Use of XML for Web-Based Query Processing of Geospatial Data

by

Ying Teng

TR00_135 June 2000

Bachelor of Science, Zhejiang University
Master of Science, University of New Brunswick

A Thesis Submitted in Partial Fulfilment of
the Requirements for the Degree of

Master of Computer Science

in the Graduate Academic Unit of Computer Science

Supervisor:	Bradford G. Nickerson, Ph.D.,Computer Science
Examining Board:	Weichang Du, Ph.D., Computer Science, Chairman Alejandro López-Ortiz, Ph.D., Computer Science David J. Coleman, Ph.D., Geodesy and Geomatics Engineering

This thesis is accepted.

Dean of Graduate Studies

THE UNIVERSITY OF NEW BRUNSWICK
June, 2000

Dedication

I would like to dedicate my thesis to my parents and my boyfriend Jason for their love, support, and continuous encouragement throughout the creation of this thesis.

Abstract

New standards for geospatial data representation are emerging. For example, the ISO (the International Organization for Standardization) geospatial metadata draft standard defines a new object-oriented representation schema. Existing collections of geospatial data and metadata need tools to transform them to the new standard. This research investigated how mapping from existing geospatial metadata standards can be formally specified and implemented using XML. In addition, we investigated how large collections of geospatial data can be indexed to permit fast search for data queries combining spatial ranges with keywords and date range.

To test out research ideas, we implemented the translation of Canadian NTDB (National Topographic Database) metadata files into the FGDC (Federal Geographic Data Committee) CSDGM (Content Standard for Digital Geospatial Metadata), which can then be translated into XML files. A tool for transforming FGDC CSDGM XML metadata files to ISO XML metadata files was designed and implemented in two ways: XSLT (eXtensible Style Language Transformations) and a Java program written for this research. A formal grammar for ISO geospatial metadata standard was proposed as a way of generating the XML DTD (Document Type Definition). Search engines for searching ISO XML metadata files on the Web by geospatial coordinates, dates and strings were developed by using a GSDindex (geospatial data index based on R-tree and AVL trees) approach and a relational(Oracle 8) database approach.

Experiments comparing the two search engines on a testbed containing 6979 geospatial metadata files showed that, on average over a set of seven search experiments, the GSDindex approach was 2.5 times faster than the Oracle database approach.

Acknowledgments

I would like to thank my supervisor Dr. Nickerson for giving me this great research opportunity, and for his knowledgeable guidance and great support. I would also like to acknowledge the support of the GEOIDE Network of Centres of Excellence, project DEC#2 Designing the technological foundations of geospatial decision-making with the World Wide Web. In addition, I would like to thank Fuqun Zhou of the Canada Center for Remote Sensing, Department of Natural Resources Canada for supplying us with the CEONet metadata for testing purposes.

Table of Contents

Dedication ii

Abstract iii

Acknowledgments iv

Table of Contents v

List of Tables ix

List of Figures x

1. Introduction 1

1.1 Problem Definition 1

1.2 Research Questions 2

1.3 Metadata 2

1.4 XML (eXtensible Markup Language) 3

1.5 Thesis Objectives 8

2. Geospatial Metadata Standards 10

2.1 FGDC (Federal Geographic Data Committee) CSDGM (Content Standard for Digital Geospatial Metadata) 10

2.2 ISO (the International Organization for Standardization) Metadata Standard ISO 19115 15

2.3 FGDC/ISO Metadata Standard Harmonization 17

2.4 Formal Grammar Representation Proposal for ISO 19115 24

3. XML for Geospatial Metadata

3.1 XML Tags for FGDC CSDGM and ISO 19115 . . . 34

3.2 Implementations . . . 34

3.2.1 Translation from Canadian NTDB (National Topographic Database) Metadata to FGDC Metadata . . . 34

3.2.2 Tools for Translating FGDC Metadata to XML Representations . . 37

3.2.3 Translation from FGDC XML Metadata Files to ISO XML Metadata Files . . . 37

4. Search Engines for Geospatial XML Metadata . . . 44

4.1 Search Types Implemented . . . 44

4.2 R-Tree and AVL Tree Index Construction . . . 45

4.3 R-Tree and AVL Tree Search Algorithms . . . 48

4.4 Oracle 8 Database Implementation . . . 50

5. Multitier Client/Server Architecture Implementation . . . 60

5.1 Introduction to Multitier Client/Server Architecture . . . 60

5.2 Multitier Client/Server Architecture Implementation . . . 61

5.3 Client User Interface . . . 63

5.4 Middle Tier Design . . . 67

5.4.1. Introduction to Java Servlet . . . 67

5.4.1.1 Java Servlet Lifecycle . . . 68

5.4.2. Java Servlet Implementation . . . 69

5.4.2.1 Applet-Servlet Communication . . . 71

5.4.2.2. Servlet-Database Communication 73

5.4.3. Java Web Server 74

5.5 Oracle 8 Database Server 75

5.6 Source Code Summary 75

6. Test Results 76

6.1 Test Data 76

6.2 Test Results of R-Tree, AVL Tree Index and Oracle 8 Database 76

7. Conclusions and Future Work 85

7.1 Related Work 85

7.2 Conclusions 85

7.2 Future Work 87

8. References 88

Appendix I: Sample NTDB Metadata 92

Appendix II: Sample NTDB FGDC Metadata 96

Appendix III: Sample NTDB FGDC XML Metadata 98

Appendix IV: Comparison of FGDC and ISO elements 101

Appendix V: Suggested formal grammar for B.2 Metadata entity set information . . . 127

Appendix VI: Suggested formal grammar for B.3 Identification information 128

Appendix VII: XML DTD for ISO metadata files 129

Appendix VIII: Selected implementation details 137

Appendix IX: XSL stylesheet 139

Appendix X : Different longitude domain casting 145

Appendix XI: Sample CEONet FGDC XML Metadata 147

Appendix XII: Sample CEONet ISO XML Metadata 151

VITA

List of Tables

Table 2.1 Comparison of FGDC and ISO metadata sections and entities 20

Table 2.2 Meaning of EBNF symbols (adapted from [OMG,1999]). 25

Table 2.3 Initial part of the table B.2 from [ISO/TC 211, 1999]. 27

Table 4.1 Oracle FileIdName Table 51

Table 4.2 Oracle Keywords Table 51

Table 4.3 Oracle Dates Table 52

Table 4.4 Oracle Coordinates Table 52

Table 5.1 Summary of all source code written for this thesis 75

Table 6.1 Time to build GSDindex data structures 77

Table 6.2 Time to load the GSDindex in servlet 77

Table 6.3 Comparison of search time for bounding box queries 78

Table 6.4 Comparison of search time for keyword queries 79

Table 6.5 Comparison of search time for date queries 79

Table 6.6 Comparison of search time for bounding box and keyword queries 80

Table 6.7 Comparison of search time for bounding box and date queries 81

Table 6.8 Comparison of search time for keyword and date queries 82

Table 6.9 Comparison of search times for bounding box, keyword and date queries . . . 83

Table 6.10 Summary of search time ratios 83

List of Figures

Figure 1.1 An example XML document describing the weather 5

Figure 2.1 FGDC metadata tools 12

Figure 2.2 An edit screen shot of tkme from [FGDC, 1999] 14

Figure 2.3 Process to translate NTDB metadata to FGDC XML 15

Figure 2.4 Metadata entity set information from [ISO/TC 211,1999] 18

Figure 2.5 MD_Identification from [ISO/TC 211, 1999] 19

Figure 2.6 Initial part of the formal grammar for table B.2 from [Nickerson and Teng, 2000] 28

Figure 2.7 Part of the formal grammar for table B.3 from [ISO/TC 211, 1999] 31

Figure 3.1 Translating FGDC XML metadata to ISO XML metadata 39

Figure 3.2 Architecture diagrams of the XML translator program 41

Figure 3.3 Using XSLT to transform two XML file structures 43

Figure 4.1 Constructing the GSDindex and Oracle search engines 45

Figure 4.2 A UML diagram of the Java class structure for R-tree construction and search 46

Figure 4.3 R-tree search algorithm 48

Figure 4.4. The algorithm for GSDindex intersection 49

Figure 4.5 A UML diagram of the Java class structure for GSDindex range search . . . 50

Figure 4.6 Relational schema of the Oracle database 52

Figure 4.7 SQL commands used to query the Oracle database 53

Figure 4.8 The connection function of JDBC Java Servlet 55

Figure 4.9 The query function of JDBC Java Servlet with example (a) of Figure 4.7. . . 56

Figure 4.10 A UML diagram of the JDBC Java servlet program 58

Figure 5.1 3-tier client/server architecture 61

Figure 5.2 Implementations of 3-tier client/server architecture for metadata query . . . 62

Figure 5.3 User interface of the 3-tier client/server architecture 64

Figure 5.4 A UML diagram of the client applet program 66

Figure 5.4 Architecture for testing combined geospatial data queries 70

Chapter 1 Introduction

1.1 Problem Definition

As the amount of geospatial data (especially digital geospatial data) continues to grow exponentially and becomes even more complex, users of all levels of experience and needs are faced with the increasingly frustrating burden of large undifferentiated geospatial data, ambiguous displays, and less-than-satisfactory search & retrieval aids [Gluck 1995]. Users are retrieving more data, but systems provide less-than-desirable support in differentiating relevant from nonrelevant geospatial objects.

