Interoperable Geo-Spatial Data Model in the Context of the Indian NSDI

Sujata Rawat

November 2003

Interoperable Geo Spatial Data Model In the Context of the Indian NSDI.

By

Sujata Rawat

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

Degree Assessment Board

Thesis Supervisor:	Dr. Sameer Saran (IIRS)
Second Supervisor:	Ir. R.L.G. Lemmens (ITC)
Thesis examiners	Chairman: Dr. Ir. R.A. de By
	External Examiner: Drs. W. Quak
	IIRS Member: Dr. P.S. Roy
	ITC Supervisor: Ir. R.L.G. Lemmens

INTERNATIONAL INSTITUTE FOR GEO-INFORMATION SCIENCE AND EARTH OBSERVATION
ENSCHEDE, THE NETHERLANDS

I certify that although I may have conferred with others in preparing for this assignment, and drawn upon a range of sources cited in this work, the content of this thesis report is my original work.

Signed …………………….

Abstract

National Spatial Data Exchange Format (NSDE) is a data exchange standard for geospatial maps in India. In the development of Indian National Spatial Data Infrastructure, NSDE has been specified as the format for data exchange in governmental GIS procurements. This is with the aim of keeping all the GIS data of India in one format to fight the problem of heterogeneity atleast at national level. But with the present scenario where interoperability is the prime issue of consideration, abiding with the NSDE format will be contradictory to the concept of openness. However, it is quite possible that the NSDE format can be mapped to a format, which is recommended as open and helps in interoperability. GML (Geographic mark-Up Language) is recommended by OpenGIS Consortium for interoperable data exchange, it is a dialect of World Wide Web recommended XML and thus inherits the qualities of XML. GML allows the domain specific user to design a flexible and extensible schema, which suits the NSDE (National Spatial Data Exchange) format requirements as well as the OGC recommendations for interoperable data exchange and transfer. This thesis covers the schematic mapping of NSDE to GML and hence moves the existing GIS bases into XML domains. It covers the experiences in converting NSDE file to GML documents, with emphasis on the requirements and design choices in mapping the NSDE data format to a GML application schema. For the demonstration purpose various visualization techniques for GML to some graphical form like SVG are also considered.

Keywords: Interoperability, Indian NSDI, NSDE, GML, SVG.

ACKNOWLEDGEMENTS

I am immensely grateful to the authorities of "GEONEDIS " Project-a collaborative program of the ITC, the Netherlands and the Indian Institute of Remote Sensing (National Remote Sensing Agency), Department of Space, Government of India, to provide me an opportunity to go under this highly specialized and unique program.

It is my privileges to express my deep and sincere gratitude to Dr. P.S. Roy, Dean, IIRS, DehraDun for imparting invaluable professional knowledge and elderly support and affection during my stay at IIRS.

My sincere thanks are reserved to all the faculty of IIRS and Geo-Informatics Division in particular; headed by Mr. P.L.N. Raju, for giving all round professional know how, co-operation and facilities during the tenure of this course.

I am really grateful to Dr. Sameer Saran, Geo-Informatics Division, supervisor at IIRS, and Mr. Rob Lemmens, supervisor at ITC, for their constant guidance, critical appraisals, valuable suggestions, and supervision, which have made the completion of this research possible well within stipulated timeframe.

I would like to express my gratitude to Dr. Marian de Vries, Delft University of Technology, Brig Girish Kumar, Mr Abhishek Sindal, Survey of India, Mr Rupesh Punjani, ESRI India, and technical people from Galdos Inc. and Safesoft FME for their technical help in my thesis.

My appreciations are due to my parents, family members, and close friends for their understanding and constant financial and moral support.

I am thankful to all those people, who have given me the moral or technical support during my course tenure.

27th Nov. 2003
Enschede, The Netherlands Sujata Rawat

TABLE OF CONTENTS

1. Introduction

1.1. Background

With the rapid development in GIS (Geographic Information System) and its applications, more and more geographical databases have been developed by different programs and applications, but data sharing and acquisition is still a big challenge for the development of GIS applications. It is not that data are not available, there is a huge amount of geographical data stored in different places and in different formats, but the aim of data reuse for new applications and data sharing gets limited with the very thought of dealing with heterogeneity among existing systems in terms of data modeling concepts, data encoding techniques and storage structures, etc. (Devogele et al, 1998).

The situation is even worse in a large and developing country like India. There is huge amount of spatial data but stored in various forms. This diversity in data storage can be seen in government departments also. There were no standards or specifications for storing or exchanging different data, but with the introduction of NSDI in India, The Department of Space and Technology, Govt. of India has recently taken initiative to resolve this issue. NSDE (National Spatial Data Exchange) is specified as a standard format for data exchange and sharing for governmental GIS procurements. However this does not fully solves the problem of Interoperability at the global level. So there should be some format above all the national level efforts for interoperability.

There are many aspects of NSDI like Institutional, political, economical, security, technical and many more to name. To overcome the problems raised due to heterogeneous data, Indian NSDI has recommended NSDE (National Spatial Data Exchange) to be used by all the GIS data providers. This research is mainly concentrated on the design issues of an Interoperable data exchange format, translation of the NSDE data format to NSDE-GML format, to make sharing through web possible.

1.2. Motivation:

The problem of heterogeneity cannot be ignored specially in developing countries like India. India is prone to many disasters, which has adverse impact on its socio-economic conditions. It could be improved with some pre disaster operations or post disaster mitigations. However for efficient planning different data from several departments are required. But the spontaneous rescue operation cannot be performed in such a heterogeneous spatial data environment. This situation is even worse than it is represented.

However, the Indian govt. has taken initiatives in the direction of developing interoperable data sets with the introduction of Indian NSDI. It has specified a data exchange format (NSDE) so it can be expected that atleast the govt. department will have their data in this format. The situation can further be improved if some fast data-sharing medium like Internet is taken into consideration, so the data can be shared and viewed on web Browser. For this data should be in a language, which can be understood by browsers, OGC recommended GML is popularly known for spatial data exchange and sharing. GML is based on W3C recommended XML and thus inherits all the specialties of XML. Online vector data visualization is another demand.

The Indian Govt. has taken initiatives to encourage the interoperability among Indian GIS data providers and users, but all this is still in discussion and has not yet implemented so it's a duty of every citizen to help the nation in his or her own way.

1.3 Literature Review:

There are several known commercial desktop GIS software systems such as ESRI ArcInfo and ArcView, Smallworld GIS, Intergraph GeoMedia, MapInfo professional; Clark Lab Idrisi, etc dominate the geographical Information (GI) industry. Different vendors have their own proprietary software designs, data models, and database storage structures. It is unlikely that all GIS applications will use the same software. Thus, geographical data designed by different softwares cannot communicate without data conversion. In order to exchange information and share computational geo-data resources among heterogeneous systems, conversion tools are usually developed to transfer data from one format to other. Furthermore, these diverse desktop GIS database structures make remote data exchange and sharing more difficult because of limited accessibility and required data conversion.

The development of the World Wide Web creates a unique environment for sharing geospatial Data. Users can use the World Wide Web to download data for viewing, analysis or manipulation. Many of commercial Internet GIS programs, such as ESRI's MapObject, and ArcIMS, AutoDesk's MapGuide, Intergraph's Geomedia WebMap, GE SmallWorld's Internet Application Server and ER Mapper's Image Web Server, are developed to offer better tools for data sharing over the Web. But like the desktop GIS software these Internet GIS programs also have problems of proprietary software designs, data models and database storage structures.

An approach to interoperability is through standardization. The definition of standard data modeling and manipulation features provide a reference point, which facilitates data exchange among heterogeneous systems (Devogele et al, 1998).

In the past, several useful standards have been developed to facilitate data exchange. Among them, Geographic Data File (GDF) and the Spatial Data Transfer Standard (SDTS) are used and accepted. SDTS was expected to become an important data format for ITS spatial data transfer or a neutral format for data archiving (Arctur et al, 1998). But several barriers block the popularity of SDTS. These barriers include the complexity of SDTS, slowness in the development of practical SDTS profiles, restriction of each SDTS dataset to a single profile, lack of a clear definition of geospatial features in SDTS, and ambiguity in the means of specifying cardinality of relationships in a data model (Arctur et al, 1998). Currently both GDF and SDTS are not so widely used as originally anticipated.

The creation of a standard data exchange format, Geography Markup Language (GML) is another important step taken by the geospatial community towards data interoperability. The GML schemas are written in XML grammar, and are used for modeling, transportation, and the storage of geographic information including both the spatial and non-spatial properties of geographic features

(OGC, 2003). It is developed as Data Exchange Standards Interface by Open GIS Consortium (OGC) to achieve data interoperability and reduce costly geographic data conversions between different systems. In the OGC spirit, interoperability is achieved by means of common specifications that programs and data must follow (Buehler and McKee, 1996).

The creation of Geography Markup Language Implementation Specification by OGC represents a significant step in the development of interoperable architectures for the use of spatial information between different applications. GML holds promise to support mapping from a wide variety of sources and enable sharing of geospatial data for on-line information exchanges.

Unlike current proprietary commercial Internet GIS programs, the OpenGIS GML specifications are the public open standard for coding and sharing spatial data. GML is a good alternative to expensive, proprietary web-based mapping solution:

- GML is an open source standard. Users can use it for free. But for other commercial Internet GIS programs, users have to buy them at high amount. For example ESRI ArcIMS Internet software is so expensive that many users cannot afford it, but if they want to provide online spatial data services GML is a good alternative for these users, with GML they can provide the online spatial data services without buying these proprietary software.
- GML data are stored in text format, which is a universal format. Thus it is easy to integrate GML data into other data across a variety of platforms and devices (Lake, 2001).
- As a standard data exchange format GML reduces the costly conversion processes among different format databases.
- GML is capable of facilitating real-time data sharing and exchange on the Web because it uses XML grammar, which is widely supported on the Web. GML can enable an accessible Geo Web (Lake, 2000, 2001; Aloisio, 1999; Kim, 2001).

In addition, GML can deliver vector data over the Internet by styling the data into Scalable Vector Graphics (SVG) format (Badard et al, 2001;Vies, M). Most current Internet GIS programs deliver spatial data through transmission of raster images such as GIF and JPEG formats over the World Wide Web. There are several advantages for delivering SVG vector GIS data over the Web compared with raster GIS data:

Compatibility: SVG also uses text-based XML format, which is compatible with other formats. It can be seamlessly integrated with current Web technologies, such as HTML, JavaScript, JSP, ASP, JPEG, GIF, etc.

Graphic quality: SVG format graphics are scalable and resolution-independent. This kind of data can be scaled without loss of quality across different platforms and devices. But coarse raster images are low quality because of a low resolution. Especially when users zoom in too many times, images become blurred and pixilated. However, a raster image with high resolution usually has a larger file size since it needs to store information as finer pixels. The speed of delivering such large files over the Web becomes slow, so it is not practical to use high-resolution images for Internet GIS. The need for delivering high quality vector graphic maps over the Internet is increasing as data availability and global sharing increases (Bertolotto et al, 2001).

Interactivity: SVG vector data can be accessed in a more interactive and dynamic way. Some dynamic functions can be integrated into SVG documents so that the SVG graphics are animated on the Web. For example, a SVG graphic can interact with users by mouse over if a 'mouserover ( )' function is added in the SVG document. By combining SVG with other web technologies like HTML, JavaScript, JSP or ASP, GML-based database can provide users an extremely rich interactive graphic interface (SVG 1.0 Spec; Tennakoon, 2002)

In general, GML-based data have many advantages compared with other alternatives.

- GML-based data can be easily shared and reused. They have no proprietary data models and data structures. Because of the proprietary software design, data created by current commercial GIS software are difficult to be shared. To share data among such data, many data conversion processes are necessary.
- GML-based data are text format; they can be easily integrated with other format data across variety of platforms. Secondly GML-based data can be shared and exchanged online in real time.
- GML-based data can let users exchange data at feature level, while current commercial Internet GIS programs cannot. For example, from a big GML-based database, users can just query and download one feature such as a specific road, while from other alternatives, users have to download the whole data set. Sharing and exchanging data at feature level in real time are especially important for emergency services; they can greatly reduce the time spend on data acquiring processes.
- Styling GML data into SVG, GML-based data can provide users a more sophisticated interactive graphic interface and deliver higher quality graphic maps over the Web than most other online alternatives.
- GML is more flexible than other alternatives. It only defines a basic geographical feature schema and geometry schema, which are convenient for users to use. *(OGC, 2003; FME, Lake, 2001)*, Based on these schemas users can define their own specific schemas for their spatial data documents. It has been widely recognized that GML will play an important role as a future Web data exchange standard *(*Lake*,* 1999*;* Meneghello*,* 2001*).*

1.4. Research Objective

The main objective of this research is:

> ***To design a Geo Spatial Data Interoperability Model that complies with Indian NSDI standards and which can allow translation of the NSDE data file into NSDE (GML- version) format, for data exchange and value addition through Internet.***

In order to achieve the main objective, understanding of type of heterogeneity, problems in data sharing due to heterogeneity in data format, need of interoperability, understanding of interoperable data exchange format, mechanism of GML for interoperability are required.

1.5. Research Questions:

- *What heterogeneity elements are the major hurdles in data accessing and sharing among GIS organizations?* **(Basic Understanding)**

- *What could be the Model and its requirements that can help in maintaining interoperability?* **(Conceptualising the model/developing Algorithm)**
- *What kind of problems can arise while translating data from one format to another?* (Implementation)
- *Which programming languages/ software are required for different components of the model?* **(Implementation)**
- *How the NSDE (Indian NSDI specified data format) geo data can be mapped on web with the help of present web map servers or other possible languages?* **(Presentation followed by Analysis/ justification)**

1.6. Material and data used:

Data	As such the research is data and application independent still to demonstrate the model a sample NSDE data is generated from Survey of India's DVD.
Software	XMLSpy (XML editor), JDK, Safe soft-FME Universal Converter and Universal Viewer, TDN Dutch GML to SVG converter, Microsoft IE 6 Browser.
Programming Languages/ used	Java, VB, XML, GML, and SVG.

1.7. Methodology

1.7.1. Study of the Indian NSDI specification, specially the technical aspect: heterogeneity in data formats.

The aim of the thesis is to make data interoperable to it be accessed by range of users at the same time, without hurdles of conversions. In order to achieve this, the Indian NSDI has developed the framework, and has specified some standards for data format. There are many aspects of NSDI, but for this research, institutional, political and economic issues are considered beyond the scope of the discussion. Primarily the research is focused on technical issue and more specifically on data format interoperability; still the understanding of other issues is required. The work started with the understanding of the NSDE format, and to make it interoperable, the NSDE data is converted to XML domain using GML, so XML and GML specifications are studied thoroughly. See Chapters 2, 3, 4.

1.7.2. Understanding the requirements of interoperability model and Web enabling of data

***Mainly two things will be required*:**

- **NSDE data file to GML document** (Figure 1.1, Part 1):

For making a geo spatial data interoperable, it should be available in an open and interoperable format like GML. GML is OGC recommended interoperable data exchange format. After literature reviews like (Aloisio, *et. al,* 1999; Badard *et al*, 2001; Buelher *et al* 1996; Chang *et al*; Henning *et al, 2001;* Kim *et al;* Kraak et *al* (2002); May; OGC 2001, 2003, Safesoft Inc, FME; Tennakoon, 2002; Vries), it was decided to use GML for data exchange format. The GML schema is designed (using XMLSpy editor) as per the NSDE format requirements (see appendix C). As GML is an XML dialect and like XML it can also be written using simple notepad, or other simple editor. To convert present NSDE data to NSDE-GML document, Java program is developed, which reads NSDE files and writes GML file. The output GML document is validated using XMLSpy. The validation means to check the structure of the GML document as per the GML schema.

- **GML document to Map** (Figure 1.1, Part 2):

As GML is text based plain file, so for the presentation part GML elements are transferred into SVG so now the GML document can be visualized as a map on web browser and other SVG viewer/software. There are other graphic formats like gif, jpeg, VML, but SVG graphics have advantage over others *(Tennakoon, 2002, SVG 1.0 Spec; Vries).* The important merits of SVG are its scalability, Resolution independence, and Text searchability (discussed in details in chapter 5). For GML to SVG, mainly Safesoft FME Universal Translator is used, which has built in XSLT and default stylesheets (Safesoft Inc, *FME*). It can partially read and understand the generic GML application schema, for that also a mapping between user defined application schemas is required. With the help of XMLSpy and the reference of the example mapping files (*GML v2 Reader and Writer*), mapping files are generated. Chapter 5, 6, and 7 explains it in details.

1.7.3. Understanding the problems and limitations while translating the NSDE data file to GML domain:

On converting one format to another, it’s always possible that the output format (GML in the context) does not represent the input format (NSDE format) fully. As the NSDE file format and GML are in different domains, so during the NSDE to GML translation also there were some problems like GML 2 does not support topology information fully. These problems are discussed in chapters 5 and 7 in general.

1.7.4. GML schema verification (Figure 1.1, Part 3):

In the context of Interoperability, a data format can best be verified if it can work in as many environments as possible. Here the GML schema and the Document are verified first at FME universal translator by translating GML document to SVG and shape files, the conversion was successful. (Details in chapter 7), and on TDN Dutch GML to SVG converter (Kraak *et al*, 2002).

1.7.5. Web Page Design (Figure 1.1, Part 3):

To make this system available to the NSDE users, it is aimed to put this work online, but as the system was developed only for demonstration purpose so at present it is not free from all the possible bugs of data translation and conversion. This could be done in near future and then the work can be uploaded on web.

1.8. Target users:

The objective of the thesis is to design a system of an interoperable data exchange format and as well as its visualization on web, in the context of Indian NSDI. So the prime users for the proposed systems are NSDI nodes (Chapter 3 section 3.3). Although other private organization, those want to translate their NSDE data file to GML document and want to visualize it on web can also benefit from this system.

2.

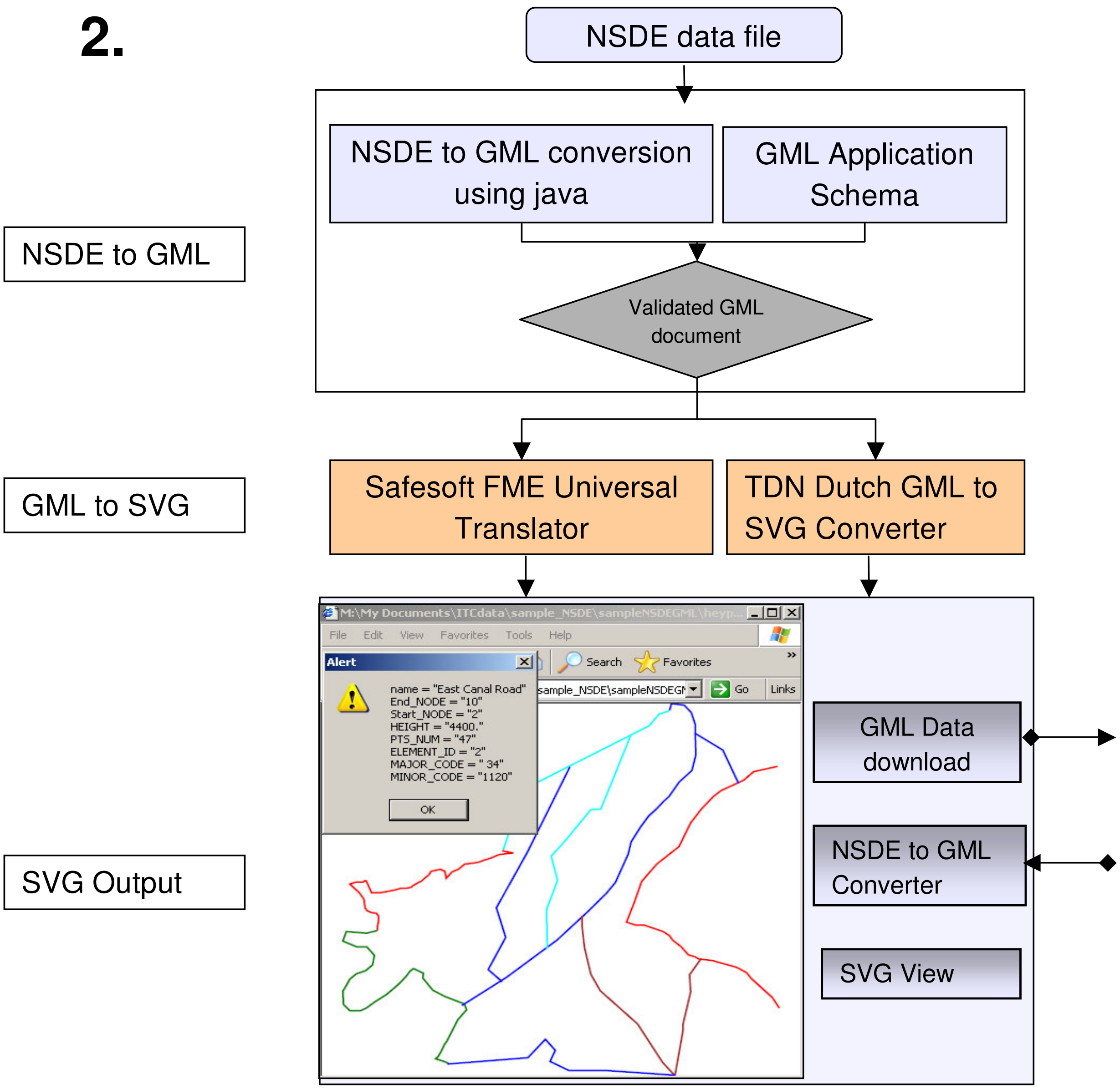


Figure 1.1: Methodology Flow

Heterogeneity vs. Interoperability in GIS domain

2.1. Heterogeniety in Geospatial Domain

There is a huge geographic data available with many organizations collecting geographic data for centuries, but some of that is still in the form of paper maps or in traditional files or databases, and with the emergence of latest technologies in the field of software and data storage some has been digitised and is stored in latest GIS systems. However, too often their reuse for new applications is a nightmare, due to poor documentation, obscure semantics of data, diversity of data sets (what information is stored, how it is represented and structured, what quality it has, which date it refers to, which scale is used…….), heterogeneity of existing systems in terms of data modelling concepts, data encoding techniques, storage structures, access functionalities, etc. [Devogele et al, 1998].

