$ **then**
$do\ Multipolygon\ to\ LineString\ conversion$
else
$nomorespatialtypeconversion$
end if
$Copy\ source\ instance\ into\ target\ using\ target\ information$
end if

4.4 CASE STUDY

we have made an experimental implementation on flood risk assessment scenario based on the information taken from SwissRe company. Swiss Re is a Swiss reinsurance company, works with corporations, governments, civil society organizations and academia to develop effective risk transfer solutions to make societies more resilient. It is operating in more than 20 countries. It complements proven reinsurance portfolio for Property, Casualty and Life, Health with

insurance-based corporate finance solutions and services for comprehensive risk management. Swiss Re works with public and private partners to design and deploy risk transfer solutions that are customized to the specific risk exposure in different parts of the world.

The risks covering large areas are mostly due to natural disasters (earthquake, flood, typhoon, hurricane etc.). These risks can be estimated by spatial modeling using GIS tools and remotely sensed data. Knowing the spatial location of any object and to estimate the risks is vital. GIS tools are necessary to identify the spatial location of objects and to estimate the risks associated with the insured objects. In order to evaluate the re-insurance pricing Swiss Re has developed a system that consists of different kind of tools and services. For modeling of risks and hazards and to evaluate objects re-insurance pricing, data from heterogeneous data sources need to be collected and combined. The collected data sources have different data models that make the unified accessibility of the data from the available data sources difficult. From this scenario the following use cases are identified:

Flood risk assessment is carried out to determine the areas at risk of flooding and possible economic losses that can be caused by flood in case, it occurs. This task requires different datasets such as land use data,Metrological data,Socio-economic data and DEM.

Based on the above information we have defined the application schema which is a common target schema for flood risk assessment. The figure depicted in 4.3 shows the application schema. The target schema reflects what information the user requires to assess flood risk for a given area(site). And, the required data come from heterogeneous data sources. We have used two schematically heterogeneous land use data sources of Enschede to be transformed into this target schema using our schema transformation method. The experimental implementation is done in two steps.

First, a schema mapping rule from the data sources to the target schema is constructed as shown below. This is done by using our OML schema shown in figure 3.1. Secondly, the interpretation of the mapping rule is performed using some of the transformation functions.

An example of a schema mapping rule specified for our use case is shown below. As shown below a cell can contain a mapping from one or more source schemas to a target schema with the required function. In the rule the source feature type,feature attribute and location of instance data of source is specified. In the same way target feature type, its attribute and target schema location. If a cell contain a mapping from more than one source then it is a many-to-one mapping in which the specified feature objects or attributes from the specified sources merged and transformed to the target feature object in the target schema.

```
<?xml version="1.0" encoding="UTF-8"?>
<Alignment xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
           xsi:noNamespaceSchemaLocation="M:\res_ref\mymappingschema2.xsd">
<map>
<cell>
<source>
    <feature>e_mainroads</feature>
    <attribute>NAME</attribute>
  <location>M:/semistr2.xml</location>
  <param>
  <name>NAME</name>
  <value>Main Roads</value>
  </param>
  <transf>'RenameAttributeFunction'</transf>
  <restriction>no</restriction>
```