Metadata is the foundation for success of web-based query processing of geospatial data. Metadata refers to data about data. Geospatial metadata refers to metadata regarding geographic objects [Gluck 1995]. There are now several standards for geospatial metadata including AACR2/USMARC [e.g., OCLC 1988, 1995], the U.S. Federal Geographic Data Committee [FGDC 1998 and OGC, 1999], and the Government Information Locator Service (GILS) [Christian 1996; OMB 1995]. Generally, each standard was devised by experts, released for public comment, revised, and implemented. Identifying, understanding, and transforming disparate geospatial data into an integrated data resource so that they can be readily shared and reused within an organization and across organizational boundaries is mandatory for developing a unified geospatial metadata standard that meets international requirements. The draft ISO geospatial metadata standards [ISO/TC 211, 1998 and 1999] were written to address these needs. Little research on the systematic, scientific transformation of different geospatial metadata standard files to the draft ISO geospatial metadata standard files has been conducted. As far as we can determine, no fully documented examples of search engines designed for ease of administration, scalability, and displaying relevant results are implemented to search ISO geospatial metadata files on the Web so as to aid discovery and browsing of geospatial data.

1.2 Research Questions

The following research questions guided the research:

(1) Are existing geospatial metadata standards adequate to describe geospatial data? Does the ISO geospatial metadata standard meet geospatial data user requirements?

(2) Is XML an effective methodology for geospatial data query processing in a web-based environment?

(3) Is geospatial data search using a relational database more efficient than a direct spatial data index for handling web-based geospatial data queries on very large volumes of geospatial data?

1.3 Metadata

The concept of metadata is familiar to most people. For example, a map legend is pure metadata. The legend contains information about the publisher of the map, the publication date, the type of map, a description of the map, spatial references, the map's scale and a symbol legend, among many other things. Geospatial metadata are simply that type of descriptive information applied to a digital geospatial file. Metadata includes traditional descriptive and subject cataloging of information objects; metadata may also include extensive descriptions of the context, quality, and accuracy of data. They are a common set of terms and definitions to use when documenting geospatial data. Some digital geospatial files now have some associated metadata [FGDC, 1999], particularly those from the U.S. and Canadian Federal governments.

Metadata helps people who use geospatial data to find the data they need and to determine how best to use it. Metadata benefits the data producing organization as well. As personnel change in an organization, undocumented data may lose their value. Later workers may have little understanding of the contents and uses for a digital database and may find they can't trust results generated from these data. Lack of knowledge about other organizations' data can lead to duplication of effort. It may seem burdensome to add the cost of generating metadata to the cost of data collection, but, in our opinion, it is worth the extra effort in the long run.

1.4 XML (eXtensible Markup Language)

1.4.1 What is XML?

XML is an ISO compliant subset of SGML (Standard Generalized Markup Language) and is designed to make it easy and straightforward to use SGML on the Web [W3C, 1998].

XML is a human-readable, machine-understandable, general syntax for describing hierarchical data, applicable to a wide range of applications (e.g. databases, e-commerce, Java, web development and searching). XML is a database-neutral and device-neutral format; it is likely that XML will play a major role in connecting heterogeneous data resources [Sall, 1998]. With XML content markup, queries are more likely to retrieve relevant files due to contextual information. Search engines could retrieve a specific portion of file; they also could be much faster if the added context eliminates numerous irrelevant matches. This facilitates more precise declarations of content and more meaningful search results across multiple platforms. In addition, XML will enable a new generation of Web-based data viewing and manipulation applications [MSDN,1999 and Norman,1999].

1.4.2 Advantages of XML

1.4.2.1 Extensible

In XML you can define an unlimited set of tags [MSDN, 1999]. While HTML tags can be used to display a word in bold or italic, XML provides a methodology for describing structured data. As XML tags are adopted throughout an organization's intranet, and by others across the Internet, there will be a corresponding ability to search for and manipulate data regardless of the applications within which it is found. Once data has been located, it can be delivered over the Internet and presented in a WWW browser in any number of ways, or it can be handed off to other applications for further processing and viewing.

1.4.2.2 Structural Representation of Data

XML provides a structural representation of data that has proved broadly implementable and easy to deploy. Industrial implementations in the SGML community and elsewhere demonstrate the intrinsic quality and industrial strength of XML's tree-structured data format [MSDN, 1999]. XML documents are easy to create. If you are familiar with HTML, you can quickly learn to author in XML. Figure 1.1 illustrates an XML document used to describe a weather report.

```
<weather-report>
         <date>March 25, 1998</date>
         <time>08:00</time>
         <area>
            <city>Seattle</city>
            <state>WA</state>
            <region>West Coast</region>
            <country>USA</country>
         </area>
         <measurements>
            <skies>partly cloudy</skies>
            <temperature units=”Celsius”>46</temperature>
            <wind>
               <direction>SW</direction>
               <windspeed units=”km/h”>6</windspeed>
            </wind>
            <h-index>51</h-index>
            <humidity>87</humidity>
            <visibility>10</visibility>
            <uv-index>1</uv-index>
         </measurements>
</weather-report>
```

Figure 1.1 An example XML document describing the weather from [MSDN, 1999].

Rather than describing the order and fashion in which the data should be displayed, the tags indicate what each item of data means (whether it is a <date> element, an <area> element, and so forth). Any receiver of this data can then decode the document, using it for their own purposes. For example, an individual might use it to make plans for the day, while a weather researcher might use it as data in a historical record of Seattle.

XML is defined by the World Wide Web Consortium (W3C), ensuring that structured data will be uniform and independent of applications or vendors [W3C, 1998]. This resulting interoperability is kick-starting a new generation of business and electronic-commerce Web applications [MSDN, 1999].

Once the data is on the client desktop, it can be manipulated, edited, and presented in multiple views, without return trips to the server. Servers now become more scalable, due to lower computational and bandwidth loads. Also, since data is exchanged in the XML format, it can be merged from different sources.

1.4.2.3 Data Separated from the Presentation Process

XML maintains the separation of the user interface from the structured data. Hypertext Markup Language (HTML) specifies how to display data in a browser, and XML defines the content. In XML, you can use stylesheets such as Extensible Style Language (XSL) and Cascading Style Sheets (CSS) to present the data in a browser. XML separates the data from the presentation process, enabling you to display and process the data as you wish by applying different style sheets and applications.

This separation of data from presentation enables the seamless integration of data from diverse sources [MSDN, 1999]. Customer Information, purchase orders, research results, bill payments, medical records, catalogue data, and other information can be converted to XML on the middle tier of a three-tier architecture. Data encoded in XML can then be delivered over the Web to the desktop. Legacy information stored in mainframe databases or documents can also be adapted to XML format for web-based access.

1.4.3 Applications of XML

1.4.3.1 XML as data

XML is gaining popularity as a data storage and exchange format as well as a document markup language [Boumphrey, etc, 1998]. Extra tags are required in an XML file besides the content, so it may not be suitable for large sets of records in a spatial data warehouse. Developers may prefer to use a traditional database, and convert its content into XML on the fly or use XML to store data extents.

1.4.3.2 Describing domain documents

XML is valuable on the Internet because it provides a universal standard mechanism for describing a “domain document” [Ken, 1998]. Different XML domain-specific vocabularies or XML dialects (e.g., mathematical, chemical, medical and geospatial metadata markup) can be invented for any purpose. These domain documents can be understood by all browsers with XML parsers so that data with specific meaning for this community can be freely exchanged and shared within the community of interest. This results in more efficient data exchange and more accurate searching on the web.

1.4.3.3 Combining different sources

XML enables structured data from different sources to be combined in an efficient manner [Boumphrey, etc, 1998]. Since XML is database-neutral, it is likely that it will play a major role in connecting heterogeneous databases [Ken, 1998]. With XML content markup, queries are more likely to retrieve relevant files due to contextual information.

Search engines could retrieve a specific portion of file; they also could be much faster if the added context eliminates numerous irrelevant matches.

1.4.3.4 Non-human interaction

XML files describe their contents, so it is designed for non-human user agents, such as the programs sent out by search engines, to process the information in the file. In the specific case of search engines, it will mean that they can provide more accurate results to queries [Boumphrey, etc, 1998].

1.5 Thesis Objectives

The overall objective of this thesis is to design and implement a client/server application which allows a user to query geospatial metadata files over the Internet.

The major objectives of this thesis are

(1) investigating the use of XML to markup geospatial metadata;

(2) exploring the differences between existing metadata standards: FGDC CSDGM and the ISO Metadata Standard ISO 19115;

(3) developing tools for transforming XML metadata files in the FGDC CSDGM standard to XML metadata files in the ISO 19115 standard;

(4) proposing a formal grammar for ISO 19115;

(5) building search engines for searching XML metadata files by geospatial coordinates, dates and strings on the Web.

(6) comparing the performance of a web-based spatial indexing data structure to an

existing relational database (e.g. Oracle) indexing the same data.