The other problem comes with the publicity of the data; it's an irony that the organization does not know about the data of other organizations. What all data does that organization carry, where to contact to get the data, how to get access to this data, what kind of data it is, which information is available, what is the quality how reliable the data is and so many such queries makes the reusability and the sharing of information from different organization, very limited. Instead of solving so many problems the other alternatives are either to go for acquiring the same data again or leave the application in the middle because of unavailability of data in spite of huge data ([Bishr et al., 2000)

The present scenario can best be represented by the below Figure 2.1. There are so many data provid-

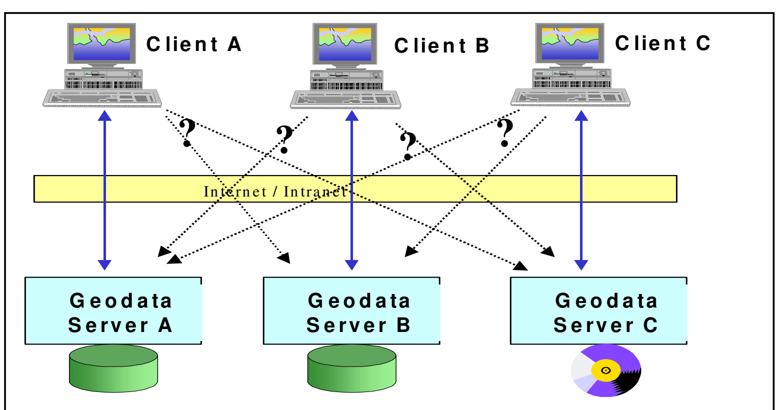


Figure-2.1 Source: OGC site

ers and so many data users also. But the data user is not aware of the data provider or the provided data is not compatible with the existing data of the user, this incompatibility can be of any type.

All this is true, but it cannot be just left like this, the reusability of the available Geospatial data is must, because lots of monetary, infrastructure and manpower is invested in the generation of data. It will be the loss of resources if the same investment is to be made again to acquire similar data. For applications like pre disaster management, mitigation & prevention strategy planning, rescue and relief operation, rehabilitation or reconstruction defence, urban planning, decision making, strategy making, business development, business promotion, market surveys or any other application, various data are required to analyse the area affected/effected, reason of affect/effect, to find optimal route to send the necessary help at the affected area or to reach the important clients in the shortest time, or to support the decision at various public and private departments. It is quite obvious that all the information can not be generated at one organization, one need to take data from various departments, like the topographic map from National Mapping Agency, housing plans from dept of urban and country planning, Road maps from road and urban planning department and many more organizations. When so many organizations are participating then the data compatibility issue is very important and should be taken seriously, otherwise the data will be available but two data cannot be used together and thus any strategy planning will be in trouble in the scarcity of proper data in spite of pool of data.

2.1.1. Aspects of information interoperability

The obvious answer to heterogeneity is interoperability. The Interoperability is the capability to communicate, execute programs, or transfer data among various functional units in a manner that requires the user to have little or no knowledge of the unique characteristics of those units.

The Interoperability in Geospatial domain is the ability, the cooperation, the compatibility of an Information system to run, manipulate, exchange and share the data of different organizations related to spatial information on, above, and below the Earth’s surface; for any kind of application to serve the society over networks.

The complexity of geographic information and processing raises the fundamental issues related to the incompatibility of representations, structures, and semantics that need to be addressed to achieve geographic information interoperation. There are three major aspects of information interoperation; each of which emphasizes resolving either *syntactic, semantic, or software incompatibility*. The syntactic approach enforces standards for encoding and interpreting geospatial information to allow one system capable of understanding the meaning of data from another system. Syntactic interoperability can be achieved by standardizing meta-data and meta-information regarding data formats and definitions to allow the data to be processed in different environments. A syntactic interoperability can make the transformation among different systems easier but it has limitations to overcome the barriers resulted from semantic gaps between communities of different cultures and histories to share geo-spatial information because of the distinct variations in conceptualisations and interpretations of geographic worlds [Bishr et al, 2000]. Diverse GIS data user group has various classification schemes, but an ideal database should allow data interoperable by different user groups, thus semantic interoperability is very important in the GIS domain. In addition to the syntactic and semantic propositions to resolve interoperation of geographic information, software interoperability is also important. For the development of Geographic Information the maintenance of all the three interoperability is important. But to start with, the present thesis aims to stress the importance of semantics to the enhancement of geographic information interoperability by developing a generic GIS data exchange format model. Semantic interoperability deals with the domain knowledge necessary for informatics services to “understand” each other’s intentions and capabilities. This is critical in order for disparate services to be able to collaborate, creating service chaining, across the network.

Geospatial data Interoperability is a very wide term and is a result of group efforts to be made by all the GIS data users. In this research thesis the stress is given on the development of interoperable data exchange format, the aim of the research is to design a data schema to fulfil the data requirement of Indian NSDI as per the recommendation of OGC for the exchange of data over internet, (chapters 3). In terms of Interoperability, the short-term expectations from the thesis are to make a data format schema so that the data can be transferred and can be visualized on the web browser before using or purchasing of the data. Eventually, in the long run, when all the GIS users will talk in terms of Interoperability then the newly acquired data and the many pieces of reused data will be integrated into a single, uniform, non-redundant data store, which will serve as the underlying database for the new applications.

2.2. Role of OGC in interoperability

Historically built as stand-alone applications, GIS services weren't made to easily communicate with other applications and systems. The standards developed by the Open GIS Consortium; called OpenGIS Specifications, support interoperability with open interfaces and protocols. The Open GIS Consortium, Inc. (OGC) is an international industry consortium of more than 260 companies, government agencies, and universities aimed at growing interoperability for technologies involving spatial information and location. The OGC mission is to deliver spatial interface specifications that are openly available for global use. Open interfaces and protocols defined by OpenGIS specifications support interoperable solutions that "geo-enable" the Web, wireless and location-based services (LBS), and mainstream information technology.

The Open GIS Consortium has six general guidelines for how geospatial information should be made available across any network, application or platform:

1. Geospatial information should be easy to find, without regard to its physical location.
2. Once found, geospatial information should be easy to access or acquire.
3. Geospatial information from different sources should be easy to integrate, combine, or use in spatial analyses, even when sources contain dissimilar types of data or data with disparate feature name schemas.
4. Geospatial information from different sources should be easy to register, superimpose, and render for display.
5. Special displays and visualizations, for specific audiences and purposes, should be easy to generate, even when many sources and types of data are involved.
6. It should be easy, without expensive integration efforts, to incorporate into enterprise information systems geo-processing resources from many software and content providers.

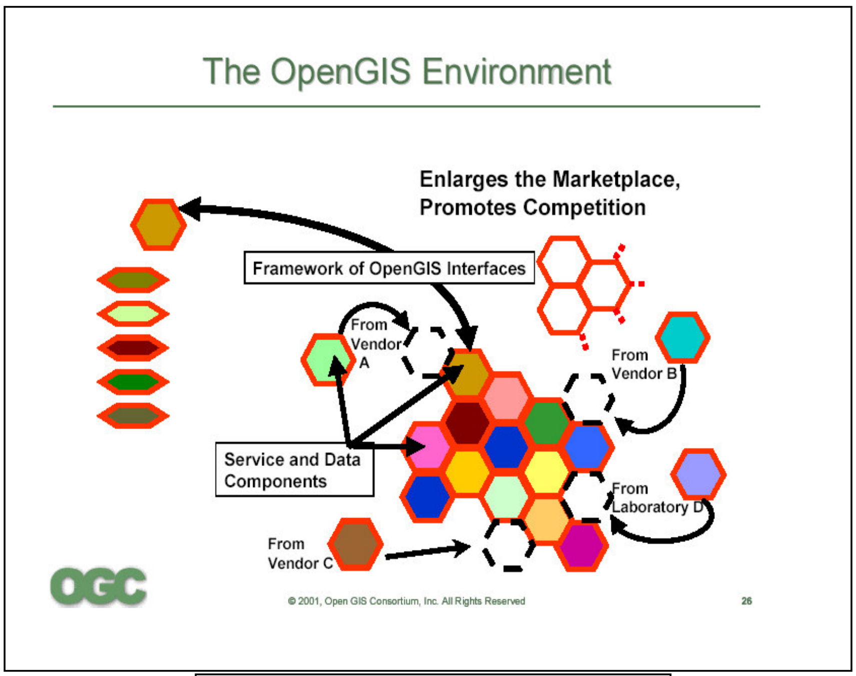


Figure- 2.2 Source: OGC

As with many standards bodies, the Open GIS Consortium has been working with Web services and XML. Recently in Feb 2003, the OGC released an approved GML (Geography Mark-up Language) Version 3.0 implementation specification. GML—an XML grammar written in XML Schema for the modelling, transport and storage of geographic information—provides a variety of object types for describing geography (detailed in 5th chapter).

With so much and such disparate geo-spatial and location information available, standards are key to the interoperability and wider use of geographic information systems technology. Standards and openness are the demands for GIS data interoperability. OGC stresses on "Openness".

What is an "*open standard*"? OGC defines an open standard as one that:

1. Is created in an open, international, participatory industry process, as described above. The standard is thus non-proprietary. It will continue to be revised in that open process in which any company, agency or organization can participate.
2. Has free rights of distribution: An "open" license shall not restrict any party from selling or giving away the specification as part of a software distribution. The "open" license shall not require a royalty or other fee.
3. Has open specification access: An "open" environment must include free, public, and open access to all interface specifications. Developers are allowed to distribute the specifications.
4. Does not discriminate against persons or groups: "Open" specification licenses must not discriminate against any person or group of persons.
5. Ensures that the specification and the license must be technology neutral: No provision of the license may be predicated on any individual technology or style of interface.

By this definition, a de facto standard established by one company or an exclusive group of companies or by a government is not an open standard, even if it is published and available for use by anyone at no charge. The Spatial Web, like the Web, needs open standards, as defined above.
So its quite clear that Openness is beneficial for one and all the GIS users and the society as a whole [OGC].

2.3. Summary

There are many kind of heterogeneity and problems in GIS domain but the solution cannot be created at the stretch with the efforts of one or two persons or organizations, there should be some standards above all the individual organizations or persons and it should be maintained by those who care about interoperability and want to take benefit out of it in many senses. It's a matter of praise that GIS data providers and users understand the importance of interoperability, development of OGC and the standards given by OGC is itself an example of the awareness among the GIS domain users. Other than this many private organization like Galdos Inc., ESRI, FME and more are coming forward to support conversion from or direct access to multiple geographic data sets in multiple formats, and are playing an active role in building interoperable technology into its products.

Still there is much to be done, data that can be shared should be kept in one common standard data exchange format, the software developers should also understand the demand of the new era and make it a point to follow the specifications given by an authority designed after the consensus. In short "*GIS interoperability a dream for users and a nightmare for system developers"* [Laurini, 1981].

3. Indian NSDI and NSDE: Format specified by Indian NSDI

3.1. Need of NSDI:

The necessity of geospatial data is understood to every one, it may be urban planning, disaster management, strategy planning, gas pipe line routing or for just any business promotion. This is also quite clear that the a data can be used for many purposes depending upon the application, for example the road information is required not only for tourism, but for urban planning, disaster management, site suitability and many more application with different interest. There are numerous organizations, which are generating geospatial data in one form or the other and with the emersion of so much scope in the geospatial field, many more companies are emerging; in such case there are quite fair chances that two organizations may not have similar data format and thus if the road data does not go with the land cover land use data then its of no use for urban planning, in the same way with other applications. In such case the other organization has to generate the same data again, which is a great loss of resources and time. In the case of temporal data it's not possible to generate such data again, and if the required data is available with some organization then it could result in good decision-making and strategy planning [Bishr et al., 2000].

If taken the case of India, it is a large country with the second highest population in the world, every year it faces huge socio-economic loss because of hazards. On the other hand there are so many organizations producing various data themes, even if consider the government departments only then also there are so many departments working on the generation of different themes for different applications. If these discrete data can be used together to fight with disasters by doing some pre disaster mitigation planning then the losses can be reduced largely. The nation today needs a NSDI much more than at any time. There are 2 major imperatives that drive the country towards establishing the NSDI (Indian NSDI strategy &Discussion, 2001):

- Enable the establishment of a national repository of a digital "warehouse" of the national map data holdings
- Facilitate Sharing and access to the digital spatial information.

3.2. Indian NSDI Components [Indian NSDI Strategy & Discussion, 2001]:

Indian NSDE components are illustrated below in figure 3.1. Starting from the foundation phase i.e.

3.2.1. NSDI Standard

It's related to defining and designing content and schemas, design and process, and defining network protocols, exchange formats standardization, and interoperable exchange and transfer, standards protocols, communication equipment, software standards and everything related to standards in consensus with the participating national agencies.

3.2.2. NSDI Metadata:

The first step in the implementation of NSDI is to develop and implement the Metadata standards and metadata files

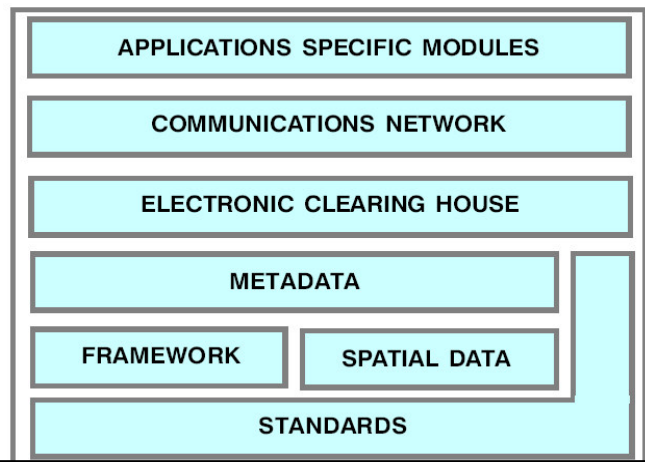


Figure 3.1: NSDI Components Source: Indian NSDI(2001)

3.2.3. NSDI Nodes and GIS servers of the actual spatial information

In conformity of the NSDI Standard, the NSDI Nodes will serve mainly as GIS based spatial databases and development oriented information systems servers - all integrated and linked to basic spatial/geographic units. The goal of NSDI is to aid as a decision-making tool and more in the context of assisting planning for developmental activities.

3.2.4. NSDI Search and Access Protocols

To enable search and location of spatial information, the protocols would provide the gateway for users to access NSDI. The basic issue in the operation of the NSDI is the backbone on which the information travels from one point to another.

3.2.5. NSDI Electronic Clearinghouse

The NSDI Clearinghouse would be the mechanism to provide access to the metadata and finally to the actual data sets. The clearinghouse is supposed to have systems to authenticate data requests and as spatial data volumes are usually large so download through networks may not be feasible. In such cases, the system should be able to generate media bearing the requested data for transmission by mail. The clearinghouse should also store information about the applications and availability of application specific modules that could be reused by other users. The clearinghouse would use the NSDI Search and access Protocols engines to look for and discover data and information.

3.2.6. NSDI User Interface

It would be the front-end interface for user queries and access of spatial information. For a completely ubiquitous NSDI, the penetration will have to reach public domain and the capabilities will include online access of information applications.

3.2.7. NSDI Outreach and Awareness programme

In order to increase the public and professional use of NSDI services and encourage e-governance concepts, The NSDI has plans to deliver awareness among data providers and data users for different applications.

3.3. NSDI stakeholders

In view of the above scenario an initiative is being taken to develop the National Spatial Database Infrastructure (NSDI) for India by the Department of Science & Technology and Department of Space in collaboration with several stakeholders of spatial community like:

- Ministry of Rural Development
- Ministry of Urban Development
- Ministry of Environment and Forests
- Department of Information Technology
- Survey of India [SOI]
- Geological Survey of India [GSI]
- Registrar General and Census Commissioner of India
- National Natural Resource Management System
- Natural Resource Database Management Systems
- Forestry Survey of India [FSI]
- National Atlas and Thematic Mapping Organisation [[NATMO]
- National Remote Sensing Agency [NRSA]
- Indian Metrology Department
- National Informatics Centre

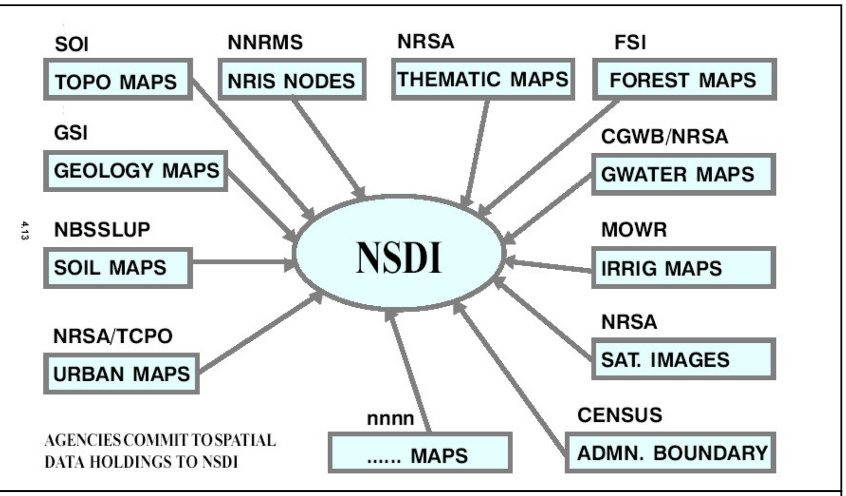


Figure: 3.2 Major Stakeholders

No doubt, India is fast moving into being information and knowledge society. The IT sector in India is developing in a drastic way. The Importance of IT is realized almost for every application, from online shopping to confidential data sharing, one can make life easy and comfortable with the involvement of online sharing and transfer or transaction of information. Its importance cannot be ignored in the geospatial domain also. India has got a huge amount of geospatial data in several themes and layers but major part of it is in the form of papers, so if one want to use some data along with its own data then he has to perform same exercise again as the pervious one did. It would have been easy if the data in digital form. As the sharing has came into discussion that means the data must be the format which can be understood by one system and there should not be so many translations from one form to another, because during translations some of the information get lost. It's very fortunate that the organizations have understood the importance of compatibility and have got the sense of sharing for the betterment of the society as well for the development of their business. But as there are so many organization and everybody has got their proprietary formats so one top level authority is required to regulate such discrepancy. With this motto India also has come up with some real initiatives and Department of Science and Technology, Government of India constituted a Task Force on National Spatial Data Infrastructure for India under the chairmanship of Surveyor General of India and members comprising from various stakeholders. Presently NSDI Task Force is the prime body responsible for developing mandate, vision, role and strategies, organization and legal framework.

Amongst the variety of datasets that are taken into prime consideration, geospatial (or map) information is a major "content". These Spatial information sets are vital to make sound decisions at the local, regional, state and central level planning, implementation of action plans, infrastructure development, disaster management support, and business development. Natural Resources management, flood mitigation, environmental restoration, land use assessments and disaster recovery are just a few examples of areas in which decision-makers are benefiting from spatial information.

As there is a rich collection of spatial data available in the country so to start with, the foundation data for NSDI could be the data available at government departments, like:

- National coverage of topographical maps on all scales of the Survey of India (SOI) toposheets
- National coverage of geological maps on all scales of the Geological Survey of India (GSI)
- National coverage of soil maps on all scales of the National Bureau of Soil Survey and Landuse Planning (NBSSLUP)
- National coverage of forest maps on all scales of the Forest Survey of India (FSI)
- National coverage of the hydrology maps on all scales of the Central Ground Water Board (CGWB)
- National coverage of landuse maps; wasteland maps; urban maps; groundwater potential maps and other thematic maps of National Remote Sensing Agency (NRSA) on all scales;
- Command area maps of Central Water Commission (CWC)
- National coverage of coastal Landuse maps of Ministry of Environment and Forests (MoEnF)
- Census maps and census data of the Census Department
- NATMOs maps

- National coverage of Satellite images all resolutions
- Indian Meteorological Department's data related to NSDI
- Naval Hydrography Charts of NHD
- Data from State Government Departments regarding, irrigation, agriculture, mines, minerals, etc.
- Data from Ministry of Transport
- Data from Ministry of Ocean Development

This is not the strict list of the organizations for participating in the NSDI database. Data may be available in various forms in many other departments/ organisations/ industry/ NGOs. It is expected that the NSDE task force is going to take steps for private data providers, so that it can be ensured that the data resources of various agencies have to be studied in detail and mechanisms should be available for ensuring that this data, whenever relevant, becomes part of the NSDI.

3.4. National Spatial Data Exchange Format:

Realization of the requirement of the National Spatial Data Infrastructure (NSDI) is the current issue of consideration. To use the data generated by various organizations, a national task force has been set up by Department of Science and Technology, which takes care of issues related to the evolution of spatial database standards at various domain servers (Data Providers) as well as the metadata standards for providing single window access to the domain databases.

The success of the NSDI concept would depend upon the ability of individual domain servers to serve the data to user community in a format, which is understood by one and all. While the individual domain servers could adopt the database design and implementation approaches based on their specific requirements of S/W and H/W tools, the exchange mechanism has to be open and neutral to specific platform. In this context it is very essential to evolve National Exchange Format. This format, once evolved has to be supplemented by individual domain servers by development of to-and-from data converters for supply of data to users and for accepting data from other domain servers.