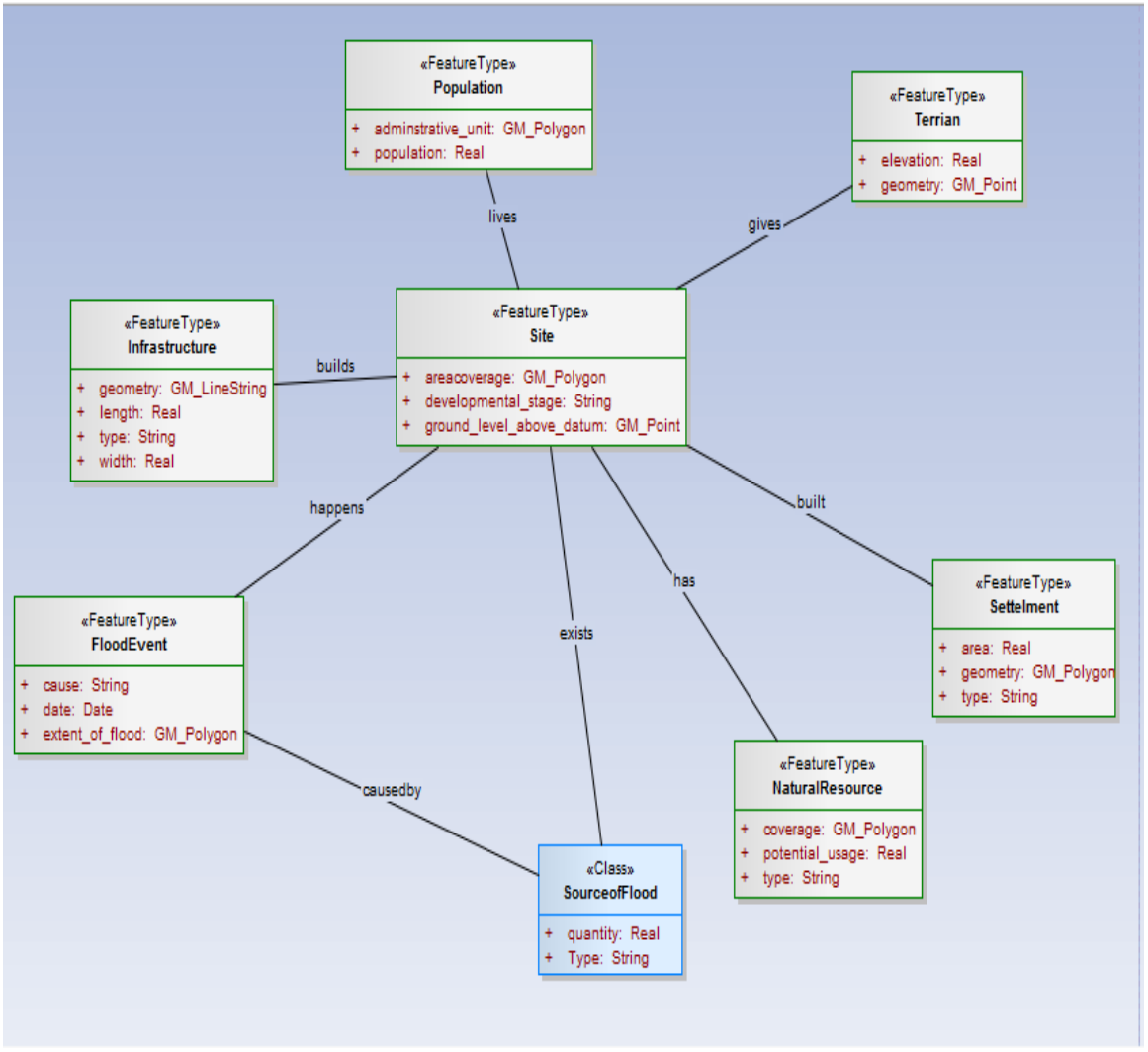

Figure 4.3: An example of target Schema for flood risk assessment

```
</source>
<source>
    <feature>e_nighbourhood</feature>
  <attribute>AREA_</attribute>
  <location>M:/testdata.xml</location>
  <param>
   <name>all</name>
   <value>all</value>
 </param>
<transf>RenameAttributeFunction</transf>
  <restriction>'no'</restriction>
</source>
<target>
  <feature>settelment</feature>
<attribute>type</attribute>
<location>table</location>
</target>
 </cell>
</map>
<map>
<cell>
<source>
    <feature>e_nighbourhood</feature>
  <attribute>AREA_</attribute>
  <location>M:/testdata.xml</location>
  <param>
  <name>all</name>
  <value>all</value>
  </param>
  <transf>RenameAttributeFunction</transf>
    <restriction>'no'</restriction>
</source>
<target>
   <feature>settelment</feature>
 <attribute>area</attribute>
 <location>table</location>
</target>
</cell>
</map>
<map>
<cell>
<source>
    <feature>e_mainroads</feature>
  <attribute>LAYER</attribute>
  <location>'M:/semistr.xml'</location>
  <param>
   <name>all</name>
   <value>all</value>
```

```
  </param>
  <transf>RenameAttributeFunction</transf>
    <restriction>'no'</restriction>
</source>
<target>
    <feature>Infrastructure</feature>
  <attribute>type</attribute>
  <location>table</location>
</target>
</cell>
</map>
<map>
<cell>
<source>
  <feature>e_mainroads</feature>
<attribute>Shape_Leng</attribute>
<location>'M:/semistr.xml'</location>
<param>
<name>all</name>
<value>all</value>
</param>
<transf>RenameAttributeFunction</transf>
  <restriction>'no'</restriction>
 </source>
 <target>
    <feature>Infrastructure</feature>
  <attribute>length</attribute>
  <location>table</location>
</target>
</cell>
</map>
<map>
<cell>
<source>
    <feature>e_mainroads</feature>
  <attribute>THICKNESS</attribute>
  <location>M:/semistr.xml</location>
  <param>
   <name>all</name>
   <value>all</value>
  </param>
  <transf>RenameAttributeFunction</transf>
    <restriction>'no'</restriction>
</source>
<target>
  <feature>Infrastructure</feature>
<attribute>width</attribute>
<location>'table'</location>
```

```
</target>
</cell>
</map>
</Alignment>
```