The remainder of the thesis is organized as follows. Chapter 2 introduces two existing geospatial metadata standards: FGDC CSDGM and ISO 19115. The proposal of a formal grammar for ISO 19115 was also presented. Chapter 3 describes how to use XML to represent geospatial metadata in these two standards and the metadata file transformations. Translation from Canadian NTDB metadata to FGDC metadata was implemented using a straightforward parsing and translation process. Transformation from FGDC XML metadata files to ISO XML metadata files was implemented using both XSLT and a Java program called XML translator developed in this research. Chapter 4 shows the building of the search engines for querying ISO standard geospatial XML metadata files. Search engines for searching ISO standard XML metadata files by geospatial coordinates, dates and strings were developed by using a GSDindex (geospatial data index based on R-tree and AVL trees) approach and a relational database (Oracle 8) approach. Chapter 5 illustrates the multi-tier client/server architecture being implemented in this research: Java Applet and Java Servlet communication was used to provide the client/server connection, the JDBC (Java Database Connectivity) API was used to access an Oracle 8 database, and a Java Applet was used to provide the user interface. Chapter 6 discusses the testing and testing results. Chapter 7 gives conclusions and proposes future work in this research area.

Chapter 2 Geospatial Metadata Standards

A geospatial metadata standard is simply a common set of terms and definitions that describe geospatial metadata.

2.1 FGDC (Federal Geographic Data Committee) CSDGM (Content Standard for Digital Geospatial Metadata)

2.1.1 General information

The Federal Geographic Data Committee (FGDC) recently adopted Content Standard for Digital Geospatial Metadata (CSDGM) as a metadata standard [FGDC, 1998]. This standard specifies the information content of metadata for a set of digital geospatial data and provides a consistent approach and format for the description of data characteristics. The standard provides a way for data users to know what data are available, whether the data meet their specific needs, where to find the data and how to access the data.

2.1.2 Purpose and goals

The objectives of the standard are to provide a common set of terminology and definitions for the documentation of digital geospatial data. The standard establishes the names of data elements and compound elements (groups of data elements) to be used for these purposes, the definitions of these compound elements and data elements, and information about the values that are to be provided for the data elements [FGDC, 1998].

The standard was developed from the perspective of defining the information required by a prospective user to determine the availability of a set of geospatial data, to determine the fitness of the set of geospatial data for an intended use, to determine the means of accessing the set of geospatial data, and to successfully transfer the set of geospatial data. The standard does not specify the means by which this information is organized in a computer system or in a data transfer, nor the means by which this information is transmitted, communicated, or presented to the user.

The standard is designed to describe all possible geospatial data. There are 334 different elements in the FGDC standard, 119 of which exist only to contain other elements. These compound elements are important because they describe the relationships among other elements. CSDGM uses both SGML and XML for structuring information.

Major sections of the CSDGM are: Identification Information, Data Quality Information, Spatial Data Organization Information, Spatial Reference Information, Entity and Attribute Information, Distribution Information, and Metadata Reference Information.

Minor sections of the CSDGM include: citation information, time period information, and contact information.

2.1.3 Tools

Figure 2.1 shows the architecture of several tools available on the Web page of metadata tools [FGDC, 1999]. Most of the tools are written in standard C.

2.1.3.1 cns

`cns` (chew and spit) is a pre-parser for formal metadata designed to assist metadata managers in converting records that cannot be parsed by `mp` into records that can be parsed by `mp`. It takes as input a poorly-formatted metadata file and, optionally, a list of element aliases, and outputs: (1) a metadata file that can be read by both `mp` and `xtme` or `tkme`; and (2) a file listing all of the lines that it couldn't figure out where to put.

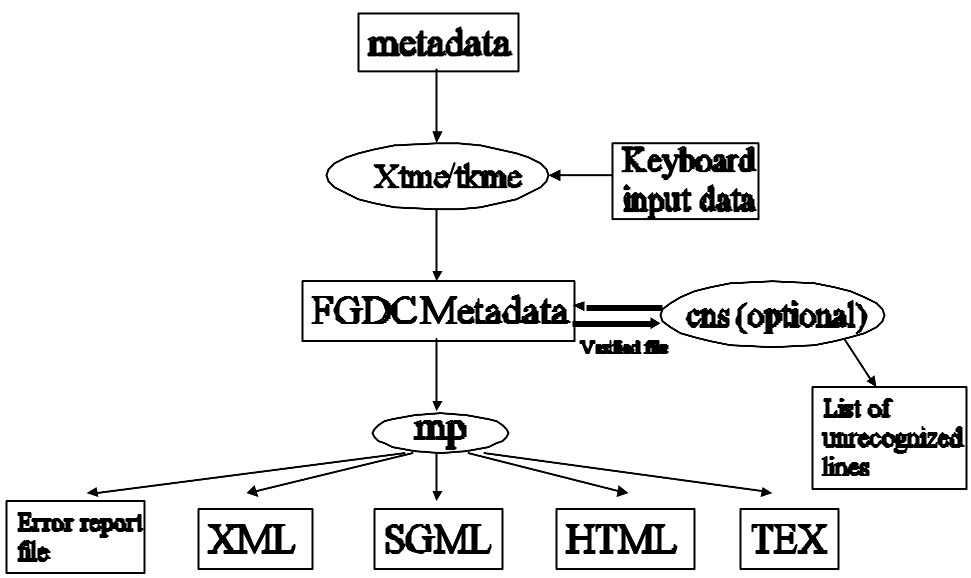


Figure 2.1 FGDC metadata tools.

2.1.3.2 mp

`mp` is designed to parse metadata encoded as indented text, check the syntactical structure against the the FGDC Content Standard for Digital Geospatial Metadata, and reexpress the metadata in several useful formats (HTML, SGML, TEXT and XML). The output is suitable for viewing with a web browser or text editor. It runs on UNIX systems and on PCs running Windows 95, 98, or NT. `mp` generates a textual report indicating errors in the metadata, primarily in the structure but also in the values of some of the scalar elements (e.g. values for latitude and longitude must be expressed as decimal fractions of degrees; values for day and month of year, and for years, must follow the calender date convention as YYYYMMDD).

2.1.3.3 xtme/tkme

`xtme/tkme` is an editor for formal metadata, that is, structured documentation conforming to the Content Standard for Digital Geospatial Metadata developed by the Federal Geographic Data Committee (FGDC). This editor is written using Standard C. `xtme` runs on UNIX systems with the X Window System, release 11, version 5 or later. While `tkme` is a port to the PC platform. The editor is intended to simplify the process of creating metadata that conform to the standard. Its output format is the input format for mp.

Figure 2.2 is an edit screen shot of tkme [FGDC, 1999].

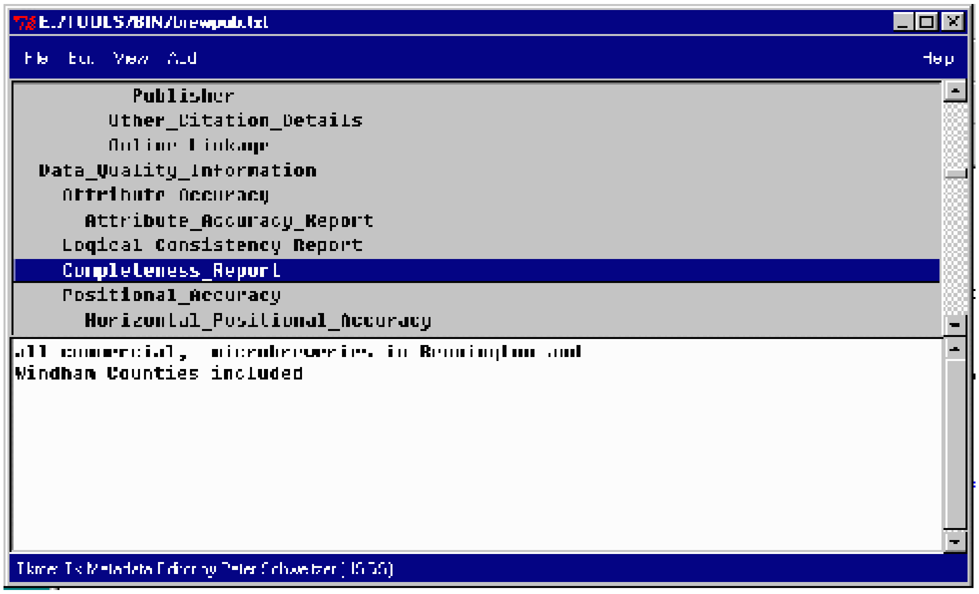


Figure2.2. An edit screen shot of tkme from FGDC [1999].

2.1.4 A small example

The FGDC metadata file (063c03.txt, see Appendix II) was obtained by using software developed for this research. It converts the demonstration data set [NTDB,1999] (063c03.met in this example, see Appendix I) metadata in NTDB (National Topographic Data Base) format to FGDC format. It is written in Borland C++ (total of 630 lines of source, including comments), using a straightforward parsing and translation process. The FGDC metadata file was converted to xml file (063c03.xml, see Appendix III) using metadata tool `mp` [Phillips, 1999]. The procedure is illustrated in Figure 2.3.

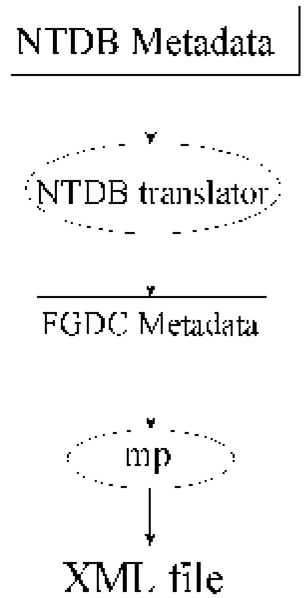


Figure 2.3 Process to translate NTDB metadata to FGDC XML.