Towards this, efforts have been made to evolve an exchange format by the sub-group on NSDI standards working, under the guidance of a NSDI task force. The format has undergone in-depth discussions and evaluation by the various participating agencies like Survey of India, Geological Survey of India, Department of Science and Technology and Department of Space. Furthermore, the veracity of the format has been tested by development of to-and-fro converters using the databases available as per NRIS data base design standards.

The format described here under, has evolved from the Digital Vector Data (DVD-3) format, which was earlier designed as the National Standard Exchange Format for Survey of India digital Cartographic vector data. This format catered for point, line and polygon topology describing relationships among spatial features. The currently proposed format has provision to include digital images acquired by satellites and Digital Elevation Model (DEM) and coded raster data. Furthermore the NSDE format also accommodates various types of thematic data sets along with the associated attribute data in tabular form.

Table 3.1: Data Structure in NSDE format		
File No.	**FILE NAME**	**DESCRIPTION**
1	VOLDIR	Volume Directory Information
2	GENINFO	General Information
3	QUALINFO	Data Quality Information
4	TOPOINFO	Toposheet specific general information
5	DATACAT	Data category records for base cate gory/layer/spectral band number 1
6	DATAFIL	Data records for base category/layer/spectral band number 1
7	DATACAT	Data category records for base cate-gory/layer/spectral band number 2
8	DATAFIL	Data records for base category/layer/spectral band number 2
---	--------	--
---	--------	--
4+2*p-1	DATACAT	Data category records for base cate-gory/layer/spectral band number p
4+2*p	DATAFIL	Data records for base category/layer/spectral band number p

The detailed NSDE format is given in Appendix – C.

3.5. Summary:

Realization of National Spatial Data Infrastructure (NSDI) is the current national focus of India. NSDI is proposed as a single window mechanism for providing access to the spatial data being generated and managed by various agencies in the country. To regulate the coordination among various stakeholders and to establish & maintain the standards for national data exchange, a national task force has been set up by Department of Science and Technology. It has given some standards, protocols, and data exchange formats (NSDE) to share national geospatial data. NSDI implementation in India has already been started with the metadata file designing. If is expected that the full fledge implementation will occur in recent future. Thus the national data can be reused to serve the nation, which will make a substantial help in the betterment of Indian economy and society as a whole.

4. XML:An answer for Internet-Based GIS

4.1. Origin

XML has got no explicitly defined data model. XML (Extensible Mark-up Language) is called a Mark-up language. In other words it is a meta-language for creating Mark-up languages. XML was itself created as a subset of SGML (Standard Generalized Mark-up Language), which, despite its name, is more oriented toward content structure rather than formatting, but it can be used for formatting. HTML (Hypertext Mark-up Language) is an application of SGML in which formatting is the major concern. XML is not a replacement for HTML, it is designed to describe data and to focus on what data is. On the other hand HTML focuses on how data looks. XML provides a means for specifying the structure of the content of documents as well as the formatting of their presentation on output devices such as computer monitor screens and printers.

The Figure 4.1 illustrates the relationship between SGML, the parent, and HTML and XML, the siblings. It also depicts the relationship of the special languages created with XML.

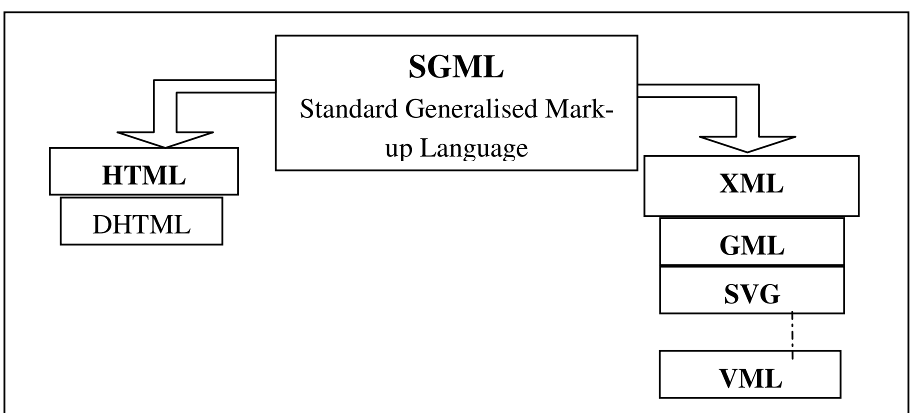


Figure- 4.1

After the design of XML several Mark-up languages came out for various applications like The XML has come up as the boon for Geospatial data transfer, “how” is explained in details latter in the same chapter.

Some Mark-up Languages Derived From XML		
Acronym	**Name**	**Function**
DrawML	Draw Mark-up Language	For creating vector graphics of simple diagrams involving rectangles and ellipses for the Web
GML	Geography Mark-up Language	For storing and sharing geographic documents such as maps
HGML	Hyper Graphics Mark-up Language	For creating vector graphics for devices such as cellular phones and handheld computers which have severely limited graphics capacities
MathML	Mathematics Mark-up Language	For storage, display and transfer of mathematical equations on the Web
PGML	Precision Graphics Mark-up Language	For creating vector graphics for the Web
OIL	Ontology Integration Language	For representation of formal semantics and logic on the Web
SVG	Scalable Vector Graphics	For creating vector graphics for the Web
VML	Vector Mark-up Language	For creating vector graphics for the Web
VocML	Vocabulary Mark-up Language	For representation of thesauri in knowledge organization systems
VoiceML	Voice Mark-up Language	For processing inputs and outputs by voice
WML	Weather Mark-up Language	For processing information in the weather-based financial derivatives industry
XGMML	Extensible Graph Mark-up and Modelling Language	For the interchange of graphs on the Web

Table 4.1

Even after so many definitions the questions is as it is, *after all what is XML and for what it is*?

So, It is a **mark-up language** much like HTML and is designed to **encrypt data** within the tags**.** Unlike HTML, tags are not predefined and any user can define them as per their requirement. XML uses a **Document Type Definition** (DTD) or an **XML Schema** to describe the data. With a DTD or XML Schema, it is designed to be **self-descriptive.** It is very strange to hear and hard to understand but its true that XML is not meant to do anything, rather it is not designed to do anything. XML is created to structure, store and to send information by encrypting it in the well defined and validated tags. The following listing is a piece of library information stored as XML:

```
<Book_List Category = "GML books">
  <Book B_ID= "GIS_IT_0101">
          <Title>GML for GIS</Title>
          <BookPrice>100 Euros</BookPrice>
          <AuthorName>Anonymous</AuthorName>
  </Book>
<Book B_ID= "GIS_IT_0102">
          <Title>XML boon for Geospatial data</Title>
          <BookPrice>100 Euros</BookPrice>
          <AuthorName>Anonymous</AuthorName>
 </Book>
</Book_List>
```

Example 4.1 Source: www.w3c.org

XML has nothing to do with this information. It is just pure information wrapped in XML tags. Someone has to write a piece of software to send, receive or display it, *as per the users requirement.* Before discussing about the rules for using XML tags, here is a list of few more merits of XML:

- XML, free and extensible

  As said above, XML tags are not predefined. One can "invent" his own tags and own document structure as per the requirement. In the above example also the tags (like <Book_List> or <BookName>) are not defined in any XML standard. These tags are "invented" by me to demonstrate the merits of XML.

- XML, a complement to HTML

  XML is not a replacement for HTML. Rather In the near future the combination of XML and HTML is going to be a miraculous combination, one for Data description and other for data presentation.

- XML, an open language

  XML is a cross-platform, software and hardware independent tool for transmitting information on the network. It is not a proprietary language, and thus the simple rules are common to one and all. With XML, data can be exchanged between incompatible systems. That's why it is expected that the XML is going to be everywhere.

- XML Separates Data independent of Visualization tools.

  With XML, the data is stored outside the visualization tool file and thus one can see the same data in numerous ways with different visualization technologies (Discussed in latter chapter), and on the fly changes in XML data will not make any affect the visualization part coding. So the data can be made available to more users and in various ways

- XML is meant to Share Data
- XML is meant to Store Data

4.2. XML Rules

The syntax rules of XML are very simple and very strict. The rules are very easy to learn, and very easy to use. Because of this, creating software that can read and manipulate XML is very easy to do. Its very true the XML rules are very strict, if the document is not valid, i.e. there is something missing or something extra then that document makes no sense and is as good as a garbage, but the interesting thing is that these rules are very simple as well and even a naïve person can easily get accustomed to these rules.[Goldfarb, Ed. ,1998]

As shown is example 4.1 Tags user defined so a user can write whatever he wishes, but as per the W3C specifications: which says:

4.2.1. All XML elements must have a starting tag [<…>] and a closing tag [/<…>]

<BookName>GIS technologies</BookName>

4.2.2. XML tags are case sensitive

Unlike HTML, XML tags are case sensitive.
<bookname> is different from <BookName>

4.2.3. All XML elements must be properly nested

Improper nesting of tags makes no sense to XML.
<Book_List>
<BookName>GIS technologies</BookName>
</Book_List>

4.2.4. All XML documents must have a root element

All XML documents must contain a single tag pair to define a root element.
Like in example [X] <Book_List> is the top class and then the 1st level child element <book> and then the second level child elements <BookName> and all. So there is only one super level parent.

4.2.5. Attribute values must always be quoted

In XML It is illegal to omit quotation marks around attribute values. Like:
<Book_List Category = "GML books">

4.2.6. With XML, white space is preserved

Unlike HTML, in XML, the white space in document is not truncated. Like:
<BookName>XML for   GIS </BookName> the space between the words will be as it is.

4.2.7. Element Naming

XML elements must follow these naming rules:

- Names can contain letters, numbers, and other characters
- Names must not start with a number or punctuation character
- Names must not start with the letters xml (or XML or Xml..)
- Names cannot contain spaces

4.2.8. XML Attributes

XML elements can have attributes, but the attribute must be enclosed inside the double quotes.

4.2.9. "Well Formed" XML documents and/or the “Valid" XML documents

A "Well Formed" document in one, which is well structured, all the tags have correct syntax properly nested and a “valid” document in one, which conforms to a DTD or its schema (defined in following section of this chapter.).

4.3. DTD

DTD stands for Document Type Definition and is meant to define the legal building blocks of an XML document. It defines the document structure with a list of legal elements of the document. A DTD can be declared inline in the XML document, or as an external reference. For internal and external DTD only the way of declaration is different. With DTD, the XML files can carry a description of its own format with itself. Independent groups of people can agree to use a common DTD for interchanging data. For the applications one can use a standard DTD to verify that the data received from the outside world is valid. DTD can be used to verify data structure.

The DTD is not covered in much depth because for this thesis Schema is used, which is a better alternative of DTD and GML is having schema not DTD. However to learn more about DTD, reference [XML handbook by Goldfarb, Ed., 1998] and [www.w3schools.com] can be explored.

4.4. Schema

XML Schema is a W3C Recommendation, originally it proposed by Microsoft, but became an official W3C recommendation in May 2001. Schema also describes the structure of an XML document. The XML Schema language is also referred to as XML Schema Definition (XSD). As said above XML Schema is an XML based alternative to DTD. Same as DTD, but schema are supposed to be the better successor of DTD. Although many XML users feel that XML Schemas are more complex then DTD, despite their complexity, W3C XML Schemas provide a lot of important capabilities and are worth using for many classes of validation rules. Web savvies think that very soon XML Schemas will be used in most Web applications as a replacement for DTDs. There are the genuine reasons behind this:

- XML Schemas are extensible to future additions
- XML Schemas support data types

  XML schemas allow you to derive new *simpleTypes* using a regular-expression-like syntax. The built-ins include string, int, float, unsignedLong, byte, and so on; but they also include some types that most programming languages lack natively: timeInstant (that is, date/time), recurringDate (day-of-year), uriReference, language, nonNegativeInteger and many more.
- XML Schemas are richer and more useful than DTDs

  The rich expressiveness of schemas makes strong validation of XML document. For example type constraints on Element contents, Attribute values and on Enumeration as well. DTDs are bit weak in this as DTD can at the most specify whether the element contains data or not, but it does not supports the type of data.
- XML Schemas support namespaces

  This is covered in more detail in subsequent section of same chapter.

  There is much more about XML schema but this is sufficient for the understanding the concept of Schema, for writing all together a new schema one is suggested to go through the references [Goldfarb, Ed. ,1998, www.w3schools.com, www.XML.com].

4.5. XML NameSpacing

Namespaces are a simple and straightforward way to distinguish names used in XML documents, no matter where they come from. However, the concepts are a bit abstract, and this specification has been causing some mental indigestion among those who read it for the first time. Basically the intension behind designing the concept of *namespacing* is to provide a method to avoid element name conflicts. The point of confusion in *namespacing* are" why its required?" and "what it points at?"

4.5.1. Why *namespacing* is required?

The actual use of the *namespacing* in this thesis will be clear in the subsequent chapter where GML schemas are discussed, there, the schemas of totally two different parties are taken together to serve one application. One is from OGC GML specification and the other in mine. But, to avoid any confusion by discussing about GML at this early stage, lets take another simple example 4.1, in this there is one Element <Title>, and as already know HTML also has <Title>, which means the page title, so when the example 4.1 application is running on web browser and the software is supposed to search for <Title> of the book then it will be difficult for it to distinguish between the both. So here does namespace comes into picture. It would have been better if something were written before the book <Title>, something like <gfm:Title> or <nsde: title>.

4.5.2. What it points at?

One of the confusing things about all this is that namespace names are URLs; it's quite possible to assume this because they're Web addresses, they must be the address of something. But, they are not; these are URLs, but the namespace draft doesn't care what (if anything) they point at. This is just the way of defining a namespace. The following example namespace is used for the schema design of this thesis.

The namespace is defined in the schema tag itself, gml and xlink is the namespace of OGC GML specifications and WWW xlink respectively. The nsde namespace is one related to my schema designed for NSDE data format of Indian NSDI (Details in coming chapter). The reason behind describing this namespace is that here in this thesis the schema OGC GML specifications are also imported (this is the another important advantage of XML) in the Schema designed in this thesis for nsde. So to distinguish any common Element of NSDE schema from that of OGC GML specified one, this namespace is going to play an important role. As the data is going to be available on web and probably anybody can access the sample data so there should not be any confusion. To avoid such confusion namespace is defined and used before every tag like *<nsde:featureCollection>, <nsde:Line_Info>* and similarly many more.

```
<schema targetNamespace="http://www.itc.nl/nsde"
xmlns:nsde=http://www.itc.nl/nsde
xmlns:xlink="http://www.w3.org/1999/xlink"
xmlns:gml=http://www.opengis.net/gml
xmlns="http://www.w3.org/2001/XMLSchema" elementFormDe
fault="qualified">
```

Example 4.2

4.6. Is XML the Answer for Internet-Based GIS?

If XML is adopted throughout an organization, or even the whole GIS industry, users will be able to search and manipulate on-line data, regardless of derived application. Retrieval of such data can allow for manipulation in spatially enabled web browsers or local XML-aware GIS applications. Just few years back when some organization had to receive a data from other organization then it was a quite a difficult task to understand the format and then to translate it back to native format, and this process is to be repeated every time a different format data is to be received. But as XML itself is a text-based format (as is HTML) that allows developers to describe, deliver and exchange data between ranges of applications. XML resembles HTML in format but unlike HTML, which has a limited set of tags to define formatting, XML allows authors to define an unlimited set of tags, such as:

```
<?xml version="1.0"?>
<feature>
<address>
<road-number>4</road-number>
<road-name> Haridwar Road </road-name>
<town>Dehradun</town>
<post-code>248001</post-code>
<country>India</country>
</address>
<spatial-attribute>
<type>point</type>
<projection>Polyconic</projection>
<node>
<X>511169</X>
<Y>197027</Y>
<Z>0</Z>
</node>
</spatial-attribute>
</feature>
```

Example 4.3

The above piece of codes is just to give an example how spatial data along with the attributes can be transferred across any platform on web. To avoid any confusion tags are also written is a human understandable language. Many works are going like the exchange of updating information between geographical information systems Using XML *(*Bedard *et al, 2001)* and XML engine querying, data extraction and evaluation over the web using XML G. Aloisio et al described about an XML based system that delivers customized remote-sensing data products to web-connected clients, and all that is required to support supervised, on-demand processing of the data. XML is also being used to set up <MEANINGFUL> Data Standards in the Geosciences (*Houlding., 2001).* In India XML is not very much used for geospatial data transfer but the metadata information is provided in XML.

Specifically for Geospatial domain, many XML dialects like GML and SVG are coming up as open specifications to share, exchange, or display GIS information on web.

4.7. Summary:

There is nothing special about XML. It is just plain text with the addition of some XML tags enclosed in angle brackets, but at the same time this is the beauty of XML, as XML is not made to do anything, and the application developer is going to decide how he can visualize or use the data. So with this and the discussion above, it is quite clear that the XML is boon for WWW, with the use of XML one can make the data useful to many users of different interest but with same data requirement, so If all the users make a consensus, then all future applications will exchange their data in XML and thus the sharable data will be available to more than one users at the same time through internet without falling in the conversions to and from to thousands of proprietary specific data formats. Just as HTML fuelled the first generation Web, XML is set to fuel the next generation.

5. GML: A language for new era of GIS

5.1. Introduction and Background

The development of widely accepted open standard language, XML has given a reliable solution for the data exchange and sharing across different applications, platforms, research groups and environment over Internet. It allows representation of data in a simple, flexible and human-readable form. As it allows application-specific customisations so many domain specific dialects of XML are currently being developed. There are a lot of useful extensions to XML and GML is one of those important extensions for Geographic data. GML is designed to support interoperability and does so through the provision of basic geometry tags (all systems that support GML use the same geometry tags), a common data model (features/properties), and a mechanism for creating and sharing application schemas.(lake. 2000,2001).

The Geography Mark-up Language (GML) is an XML based OGC specification for the transport and storage of geographic information, including the geometry as well the properties of geographic features. This specification defines the mechanisms and syntax that GML uses to encode geographic information in XML. With the popularity of GML, it is very much expected that the GML will make a significant impact on the ability of organizations to share geographic information with one another (OGC 2001). The GML came into existence with the publication of GML 1.0 in May 2000 by Open GIS.GML 1.0 makes use of DTD. Since then rapid developments has been carried out in the enhancement of GML specifications. Soon in Feb 2001 GML 2.0 was published with as a successor of GML 1.0 and this time it was GML schema (The benefits of using Schema over DTD are already discussed in chapter 4). GML 2 has 3 schemas Feature.xsd, Geometry.xsd, and Xlink.xsd (Discussed in section 5.4) The latest version of GML, GML3 published in June 2003 has specified more requirements Geospatial Data, like it has provision for describing 3d data, topology information among the feature, which was missing in GML2. Now GML3 has separate schema defined for coverage, datum, dynamic features, Data quality, direction, projection system and many more, which were missing in GML 2. GML 2 and now GML 3 can take advantage of the XLink and XPointer mechanisms enabling linking of features in one GML file to those in another file through use of hyperlinks. The enhancement of GML3 over GML2 is that in GML2 as such only real world objects that can be captured as a vector can be described using GML2 and that to only 0D, 1D, and 2D objects only. As of GML 3, it is possible to represent coverage type data such as remote sensing or distribution functions. Next to coverage data features with complex 2D and 3D geometries can now be captured in GML as well as temporal features, observation data and dynamic features.

At least two examples of data providers using GML (GML2) today are the British Ordnance Survey with their Digital National Framework dataset and the Dutch Topographic Service (TDN). For this thesis GML 2 is used and is described in little more details in subsequent sections.

5.2. Benefits of GML (Galdos Inc):

1. **Open Specification:** As GML is a dialect of XML, thus it can also provide both a vendor neutral as well as implementation neutral format that is optimally suitable for distribution over a network.
2. **Better quality maps (Galdos):** GML separates data information from data presentation and thus the geospatial maps out of the GML file can be displayed as fine as one wishes it to be.
3. **Good for Web:** As GML is based on XML; most of the web browsers that understand the XML will also understand GML. So it is not required to purchase any client specific software.
4. **Serves more users of different interest:** A GML file can be used by many users, and can be visualized in various ways using different types of stylesheets.
5. **More sophisticated linking capabilities (Galdos Inc):** Links can be associated with features in GML file. If some feature is having associated detailed information at some other location or in any other document then it is possible to link it with the feature.
6. **Better query capability:** GML enables the querying, so a GML document can be searched for a particular feature, so that only the relevant information can be chosen.
7. **Animation is possible.** Change presentation is sometimes a main theme of GIS application, such an operation can be accommodated in GML, and using SVG it can be rendered as animated graphics.
8. **Control over Features**: Information can be provided about the ability to share each feature and thus when an end user downloads the data only the sharable features from a file can be downloaded.
9. **Good for any target:** Not only to web browsers, GML file like XML can be sent on PDAs and cell phones.
10. **Easily editable:** GML is again a text based file format so it can easily be opened and edited in any simple editor like notepad. However many software companies like XMLSpy, XMLWriter, and many free software are also available.

5.3. Design Goals of GML(OGC 2001)

GML was developed with a number of explicit design goals, as discussed above that GML is a dialect of XML and that's why these goals overlap the objectives of XML itself:

Goals:

- Provide a means of encoding spatial information for both data transport and data storage, especially in a wide-area Internet context;
- Be sufficiently extensible to support a wide variety of spatial tasks, from portrayal to analysis;
- Establish the foundation for Internet GIS in an incremental and modular fashion;
- Allow for the efficient encoding of geo-spatial geometry (e.g. data compression);
- Provide easy-to-understand encoding of spatial information and spatial relationships, including those defined by the OGC Simple Features model;
- Be able to separate spatial and non-spatial content from data presentation (graphic or otherwise);
- Permit the easy integration of spatial and non-spatial data, especially for cases in which the non-spatial data is XML-encoded;
- Be able to readily link spatial (geometric) elements to other spatial or non-spatial elements.
- Provide a set common geographic modelling objects to enable interoperability of independently developed applications.