c

Chapter 5
Discussion and recommendations

5.1 CONCLUSION

5.1.1 Introduction

Nowadays different spatial applications require data from diverse data sources which are represented with different schemas. The diversity of data source schemas can be categorized into three such as structured, semi-structured and unstructured data based on the characteristics of their schemas. Structured data has a fixed and regular schema where as semi-structured data does not have fixed and regular schema. Unstructured data is a kind of data with no schema. This research discussed how to resolve the schematic heterogeneity of structured and semi-structured data sources so as to facilitate data integration.

The availability of schematically heterogeneous data sources pose a challenge in the data integration process so that it is difficult to get an integrated information from different data sources. Hence, to resolve the problem of integrating schematically heterogeneous data sources, schema transformations is have to be carried out before datasets can be integrated. In the previous work schema transformation process performed only between one source and one target schema. In this study we proposed an approach towards schema transformation process in which the schema mapping process is performed from multiple source schemas to a target schema so that we will be able to integrate a bunch of source datasets in the required application schema. To this end, our schema transformation approach resolves some of the discrepancies with data harmonization process from the user's perspective and deliver the integrated data to the user following his/her requirements. That is the user does not need to have knowledge about how the source data is provided, how the schema transformation process taken place on source data sets etc.ẇhich is the important part of the proposed approach.

The conclusions of this study can be summarized based on the research questions as follows:

1. *What classification mechanism can be defined to classify the heterogeneous data source in terms of structured, semi-structured and unstructured data to facilitate formalization of schema mappings and transformations of the data sources?*

   The classification of heterogeneous data sources is done based on the characteristics of their schema. Different schemas have different structural characteristics to represent a real world object. Structured data are those ones which are organized in a structure so that it is identifiable. The characteristics of structured data is that data is organized in semantic entities,similar entities are grouped together, entities in the same group have the same descriptions and description for all entities in a group have the same defined format and a predefined length and all are present and follow the same order. Data organized in a relational database can be mentioned as a representative example of structured data. Semi-structured data is a kind of data that does not conform to a fixed or regular schema. In semi-structured data the information associated with a schema is contained within the data. Semi-structured data have the characteristics such as entities in the same group may not have the same at-

tributes, Order of attributes not necessarily important, not all attributes may be available, size of same attributes in a group may defer, type of same attributes in a group may defer, no fixed schema(schema less) and contains structure information in itself (self-describing) and structure is implicit and irregular. Data in XML schema can be mentioned as an example. Unstructured data those which do not have a schema at all.

2. *What is the suitable schema mapping and transformation approach to transform data from source to target schema?*

   Schema transformation process plays a great role to address the schematic heterogeneity of the data sources that hinder data integration. The suitable schema transformation approach is the one which can be done in two process steps such as schema mapping process at the schema level performed for the specification of schema mapping rules and the exclusion of transformation functions at the data level for the interpretation of specified schema mapping rules to transform data from multiple source schemas to a target schema.