2.2. ISO (The International Organization for Standardization) Metadata Standard

2.2.1 General information

International standard ISO 19115 was prepared by Technical Committee ISO/TC 211, Geographic information/Geomatics [ISO/TC 211,1999].This standard identifies the metadata required to describe digital geographic data. Metadata is applicable to independent datasets, aggregations of datasets, individual geographic features, and the various classes of objects which compose a feature. Metadata is mandatory (required) for each geographic dataset and may, optionally, be provided for aggregations of datasets, features, and parts of features.

This International Standard defines [ISO/TC 211,1999]:

- mandatory and conditional metadata packages, metadata entities, and metadata elements — the minimum set required to serve the full range of metadata applications (data discovery, determining data fitness for use, data access, data transfer, and use of digital data).
- optional metadata elements — to allow for a more extensive standard description of geographic data, if required.
- a method for extending metadata to fit specialised needs.

Though this International Standard is applicable to digital data, its principles can be extended to many other forms of geographic data such as maps, charts, and textual documents.

2.2.2 Purpose and goals

The objective of the standard is to provide a structure for describing digital geographic datasets and a procedure to extend the existing structure so that users will be able to locate, select, purchase, and access geographic data, determine whether the data in a holding will be of use to them, and use it in the most efficient way. By establishing a common set of metadata terminology, definitions, and extension procedures, this standard will promote the proper use and effective retrieval of geographic data. Supplementary benefits of this standard for metadata are to facilitate the organization and management of geographic data and to provide information about an organization's dataset to others. This standard for the documentation of data furnishes data producers the appropriate information for them to characterize geographic data, and it makes possible dataset cataloguing enabling data discovery, retrieval and reuse.

The metadata are specified using UML in a hierarchy to establish relationships and an organization for the information [ISO/TC 211,1999]. The metadata is categorized into the following main sections: Identification Information, Maintenance Information, Metadata Extension Information, Reference System Information, Spatial Representation, DataQuality, Data Constraint Information, Distribution Information, Feature catalogue Information, Portrayal catalogue Information and Application Schema Information. The sections are subdivided into entities, which are further divided into metadata elements that carry the individual units of metadata. The supporting repeatable entities provide common information called for by main sections, and are never used alone [ISO/TC 211,1998].

Figure 2.4 defines the class “Metadata” and shows containment relationships with other metadata classes which, in aggregate, define geographic metadata.

Figure 2.5 defines the metadata classes required to identify a dataset. It also defines a specialization subclass for identifying images.

2.3. FGDC/ISO Metadata Standard Harmonization

2.3.1 Comparison of the two standards

ISO and FGDC metadata standards have some differences.

The ISO standard addresses known deficiencies in the FGDC CSDGM (e.g.

Raster and Imagery data types are both available in ISO standard but only raster is available in the FGDC standard). For example, the FGDC CSDGM does not provide a compound element that contains the coordinates of the four corners of satellite images.

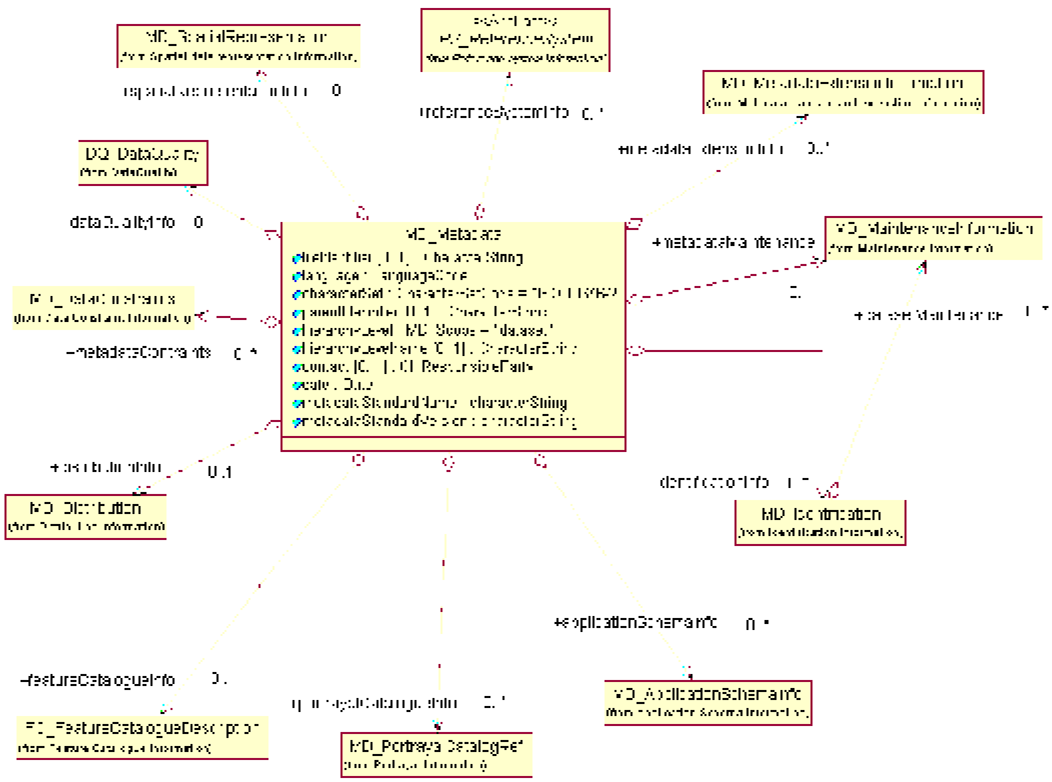


Figure 2.4. Metadata entity set information from ISO/TC 211 [1999].

ISO has 2 levels of compliance: essential and comprehensive. Essential metadata is required to uniquely identify a dataset, and includes 7 elements from MD_Metadata class and 12 from the MD_Identification class. Essential metadata is used for the purposes of cataloguing datasets and dataset series and to support data clearinghouse activities facilitating data discovery. Comprehensive metadata provides the metadata required

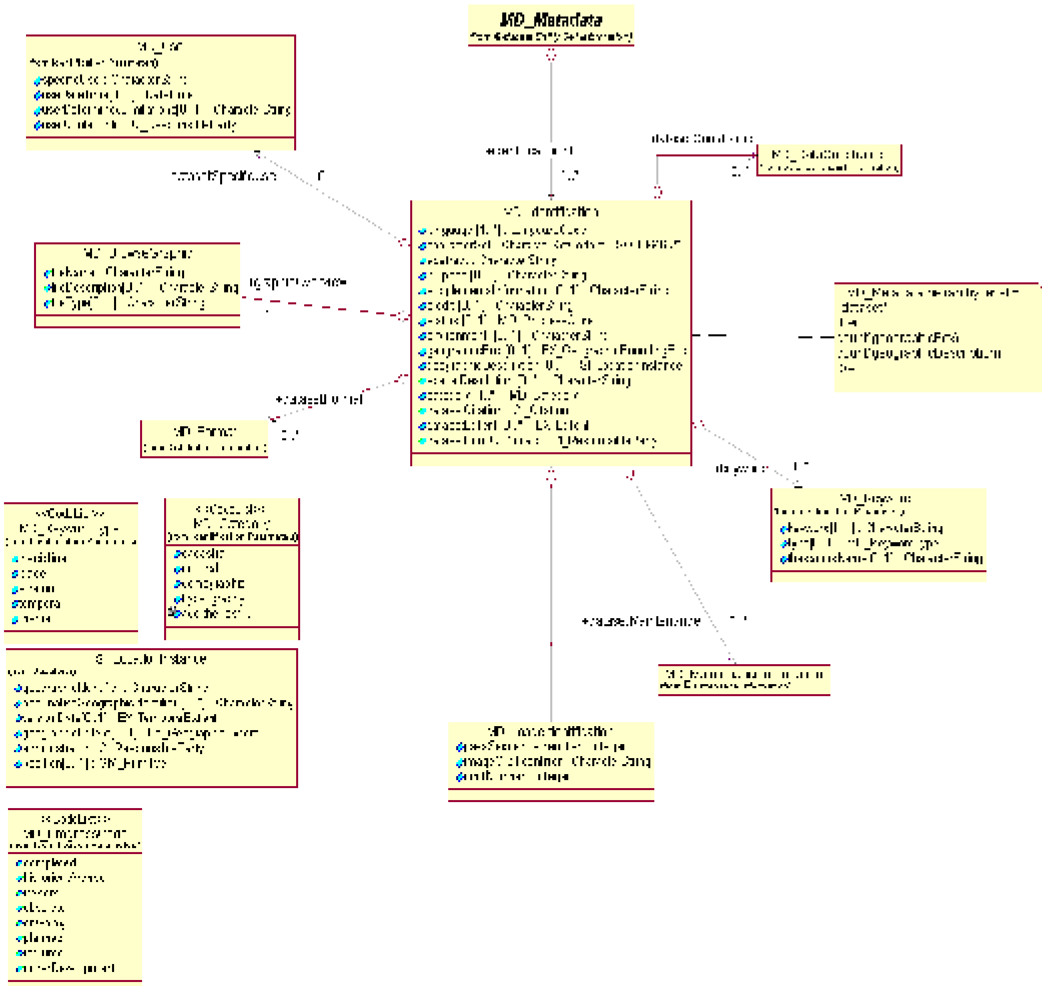


Figure 2.5. MD_Identification from ISO/TC 211 [1999].

to document a dataset completely (independent dataset, dataset series, or individual geographic features and attributes). This level of conformance fully defines the complete range of metadata required to identify, evaluate, extract, employ, and manage geographic information. A data producer would typically provide metadata at this level.