GML is designed to support interoperability and does so through the provision of basic geometry tags.

5.4. GML schemas

For GML2 OGC has provided three base XML Schema documents:

feature.xsd defines the general feature-property model, *geometry.xsd* includes the detailed ge-

o

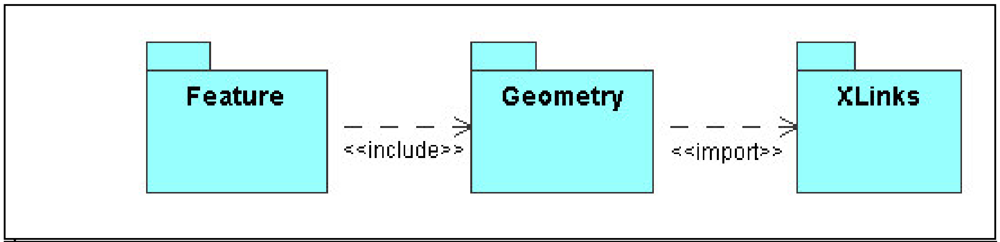


Figure 5.1: Three base Schemas of GML 2

components, and *xlinks.xsd* provides the Xlink attributes used to implement linking functionality. These schema documents provides base types and structures, which may be used by an application schema as per the requirement of the individual user. The feature schema imports the Geometry schema and geometry schema imports the Xlink schema.

5.4.1. The Feature.xsd:

A geographic feature is essentially a named list of properties. Some or all of these properties may be geospatial, describing the position and shape of the feature. Each feature has a type, which is equivalent to a class in object modelling terminology, such that the class-definition. A feature is encoded as an XML element whose name is the feature type according to some classification. The feature instance contains feature properties, each as an XML element whose name is the property name. Each of these contains another element whose name is the type of the property value or instance; this produces a "layered" syntax in which properties and instances are interleaved (OGC, 2001). The feature schema UML diagram (feature schema) shown in Appendix.

5.4.2. The Geometry schema

The actual definition of feature properties depends upon in the domain specific application schemas. Still as OGC abstract specification defines a set of basic geometries, so GML defines a set of geometric property elements to associate these geometries with features. GML Geometry property names are described in GML Feature schema. Overall, there are three levels of naming
Geometry properties in GML:

1. **Formal names:** that denote geometry properties in a manner based on the type of geometry allowed as a property value
2. **Descriptive names:** that provide a set of standardised synonyms or aliases for the formal names; these allow use of a more user-friendly set of terms.
3. **Application-specific names:** chosen by users and defined in application schemas based on GML

The formal and descriptive names for the basic geometric properties are listed in Table below:

Table 5.1 basic Geometry Properties:

Format Name	Descriptive Name	Geometry Type
boundedBy	-	Box
pointProperty	Location, position, centerOf	Point
lineStringProperty	centerLineOf, edgeOf	LineString
PolygonProperty	ExtentOf, Coverage	Polygon
geometryProperty	-	*any*
multiPointProperty	multiLocation, multiPosition, multiCenterOf	MultiPoint
multiLineStringProperty	multiCenterLineOf, multiEdgeOf	MultiLineString
multiPolygonProperty	multiExtentOf, multiCoverage	MultiPolygon
multiGeometryProperty	-	MultiGeometry

5.4.3. The Xlink Schema

The World Wide Web Consortium recommended XML Linking Language (Xlink) allows elements to be inserted into XML documents and thus create linking between resources; such links can be used to reference remote properties. Xlink uses XML syntax to create structures that can describe links similar to the simple unidirectional hyperlinks of HTML. The Xlink schema in GML provides the Xlink attributes to support linking functionality in GML documents.

In general:

- Features have an Id, name and feature type and comprise:
    - simple properties (booleans, integers, reals, strings)
    - 2D geometry with linear interpolation between coordinates
- Geometric constructs:
    - Coordinates (tuples of decimal numbers with decimal, coordinate, and tuple separators defined)
    - Box (defined by two coordinates) Point (coordinates with an Id)
    - LineString (at least two coordinates with an Id, the resulting lines can cross)
    - LinearRing (at least four coordinates, the first and last identical, with an Id)

- Polygon (one or more linear rings, an external boundary and possibly one or more internal boundaries, which mustn't intersect)
- MultiPoint (a collection of points)
- MultiLineString (a collection of line strings)
- MultiPolygon (a collection of polygons)
- GeometryCollection (a collection of geometric constructs)
- There is an open-source Java API GML4J implementing these features.

5.4.4. GML Application schemas

GML schema provides the base for application schema and the application schema declares the actual feature types and property. In the present thesis the application schemas are defined as per the requirement of Indian NSDI exchange data format (NSDE). The NSDE, basically has got three basic feature types: line, node and Area. These three types can further be divided into sub feature types like road, school, land parcels for each of the three basic feature types respectively of NSDE. But due to time constraints and sample data unavailability this is not done and only basic three data types are defined as Line_info, Point_Info and Area_Info. Although text information is also defined but it is not demonstrated in output GML data.

Broadly, the application-specific type definitions are derived from types in the standard GML schemas, or by directly including elements and types from the standard GML schemas. GML alone do not provide a schema suitable for constraining data instances; they actually provide base types and structures, which may be used by an application schema. An application schema declares the actual feature types and property types of interest for a particular domain, using components from GML in standard ways.

The base GML schemas effectively provide a meta-schema, or a set of foundation classes, from which an application schema can be constructed. User-written application schemas may declare elements and/or define types to name and distinguish significant features and feature collections from each other; the methods used to accomplish this are given in details in OGC specification.

Although GML is an extensible language and any user can flexibly design a schema as per the domain requirements still there are some basic conformance requirements that every application schema must satisfy. Specifically, a conforming GML application schema must follow the following general requirements:

1. An application schema must adhere to the detailed schema development rules for defining feature and geometry:

- Defining new feature types:

  Application schemas Developers can create their domain specific feature or feature collection types, but it must be ensure that these concrete feature and feature collection types are subtyped (either directly or indirectly) from the corresponding GML types: *gml:AbstractFeatureType* or gml:AbstractFeatureCollectionType. (OGC, 2001).

- Defining new geometry types:

  In the same way developers can create their application geometry types if GML do not have already defined geometry type construct. But, it must be ensure that these concrete geometry and geometry collection types are subtyped (either directly or indirectly) from the corresponding GML types: *AbstractGeometryType* or *GeometryCollectionType*.

- an application schema must not change the name, definition, or data type of mandatory GML elements.

- abstract type definitions may be freely extended or restricted.

- the application schema must be made available to anyone receiving data structured according to that schema.

- the relevant schemas must specify a target namespace, but it *must not* be the gml namespace i.e. http://www.opengis.net/gml.

5.5. GML Application Schema and document Structure

Just like any XML schema, GML schema has a well-defined structure, although the elements vary from application to application but overall the structure remains same. In general it has header, and body:

5.5.1. Header:

It starts with XML declaration including XML version and encoding attributes. The next tag of header is <schema> open tag, which has namespace target declaration and namespaces used in the document. An important part of XML schema is the ability to use the schema of other applications, such import and include information is included in the header. In present schema a metadata related schema metadata.xsd (Figure 7.2.a and 7.2.b) is separately designed in nsdeschema.xsd (Figure 7.4.and 7.5) it is included to make use of it. Annotation and documentation about the application also usually comes under the header.

```
<? xml version="1.0" encoding="UTF-8"?>
<schema targetNamespace="http://www.itc.nl/nsde"
            xmlns="http://www.w3.org/2001/XMLSchema"
             xmlns:gml="http://www.opengis.net/gml"
             xmlns:xlink="http://www.w3.org/1999/xlink"
             xmlns:nsde="http://www.itc.nl/nsde" elementFormDefault="qualified">
<import namespace="http://www.opengis.net/gml" schemaLocation="feature.xsd"/>
<annotation>
            <appinfo>nsdeschema.xsd, gfm 2003</appinfo>
            <documentation xml:lang="en">This schema is designed as the part of
            MSc (Geoinformatics)- 2003 thesis for Indian NSDI by Sujata Rawat
</documentation>
</annotation>
<include schemaLocation="generaltest.xsd"/>
```

Example 5.1: Header of GML application schema

5.5.2. Root Element of GML schema tree (Example 5.2)

Just like HTML and more precisely just like XML, GML also maintains the tree structure among its elements and there can be only one root element declaration allowed in GML schema and document, all the other elements comes under the umbrella of this root elements.

```
<element name="FeatureCollection" type="nsde:FeatureCollectionType" substi-
tutionGroup="gml:_FeatureCollection"/>
        <complexType name="FeatureCollectionType">
                <ComplexContent>
                        <extension base="gml:AbstractFeatureCollectionType">
```

Example 5.2:Root Element

5.5.3. Feature Definitions (Example 5.3)

These are the abstract definition of real world objects. Actually these features visibly come on the map on display. Here is the nsdeschema.xsd schema basically there are three Top-level feature types, Line_Info, Point_Info and Area_Info. These features comes under the substitution group *“gml:_feature”*.

```
<complexType name="FeatureCollectionType">
        <complexContent>
            <extension base="gml:AbstractFeatureCollectionType">
                <sequence>
                    <element name="Metadata" type="nsde:MetadataType" minOccurs="0"/>
                </sequence>
            </extension>
        </complexContent>
    </complexType>
    <element name="Line_Info" type="nsde:Line_InfoType" substitutionGroup="gml:_Feature"/>
    <complexType name="Line_InfoType">
    <complexContent>
            <extension base="gml:AbstractFeatureType">
                <sequence>
                    <element name="Start_NODE" type="string"/>
                    <element name="End_NODE" type="string"/>
                    <element name="PTS_NUM" type="string"/>
                    <element name="HEIGHT" type="string"/>
                    <element name="ELEMENT_ID" type="string"/>
                    <choice>
                        <element ref="gml:lineStringProperty"/>
                        <element ref="gml:multiLineStringProperty"/>
                    </choice>
                    <element name="MAJOR_CODE" type="string"/>
                    <element name="MINOR_CODE" type="string"/>
                    <element name="OBJECT_ID" type="string" minOccurs="0"/>
                    <element name="FID" type="string" minOccurs="0"/>
                </sequence>
            </extension>
        </complexContent>
    </complexType>
```

Cont…….

```
Cont…
<element name="Point_Info" type="nsde:Point_InfoType" substitutionGroup="gml:_Feature" id="NSDE">
        <annotation><documentation>Information about Point type data </documentation></annotation>
    </element>
    <complexType name="Point_InfoType">
        <sequence>
            <sequence>
                <element ref="gml:description" minOccurs="0"/>
                <element ref="gml:name" minOccurs="0"/>
                <element ref="gml:boundedBy" minOccurs="0"/>
            </sequence>
<element name="Element_ID" type="string"/>
            <element name="Angle" type="double"/>
            <choice>
                <element ref="gml:pointProperty"/>
                <element ref="gml:multiPointProperty"/>
            </choice>

        </sequence>
    </complexType>
    <complexType name="MetadataType">
        <sequence>
            <element ref="nsde:general_Info"/>
        </sequence>
    </complexType>
</schema>
```

Example 5.3 feature Definition

5.5.4. Enumerations and other components of Schema definition

As discussed in chapter 4 about comparison of DTD and Schema, in schema it is possible to declare enumeration for property values and thus constraints can be made on the possible input values of a property. Grouping the common properties, which are going to be used by more than one feature are some of the features of GML.

```
<element name="Data_Classification">
        <annotation>
          <documentation>0-TOpologiclally structutred
1-Topologically unstrutured</documentation>
        </annotation>
        <simpleType>
          <restriction base="integer">
            <enumeration value="1"/>
            <enumeration value="0"/>
          </restriction>
        </simpleType>
      </element>
```

Example 5.4: Enumeration and constraints on property values

5.5.5. **GML document structure** (Example 5.5):

Like Any XML document, GML document also starts with the standard xml header (Pointer 1) in which the character *encoding* and xml *version* are mentioned. Then the root element (Pointer 2) of the GML document is appeared with all namespace definitions used in the GML document and schema location. The <gml: boundedBy> element (Pointer 3) contains the bounding box of all the features in the GML document. After the bounding box of all feature collections, then starts the *featureMember*. All the application specific or general GML specified feature types are the child elements of *featureMember*. A GML document fragment based on the above application schema is given below:

```
<?xml version="1.0" encoding="UTF-8"?> --------------------------------------------1
<nsde:FeatureCollection xmlns:nsde="http://www.itc.nl/nsde"
xmlns:gml="http://www.opengis.net/gml"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://www.itc.nl/nsde shp2gmltestMod.xsd"> -------------2
        <gml:boundedBy> ---------------------------------------------------------------3
                <gml:Box srsName="">
                        <gml:coordinates>1283260.90866037,3351918.49122569
1301431.46162992,3369044.48032131</gml:coordinates>
                </gml:Box>
        </gml:boundedBy>
        <gml:featureMember>-----------------------------------------------------------4
                <nsde:Line_Info>
                        <gml:name>Outer Ring Road</gml:name>
                        <nsde:Start_NODE>1</nsde:Start_NODE>
                        <nsde:End_NODE>7</nsde:End_NODE>
                        <nsde:PTS_NUM>7</nsde:PTS_NUM>
                        <nsde:HEIGHT>4400.</nsde:HEIGHT>
                        <nsde:ELEMENT_ID>1</nsde:ELEMENT_ID>
                        <gml:lineStringProperty>
                                <gml:LineString srsName=" ' '">
                                        <gml:coordinates>1296840.24,3368704.39
1296427.27,3368558.64 1296014.30,3368024.21 1292007.13,3365782.31
1291162.92,3365310.00 1290784.10,3365098.06 1289916.97,3362242.67
</gml:coordinates>
                                </gml:LineString>
                        </gml:lineStringProperty>
                        <nsde:MAJOR_CODE> 34</nsde:MAJOR_CODE>
                        <nsde:MINOR_CODE>1120</nsde:MINOR_CODE>
                </nsde:Line_Info>
        </gml:featureMember>
</nsde:FeatureCollection>
```

Example 5.5 : Structure of the GML Document

5.6. How Interoperability is assisted in GML?

Interoperability in GML is assisted in the following ways:

GML provides a common model for writing schemas (so called object-property or feature-property model). This ensures that software that can read a GML schema can read ANY GML schema and interpret that schema to determine which elements are features and which are properties. This is not possible with a relational DBMS encoding where one cannot tell after the fact which tables represent entities (or classes) and which represent relationships. This can always be determined in GML simply by processing the schema.

GML gives extensibility as per the user demand. One person may define a feature called ROAD where another might use STREET. GML does not constrain how such objects are named, or define what properties they have. Users can however readily compare schemas on the Internet and provide *mapping* from of data from one schema to another. This has been done already in a number of pilot implementations and will become a standard part of future Web Feature Servers. To visualise in FME SafeSoft software one need to have an .xmp file (see Appendix D), which actually maps user application specific schema with that of FME GML schema (see section 6.4.1 and 7.5.1). For Ordinance Survey, UK data already there is a built in *OS (GB) Master map* which can visualise or convert OS data in FME and for TDN, The Netherlands data there is one mapping file *Top10vector.xmp [FME]*, which actually maps TDN GML Schema into FME Schema.

GML provides a set of core components for things like geometry, topology, reference systems, coverages, observations, units of measure, and map styles that are used in the creation of application schemas. Schema parsers can determine what types of GML components are being used even when a schema derives from the core GML components, these core components are key to achieving interoperability at the geometry level, topology level, etc, while the object-property model is key to interoperability at the feature level.

In terms of portrayal - there is a need to determine some sort of portrayal rule - e.g. how should one portray a road or river? A black line? How thick? A shaded polygon? What type of shading? etc. These are determined by properties of the road and river and by the users styling choices. GML does not constrain on it and as such its not a problem related to interoperability.

Interoperability is critically dependent on Extensibility, this ability of GML enables the presentation of user defined object types and that's the reason how different languages and data formats can be represented in GML (Email Discussion with Rone Lake, Galdosinc).

In the present thesis the NSDE data format is a new format, but because of the extensible and interoperable properties of GML it has become possible to represent this data in GML.

5.7. Demerits of GML:

Topology not supported: In versions earlier than GML 3 topology is not supported; this means that overlay, neighbours and other functions like these cannot take advantage of this. Although the latest version of GML 3 does support topology, but most of the available software for GML data visualization has not yet incorporated this functionality, this is one of the reasons for using GML 2 in present thesis.

Bulky File Size: One more problem with XML and thus GML in general is that files tends to be bulky and can take quite some time to parse files for processing.

5.8. Summary:

GML is an OGC recommended text based language; it is designed for transport and exchange of geographic and is not concerned with the visualization of that data. Though not specifically intended for Internet GIS, GML is ideally suited format for data transfer and exchange over the Internet GIS.

6. GML Visualization Technique

6.1. Introduction:

As discussed in chapter 5, GML is appreciated for being an easy human readable language; still the importance of graphical maps over the text cannot be ignored. As GML is concerned with representation of geographic data contents only and it has nothing to with the visual graphical presentation. So the conversion of GML document data to map has to be accomplished with the help of some rendering tools, which can interpret GML data and a styling language to style GML elements into a form that can be understood by geographic display viewers. Some potential displays formats know are:

- Scalable Vector Graphics (SVG);
- Vector Mark-up Language (VML);
- World Wide Web Consortium's X3D.

6.2. GML to SVG

Typically the visualization of GML geographic contents into suitable graphical contents involves the interpretation of the GML contents using graphical symbols, line styles, and area or volume fill. So, there must be an additional document that provides information on how to present or otherwise process the XML document. XSL (eXtensible Stylesheet Language) is known for such styling of XML files. XSL is a specification being developed within World Wide Web Consortium (W3C) for applying formatting to XML documents in a standard way. The specification defines; XSL is a language for expressing stylesheets. Stylesheets are used to describe how the content of a given structured document should be presented.

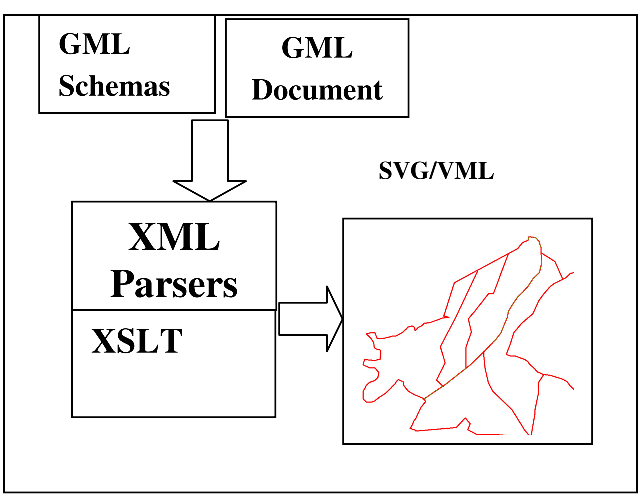


Figure 6.1, GML to Map

XSL Transformation is the important part of XSL, it transforms To do this conversion from text to a graphical output, some sort of translations is required; XSLT (eXtensible Stylesheet Language Translation) [Tennakoon, 2002] is known for such translation from one XML to other XML Document.

With the increasing requirement of displaying text based data formats, varieties of graphical render programs are available for the various XML graphical formats as:

1. native in the web browser, like Internet Explorer 5.0 + built in VML processor,
2. plug-in for the browser like (Adobe SVG viewer),
3. stand-alone viewer like Ionic Software has java applet SVG viewer, or
4. library of functions

The map styling operation can be executed either on the server or on the client side, by means Map Style Sheets completely portable from server to client and from client to client. The translation behind text GML data to SVG is a kind of mapping between GML tags and SVG definitions. Table 1 illustrates the same thing:

Correspondence between GML and SVG elements	
GML geometry element	**SVG element**
Box	rect
Point	rect, circle or path
LineString	polyline, path
Polygon	path

Table 6.1

In this translation the objects and features in GML are one to one mapped to SVG equivalent tags, the browsers are familiar with these tags and thus can interpret the meaning of GML information for graphical presentation.

6.3. Benefits of SVG:

SVG has many benefits for displaying Geographic data over other known graphics formats:

1. Easy editing: Like XML, SVG is also plain text based data format and thus it can be edited using simple text editors. See example below:

```
<g id="Line_Info">
<path fme:name="Outer Ring Road" fme:End_NODE="7"
fme:Start_NODE="1" fme:HEIGHT="4400." fme:PTS_NUM="7"
fme:ELEMENT_ID="1" fme:MAJOR_CODE=" 34"
fme:MINOR_CODE="1120"
d="M 13579.3,16785.9 l -412.97,-145.75 -412.97,-534.43 -4007.17,-2241.9 -
844.21,-472.31 -378.82,-211.94 -867.13,-2855.39 "/>
  </g>
```

Example 6.1 SVG in Text Editor

2. Better visibility: Unlike other raster based graphic formats like jpeg and gif,, which shows blurred image on zooming, SVG gives better clarity and sharp output.
3. Interactivity: Besides Zooming, Panning, SVG has unparalleled interaction properties. SVG images can be styled to respond to users actions with highlighting, tool tips, and many special effects.
4. Test Searchability: In SVG file the text remains text, and thus user can edit and search it. Querying is possible in SVG file.
5. Animation and graphic filter effects: SVG features can be animated to make it more presentable.