3. *What is the best and suitable approach for schema merging process to derive integrated data model to facilitate data integration in a unified way?*

   The suitability of schema merging process is determined by the effectiveness of the specified schema mapping rules. And the specification of schema mapping rules depends on the definition of schema mapping. To this end, the definition of schema mapping depends on the schema mapping language used. The characteristics of a schema mapping language being declarative,expressive enough , not being limited with a specific schema language, and being extensible can make the schema merging process suitable for the purpose of this study. Thus, we have used Ontology Mapping Languages(OML) as it has the capability of being declarative , expressiveness and loose coupling.

4. *How to specify complete, correct and functional querying mechanism to retrieve information from the underlying data sources through the integrated model?*

   The querying mechanism is performed through the proposed schema transformation process in which the query posed on the target schema(user's application) is analyzed during the schema mapping process to find out which source datasets needed to be involved to answer the query in the target schema, that means retrieving the data from the underlying source datasets and exposed it to the transformation functions. Then, the user is provided with the integrated data for his/her application. The completeness correctness and functionality of the querying mechanism is related with the efficiency of the proposed schema transformation process and we have done a case study to see its functionality.

5. *How to evaluate and test the proposed schema transformation approach?*

   The proposed schema transformation process is tested by using a use case called Flood Risk Assessment. For that we have defined target schema. We use two datasets about Enschede landuse data one is structured data and the other one is a semi-structured data based on our data source schema characterization. We make use of our OML schema to specify the schema mapping rules from the two datasets to the target schema. Here the target schema is a relational database schema. Then, we defined some of the transformation functions to interpret the specified schema mapping rules.

5.2 RECOMMENDATION AND FURTHER WORK

We would like to suggest some works to be done in the future. They are explained as follows:

- The modification of schema mapping process to be able to specify the schema mapping rules used to transform data from unstructured data source to the target schema. The fact that unstructured data is a schema less and the suggested schema mapping process works with those data sources that structured and semi-structured data, there should be a mechanism that is used to be able to retrieve data from such type of data source. This can be done by retrieving some information from the Meta data and generate a schema for the retrieved information and expose it to involve in the schema transformation process.

- The quality assessment of the integrated data at the target schema is an important issue as it is important to make significant decisions for different spatial applications. Thus,we would like to suggest quality assessment of the integrated data to achieve accurate and efficient integration of spatial data. The importance of assessing the quality of integrated data gets increased with the increase number of source datasets as it gets more complex. The quality of the transformed data can be dealt by including some quality parameters in the schema mapping rules specification task.

- The other future work we would like to suggest is a schema mapping process which can generate schema mapping rules from multiple data sources to a target schema with generic and automatic characteristics. To design the schema mapping process in a way that will have the ability to generate the schema mapping rules from multiple and disparate source schemas to a target schema, an effective conceptual schema transformation including ontology-based semantic approach for schema matching purpose to handle the automation part, is required.

LIST OF REFERENCES

[1] A.R. Beck, A.G. Cohn, B. Bennett, G. Fu, S. Ramage, M. Sanderson, JG Stell, and C. Tagg. UK utility data integration: overcoming schematic heterogeneity. In *Proceedings of SPIE*, volume 7143, 2008.

[2] M. Boyd, S. Kittivoravitkul, C. Lazanitis, P. McBrien, and N. Rizopoulos. AutoMed: A BAV data integration system for heterogeneous data sources. In *Advanced Information Systems Engineering*, pages 511–524. Springer, 2004.

[3] M. Butenuth, G. G
"osseln, M. Tiedge, C. Heipke, U. Lipeck, and M. Sester. Integration of heterogeneous geospatial data in a federated database. *ISPRS Journal of photogrammetry and Remote sensing*, 62(5):328–346, 2007.

[4] J. Cardoso. Semantic Web: Technologies and Applications. *International Journal on Digital Libraries*, 3(3):237–248, 2007.

[5] S. Castano and V. De Antonellis. Global viewing of heterogeneous data sources. *Knowledge and Data Engineering, IEEE Transactions on*, 13(2):277–297, 2002.

[6] S. Castano, V. De Antonellis, MG Fugini, and B. Pernici. Conceptual schema analysis: Techniques and applications. *ACM Transactions on Database Systems (TODS)*, 23(3):286–333, 1998.

[7] Mehdi Essid, Omar Boucelma, François-Marie Colonna, and Yassine Lassoued. Query processing in a geographic mediation system. In *GIS '04: Proceedings of the 12th annual ACM international workshop on Geographic information systems*. ACM, 2004.