FGDC CSDGM and the ISO standard also have terminology and element name differences. For example, the bounding box of a metadata file in the FGDC CSDGM standard has four elements: West Bounding Coordinate, East Bounding Coordinate, North Bounding Coordinate and South Bounding Coordinate and the four elements have the following short names : westbc, eastbc, northbc and southbc. In contrast, the bounding box of a metadata file in the ISO 19115 standard has four elements: westBoundLongitude, eastBoundLongitude, northBoundLatitude and southBoundLatitude. These four elements have the short names as follows: westBL, eastBL, northBL and southBL. The only difference, then, is the element short name. The semantics are identical. Appendix IV gives a complete description of the ISO metadata elements and their corresponding FGDC equivalents (if any). Table 2.1 shows the structural comparison of these two geospatial metadata standards.

There are a total of 509 elements in the ISO metadata definition, and 334 elements in the FGDC metadata. All FGDC elements have an equivalent element in the ISO standard.

Table 2.1 Comparison of FGDC and ISO metadata primary sections.

FGDC section name	**ISO section name**	**Description**	**FGDC obligation**	**ISO conformance level**
Metadata	Metadata	Aggregate of the sections below	M	E
Identificationinformation	Identification information	Information to identify the data	M	E
Data quality	Data quality	Assessment of the quality of the data	O	C

FGDC section name	ISO section name	Description	FGDC obligation	ISO conformance level
Spatial data organization information	Spatial representation information	Information about the mechanisms used to represent spatial data	O	C
Spatial reference information	Reference system information	Spatial and temporal reference used in a dataset	O	C
Entity and Attribute information	Feature catalogue	Information identifying the feature catalogue used	O	C
Distribution information	Distribution information	Information about the distributor of, and options of obtaining a dataset	O	C
Metadata reference information	Metadata extension information	Information on the currentness of the metadata information and the responsible party	M	C
	Maintenance information	Information about the scope and frequency of updating data		C
	Data constraint information	Information concerning the restrictions placed on data.		C
	Portrayal catalogue information	Information identifying the portrayal catalogue used		C
	Application schema information	Information about the application schema used to build a dataset		C

M (Mandatory) and E (Essential) have the same meaning; O (Optional) and C(Comprehensive) also have the same meaning.

2.3.2 Activities in Progress

Since 1996, the International Organization for Standardization (ISO) Technical Committee 211, project item 19115 (formerly 15046-15), has been drafting an International Metadata Standard. The FGDC, through the United States ANSI L1 committee, has been an active participant in this project and has had a significant impact on the working drafts of the ISO Metadata Standard. Much of the draft ISO Metadata Standard is closely aligned with the CSDGM [FGDC,1998]. In June 1998, the status of the ISO Metadata Standard progressed from that of a working draft to a committee draft [ISO/TC 211,1998]. With the status changed to that of a committee draft, the proposed ISO Metadata Standard was reviewed in the United States by over 230 individuals; with over 500 comments submitted by the reviewers. The ISO Metadata Standard has been modified based upon the US and other national comments and has now been reissued as the 2nd Committee Draft [ISO/TC 211,1999]. This document is now ready to be reviewed again by the national bodies.

Numerous organizations, including many in Canada, plan to use the ISO Metadata Standard once it has been approved by the ISO Standards Committee. The FGDC is committed to harmonize the CSDGM with the ISO Metadata Standard and has numerous activities ongoing to assure that the harmonization is successful [FGDC, 1998]. To protect the significant existing metadata investment, it is important to assure that the proposed ISO Metadata Standard allows the maximum compatibility with existing FGDC compliant metadata records.

2.3.3 Discussion

The ISO standard is necessary to meet “international” requirements. This International Standard defines the schema required for describing geographic information and services. It provides information about the identification, the extent, the quality, the spatial and temporal schema, spatial reference, and distribution of digital geographic data. This International Standard is applicable to the cataloguing of datasets, clearinghouse activities, and the full description of datasets. This International Standard is also applicable to geographic datasets, dataset series, and individual geographic features and feature properties.

Since XML is a meta-markup language that provides a format for describing structured data, it is a suitable technique for the implementation of the ISO metadata standard.

Tools that are useful for the ISO metadata standard would be similar to those for the FGDC metadata; e.g. editors that can create the ISO metadata file from keyboard input or from FGDC metadata files, and software that can create XML, text, HTML or SGML files from the ISO metadata files [Teng and Nickerson, 2000]. These tools make it much easier for users to create machine-readable XML files for their ISO geospatial metadata. In addition, user acceptance of the new ISO standard will be much greater if tools exist to automatically transform existing metadata files to XML files conforming to the ISO standard.

2.4 Formal Grammar Representation Proposal for ISO 19115

This section summarizes the work given in [Nickerson and Teng, 2000].

2.4.1. Introduction

A context free grammar (CFG) is a formal system that describes a language L by specifying how any legal sentence can be derived. Noam Chomsky [1956] first investigated the application of a formal mathematical model to the structure of natural languages. Computer scientists immediately recognized the value of Chomsky's work for formally representing computer languages [e.g. Aho, 1986, Appel, 1998 and Sebasta, 1996]. A language L is a subset of the closure set of an alphabet. Valid sentences of a language are generated using a grammar $G = (V_N, V_T, S, M)$, where V_N = the set of nonterminal symbols, V_T = the set of terminal symbols, S = the starting nonterminal symbol, and M = the set of productions.

Productions define rules on how proper sentences in the language can be derived. For example, the following grammar defines rules for the language of expressions in a large number of computer languages:

<E> ::= <T> | <E> {“+” | “-”} <T>

<T> ::= <F> | <T> {“*” | “/”} <F>

<F> ::= <I> | “-”<I>

<I> ::= “(“ <E> “)” | <id> (2.1)

The <id> nonterminal stands for any legal identifier. Note that <id> can be considered a terminal in practice as it is returned directly as a token from the lexical analyzer. The notation used here is the Extended Backus Naur Form (EBNF), and is summarized in Table 2.2.

Table 2.2. Meaning of EBNF symbols (adapted from [OMG,1999]).

Symbol	Meaning
::=	is defined to be
|	Alternatively
<text>	Nonterminal
“text”	Literal (terminal)
*	The preceding syntactic unit can be repeated zero or more times
+	The preceding syntactic unit can be repeated one or more times
{}	The enclosed syntactic units are grouped as a single syntactic unit
[]	The enclosed syntactic unit is optional (may occur zero or one time)

For a context free grammar, M contains only productions of the form " ::= $, where " is a single element from V_N and $ has one or more elements from {V_N, V_T}. The way in which context free grammars are used is powerful. For example, the way in which the above grammar (2.1) is stated ensures that, when the parse tree is constructed, multiplication and division have a higher priority than addition and subtraction, that unary minus has a higher priority than multiplication and division, and that expressions in parentheses are always evaluated first.

Due to the fact that the grammar above is unambiguous, there is only one possible way to construct the parse tree for the sentence <id> + <id> * <id>. Context free grammars are the formal basis of almost all modern day computer languages and computer representation schemes, including Java [Gosling, 1996] and C++ [ISO/IEC,1998 and Stroustrup, 1997].

2.4.2 A context-free grammar for geographic information metadata

It is important for any computer representation scheme such as the ISO geographic information metadata standard CD 19115 [ISO/TC 211, 1999] to use a formal grammar. In fact, a formal grammar is the only way to clearly define the language of geographic information metadata. This allows the development of tools that can accurately parse the metadata information and that can supply good error messages back to the user defining the metadata files. Currently, the ISO standard [ISO/TC 211, 1999] uses UML diagrams (see e.g. [Teng, and Nickerson, 2000] as well as [ISO/TC 211, 1999]) to define how the metadata objects relate to one another.

As a start at defining a formal grammar for the ISO geographic information metadata, two tables (B.2 Metadata entity set information and B.3 Identification information) from [ISO/TC 211, 1999] have been translated into a formal context free grammar. Table 2.3 shows the initial part of the Table B.2).

Table 2.3 Initial part of the Table B.2 from [ISO/TC 211, 1999].