Figure 6.2.a: gif Image (Pixilated)

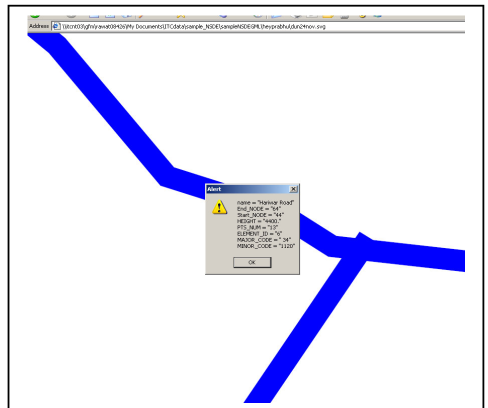


Figure 6.2.b: SVG image (better Clarity)

6.4. Available software for GML visualization:

It is substantially difficult to write a software component that works on any arbitrary GML user application schema because the component would be expected to understand any GML dataset. Reading GML documents into a system is trivial since users can have any application specific tags. So it will be difficult for a software developer to design generic software, which can understand all the application specific GML schemas. However there are some software, which can help in visualizing user-defined schema.

6.4.1. FME:

Feature Manipulation Engine: it's a FME Safesoft Inc product. It has got the functionalities for translating most of the known data formats to other data formats. Translating any data format to GML format is easy, but it will be with the namespace of fme and in the default fme schema designed.

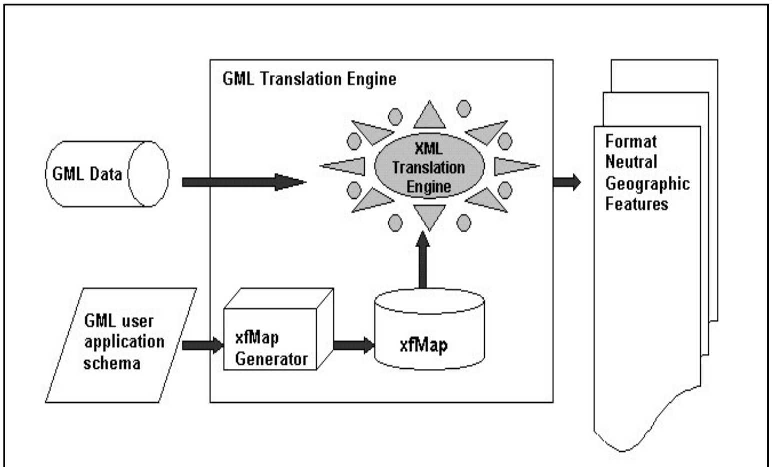


Figure 6.3: FME GML Translation Engine source: FME Site

Thus the data mapped to GML will be as per the FME schema.

However it is comparatively flexible for conversions from GML to any other data format. For that user need to design one .xmp file (Appendix D), which will map the user-defined schema to that of fme defined schema. For Ordnance Survey data and TDN Dutch data FME already has designed these mapping files. During translation one need to give the path of this mapping file in the setting of source files.

With the help of FME one can convert GML to SVG with the flexibility of using default Stylesheet or user-designed Stylesheet. It has conversion facility for most of the known formats. For other proprietary formats also this work fine, Including WFS data source. It has got quite good functionalities to work in heterogeneous format, but it is a Safe Soft Inc proprietary software.

6.4.2. FreeStyler:

FreeStyler is a Galdos Inc product and they say that it is an open-standards solution for the web based mapping of GML data. Its components provide all the needed functionality for designing interactive maps, editing and managing symbols, managing map styles, and creating Scalable Vector Graphics (SVG) or image-based maps.

FreeStyler provides the following key features:

- Standards-based web mapping, compliant with the OGC Web Feature Services (WFS), Styled Layer Descriptor, GML, W3C XSLT and SVG.
- XSLT Style Sheets for standard SVG or image-based map generation.
- Plug-ins for popular graphics editors such as Adobe Illustrator, allowing easy and rapid creation of map symbols from a host of graphical data formats.

6.4.3. The Future of SVG

The SVG format has already found grassroots support among many Web designers who are looking for an open standard to create vector graphics on the Web. It's a promising technology for geographic data in the form of maps. With the help of external CSS the appearance of SVG document can be varied easily.

Although its lack of a wide browser install base is an obvious limitation right now, SVG is being positioned as a major graphic format for the future. As with all technologies there will be an inevitable breaking-in period as SVG gains wider acceptance and Web designers learn not only how to use it, but also the best ways to use it.

7. Implementations, Results and discussions

7.1. Introduction:

Heterogeneity vs. Interoperability is the major issue of consideration among the academia of the geospatial domain. Throughout the thesis also there is a discussion about the concepts, standards and technologies to overcome the heterogeneity in geospatial data. All the technologies seem very promising to fight with the problem and to enhance the Interoperability in GIS data. This thesis started with an objective to design an interoperable data model in the context of Indian NSDI Spatial data exchange format (NSDE), for data exchange and sharing over Internet in a wider perspective. This chapter covers the requirements and design choices experiences faced in mapping the NSDE data format to a GML application schema.

7.2. Sample NSDE data generation:

As discussed in chapter 3 NSDE is an Extension of DVD (Digital Vector Data) format of Survey Of India. DVD is a FORTRAN based format. With the consensus of participating nodes of Indian NSDI, NSDE format came in to existence. It has separate files for general metadata kind of information and spatial information and each record in file is 72 bytes long. As Indian NSDI came into existence in Jan 2001 [NSDI document] so the NSDE format is not yet used widely and thus no sample data was available but with the help of DVD and some arbitrary information at the required fields, a sample NSDE data was generated for the road map and facility locations map of Dehradun City in India.

In NSDE format, there are four metadata related files (VOLDIR, GENINFO, QUALINFO, TOPOINFO) and *n* numbers of DATACAT and DATAFIL files, where *n* is the number of themes. For the demonstration purpose VOLDIR, GENINFO, DATACAT, DATAFIL files are generated, but as DVD do not have information about QUALINFO and TOPOINFO in its base format, so these files are not generated but as such their unavailability is not going to make much difference on the demonstration of the output.

The data was generated using small visual basic programs and sometimes manually also. The data generated does not represent actual information but if actual correct data files are available then also the system will work is a similar way. Below are the samples of data for VOLDIR and DATAFIL files; every record in the file is 72 bytes long. The files have fillers (empty bytes) to cater the future extension or modifications in NSDE. (Appendix C).

A: VOLDIR FILE

```
52H                                                  --1.0.1 --000100010001
voldir --     0 0      0            15   0 0                              x
    12       0 0 0
     1   0 52HBKW...BLACK                                           1      0
     2   0 52HGRW...GREEN                                           1      0
     3   0 52HCW...CONTOUR                                          1      0
     4   0 52HRW...RED                                              1      0
     5   0 52HBLW...BLUE                                            1      0
     6   0 52HCROS...CROSS-MARKS                                    1      0
     7   0 52HADMN...ADMINISTRATIVE INDEX                           1      0
     8   0 52HGDW...GRID                                            1      0
     9   0 52HNW...NAME                                             1      0
    10   0 52HCOMP...COMPILATION INDEX                              1      0
    11   0 52HGLW...GLACIER                                         1      0
    12   0 52HSPLF...SPECIAL FOOT-NOTES                             1      0
```

B: DATAFIL FILE (Line Data)

```
LINE        1          1          7          7     4400.
  1296840.24   3368704.39   1296427.27   3368558.64   1296014.30   3368024.21
  1292007.13   3365782.31   1291162.92   3365310.00   1290784.10   3365098.06
  1289916.97   3362242.67
 34            1120          CINTBR
LINE        2          2         10         47     4400.
  1289892.67   3362266.96   1290354.23   3362169.79   1289771.21   3362096.92
  1289455.42   3361951.16   1289261.08   3361878.29   1289115.32   3361829.70
  1288945.28   3361756.83   1288726.65   3361708.24   1288337.97   3361659.66
  1288070.76   3361416.73   1287803.55   3361295.27   1287706.38   3361173.81
  1287560.63   3361003.77   1287512.04   3361319.57   1287317.70   3361416.73
  1286929.03   3361319.57   1286831.86   3361149.52   1286686.11   3361028.06
  1286540.35   3360882.31   1286370.31   3360785.14   1286175.97   3360663.68
  1286103.09   3360590.80   1285884.46   3360542.22   1285665.83   3360542.22
  1285447.21   3360469.34   1284864.19   3360347.88   1284451.23   3360347.88
  1284232.60   3360882.31   1284159.72   3360882.31   1284111.14   3360833.72
  1283819.63   3360833.72   1283552.41   3360736.55   1283819.63   3360469.34
  1283941.09   3360347.88   1284111.14   3360080.66   1284159.72   3359959.20
  1284159.72   3359837.74   1284184.01   3359643.41   1284256.89   3359521.94
  1284451.23   3359424.78   1284645.56   3359400.48   1284742.73   3359279.02
  1284742.73   3359133.27   1284742.73   3358987.52   1284718.44   3358866.06
  1284694.15   3358793.18   1284694.15   3358598.84
 34            1120          CINTBR
```

Figure 7.1: Sample data for VOLDIR and DATAFIL

In Figure 7.1:A, first record include map sheet Code=52H, NSDE Version=1.0.1, NSDE layer=0001 and NSDI sub-layer=0001 Information. Similarly the rest of the records have information. Please see the NSDE data format in Appendix C for the meaning of information in corresponding record.

7.3. GML Application Schema Design:

OGC recommended GML (Geographic Markup Language) is popularly accepted as a tool for geospatial data exchange and transfer, as discussed in chapter 5, it is based on W3C recommended XML and thus inherits all the properties of it. Like XML, GML is also extensible and allows users to define their domain specific elements, feature types and geometry types. So for this thesis also, a schema is designed by keeping the requirements of NSDE in mind and the recommendation of OGC. So this phase started with the understanding of GML and mapping of NSDE in GML domain. For the demonstration purpose and due to time constraint the schema is designed with a moderate approach. Not all the information is mapped but this is assured that the things will work fine if a full fledge schema is designed on similar pattern.

For Metadata and attribute information, GML gives flexibility to define own tags, so these are designed according to NSDE. For spatial information also user defined feature types and geometry types can be defined. In the present schema, three feature types are defined and no geometry is defined at present. There are two schemas designed one for metadata kind of information and other for feature information. The schemas are designed with the help of XMLSpy.

7.3.1. Metadata Schema:

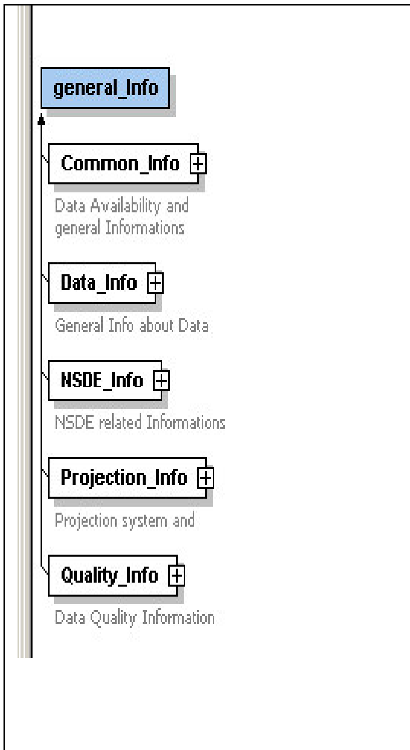


Figure 7.2.a: Overall Metadata schema

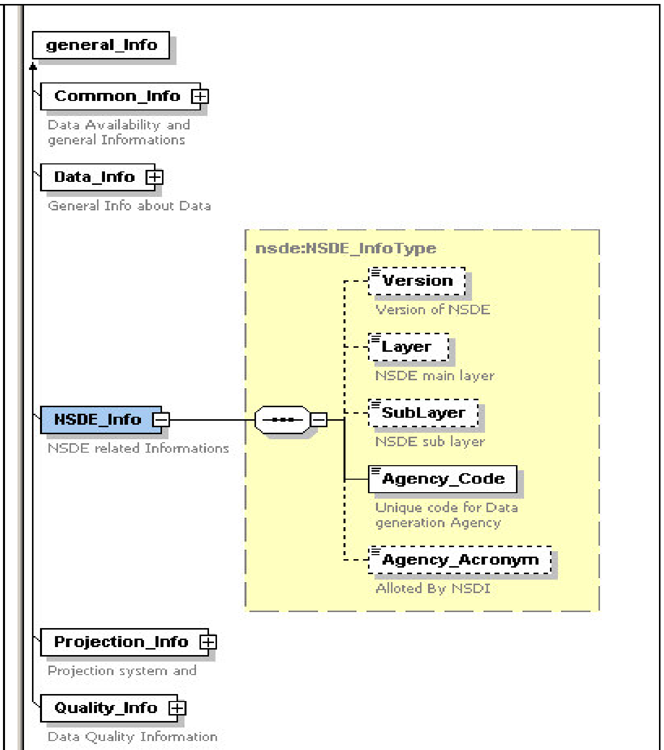


Figure 7.2.b: NSDE_Info Expanded.

In the above figures, the left image is the overall metadata schema and the right contains the expanded details of *NSDE_Info* type information. All information is divided into five groups (*Common_Info*, *Data_Info*, *NSDE_Info*, *Projection_Info, Quality_Info).* All these groups are made complex type, so if any other user wants to import this schema than he can make his tags of the *types* defined in this schema.

7.3.2. Feature Schema:

Rules for creating feature schema are discussed in chapter 5 In that it was described that all the features of a schema should be contained in one *featureCollection* defined with the user specified namespace. In this schema also the *FeatureCollection* is an application-defined *featureCollection* and it has *nsde* namespace. It is made complex type (*nsde: featureColectionType*) and belongs to the substitution group *gml:_featureCollection* (acts as a placeholder in the definition of an actual element type),.

There are four application specific feature types (*Area_info, Line_Info, Point_Info and Text_Info*). These *feature types* are derived from *gml: AbstractFeatureType* and thus inherits the property of it. All these feature types have *gml:_feature* as the head of a substitution group [see OGC 2001 specifications]. Figure 7.3 is the graphical representation of over all feature schemas.

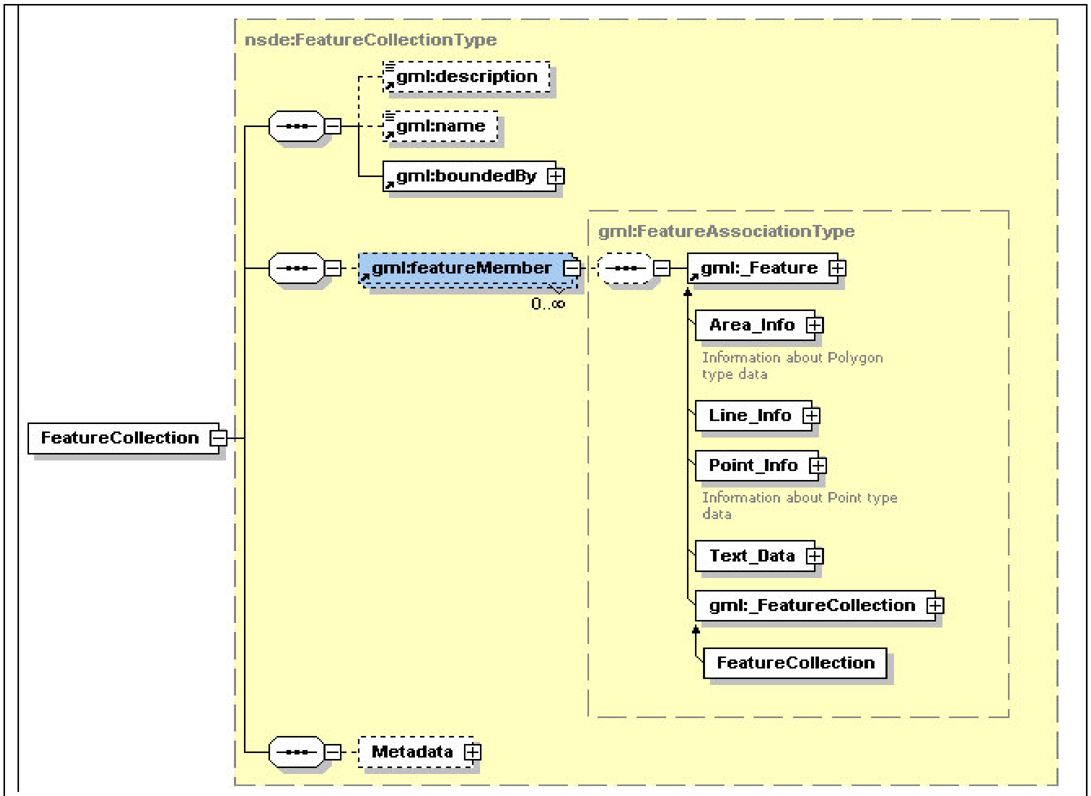


Figure 7.3: Overall Feature Schema Screen shot of XMLSpy view

Figure 7.4, graphically representing the details for line feature type (*Line_Info type*). Area feature, point feature has similar structure with their specific properties of being a polygon and a point respectively. Each feature type has both attribute information and the spatial information. The attribute information elements are named in such a manner that the meaning of the tag could easily be understood to the user on the other end. However, for the spatial Information, GML tags are used. This information can be provided in six ways (gml:lineStringProperty, gml:centerLineOf, gml:edgeOf, gml:multiLineStringProperty, gml:multiCenterLineof, gml:multiEdgeOf,) so as to take care of diverse group of users, who may want to import this schema for their applications. In the NSDE file format, the spatial information is given as a continuous string of X and Y. So here for demonstration purpose the gml:lineStringProperty is used see figure 7.5.

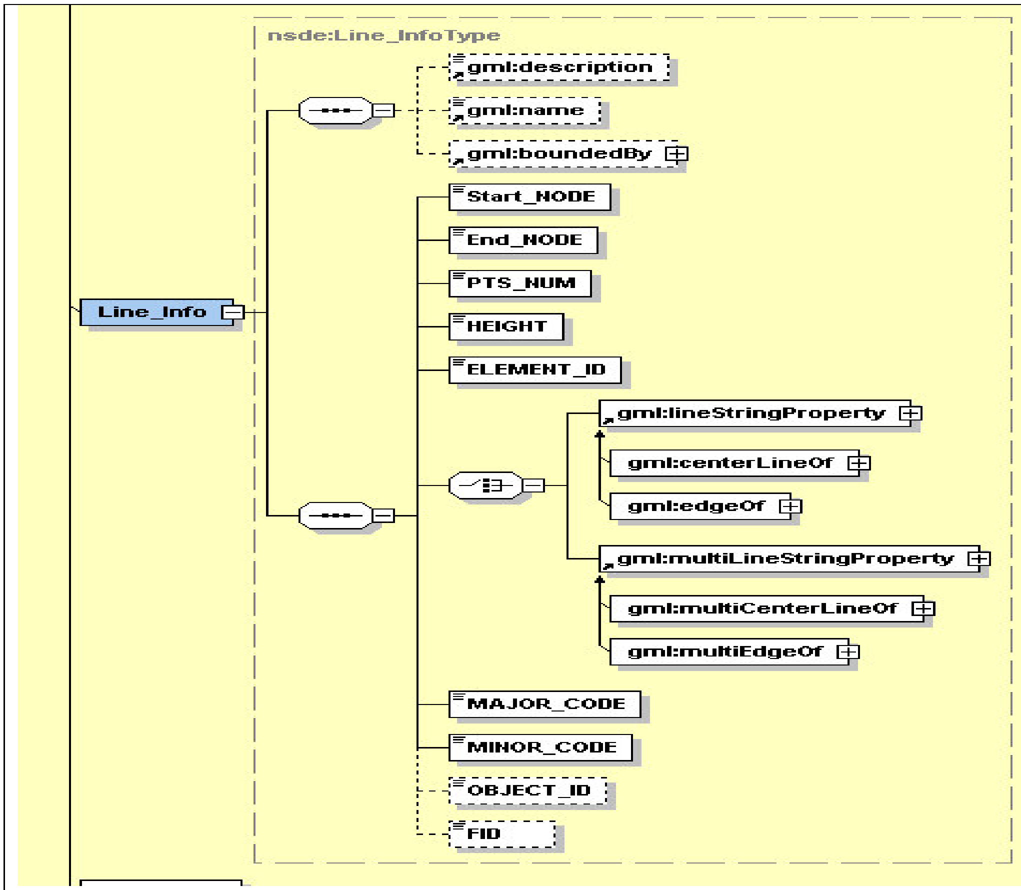


Figure 7.4 Expanded Line feature Screen shot of XMLSpy view

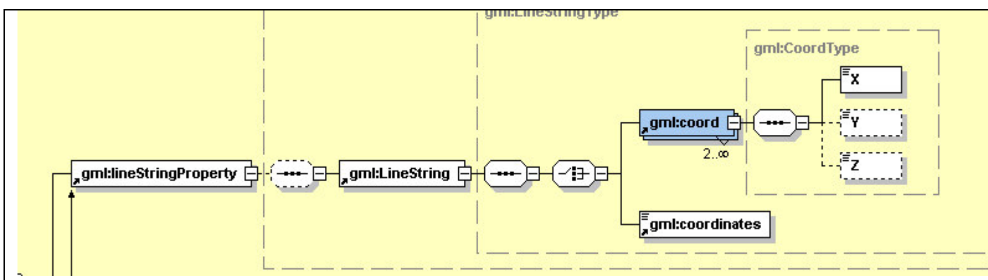


Figure 7.5: gml:lineStringProperty Details Screen shot of XMLSpy view

7.4. NSDE to GML document Using Java:

The GML document can be written with a text editor. For converting the NSDE data file to GML document, Java program is used, which writes a GML document as text file with the file extension ".gml". The rational behind using java is that it is well known for being a platform independent language, as it makes byte codes, which can run on any platform using the java virtual machine. Along with other basic classes, the I/O class is primarily used.

Java program is designed to read the input NSDE files and to write the corresponding output GML tags. The tags are in the pattern of the already designed schema. The structure of the output GML document is described in Example 5.5 of the thesis.