[8] R. Fagin, P.G. Kolaitis, A. Nash, and L. Popa. Towards a theory of schema-mapping optimization. In *Proceedings of the twenty-seventh ACM SIGMOD-SIGACT-SIGART symposium on Principles of database systems*. ACM, 2008.

[9] D. Ferrucci, R.L. Grossman, and A. Levas. PMML and UIMA based frameworks for deploying analytic applications and services. In *Proceedings of the 4th international workshop on Data mining standards, services and platforms*. ACM, 2006.

[10] L. Harrie. Generalisation and integration for location-based services. *Schloss Dagstuhl*, 29.

[11] S. Jacoby, J. Smith, L. Ting, and I. Williamson. Developing a common spatial data infrastructure between State and Local Government–an Australian case study. *International Journal of Geographical Information Science*, 16(4):305–322, 2002.

[12] S.G.L. Julkaisuja, V. Des, F.G. Institutes, and L. Lehto. *Real-Time Content Transformations in a Web Service-Based Delivery Architecture for Geographic Information*. 2007.

[13] A. Kap, B. van Loenen, and M. de Vries. Harmonized Access to Heterogeneous Content: Towards a European SDI. In *AGILE Conference*, 2004.

[14] D. Kensche, C. Quix, M. Chatti, and M. Jarke. GeRoMe: A generic role based metamodel for model management. *Journal on data semantics VIII*, pages 82–117, 2007.

[15] L. Lehto. Schema translations in a Web service based SDI. In *10th AGILE International Conference on Geographic Information Science, The European Information Society: Leading the way with geo-information, Aalborg, Denmark*, 2007.

[16] L. Lehto and T. Sarjakoski. Schema Translations by XSLT for GML-Encoded Geospatial Data in Heterogeneous Web-Service Environment. In *Proceedings of the XXth ISPRS Congress, July*, 2004.

[17] Y. Li, D. Liu, and W. Zhang. A Data Transformation Method Based On Schema Mapping. In *Proc. of the 3rd International Conference ISTAŠ2004: Information Systems Technology and its Applications, Salt Lake City, GI Lecture Notes in Informatics P-48, 2004, pp. 67-79*.

[18] M. Magnani and D. Montesi. A unified approach to structured, semistructured and unstructured data, 2004.

[19] I. Manolescu, D. Florescu, and D. Kossmann. Answering xml queries on heterogeneous data sources. In *Proc. of VLDB*, volume 2001, 2001.

[20] Astrid Fichtinger (TUM) Marian de Vries (TUD). Models and schemas in humboldt, 2001.

[21] P. McBrien and A. Poulovassilis. Data integration by bi-directional schema transformation rules. In *19th International Conference on Data Engineering, 2003. Proceedings*, 2003.

[22] H. Mohammadi. Spatial data integration: a necessity for spatially enabling government. *Towards a Spatially Enabled Society*, pages 333–341, 2007.

[23] H. Mohammadi and A. Rajabifard. Multi-source Spatial Data Integration within the context of SDI initiatives. *International Journal of Spatial Data Infrastructures Research. v4*, 18, 2008.

[24] C. Quix, D. Kensche, and X. Li. Generic schema merging. In *Advanced Information Systems Engineering*. Springer, 2007.

[25] T. Reitz, U. Sch
"affler, E. Klien, and D. Fitzner. Efficient Conceptual Schema Translation for Geographic Vector Data Sets. 2010.

[26] S. Schade, C. Keßler, and A. Nandipati. Studying Data Transformation in INSPIRE.

[27] F. Scharffe. Ontology Alignment Specification Language. In *Knowledge Web PhD Symposium 2007*, 2007.

[28] F. Scharffe, J. Euzenat, and D. Fensel. Towards design patterns for ontology alignment. In *Proceedings of the 2008 ACM symposium on Applied computing*. ACM, 2008.

[29] A. Tolk and S. Diallo. Model-based data engineering for web services. *Evolution of the Web in Artificial Intelligence Environments*, pages 137–161, 2008.

[30] H. Xiao. *Query processing for heterogeneous data integration using ontologies*. PhD thesis, University of Illinois, 2006.