	Name / Role name	Short Name	Definition	Obligation / Condition	Maximum occurrence	Data type	Domain
1	MD_Metadata	Metadata	Information about the metadata	M	1	Class	Lines 2-22
2	fileIdentifier	fileID	Unique identifier for this metadata file	O	1	CharacterString	Free text
3	language	lang	Language used for documenting metadata	C / Not defined by encoding?	1	Class	LanguageCode(ISO 639)
4	characterSet	charSet	Full name of the ISO character coding standard used for the metadata set	C / ISO 10646-2 not used?	1	Class	CharacterSetCode(ISO 10646-2 ISO 8859)
5	parentIdentifier	parID	Unique identifier of the parent metadata file	O	1	CharacterString	Free text
6	hierarchyLevel	hierLev	Scope to which the metadata applies (see informative Annex J for more information about metadata heirarchy levels)	C / Scope is not "dataset"?	1	Class	MD_Scope <<CodeList>>
7	hierarchyLevelName	hierLevName	Name of the hierarchy levels	C / Scope is not equal to "dataset"?	1	CharacterString	Free text
8	contact	contact	Party responsible for the metadata information	O	1	Class	CI_ResponsibleParty <<DataType>>
9	date	date	Date that the metadata were created or last updated	O	1	Date	ISO 19108
10	metadataStandardName	mdStanName	Name of the metadata standard used	O	1	CharacterString	Free text
11	metadataStandardVersion	mdStanVer	Version of the metadata standard used	O	1	CharacterString	Free text
12	*Role name:* spatialRepresentationInfo	spatRepInfo	Digital mechanism used to represent spatial information in the dataset	O	N	Association	MD_SpatialRepresentation

Figure 2.6 below shows the initial part of the grammar for Table B.2 from [ISO/TC 211, 1999]. The suggested complete grammar for Table B.2 and B.3 from [ISO/TC 211, 1999] is given in Appendix V and VI.

```
(B.2)  <MD_Metadata> ::= “<Metadata>”
           [ “<fileID>” <string_literal> “</fileID>” ]
           [ “<lang>” <LanguageCode> “</lang>” ]
           [ “<charSet>” <CharacterSetCode> “</charSet>” ]
           [ “<parID>” <string_literal> “</parID>” ]
           [ {“<hierLev>” “dataset” <MD_Scope> “</hierLev>”} |
                {“<hierLev>” “notdataset”
                     “<hierLevName>” <string_literal>
                      “</hierLevName>”
                “</hierLev>”} ]
           [ “<contact>” <CI_ResponsibleParty> “</contact>” ]
           [ “<date>” <Date> “</date>” ]
           [ “<mdStanName>” <string_literal> “</mdStanName>” ]
           [ “<mdStanVer>” <string_literal> “</mdStanVer>” ]
```

```
{ “<spatRepInfo>” <MD_SpatialRepresentation>
 “</spatRepInfo>” }*
{ “<refSysInfo>”<RS_ReferenceSystem>“</refSysInfo>” }*
{ “<metExtensInf>” <MD_MetadataExtensionInformation>
“</metExtensInf>” }*
{ “<idInfo>” <MD_Identification> “</idInfo>” }+
{ “<featColl>” <FT_FeatureCollection> “<featColl>” }+
{ “<featCatInfo>” <FC_FeatureCatalogueDescription>
“</featCatInfo>” }*
[ “<distInfo>” <MD_Distribution> “</distInfo>” ]
```

Figure 2.6. Initial part of the formal grammar for Table B.2 from Nickerson and Teng [2000].

In summary, the systematic mapping from ISO metadata standard to the context free grammar (CFG) was conducted according to the following rules:

(1) Every table in Annex B of the ISO CD 19115 [ISO/TC 211, 1999] is mapped to one production in the CFG.

(2) Every entry of the table is mapped to one term of the corresponding production and the table number is mapped to the matching production number.

(3) Employ XML tag representation that uses the "Short Name" attribute of the tables for the beginning and ending tag definition.

(3) Entries with “O” in the “Obligation/Condition” column and “1" in the “Maximum occurrence” column are represented as terms within square brackets [].

(4) Entries with “O” in the “Obligation/Condition” column and “N" in the “Maximum occurrence” column are represented as terms within curly braces followed by an * (i.e. {}*).

(5) Entries with “M” in the “Obligation/Condition” column and “N" in the “Maximum occurrence” column are represented as terms within curly braces followed by a $^+$ (i.e. $\{\}^+$). (6) Entries with “C” in the “Obligation/Condition” column are treated as those with “O” in the “Obligation/Condition” column.

(7) Entries with “Free text” in the “Domain” column are mapped to the terms with <string_literal> between beginning and ending tags. Similar terminal entries (i.e. integer, real, boolean etc) are also mapped to token classes.

(8) Entries with a class, association or set in the “Data type” column are mapped to a nonterminal with the name given in the “Domain” column between beginning and ending tags. (e.g. 12 spatialRepresentationInfo has “MD_SpatialRepresentation” in the “Domain” column. This entry will be mapped to a term with <MD_SpatialRepresentation> between its beginning and ending tags as shown in Figure 2.6).

Conditional items are problematic. For example, item 3 "language" from Table B.2 of [ISO/TC 211, 1999] is "Conditional (C) / not defined by encoding?". The meaning of this "obligation" is unclear, as is the meaning of the following item 4 "characterSet" "C/ISO 10646-2 not used?". For the time being, we have simply indicated that these items are optional (0 or 1 time). To be useful, the definition of these conditional items should be replaced as either Optional (O) or Mandatory (M). For example, we have changed the syntax slightly for entities 6 and 7 in Table B.2 from [ISO/TC 211, 1999]. Instead of having the obligation "C/ Scope is not equal to "dataset"?", we have the keyword "dataset" indicating that a <MD_Scope>

nonterminal production follows. The keyword "notdataset" indicates that a “<hierLevName>” <string_literal> “</hierLevName>” entity follows that gives the name of the hierarchy level. This is one example that illustrates how a context free grammar can precisely and clearly define how the metadata must be constructed.

Another instance where we have used the context free grammar to modify the syntax of [ISO/TC 211, 1999] to make it precise is shown below in Figure 2.7.

```
(B.3)  <MD_Identification>  ::= “<ID>”
           { “<lang>” <LanguageCode> “</lang>” }+
           [ “<charSet>” <CharacterCodeSet> “</charSet>” ]
           “<abstract>” <string_literal> “</abstract>”
           [ “<purpose>” <string_literal> “</purpose>” ]
           [ “<suppInfo>” <string_literal> “</suppInfo>” ]
           .
           . /* stuff snipped */
           .
           { “<descKey>” <MD_Keywords> “</descKey>” }*
           { “<dsSpecUse>” <MD_Use> “</dsSpecUse>” }*
           [ “<dsConst>” <MD_DataConstraints> “</dsConst>” ]
           [ “<ImageID>” <MD_ImageIdentification>
                “<passSeqID>” <integer_literal> “</passSeqID>”
```

```
        "<imagOrbID>" <string_Literal> "</imagOrbID>"
        "<orbNum>" <integer_literal> "</orbNum>"
        "</ImageID>" ]
    "</ID>"
```

Figure 2.7. Part of the formal grammar for Table B.3 from [ISO/TC 211, 1999].

Row 51 of Table B.3 from [ISO/TC 211, 1999] contains the "C/ Image series exists?" obligation. Our approach is to precisely define this in the grammar; i.e. if the "<ImageID>" … "</ImageID>" tags exist (they are optional as indicating by the enclosing [ ]), then the pass sequence number, image orbital identifier and orbit number are all required to be given. There is no confusion about how the conditional obligation applies.

2.4.3. Discussion

We have made a first attempt at defining a formal grammar for geographic information metadata. This formal grammar uses XML tags to precisely define the limits or each metadata entity. We deliberately chose the XML syntax to simplify understanding the often long metadata text descriptions. Beginning and ending tags for each distinct metadata entity are extremely helpful for syntactic and semantic analysis of computer languages. A formal context free grammar representation is required to precisely define how the geographic information metadata sentences are correctly constructed.

The systematic mapping from ISO metadata standard to the context free grammar (CFG) described above are valid because the grammar describes the correct hierarchical syntactic and semantic structure of the ISO metadata standard. The mapping is complete as all possible ISO data types have a mapping. Some open questions and problems remain. For example, a formal grammar defines production rules in a specific order. This means that for the example grammar given in Appendices V and VI, the metadata tags must be given in the same order as they are specified in the grammar. Conditional entities in the metadata data dictionary are problematic. As mentioned in Appendix III of [Nickerson and Teng, 2000], it is important for these tags to be changed to either optional (O) or mandatory (M), and moved to their appropriate place within the other parts of the data dictionary definitions. This will enable accurate and meaningful error messages to the end user from toolkits used to parse metadata files.

We have found an inconsistency; in Table B.2 from [ISO/TC 211, 1999], feature Collection (item 16) is listed as mandatory, but in the UML model A.2 of Metadata schemas (Annex A of [ISO/TC 211, 1999]) it is not displayed. In addition, it would be very helpful if XML type tags were mandatory (and not optional as explained on p.27 of [ISO/TC 211, 1999]) to make parsing the metadata easier and more consistent with metadata representation in other domains (e.g. medical and manufacturing electronic data interchange). A formal presentation of comments for the ISO/TC 211 Secretariat is given in Appendix IV of [Nickerson and Teng, 2000].

A good start has been made in formally defining geographic information metadata. It is important that work continue to formally define the standard [ISO/TC 211, 1999] to benefit from the modern approaches to representing a language (e.g. that used in [OMG,1999]).

The context free grammar is useful in the development of DTD (Document Type Definition), a formal definition of the structure of a document, of ISO XML metadata files because the ISO standard does not define the order of the data elements. The XML DTD, on the other hand, restricts the XML tags are in the same order as defined in the XML DTD. The formal grammar for the ISO metadata standard provides an unambiguous mapping for the ISO XML DTD.

Chapter 3 XML for Geospatial Metadata

3.1 XML Tags for FGDC CSDGM and ISO 19115

Although HTML describes how to display the data in a Web page, we use XML to describe the data itself. In other words, XML is most often used as a data description language, allowing us to organize data into data structures---even complex data structures, if we so choose. You can tailor the data as you want it; the most attractive feature of XML is that you can create your own tags. This lets you structure the data in an XML document as you like [Holzner, 1998]. XML tags for both FGDC CSDGM and ISO 19115 XML metadata come from the short names of the metadata elements provided in the standards.