7.5. GML data visualization

As discussed in chapter 7, that the GML document is required to be transformed to a form, which can be interpreted by a graphic viewer like Adobe SVG viewer as plug in with HTML or any standalone software [FME; Quak et al, 2002]. The transformation of GML to SVG is already discussed in chapter 6. In this thesis, for the demonstration purpose the GML data is viewed in following ways:

7.5.1. GML to SVG conversion using FME:

As discussed in chapter 6, FME supports GML2 but it cannot understand the all the generic GML application schema. If one wants to visualize his data using FME then he has to design one mapping file, which actually maps application schema with the schema defined by FME. Mapping is basically about making FME understand the namespace of the user and the pattern or the tree of the application schema. Once the mapping is done then FME can help in translating in most of the known formats with its built in XSLT. For FME mapping file see appendix [Appendix-D]. The path of this mapping file is required at the time of conversion. It can be done by changing the *settings* of the source dataset [FME].

To understand the difference between various graphical formats, GML data was translated to GIF, VML, SVG, shape. There are few finding from these translations:

GIF:

- It can be viewed on any viewer, which supports images.
- It’s a raster based format, so individual feature cannot be taken.
- On Zooming the information gets pixilated and blurred.
- Attribute information cannot be associated with the features.

VML:

- Microsoft IE 5.0 and higher browsers support it.
- Solving the problem of pixilation cannot be demonstrated, because there was no provision for zooming and panning in default, however the possibilities are not checked for adding it with the help of some scripting.
- This is a plain text data format, and thus can be modified simply by changing the property values. But it’s comparatively difficult to modify and as the position of the features primarily depends upon the styling sheet so after changes are difficult to incorporate.

Shape:

- It creates three files (.shp, .dbf, .shx) of shape format shape files. These can be viewed with the help of shape file viewer.
- The translation was done to check the effect of conversion on data, for this one original shape file was converted to GML and then back to shape, as such there was no prominent difference in that, only the name of feature file was changed but that is not a problem. However this fact cannot be ignored that it cannot be viewed on web.

SVG:

- The pros and cons of SVG are already discussed in chapter 7. SVG is a W3C standardization effort for an XML-based vocabulary for vector graphics. It seems to be more promising as compared to its rival formats.
- SVG can be viewed in Internet explorer and there is a freely downloadable ADOBE SVG viewer.
- All the discussed benefits are not tested, but the files are easy to understand and edit. Few experiments were done with varied stroke color and line width (see Image 7.6.c and d).

7.5.2. GML visualization using TDN Dutch Software:

The Delft University of Technology, Netherlands, developed a software [Vries] for visualizing TDN GML data. The software was developed using an xslt-stylesheet, with the Microsoft xslt-processor on the DOS prompt-msxsl.exe (Discussion with Ms. Marian Vries). The NSDE application schema and data was visualized through that software. The results were satisfactory, although it could read only ID attribute.

7.5.3. Output Images from Different Sources:

Below are the Outputs of different graphic languages in different viewer: There are differences in the output SVG file as per the discussion above.
Image 7.6.e: is viewed in FME universal viewer to see the of GML data and Shape files different format data of same location, similar output came on overlaying shape file converted GML and another shape file.

Image 7.6.a: GIF image

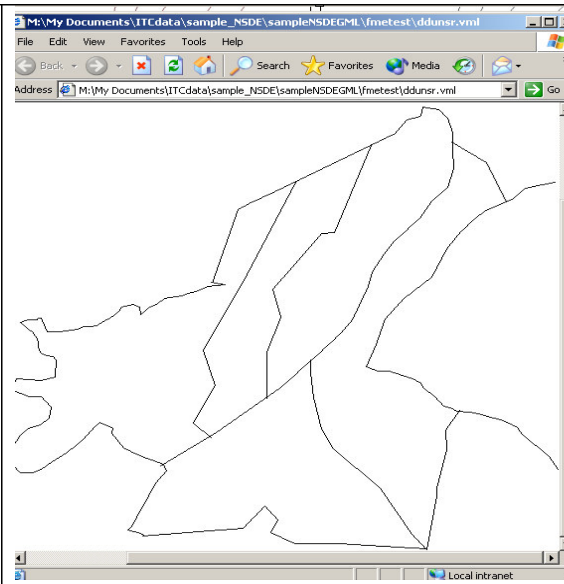
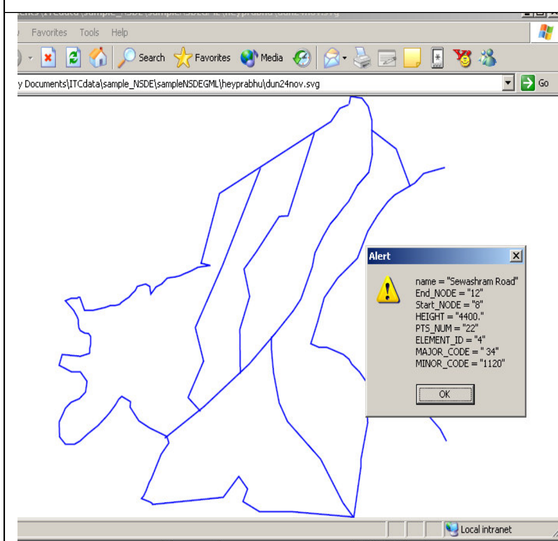


Image 7.6.b: VML O/p (Microsoft IE View)



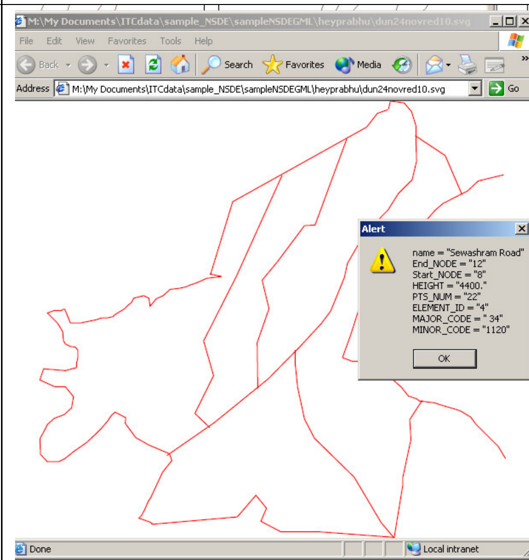


Image 7.6.c and d: SVG Output From FME (Microsoft IE View)

With Blue and Red Stroke respectively and “onclick” functionality to see the attributes.

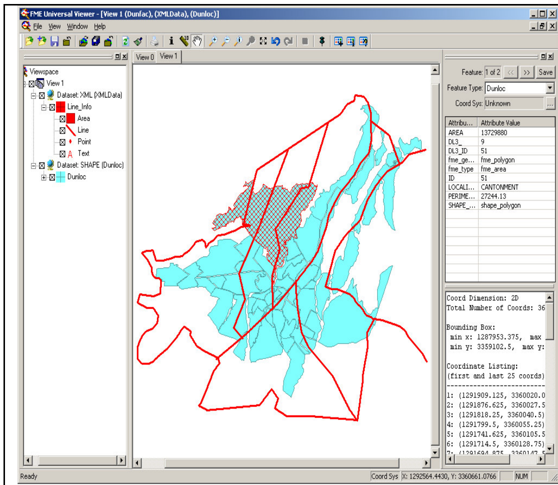


Image 7.6.e: XML and Shape files as Viewed in FME universal viewer.

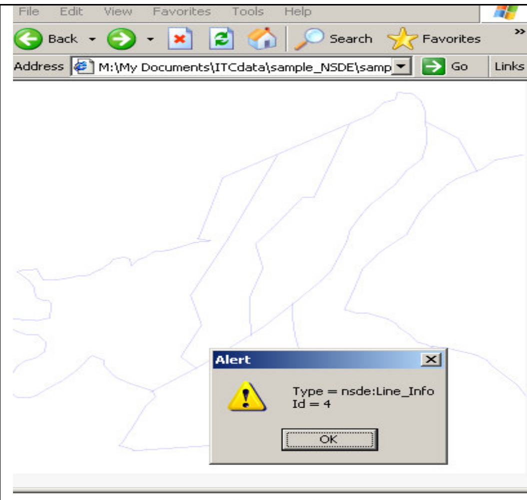


Image 7.6.f: SVG o/p from TDN Dutch Software as viewed in Microsoft-IE.

7.6. Web Page:

It is planned to put this work on some server, from where the Indian NSDE data format users can translate their data to GML. As at present no GML to SVG translator is available for online translation, in the present web page, already uploaded GML can be kept for online visualization, which is to be done separately using some software like Safesoft FME [Juan Chu Chow]. If time was not the major constraint, this also could have been done for at least NSDE- GML version specific format and hopefully for generic schema as well.

7.7. How Indian NSDI can benifit from this System?

The thesis is aimed to serve the Indian NSDI specified NSDE data file users. As discussed in earlier chapters, the GML application schema is primarily designed as per the NSDE format. So the NSDI nodes or any with NSDE data file user can use this system for online exchange or transfer of data in a platform independent interoperable format (GML). As I could not find any GML to SVG online translator, so this data cannot be visualized at present. However already GML to SVG translated files can be uploaded and viewed on web browser, with the help of this system. This system can be adopted by NSDI, for NSDE file to GML file, there is a provision in this system, and for GML to SVG either a independent software can be developed, which can do this translation online, or NSDI can do this job offline for the time being using existing softwares like FME and can upload the corresponding SVG files on the web.

7.8. Summary

The aim of performing translation from GML to other formats was to compare various known graphic formats and also to compare and check the possibility of visualizing an application schema defined GML data by different ways. There is much heterogeneity at different level but it cannot be denied that GML holds promise in providing a standard way to share and use the spatial data. For Visualization purpose SVG is considerably promising, things could have been better if there was any generic XML reader and SVG Converter.

8. Conclusion and Recommendations

8.1. Conclusions

This thesis introduces the issues of data interoperability, advantages of GML, its mechanism for data interoperability and the design issue of a GML application schema model for the exchange and sharing of spatial data in Indian organizations as per the Indian NSDI specified NSDE format.

Sample data of Indian NSDI specified national Spatial Data exchange format (NSDE) is taken for conversion from NSDE to GML. The GML application schema is designed as per the NSDE format (see Appendix C). For the demonstration purpose an NSDE sample data is converted to GML document, using java coding, however due to unavailability of sample NSDE data, the data contains arbitrary values and the whole data is not converted but this will not affect the output. GML data is displayed into SVG and the demonstration shows that the GML-based interoperable data can be displayed as a SVG map. SVG is scalable and resolution-independent and so have high quality maps even on zooming for multiple times. The SVG maps can be viewed on Internet Explorer and the Interactivity can be made as per the efficiency of the programmer. The translation from GML to SVG can be done by XSLT and can be modified with stylesheets. There are some readymade software available, which can perform GML to SVG conversion.
As interoperability standard, GML allows us to bridge the gaps among different data sources, vendors, databases and formats. GML gives users the capability to easily and dynamically publish and exchange data in an open, non-proprietary industry-standard format on the Web, thus maximizing the re-use of geospatial data, eliminating time-consuming data conversion and reducing associated costs. A range of programs on different platforms via the Internet can access the information. The high quality and colorful SVG map transformed from the GML-based data shows an attraction to users, which can improve the public accessibility to existing data. GML holds promise to lead an exciting interoperable future via online interactive Web maps and spatial Web services. But because the development of support software systems for GML-data is still at its beginning stage, the advantages of GML-based data are not fully adapted by GIS data providers. During the literature review, very few data providers like Ordnance Survey UK and Dutch topographic data services are found those have implemented the usage of GML In the database.

As a new interoperability approach, GML still has some limitations. GML is not intended to solve all geo-processing interoperability problems. It still cannot fully solve the problem of semantic interoperability. For example, GML provides users the ability to create application schemas to model their data, but different users (i.e., data providers) may use different names to represent the same feature, e.g., one user may decide to create a GML schema with a *building* feature while another user may use a *house* feature for essentially the same entities. Thus the second user must know the schema created by the first user in order to integrate the data from the first user into his. The same problem arises in the case of software also, as one software cannot read the entire range of possible schemas in the world and thus a mapping is required between the application schema and the software schema. Otherwise users and software cannot fully understand what the GML represents without understanding these schemas.

The initiatives towards Interoperability are appreciable; still there is much to do. The real data Interoperability is to provide seamless communication between remote GIS databases without having prior knowledge of their underlying semantics. A real interoperable GIS database should provide transparent communications at data model and application semantics level (Bishr, 1998).

8.2. Recommendation

In the present scenario it is not possible to make full use of available software for GML data generation and GML to other data and vice versa conversion. There are two reasons behind it, first, as these software do not support generic schema and thus could not always represent the whole meaning of the GML elements. However mapping between the user application schema and software application schema can be done partially, but then this is required to be done for all software. It will be easier for the users if such mapping can be standardized. So that only one mapping file for one schema will work. The second reason is high cost of proprietary software, so some open source software could be made available to the users, for generic GML schema based data translation from other GIS formats. To encourage the attitude of Interoperability among GIS data providers and users, these softwares can be provided free of charge.

Indian NSDI specified data format is also not fully object oriented, as it’s the early stage of its designing so it will be better if instead of line, area and points, the NSDE format is defined in terms of objects like ROADS, BUILDINGS etc. It is possible to do this with slight modifications in MAJOR CODE and MINOR CODE list, by grouping some of the similar features in one MAJOR CODE and making MINOR CODE as one of the attribute to distinguish features within the group.

Indian NSDI can use the proposed system for online exchange of data through a platform independent Interoperable format. For making this system fully working, the Indian NSDI can develop free source software for NSDE (GML) version to SVG conversion on the fly.

9. References

- Aloisio, G., Milillo, G., Williams, R., (1999) An XML architecture for high-performance web-based analysis of remote-sensing archives, *Future Generation Computer Systems*, *Vol.16, pp. 91–100.*

- Arctur, D., Hair, D., Timson, G., Martin, E.P., and Fegeas, R. (1998) Issues and prospects for the next generation of the spatial data transfer standard (SDTS). *Int. J. Geographical Information Science, vol. 12, no. 4, pp. 403-425.*

- Badard, T. and Richard, D. (2001). Using XML for the exchange of updating information between geographical information systems. *Computers, Environment and Urban Systems, Vol. 25, pp.17-31.*

- Bertolotto, M. and Egenhofer, M. J. (2001) Progressive Transmission of Vector Map Data over the World Wide Web. GeoInformatica. 2001. vol. 5, no. 4. pp 345-373.

- Bishr, Y. (1998) Overcoming the semantic and other barriers to GIS interoperability. *Int. J.Geographical Information Science, vol. 12, no. 4, pp. 299-314.*

- Bishr, Y., and Radwan, M.(2000), *GDI architectures* in Geospatial Data Infrastructure, Ed-Richard Groot and john McLaughlin, pp 130-149.

- Buelher, K. and McKee, L. (1996) Introduction to Interoperable Geoprecessing. The OpenGIS Guide, Open GIS Consortium.

- Chang, C., Chang, Y., Chuang T., (Bridging Two Geography Languages: Experience in Mapping SEF to GML, http://homepage.iis.sinica.edu.tw/~trc/papers/gml2003 (last accessed 22nd Nov 2003).

- Devogele, T., Parent, C., and Spaccapietra, S. (1998) On spatial database integration. *Int. Journal of Geographical Information Science, vol. 12, no.4, pp335-352.*

- *Galdos Inc, www.galdos.com ,(Last Accessed on 20th Nov 2003)*

- Goldfarb, C., and Prescod, P., (1998), *The XML Handbook.* Prentice-Hall, Inc., Upper Saddle River, NJ 07458,.

- ETMIE White Paper: chapter on Interoperability.

  http://wwwlmu.jrc.it/etemii/reports/chapter3.pdf (last Accessed 26th Nov 2003).

- FME Safesoft, GML v2 Reader/Writer

http://www.safe.com/reader_writerPDF/gml2.pdf (Last accessed 20th Nov 03)

- Georgiadou Yola (2003), **Is Survey Of India ready for the NSDI?,** *GIS Development, Vol. 7, Issue 2, February 2003, pp. 38-39"*
- Hansen Trevor, Marriott Kim, Meyer Bernd And Stuckey P.(2002), Flexible Graph Layout For The Web, *Journal of Visual Languages and Computing* , Vol. 13, pp 35-60.
- Henning, S., Geography Markup Language- the foundation of Geo-Spatial Interoperability?, Nordic GIS Conference 2001 Helsinki, 8-10. October 2001.
- Houlding, S., 2001,XML—An opportunity for <meaningful> data standards in the geosciences. Computers & Geosciences, 27:839–849.
- Indian National Spatial Data Infrastructure (NSDI) Strategy and action Planning (2001), NSDI - Strategy and Action Plan, ISRO-NNRMS-SP-75-2001 Discussion Document.
- Indian NSDI, www.nsdiindia.org (Last accessed 17th Nov 03)
- Kim, H., (2001), An XML-based modelling language for the open interchange of decision models, Ž. Decision Support Systems Vol. 31 pp. 429–441
- Kim H. and Kim M., XML and Inter-Operability in Distributed GIS, http://www.fig.net/figtree/pub/fig_2003/TS_10/PP10_4_Kim_Kim.pdf (Last accessed 11th Nov 2003).
- Kraak, M., Knippers, R., kolk, B., Bakker, N., Bruns B., Oosterrom, P., Quak, C., Bregt, A., Bulens J., Zeeuw,C., Heres, L.,. (2002), Top10Vector Object Oriented Design Data model version 1.1.2., http://kartoweb.itc.nl/top10nl/TOP10NL_eng/download/pdf/Design.pdf (last accessed on 11th Nov 2003).
- Lake, R. (1999) Introduction to GML Geography Markup Language, http://www.w3.org/Mobile/posdep/GMLIntroduction.html (last Accessed 26th Nov 2003).
- Lake, R. (2000) Making Maps With Geography Markup Language (GML),

  http://www.galdosinc.com/files/MakingMapsInGML2.pdf ((last Accessed 26th Nov 2003).
- Lake, R. (2001), GML 2.0 – Enabling the Geo-spatial Web, http://www.galdosinc.com/files/GML2-PoweringTheGeoWeb.pdf (last Accessed 26th Nov 2003).
- Laurini, R. (1998) Spatial multi-database topological continuity and indexing: a step towards seamless GIS data interoperability. *Int. J. Geographical Information Science, vol. 12, no.4, pp. 373-402.*
- Maruyama H, Tamura K. and Uramoto N., XML and Java Developing Web Applications, Addison Wesley, ISBN 817-808-023-0.

- May Y., Development of a Global Conceptual Schema for Interoperable Geographic Information, http://www.ncgia.ucsb.edu/conf/interop97/program/papers/yuan/yuan.html, (last Accessed 26th Nov 2003).

- Meneghello, M. (2001) XML (eXtensible Markup Language): The New Language of Data Exchange, *Cartography, Vol. 30, No. 1.*

- Neumann A. and A.M. Winter, editors. Time for SVG—Towards High Quality Interactive WebMaps. International Cartographic Association, 2001.

- Open GIS Consortium, Inc. Open GIS specification - geography markup language (GML). Technical Report version 2.0 (01-029), OGC, March 2001.

- OGC (2003) http://www.opengis.org/techno/documents/02-023r4.doc

- Quak, W., Vries, M., Tijssen, T., Stoter, J. and Oosterom, P (2002), GML for exchanging topographic data, 5th AGILE Conference on Geographic Information Science, Palma (Balearic Islands, Spain) April 25th-27th 2002.

- Safesoft Inc, FME, An XML-Driven Data Translation Engine for GML 2.0, White paper, http://www.safe.com/solutions/whitepapers/gml_data_translation.htm (Last Access 26th Nov 2003).

- Scalable Vector Graphics (SVG) 1.0 Specification, http://www.w3.org/TR/SVG/ (Last access 25th Nov 2003)

- Tennakoon, W. (2002), Visualization of GML data using XSLT, http://www.itc.nl/library/Papers_2003/msc/gim/tennakoon.pdf (Last Access 20th Nov 2003).

- Visser, U., Stuckenschmidt H., Schlieder C. (2002), Interoperability in GIS – Enabling Technologies, 5th AGILE Conference on Geographic Information Science, Palma (Balearic Islands, Spain) April 25th-27th 2002.]

- VOISARD, A. and SCHWEPPE H. (1998), Abstraction and decomposition in interoperable GIS, *Int. j. geographical information science, , vol. 12, no. 4, 315-333.*

- Vries Marian, GML, and SVG: from content to presentation, **http://www.svgopen.org/abstracts/de_vries__gml_and_svg.html** (Last Access 24th Nov 2003).

- Web Feature Server, Galdos Inc, http://dali.galdosinc.com/wms/client (Last access 15th Nov 2003).

- XML Schema, http://www.w3schools.com/schema/default.asp (Last Access 20th Nov 2003).

- XML, http://www.xml.com (Last Access 26th Nov 2003).

Personal Communication:

- Achyut P Rao, ESRI India.
- Amarnath Gupta, San Diego Supercomputer Center, University of California San Diego
- Brig. Girish Kumar, Deputy Surveyor General of India, Survey of India.
- Juan Chu Chow, Safesoft FME.
- Marian de Vries, Delft University of Technology, Delft, Netherlands.
- Milan, Galdos Inc.
- Ordnance Survey UK, Technical help Line.
- Rone Lake, Galdos Inc.
- Tyng-Ruey Chuang, Academia Sinica, Institute of Information Science, Taiwan.

Appendix A

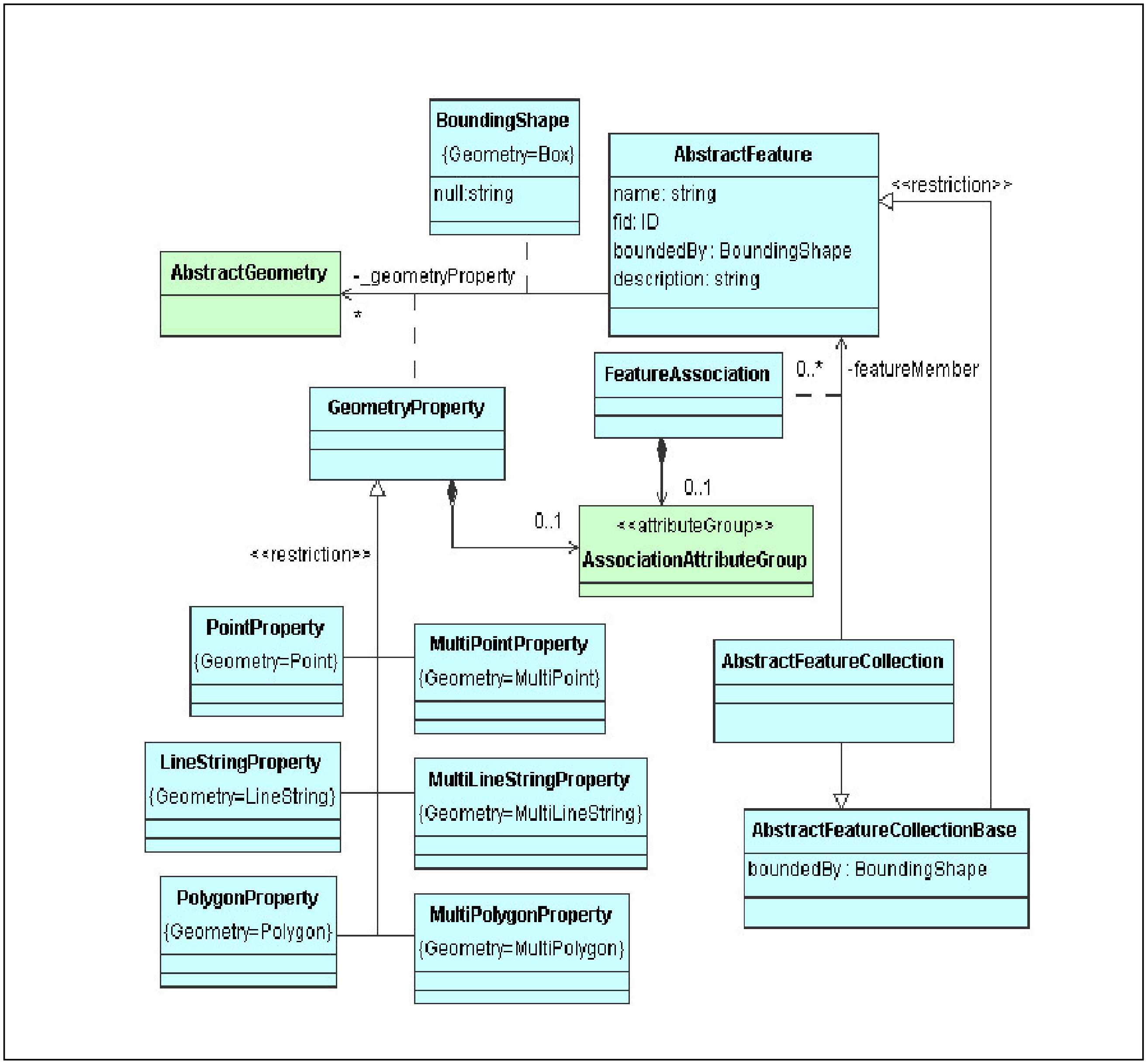


Figure A: UML Model of the GML feature Schema. Source: GML2, OGC, 2001

Appendix B

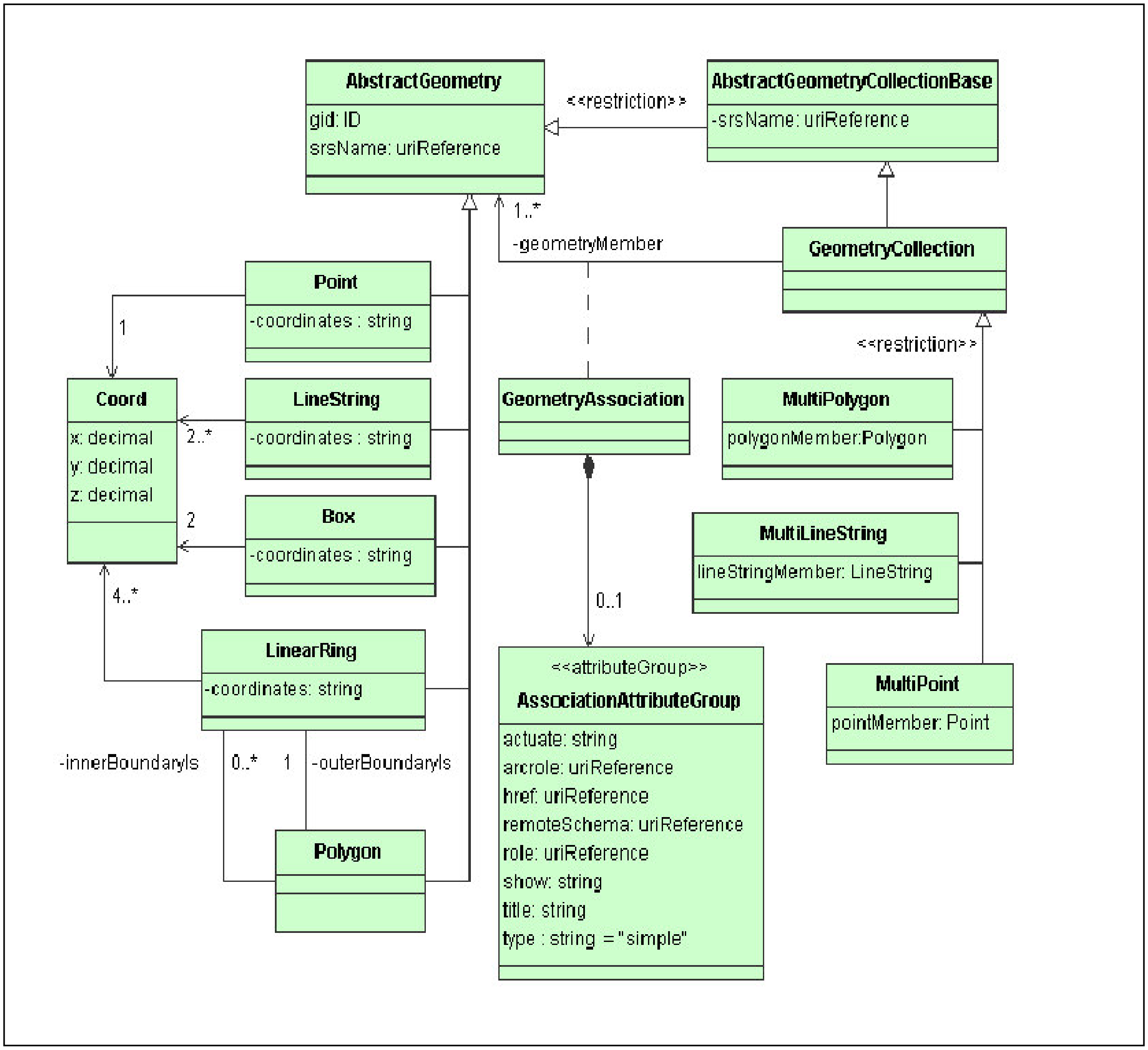


Figure B: UML Model of GML Geometry Schema Source: GML2, OGC, 2001

Appendix C

NATIONAL SPATIAL DATA EXCHANGE FORMAT (ORGANIZATION)

File Structure

Figure 1 : Data Organisation under NSDE

File No.	FILE NAME	DESCRIPTION
1	VOLDIR	Volume Directory Information
2	GENINFO	General Information
3	QUALINFO	Data Quality Information
4	TOPOINFO	Toposheet specific general information
5	DATACAT	Data category records for base category/layer/spectral band number 1
6	DATAFIL	Data records for base category/layer/spectral band number 1
7	DATACAT	Data category records for base category/layer/spectral band number 2
8	DATAFIL	Data records for base category/layer/spectral band number 2
---	--------	---
---	--------	---
4+2*p-1	DATACAT	Data category records for base category/layer/spectral band number p
4+2*p	DATAFIL	Data records for base category/layer/spectral band number p

9.1. Table 1 : DESCRIPTION AND FORMAT OF FIRST TWO RECORDS COMMON TO ALL FILES

Rec. No	Field	Contents	Type (Fortran)	Format	Starting Byte	Ending Byte	Remarks
1	1	Map Sheet Code	CHARACTER	A40	1	50	Topo Sheet Number as per Scale e.g. 56G/12
		Filler	-	-	51	52	2 Spaces
	2	NSDE Version	CHARACTER	A6	53	58	Version of the exchange format in which data is organized e.g. 1.0.1
		Filler	-	-	59	60	2 Spaces
	3	NSDI Layer	INTEGER*2	I4	61	64	NSDI Main Layer ,e.g. Topography-001, Geology/Mining-002 etc.
	4	NSDI Sub-Layer	INTEGER*2	I4	65	68	Sub-layer Classification e.g. Coastal Geomorphology etc.
	5	Agency Code	INTEGER*2	I4	69	72	Unique Code of Data Generating Agency
2	1	File Name	CHARACTER	A7	1	7	7- Digit file name as per figure 1.
	2	Filler	-	-	8	9	2 Spaces
	3	Volume Code	INTEGER*4	I6	10	15	6- Digit Code, Unique for Data Set. Provided for future use.
	4	Data Type	INTEGER*2	I2	16	17	Indicates type of Spatial Data 1 if data is base category wise (Vector Format). 2 if data is Layer Wise (Vector Format) 3 if raster multiband image (band wise) 4 if Single band raster image values 5 if Single band coded raster image
		Filler	-	-	18	18	1 Space

9.1. Table 1 : DESCRIPTION AND FORMAT OF FIRST TWO RECORDS COMMON TO ALL FILES							
Rec. No	**Field**	**Contents**	**Type (Fortran)**	**Format**	**Starting Byte**	**Ending Byte**	**Remarks**
	5	Serial Number of Media	INTEGER*2	I6	19	24	For data on Multiple Media only. Other-wise this value is zero
	6	Total Number of Records in the file	INTEGER*4	I12	25	36	
	7	Base Category / Layer serial No.	INTEGER*2	I6	37	42	Corresponds to field 1 of Record number 4 to 3+p of VOLDIR. Applicable only for files DATACAT and DATAFIL. FOR Other files this value is Zero.
		Filler	-	-	43	44	2 Spaces
	8	Data Classification	INTEGER*2	I2	45	46	0 Topologically Non-structured 1 Topologically Structured
		Filler	-	-	47	72	26 Spaces

Table 2 : VOLUME DIRECTORY (VOLDIR) RECORDS							
Rec . No	**Field**	**Contents**	**Type (For-tran)**	**Format**	**Starting Byte**	**End Byte**	**Remarks**
3	1	Availability of Data as One Single file or Multiple Files Or No. of Layers Or No. of Bands (p)	INTEGER*2	I6	1	6	This Value is zero for single file and non-zero (p) for multiple files, where p is the total number of base categories/ Layers mentioned in the volumes. Record no. 4 onwards will appear only if this value is non-zero. Also the next field of this record is meaningful only if this value is non-zero. In case of Raster data, each file will contain data for a particular band. In case of Layer data, the file will contain data pertaining to a layer.
	2	Whether data is base category wise / layer wise/ Image band wise	INTEGER*2	I6	7	12	This value is 1 if data is base category wise (Vector Format). 2 if data is Layer Wise (Vector Format) 3 if raster multiband image (band wise) 4 if Single band raster image values 5 if Single band coded raster image
	3	Whether File TO-POINFO exists	INTEGER*2	I2	13	14	Relevant only for Topographic data 1 for yes 0 for no
	4	Whether File QUALINFO exists	INTEGER*2	I2	15	16	Flag for existence of file containing Quality processes and parameters; 1 for yes 0 for no
		Filler	-	-	17	72	46 Spaces
4 to	1	Base Category/	INTEGER*2	I6	1	6	Order of listing of Base Categories, Layers,

Table 2 : VOLUME DIRECTORY (VOLDIR) RECORDS							
Rec . No	**Field**	**Contents**	**Type (For-tran)**	**Format**	**Starting Byte**	**End Byte**	**Remarks**
3+p		Layer/ Band Serial Number					Bands
	-	Filler	-	-	7	7	1 Space
	2	Whether Attribute table Exist	INTEGER*2	I2	8	9	0 indicates no attribute data attached in DATA-FIL, ELSE number of attribute tables associated
	-	Filler	-	-	10	10	1 Space
	3	Description of Base Categories/ Layers/ Bands	CHARAC-TER	A50	11	60	As per Section 3. Description of Base Categories, in case of vector topographic data Description of Layers, in case of user domain Layers in vector or raster format Description of Spectral Bands in case of multiple band image data (raster)
	4	Whether Base Category/ Layer/ Band included in the volume	INTEGER*2	I6	61	66	1, if includes Otherwise 0
	5	Serial Number of Media in which the Base Category/ Layer/ Band is contained	INTEGER*2	I6	67	72	Applicable For data on Media only

Table 3 : GENERAL INFORMATION (GENINFO) RECORDS							
Rec No	**Field**	**Contents**	**Type (Fortran)**	**Format**	**Starting Byte**	**End Byte**	**Remarks**
3	1	Agency Acronym	CHARACTER	A10	1	10	Allotted by NSDI (ref NSDI metadata document 1.1.1.1)
	2	Layer Name	CHARACTER	A30	11	40	Allotted by NSDI (ref NSDI metadata document 1.1.1.2)
	3	Layer Type	CHARACTER	A20	41	60	Allotted by NSDI (ref NSDI metadata document 1.1.1.2)
		Filler	-	-	61	61	1 Space
	2	Number of States/ UT = (m)	INTEGER*2	I2	62	63	As per Central Heading. Mostly applicable for SOI topographic data
	3	Source Type	INTEGER*2	I9	64	65	0 – Original 1 – Compiled or Aggregated …*(Add new table)*
		Filler	-	-	66	72	7 Spaces
4	1	Scale	INTEGER*4	I9	1	9	Scale of the map
		Filler	-	-	10	11	2 Space
	2	Edition Number	INTEGER*2	I4	12	16	Edition of the source material
		Filler	-	-	17	18	2 Space
	3	Edition Year	INTEGER*2	I8	19	26	Edition year e.g. 2001
		Filler	-	-	27	72	46 Spaces
5 to 5+m	1	StateName	CHARACTER	A68	1	64	One state per record (ref NSDI metadata document 1.1.1.4)
	2	Number of Districts (n)	INTEGER*2	I4	65	68	No. of Districts in the State
	3	Number of Location points (o)	INTEGER*2	I4	69	72	No. of unique reference locations

Table 3 : GENERAL INFORMATION (GENINFO) RECORDS							
Rec No	**Field**	**Contents**	**Type (Fortran)**	**Format**	**Starting Byte**	**End Byte**	**Remarks**
5+m to 5+m+n	1	DistName	CHARACTER	A72	1	72	Districts will be listed One District per record (ref NSDI metadata document 1.1.1.5)
5+m+n to 5+m+n +o	1	LocPtName	CHARACTER	A72	1	72	One Location name per record (ref NSDI metadata document 1.1.1.6)
6+m+n +o	1	Surveyed Year/ Date of acquisition	CHARACTER	A60	1	60	Year(s) of Survey of the source material e.g. 1968-69 Date of Acquisition in case of Satellite raster image
		Filler	-	-	61	68	8 Spaces
	2	Security classification of dataset	INTEGER*2	I4	69	72	0 Not Classified 1 Classified
7+m+n +o	1	Data Type Code	INTEGER*2	I2	1	2	1 if data is base category wise (Topographgic Vector). 2 if data is Layer Wise (Vector Format) 3 if raster multiband image (band wise) 4 if Single band raster image values 5 if Single band coded raster image
	-	Filler	-	-	3	8	6 Spaces
	2	MapProjection	CHARACTER	A4	9	12	POLY – Polyconic, UTM

Table 3 : GENERAL INFORMATION (GENINFO) RECORDS

Rec No	Field	Contents	Type (Fortran)	Format	Starting Byte	End Byte	Remarks
							LAMB - Lambert conical orthomorphic SOM - Space Oblique Mercater Any Other
		Filler	-	-	13	14	2 Spaces
	3	Units	CHARACTER	A6	15	20	Units of measure for ground planimetric co-ordinates. Its Value is METERS
		Filler	-	-	21	22	2 Spaces
	4	Grid Size (Resolution)	REAL*4	F8.2	23	30	The true ground distance in meters.
	5	Number of Control (Tick) Points (q)	INTEGER*2	I6	31	36	Usually 4 corners of the sheet for Topographic data. Could be more for User Domain data.
		Filler	-	-	37	72	36 Spaces
8+m+n +o	1	CentralLongitude	REAL*4	F12.5	1	12	Projection Parameters
	2	CentralLatitude	REAL*4	F12.5	13	24	Projection Parameters
	3	StandardParallel 1	REAL*4	F12.5	25	36	Projection Parameters
	4	StandardParallel 2	REAL*4	F12.5	37	48	Projection Parameters
	5	ScaleFactor	REAL*4	F6.4	49	54	Projection Parameters
		Filler	-	-	55	56	2 Spaces
	6	Spheroid	CHARACTER	A8	57	64	EVEREST, WGS84, WGS72 etc.
		Filler	-	-	65	72	8 Spaces
9+m+n	1	OriginLatitude	REAL*4	F12.5	1	12	Projection Parameters

Table 3 : GENERAL INFORMATION (GENINFO) RECORDS

Rec No	Field	Contents	Type (Fortran)	Format	Starting Byte	End Byte	Remarks
+o							
	2	OriginLongitude	REAL*4	F12.5	13	24	Projection Parameters
	3	OriginEasting	REAL*4	F12.2	25	36	Projection Parameters
	4	OriginNorthing	REAL*4	F12.2	37	48	Projection Parameters
		Filler	-	-	49	72	24 Spaces
10+m+ n+o to 10+m+ n+o+q	1	Control (Tick) point Label	INTEGER*2	I4	1	4	1 = SW/LL, 2 = SE/LR, 3 = NE/UR, 4 = NW/UL 999 = Reserved for Identifying the origin of Raster Image (For Raster data) 5 to 998 can be used for identifying other control (Tick) points within the sheet/ dataset
	2	Raster Image Orientation	CHARACTER	A2	5	6	Relevant for Control (Tick) Point Label = 999 (i.e. raster data) BL = Indicates raster image origin at Bottom left and rasterization is bottom up (row/ scan lines) and Left-right (Columns/ Pixels) TL =Indicates raster image origin at Top left and rasterization is Top-down (row/ scan lines) and Left-right (Columns/ Pixels)
	3	Latitude	REAL*4	F12.6	7	18	In Degrees of Tick Points
	4	Longitude	REAL*4	F12.6	19	30	In Degrees of Tick Points

Table 3 : GENERAL INFORMATION (GENINFO) RECORDS

Rec No	Field	Contents	Type (Fortran)	Format	Starting Byte	End Byte	Remarks
	-	Filler	-	-	31	36	6 Spaces
	5	X Co-ordinate	REAL*4	F12.6	37	48	Projected in ground terms in meters
	6	Y Co-ordinate	REAL*4	F12.6	49	60	Projected in ground terms in meters
	7	User Tick Label	CHARACTER	A10	61	70	User defined label for Tick point
	-	Filler	-	-	71	72	2 Spaces
11+m+n+o+q	1	O.S. Name	CHARACTER	A18	1	18	Specifies name of source Operating System
	2	GIS Package	CHARACTER	A18	19	36	Specifies name of source GIS Software
	3	DBMS Name	CHARACTER	A18	37	54	Specifies name of source Database
	-	Filler	-	-	55	72	18 Spaces

Table 4 : QUALITY INFORMATION (QUALINFO) RECORDS

Rec No	Field	Contents	Type (Fortran)	Format	Starting Byte	End Byte	Remarks
3	1	Process Citation	CHARACTER	A69	1	69	General statement on data generation processes
	2	No. of Process stages (m)	INTEGER*2	I3	70	72	Number of distinct process stages
4 to 4+2*m	1	Process Description	CHARACTER	A62	1	62	Processes description
		Filler	-	-	63	64	2 Spaces
	2	Process Year	INTEGER*4	I8	65	72	Process/ Sorece Data Year

Table 4 : QUALITY INFORMATION (QUALINFO) RECORDS							
Rec No	**Field**	**Contents**	**Type (Fortran)**	**Format**	**Start-ing Byte**	**End Byte**	**Remarks**
	1	PAccHorX	REAL	F12.2	1	12	Positional accuracy in X direction
	2	PAccHorY	REAL	F12.2	13	24	Positional accuracy in Y direction
	3	PAccVer	REAL	F12.2	25	36	Positional accuracy in Z direction
	4	TAcc	REAL	F12.2	37	48	Thematic accuracy in %
	5	TAccConf	REAL	F12.2	49	60	Confidence level for thematic accuracy
	6	TAccArea	REAL	F12.2	61	72	Thematic Area accuracy in %
5+2*m	1	PAC-CLayer*(horizontal)	REAL4	F12.2	1	12	Cumulative RMS Horizontal Positional accuracy (ref. NSDI metadata document 1.1.1.9)
	2	PACCLayer*(vertical)	REAL4	F12.2	13	24	Cumulative RMS vertical Positional accu-racy (ref. NSDI metadata document 1.1.1.9)
	3	TACCLayer*	REAL4	F12.2	25	36	Final % thematic accuracy (ref. NSDI metadata document 1.1.1.9)
	4	TACCLayer*(area)	REAL4	F12.2	37	48	Final thematic area accuracy
		Filler	-	-	49	72	24 spaces