3.2 Implementations

3.2.1 Translation from Canadian NTDB (National Topographic Database) Metadata to FGDC Metadata

My first investigation explored the availability of metadata for the NTDB (National Topographic Data Base). Metadata for the NTDB exists, but it uses its own metadata definition which is different from the others. FGDC metadata files for NTDB metadata were obtained using software we developed called "NTDB translator". NTDB translator converts the metadata [NTDB,1999] to FGDC format. The NTDB translator is written in Borland C++, using a straightforward parsing and translation process. An example of NTDB metadata before and after translation is shown in Appendix I and II respectively.

The translation rules used in the NTDB translator are:

(1) *Semantic Mappings*

Foremost in the transformation is the determination of the semantic mapping of elements between the source and target metadata standards [St. Pierre and LaPlant, 1998]. The task involves specifying a mapping of each element in the source metadata standard with a semantically equivalent element in the target metadata standard. The prerequisite to a meaningful mapping requires a clear and precise definition of the elements in each standard. The NTDB metadata standard is defined in [NTDB, 1998] and the FGDC CSDGM standard is specified in [FGDC, 1998].

(2) *Content Conversion*

Metadata standards typically restrict the contents of each metadata element to a particular data type, range of values, or controlled vocabulary. For example, conversions are required between NTDB dates (e.g. 23-SEP-98) and FGDC dates (e.g. 19980923).

The following cases can occur during the transformation that may affect the accuracy of the resulting target metadata files.

(a) *Many to One*. There is only one semantically equivalent element in the target metadata standard for more than one element in the source metadata standard. If the values of all source elements are allowed to be appended to a single value in the target element, there is no problem. Alternatively, if only one source element value is allowed to be mapped to the target, there is a consequence of information loss. For example, NTDB polygon details are mapped to one FGDC element called Entity_Type_Label.

(b) *Extra Elements in Source*. Another important case is the handling of a source element that does not map to any appropriate element in the target standard. For example, the NTDB “PCT_OF_LAND” element (percent of land) was left out of the FGDC target standard.

(c)*Unresolved Mandatory Elements in Target*. In some cases, there may be mandatory elements in the target that have no corresponding mapping in the source metadata standard. Because the target requires a value for the mandatory elements, the translation must provide its value from the available resources of the source metadata files. This value may not be as accurate as they should be. For example, in FGDC, the Metadata Reference Information is mandatory, but was not provided in NTDB. I used “FGDC Content Standards for Digital Geospatial Metadata” as the content of Metadata_Standard_Name element.

Other than the cases discussed above, the “one to one” semantic mapping of the data items of both source and target metadata files are correct because

(1) It maps the data elements from the NTDB source metadata file to the FGDC target metadata file based on the semantics of the data elements as defined in both metadata standards .

(2) It implements proper data type and domain changes for the data elements during the translation.

(3) The result metadata files meet the requirements of the FGDC CSDGM.

No one-to-many mappings were encountered.

3.2.2 Tools for Translating FGDC Metadata to XML Representations

The FGDC metadata files were converted to FGDC XML files using metadata tool `mp` [Phillips, 1999]. The procedure is illustrated in Figure 2.3. The result of translating the NTDB metadata in Appendix II is shown in Appendix III.

3.2.3 Translation from FGDC XML Metadata Files to ISO XML Metadata Files

The ISO standard is necessary to meet "international" requirements. Numerous organizations plan to use the ISO metadata standard once it has been approved by the ISO Standards Committee. This research used both XSLT (eXtensible Style Language Transformations) and a Java program called XML translator to translate FGDC XML metadata files to ISO XML metadata files because of the significant already existing FGDC metadata investment.

An XML document may optionally contain a document type definition (DTD) that specifies which elements are and are not allowed in an XML document. The DTD specifies the exact context and structure of those elements that are allowed. A validating parser can read a document and compare it to its DTD, and report any mistakes it finds. This allows document authors to check that their work meets any necessary criteria. DTDs are essential for large document management projects. One thorny issue when managing large volumes of documents (up to several thousands of pages) is to enforce coherence. Due to the large amount of geospatial metadata (6979 FGDC metadata files of about 100MB), an XML DTD was generated according to the formal grammar described in section 2.4. This XML DTD is listed in Appendix VII.

3.2.3.1 Transformation Rules

The transformation from FGDC XML metadata files to ISO XML metadata files are carried out by applying the following rules to map from the FGDC XML DTD [Nebert, 2000] to the ISO XML DTD (see Appendix VII):

(1) *Semantic Mappings*, as described in section 3.2.1, except that the source standard is FGDC XML DTD and the target standard is ISO XML DTD. Appendix IV shows the complete mapping.

(2) *Content Conversion*, as described in section 3.2.1.

(3) *Hierarchy*

Hierarchy of the metadata is the organization of metadata elements relative to each other, e.g. hierarchical parent-child relationships. FGDC and ISO XML DTDs have a different hierarchy, but there is a semantic "one to one" element mapping for all elements. The hierarchical structure of the source metadata file does not affect that of the target metadata file.

(4) *Properties*

The FGDC metadata standard indicates a lower and upper bound on the number of times an element may occur. The ISO metadata standard indicates whether or not an element is mandatory or optional. According to the grammar defined in Chapter 2 for the ISO metadata standard, a lower bound of zero indicates an optional element, whereas an inclusive lower bound of one indicates that the element must occur at least once and thus is mandatory.

The discussion of the correctness of the NTDB translator also applies to the XML translator, except that the result metadata files strictly meet the requirements of the ISO XML DTD.

3.2.3.2 Transform Implementation

3.2.3.2.1 XML Translator

Microsoft XML parser MSXML [Microsoft, 2000] was used to parse XML files. The building of the Java program was conducted according to the structure of an ISO metadata file illustrated by the Universal Modelling Language (UML) diagram (see Figure 2.4 for MD_Metadata) from the ISO TC/211 Geographic Information/Geomatics Metadata draft standard [ISO/TC 211, 1999]. Figure 3.1 illustrates the transformation from CEONet FGDC metadata XML files to ISO 19115 metadata XML files.

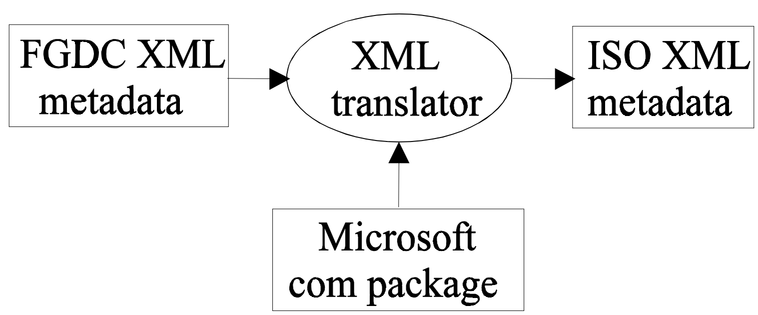
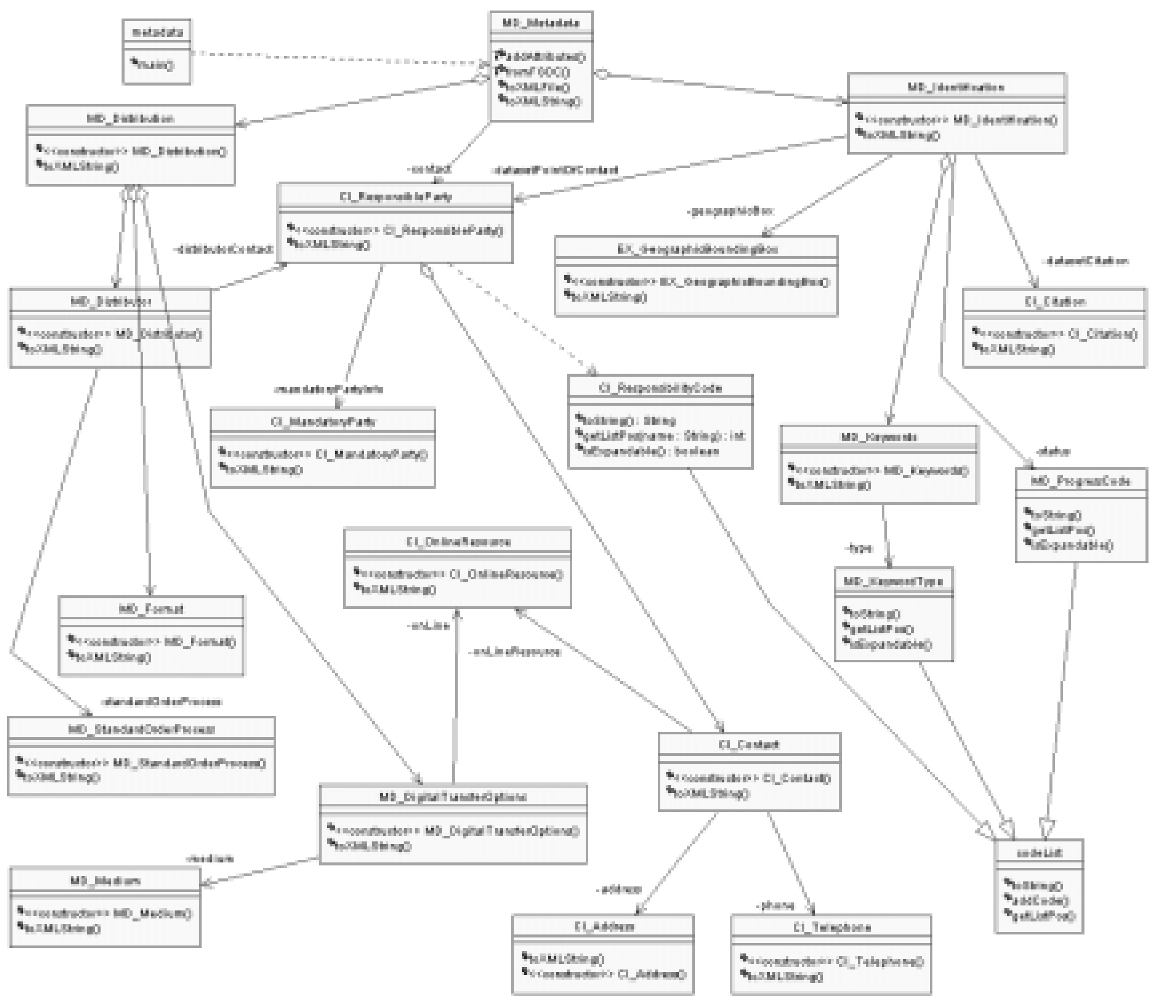