Table 5 : GENERAL TOPOGRAPHIC INFORMATION (TOPOINFO) RECORDS							
Rec No	**Field**	**Contents**	**Type (Fortran)**	**Format**	**Start-ing Byte**	**End Byte**	**Remarks**
3	1	Whether COMPIND Data Exists	INTEGER*2	I2	1	2	1 for yes, 0 for NO
	2	Whether ADMIND Ex-	INTEGER*2	I2	3	4	1 for yes, 0 for NO

Table 5 : GENERAL TOPOGRAPHIC INFORMATION (TOPOINFO) RECORDS							
Rec No	**Field**	**Contents**	**Type (Fortran)**	**Format**	**Start-ing Byte**	**End Byte**	**Remarks**
		ists					
	3	Number of SPECIAL FOOTNOTES lines (m)	INTEGER*2	I2	5	6	
	4	Magnetic Variation E or W of True North	CHARAC-TER	A2	7	8	E = East of True North W = West of True North
	5	Annual Change in Magnetic Variation with sign	INTEGER*2	I4	9	12	+ for increasing, - for decreasing
		Filler	-	-	13	14	2 Spaces
	6	Year of Magnetic Varia-tion	INTEGER*2	I6	15	20	
	7	Copyright	CHARAC-TER	A24	21	44	Example : Govt of India Copyright
	8	Year of Copyright	INTEGER*2	I6	45	50	Example : 1975
	9	Contour Interval	INTEGER*2	I6	51	56	Contour interval in meters
	8	Mean Grid North Alignment	INTEGER*2	2(I3)	57	62	In degrees and minutes. For un-gridded sheets this field and the next are blank
		Filler	-	-	63	64	2 Spaces
	9	East and West of True North	CHARAC-TER	A1	65	65	E = East of True North W = West of True North
		Filler	-	-	66	67	2 Spaces
	10	Name of Grid	CHARAC-TER	A5	68	72	User Defined

Table 5 : GENERAL TOPOGRAPHIC INFORMATION (TOPOINFO) RECORDS							
Rec No	**Field**	**Contents**	**Type (Fortran)**	**Format**	**Starting Byte**	**End Byte**	**Remarks**
4	1	Authority	CHARACTER	A64	1	64	Name of Authority under whose Direction the data is published/ generated
	2	Admn. Boundary Verification Remark	INTEGER*2	I2	65	66	1 = Adm. Boundaries are Verified 0 = Adm. Boundaries are not verified from appropriate Authority
	-	Filler	-	-	67	68	2 Spaces
	3	Triangulated heights and contours adjusted with spirit-leveled heights	CHARACTER	A1	69	69	Y = Adjusted N = Not Adjusted
	-	Filler	-	-	70	71	2 Spaces
	4	Territorial water upto 12 Nautical miles from the coastline	CHARACTER	A1	72	72	Y in case of data having coastline; N in case of data not having coastline.
5 to 5+m	1	SPECIAL FOOTNOTES RECORDS	CHARACTER	A72	1	72	

Table 6 : DATA CATEGORY (DATACAT) RECORDS							
Rec No	**Field**	**Contents**	**Type (Fortran)**	**Format**	**Starting Byte**	**End Byte**	**Remarks**
3	1	Base Category/ Layer/ Band Serial Number	INTEGER*2	I6	1	6	Order of listing of Base Categories, Layers, Bands (Corresponds to field 1 of records 4 to 3+p ob Table 2)
	2	Number of major	INTEGER*2	I6	7	12	• Number of Base categories in case of Topog-

Table 6 : DATA CATEGORY (DATACAT) RECORDS							
Rec No	**Field**	**Contents**	**Type (Fortran)**	**For-mat**	**Starting Byte**	**End Byte**	**Remarks**
		categories / Layer Categories (m)					raphic data • Number of Layer Categories in case of user domain thematic data in vector or raster form • Not relevant in case of Image data
	3	Number of tables (nt)	INTEGER*2	I6	13	18	• Number of attribute tables attached to the Layer
	-	Filler	-	-	19	72	54 Spaces
4 to 3+m	1	Category Code	INTEGER*2	I2	1	2	Major Category/ Layer Category Code
	-	Filler	-	-	3	8	6 Spaces
	2	Category Name	CHARAC-TER	A64	9	72	Full category name as per Appendix A Major Category/ Layer category Name/ Band-width
4+m	1	Number of Nodes (nn)	INTEGER*4	I8	1	8	Set to Zero for Raster Data
	2	No. of Lines (nl)	INTEGER*4	I8	9	16	Set to Zero for Raster Data
	3	Number of Areas (na)	INTEGER*4	I8	17	24	Set to Zero for Raster Data
	4	No. of Texts (nx)	INTEGER*4	I8	25	32	Set to Zero for Raster Data
	5	No. of Rows (nR)	INTEGER*4	I6	33	38	Set to Zero for Vector data
	6	No. of Columns (nC)	INTEGER*4	I6	39	44	Set to Zero for Vector data
	7	No. of Pixel Bytes (nBytes)	INTEGER*2	I2	45	46	Set to Zero for Vector data, relevant to Image, Grid data 1- Single byte, binary for images

Table 6 : DATA CATEGORY (DATACAT) RECORDS							
Rec No	**Field**	**Contents**	**Type (Fortran)**	**For-mat**	**Starting Byte**	**End Byte**	**Remarks**
							2- Two byte, binary for images/ coded grids 4- Four byte, binary for DEM
	8	Byte Order (depending upon architecture)	CHARAC-TER	A2	47	48	II / MM for more than one Byte data II indicates Intel, Big Endian architecture MM indicates Motorola, Little Endian Architecture
	9	Record Type	CHARAC-TER	A6	49	54	RASTER or VECTOR
	10	Format of Value Records	CHARAC-TER	A10	55	64	Content would be LOGICAL*1 for Single byte, binary images INTEGER*2 for two byte, binary images/ coded grids REAL*4 for four byte, binary DEM data (as per the details in field 7 above)
		filler			65	72	8 spaces

Table 7 : RASTER (GRID) DATA RECORDS							
Rec No	**Field**	**Contents**	**Type (Fortran)**	**Format**	**Starting Byte**	**End Byte**	**Remarks**
Value Re-cords3 to (nR*nC* nBytes)/	1	Raster Values • Reflectance Val-ues for image data • Elevation Values for DEM	LOGICAL*1 or INTE-GER*2 or REAL*4 (As per record 4+m, field 7 in	Binary	1	NR*Nc* nByte	Order of Grid cell locations for Values Row-1 :Col-1, col-2, …col-nC Row-2: ……………………. …….. Row-nr: …………………….

Table 7 : RASTER (GRID) DATA RECORDS							
Rec No	**Field**	**Contents**	**Type (Fortran)**	**Format**	**Starting Byte**	**End Byte**	**Remarks**
72 + 3		• Coded Values for Coded Raster Layer	table 6)				

Table 8 : NODE RECORDS							
Rec No	**Field**	**Contents**	**Type (Fortran)**	**Format**	**Start-ing Byte**	**End Byte**	**Remarks**
3	1	Record Type	CHARACTER	A4	1	4	NODE
	2	Element Internal Id No.	INTEGER*4	I8	5	12	This is Unique value for each Node
	3	Co-ordinates of Node Point	REAL*4	2F12.2	13	36	X,Y co-ordinate in ground meters
	4	Angle	REAL*4	F12.4	37	48	Orientation of point feature with X-axis in degrees (Relevant for Topog-raphic Data)
	-	Filler	-	-	49	72	24 Space

Table 9 : LINE RECORDS							
Rec No	**Field**	**Contents**	**Type (Fortran)**	**Format**	**Starting Byte**	**End Byte**	**Remarks**
3	1	Record Type	CHARACTER	A4	1	4	**LINE**
	2	Element Internal Id	INTEGER*4	I8	5	12	This is Unique value for each Line
	3	Starting Node	INTEGER*4	I8	13	20	Internal ID Number. This refers to data element 2 of the Node records
	4	Ending Node	INTEGER*4	I8	21	28	- do-
	5	Number of points in the line (p)	INTEGER*4	I8	29	36	Number of x-y Co-ordinate pairs (excluding the end points)
	6	Height	REAL*4	F8.2	37	44	Height in Meters
	7	Number of Feature Codes or Levels (q)	INTEGER*2	I4	45	48	
	8	Object ID	INTEGER*4	I8	49	56	
	-	Filler	-	-	57	72	16 Spaces
Co-ordinate records	1 to 2p	x,y co-ordinate string	REAL*4	3(2F12.2)	1	.....	Ending byte number is as per the number of points in the line
Code records	1 to 2q	Major& Minor Code List or	INTEGER*4	6(2I6)	1	....	• Major and Minor Codes for topographic data • Will extend to next line as per the number of feature codes
		Layer Category Code List	INTEGER*4	2(I36)	1	.....	• Layer category Codes for User Domain Thematic data • Will extend to next line as per the number of feature codes

Note: All **POINT** features included in **LINE** type records. A **POINT** feature is distinguished from a **LINE** feature as follows:

- A **POINT** feature has the same start and end node ID (Data elements 3 and 4) and
- The number of points (Data element 5) for a **POINT** feature is zero (0)

The key for linking with the text in the file can also be used for linking with non-graphic data outside the file

Table 10 : AREA RECORDS (TOPOLOGICALLY STRUCTURED)							
Rec No	**Field**	**Contents**	**Type (Fortran)**	**Format**	**Starting Byte**	**End Byte**	**Remarks**
3	1	Record Type	CHARACTER	A4	1	4	**AREA**
	2	Element Internal Id No.	INTEGER*4	I8	5	12	This is Unique value within all areas (polygons)
	3	Co-ordinates of Area Point	REAL*4	2F(12.2)	13	36	The area point (polygon label) is always within the polygon it represents
	5	Number of lines bounding the area (b)	INTEGER*2	I4	37	40	Number of x-y Co-ordinate pairs
	6	Number of Feature Codes (r)	INTEGER*2	I4	41	44	
	7	Object ID	INTEGER*4	I8	45	52	The key for linking with the text in the file, can also be used for linking with non-graphic data outside the file
	-	Filler	-	-	53	72	20 Spaces
Line ID records	1 to b	List of Line ID's	INTEGER*4	9(I8)	1	.....	Internal ID number of lines bounding the area. Ending byte number is as per the number of lines in the area (polygon)
Code	1to 2r	Major& Minor	INTEGER*4	2I6	1	12	• Major and Minor Codes in pairs for to-

Table 10 : AREA RECORDS (TOPOLOGICALLY STRUCTURED)

Rec No	Field	Contents	Type (Fortran)	Format	Starting Byte	End Byte	Remarks
re-cords		Code List					pographic data • Will extend to next line as per the number of feature codes
		Filler	-	-	13	18	6 spaces
		Feature Name	Character	A54	19	72	
		OR					
		Layer category Code List	INTEGER*4	2(I36)	1	.....	• Layer category Codes for User Domain Thematic data • Will extend to next line as per the number of feature codes

Table 11 : AREA RECORDS (TOPOLOGICALLY NON- STRUCTURED)

Rec No	Field	Contents	Type (Fortran)	Format	Starting Byte	End Byte	Remarks
3	1	Record Type	CHARACTER	A4	1	4	**AREA**
	2	Major Code	INTEGER*4	I6	5	10	Major Code of the Feature
	3	Minor Code	INTEGER*4	I6	11	16	Minor Code of the Feature
	6	Feature Type	CHARACTER	A52	17	68	
	5	Number of vertices (p) excluding end points	INTEGER*4	I4	69	72	Number of vertices
Co-ordinate records	1 to 2p	x,y coordinates of Area Points	REAL*4	3(F12.2)	1	……	Ending byte as per the number of vertices in the area feature

Table 12: TEXT RECORDS							
Rec. No	**Field**	**Contents**	**Type (Fortran)**	**Format**	**Starting Byte**	**End Byte**	**Remarks**
3	1	Record Type	CHARACTER	A4	1	4	**TEXT**
	2	Element Internal Id	INTEGER*4	I8	5	12	This is Unique value for all text records
	3	Text Size	INTEGER*2	I4	13	16	In one hundredth of an inch
	4	Text Justification	INTEGER*2	I4	17	20	0 for left, 1 for right, 2 for centre justification
	5	Angle	REAL*4	F12.4	21	32	Angle of Text in Degrees
	6	x,y Co-ordinates of reference Point	REAL*4	2F(12.2)	33	56	
	7	Number(r) of attribute codes	INTEGER*2	I4	57	60	Number of x-y Co-ordinate pairs
	8	Number Text Character strings	INTEGER*2	I4	61	64	
	9	Object ID	INTEGER*4	I8	65	72	The key for linking with the object in the LINE or Area Records
	-	Filler	-	-	53	72	20 Spaces
Second	1	Text Characters	CHARACTER	A72	1	72	This record contains the text data describing Object ID
Third	1-2r	Major& Minor Code List	INTEGER*4	2I6	1	12	• Major and Minor Codes for topographic data • Will extend to next line as per the number of feature codes
		Filler	-	-	13	18	6 spaces
		Feature Name	Character	A54	19	72	
		OR					
		Layer category Code List	INTEGER*4	2(I36)	1	.....	• Layer category Codes for User Domain Thematic data • Will extend to next line as per the number of feature codes

Table 13 : ATTRIBUTE RECORDS

Rec. No	Field	Contents	Type (Fortran)	Format	Start-ing Byte	End Byte	Remarks
3	1	Type	CHARACTER	A4	1	4	Indicates attribute table type (ATTR)
	2	Attribute Table Id.	Integer	I6	5	10	Attribute table Id.
	3	Name of Attribute Table	CHARACTER	A12	1	24	Indicates name of the attribute table as per source specification.
	4	No. of fields (af)	INTEGER	I6	11	16	Indicates number of columns in attribute table
	5	No. of records (ar)	INTEGER	I6	17	22	Indicates number of records in attribute table
	-	Filler	-	-	23	72	50 Spaces
4 to 4+af	1	Field name	CHARACTER	A12	1	12	Represents Field Item name
	2	Field type 1	CHARACTER	A12	13	24	Definition of Field giving Width, Output, Type, number of decimal places separated by comma.
	3	Field type 2	CHARACTER	A8	25	32	Represents ANSI standard format
	4	Key-Field	CHARACTER	I2	33	34	Indicates 1 if the field is Key-indexed, ELSE 0
	-	-	-	-	35	36	2 Spaces
	5	Remarks	CHARACTER	A36	37	72	Description/Remarks of the Field
							•
Data\Records 5+af to	1-af	List of field values	CHARACTER	2(A36)	1	....	• List of field data values • Will extend to next line as per the number of filed values in attribute re-

Table 13 : ATTRIBUTE RECORDS

Rec. No	Field	Contents	Type (Fortran)	Format	Start-ing Byte	End Byte	Remarks
3	1	Type	CHARACTER	A4	1	4	Indicates attribute table type (ATTR)
	2	Attribute Table Id.	Integer	I6	5	10	Attribute table Id.
	3	Name of Attribute Table	CHARACTER	A12	1	24	Indicates name of the attribute table as per source specification.
	4	No. of fields (af)	INTEGER	I6	11	16	Indicates number of columns in attribute table
	5	No. of records (ar)	INTEGER	I6	17	22	Indicates number of records in attribute table
	-	Filler	-	-	23	72	50 Spaces
5+af+ar							cords
Records numbered 3 onwards … will be repeated as many times as the number of attribute tables indicated by number nt in data-cat file.							

Source: Indian NSDI (www.nsdiindia.org)

Appendix D

FME: Xmap file


```
<?xml version="1.0" encoding="UTF-8"?>
<xfMap>
    <feature-map>
        <mapping match="nsde:Line_Info">
            <feature-type>
                <literal expr="Line_Info"/>
            </feature-type>
            <attributes>
                <attribute>
                    <name>
                        <literal expr="DESCRIPTION"/>
                    </name>
                    <value>
                        <extract expr="./nsde:DESCRIPTION"/>
                    </value>
                </attribute>
                <attribute>
                    <name>
                        <literal expr="name"/>
                    </name>
                    <value>
                        <extract expr="./gml:name"/>
                    </value>
                </attribute>
                <attribute>
                    <name>
                        <literal expr="Start_NODE"/>
                    </name>
                    <value>
                        <extract expr="./nsde:Start_NODE"/>
                    </value>
                </attribute>
                <attribute>
                    <name>
                        <literal expr="End_NODE"/>
                    </name>
                    <value>
                        <extract expr="./nsde:End_NODE"/>
                    </value>
                </attribute>
                <attribute>
                    <name>
                        <literal expr="PTS_NUM"/>
                    </name>
                    <value>
                        <extract expr="./nsde:PTS_NUM"/>
                    </value>
                </attribute>
                <attribute>
                    <name>
                        <literal expr="HEIGHT"/>
                    </name>
                    <value>
                        <extract expr="./nsde:HEIGHT"/>
                    </value>
                </attribute>
                <attribute>
                    <name>
                        <literal expr="ELEMENT_ID"/>
                    </name>
                    <value>
                        <extract expr="./nsde:ELEMENT_ID"/>
```

```
                </value>
            </attribute>
            <attribute>
                <name>
                    <literal expr="MAJOR_CODE"/>
                </name>
                <value>
                    <extract expr="./nsde:MAJOR_CODE"/>
                </value>
            </attribute>
            <attribute>
                <name>
                    <literal expr="MINOR_CODE"/>
                </name>
                <value>
                    <extract expr="./nsde:MINOR_CODE"/>
                </value>
            </attribute>
            <attribute>
                <name>
                    <literal expr="OBJECT_ID"/>
                </name>
                <value>
                    <extract expr="./nsde:OBJECT_ID"/>
                </value>
            </attribute>
            <attribute>
                <name>
                    <literal expr="FID"/>
                </name>
                <value>
                    <extract expr="./nsde:FID"/>
                </value>
            </attribute>
            <attribute>
                <name>
                    <literal expr="FID"/>
                </name>
                <value>
                    <extract expr="./nsde:FID"/>
                </value>
            </attribute>
        </attributes>
    </mapping>
    <mapping match="nsde:Metadata">
        <feature-type>
            <literal expr="Metadata"/>
        </feature-type>
    </mapping>
    <mapping match="nsde:NSDE_Info">
        <feature-type>
            <literal expr="NSDE_Info"/>
        </feature-type>
        <attributes>
            <attribute>
                <name>
                    <literal expr="Version"/>
                </name>
                <value>
                    <extract expr="./nsde:Version"/>
                </value>
            </attribute>
            <attribute>
                <name>
                    <literal expr="Layer"/>
                </name>
                <value>
                    <extract expr="./nsde:Layer"/>
                </value>
            </attribute>
            <attribute>
                <name>
                    <literal expr="Sub_Layer"/>
                </name>
                <value>
```

```
                        <extract expr="./nsde:Sub_Layer"/>
                    </value>
                </attribute>
                <attribute>
                    <name>
                        <literal expr="Agency_Code"/>
                    </name>
                    <value>
                        <extract expr="./nsde:Agency_Code"/>
                    </value>
                </attribute>
                <attribute>
                    <name>
                        <literal expr="Agency_Acronym"/>
                    </name>
                    <value>
                        <extract expr="./nsde:Agency_Acronym"/>
                    </value>
                </attribute>
            </attributes>
        </mapping>
    </feature-map>
    <feature-content-map>
        <mapping match="gml:pointProperty/gml:Point">
            <geometry activate="xml-point">
                <data name="coord-separator">
                    <extract expr="./gml:coordinates[@ts]" default="whitespace"/>
                </data>
                <data name="axis-separator">
                    <extract expr="./gml:coordinates[@cs]" default=","/>
                </data>
                <data name="decimal">
                    <extract expr="./gml:coordinates[@decimal]" default="."/>
                </data>
                <data name="data-string">
                    <extract expr="./gml:coordinates"/>
                </data>
            </geometry>
        </mapping>
        <mapping match="gml:LineString">
            <geometry activate="xml-line">
                <data name="coord-separator">
                    <extract expr="./gml:coordinates[@ts]" default="whitespace"/>
                </data>
                <data name="axis-separator">
                    <extract expr="./gml:coordinates[@cs]" default=","/>
                </data>
                <data name="decimal">
                    <extract expr="./gml:coordinates[@decimal]" default="."/>
                </data>
                <data name="data-string">
                    <extract expr="./gml:coordinates"/>
                </data>
            </geometry>
        </mapping>
        <mapping match="gml:LinearRing">
            <geometry activate="xml-area">
                <data name="coord-separator">
                    <extract expr="./gml:coordinates[@ts]" default="whitespace"/>
                </data>
                <data name="axis-separator">
                    <extract expr="./gml:coordinates[@cs]" default=","/>
                </data>
                <data name="decimal">
                    <extract expr="./gml:coordinates[@decimal]" default="."/>
                </data>
                <data name="data-string">
                    <extract expr="./gml:coordinates"/>
                </data>
            </geometry>
        </mapping>
        <mapping match="gml:Polygon">
            <geometry activate="xml-area"/>
        </mapping>
        <mapping match="gml:multiPolygonProperty/gml:MultiPolygon">
```

```
            <attributes>
                <attribute>
                    <name>
                        <literal expr="xml_type"/>
                    </name>
                    <value>
                        <literal expr="xml_area"/>
                    </value>
                </attribute>
            </attributes>
            <geometry activate="xml-aggregate"/>
        </mapping>
        <mapping match="gml:multiLineStringProperty/gml:MultiLineString">
            <attributes>
                <attribute>
                    <name>
                        <literal expr="xml_type"/>
                    </name>
                    <value>
                        <literal expr="xml_line"/>
                    </value>
                </attribute>
            </attributes>
            <geometry activate="xml-aggregate"/>
        </mapping>
    </feature-content-map>
</xfMap>
```

Source: Safesoft FME universal Translator.