Figure 3.1. Translating FGDC XML metadata to ISO XML metadata.

There is a total of 2450 lines of Java source code in the XML translator program. An architecture diagram explaining how the different classes interact is shown in Figure 3.2. Figure 3.2 (a) displays the relation between the two packages XML translator and com.ms.xml.om. Every class of the XML translator package has to import the com.ms.xml.om package to enable the parsing of XML files. Figure 3.2 (b) shows a set of the classes drawn from the implementation of the XML translator package. The figure focuses on the classes involved in the identification and distribution of the matadata files. The XML translator is an object-oriented program and the hierarchy of the program is constructed according to that of the ISO XML DTD. For example, in ISO XML DTD, a metadata tag has *idInfo* and *distInfo* etc as its direct sub-tags. So, in the XML translator program, class MD_Metadata has class MD_Identification and class MD_Distribution as its aggregate classes. Every class except class codeList and its subclasses has a constructor and a toXMLString function. The constructor of each class parses the equivalent part of the class in the hierarchy of the XML file and return an object of this class. The toXMLString function takes an object of this class and return the part of the text in the result XML file that represents this class. Each subclass of the abstract class codeList is a flexible enumeration for expressing a long list of potential values, such as MD_Progresscode list (e.g. completed, obsolete, planned, in Work etc).

(a) Overview of the package relation.



(b) A UML diagram of part of the XML translator program.

Figure 3.2 Architecture diagrams of the XML translator program.

3.2.3.2.2 XSLT(eXtensible Style Language Transformations) implementation

The Extensible Style Language (XSL) includes both a transformation language and a formatting language [Harold, 1999]. Each of these, naturally enough, is an XML application. The transformation language provides elements that define rules for how one XML document is transformed into another XML document. The transformed XML document may use the markup and DTD of the original document or it may use a completely different set of tags.

In an XSL transformation, an XSL processor reads both an XML document and an XSL style sheet. Based on the instructions the processor finds in the XSL style sheet, it outputs a new XML document or fragment thereof. There's also special support for outputting HTML. With some effort it can also be made to output essentially arbitrary text, although it is designed primarily for XML-to-XML transformations.

To implement the transformation of XML metadata files from the FGDC standard to the ISO standard, the author used LotusXSL (version 0.19.1), an XSL processor - a software component that implements the XSL standard. This XSL processor is available at [IBM, 2000]. Like most XML tools, LotusXSL is written in Java. The download and install details are listed in Appendix VIII. The XSL style sheet used in the transformation was written according to the translation rules discussed in section 3.2.3.1 and the XSL style sheet is listed in Appendix IX. Figure 3.3 illustrates the process.

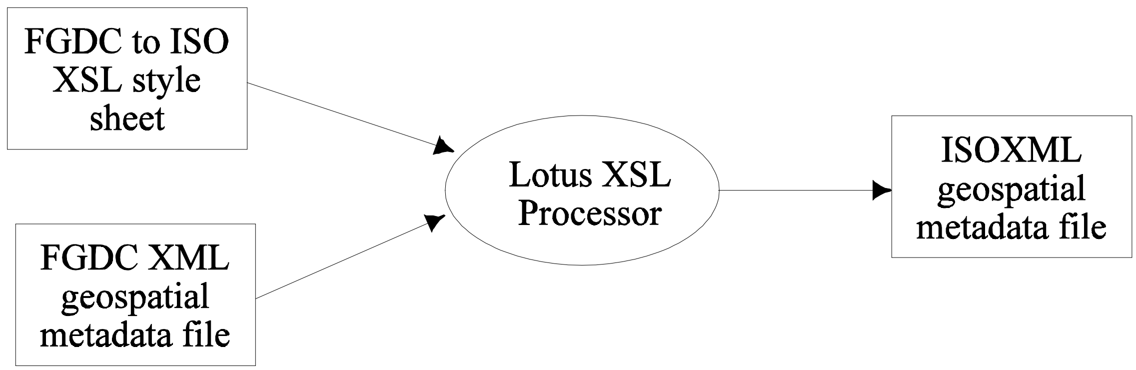


Figure 3.3. Using XSLT to transform two XML file structures.

Chapter 4 Search Engines for Geospatial XML Metadata

4.1 Search Types Implemented

There are three search types that have been implemented in this research: geospatial coordinates, strings and data production dates.

"Geospatial coordinates" represents the geographic areal domain of the dataset: western-most and eastern-most longitude of the limit of the dataset extent expressed in decimal degrees; southern-most and northern-most latitude of the limit of the dataset extent expressed in decimal degrees. Both the FGDC CSDGM and the ISO 19115 have [-180.0, 180.0] as the domain for longitude, but the user interface uses [0.0, 360.0] as the domain for longitude. The correct casting between these two different domain representations is discussed in Appendix X. "Strings" are the theme and place keywords which are the common-use words or phases used to describe the theme and place of the dataset. "Date" is the date when the dataset was produced.

Two different types of search engines were developed for web-based querying of XML metadata files by geospatial coordinate bounding boxes, strings and dates. The first approach is called the "GSDindex" method as it uses classical data structures (the R-tree [Guttman, 1984] and AVL tree [e.g. Weiss, 1996]) and search techniques to answer combined geospatial data queries. The second approach was to use the Oracle 8i relational database to build a standard relational database index that can subsequently be queried using SQL statements issued through the JDBC interface. Figure 4.1 shows the procedure used to build these two search engines. The remaining sections of this chapter describes the implementation details of these two search engines.

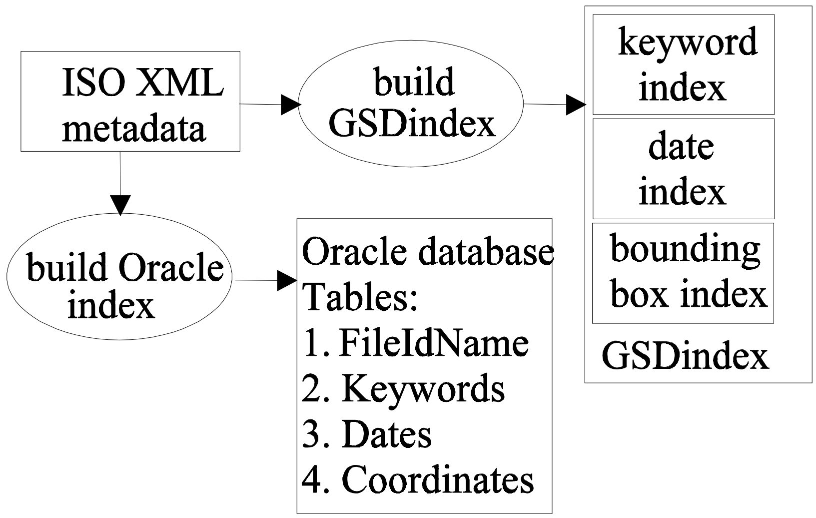


Figure 4.1. Constructing the GSDindex and Oracle search engines.

4.2 R-Tree and AVL Tree Index Construction

An R-tree is a height-balanced tree similar to a B-tree with leaf nodes containing pointers to data objects [Guttman, 1984]. Let M be the maximum number of entries that will fit in one node and let m = $\left\lceil \frac{M}{2} \right\rceil$ be a parameter specifying the minimum number of entries in a node. An R-tree satisfies the following properties:

(1) Every leaf node contains between m and M entries unless it is the root.

(2) Each entry in a leaf node has the smallest rectangle that spatially contains the n-dimensional data object represented by this entry.

(3) Every non-leaf node contains between m and M children unless it is the root.

(4) Each entry in a non-leaf node has the smallest rectangle that spatially contains the rectangles in the child node.

(5) The root node has at least two children unless it is a leaf.

(6) All leaves appear on the same level.

The source code for R-tree index construction consists of 650 lines of Java code. Figure 4.2 shows the architecture of the Java class structure for R-tree construction and search.

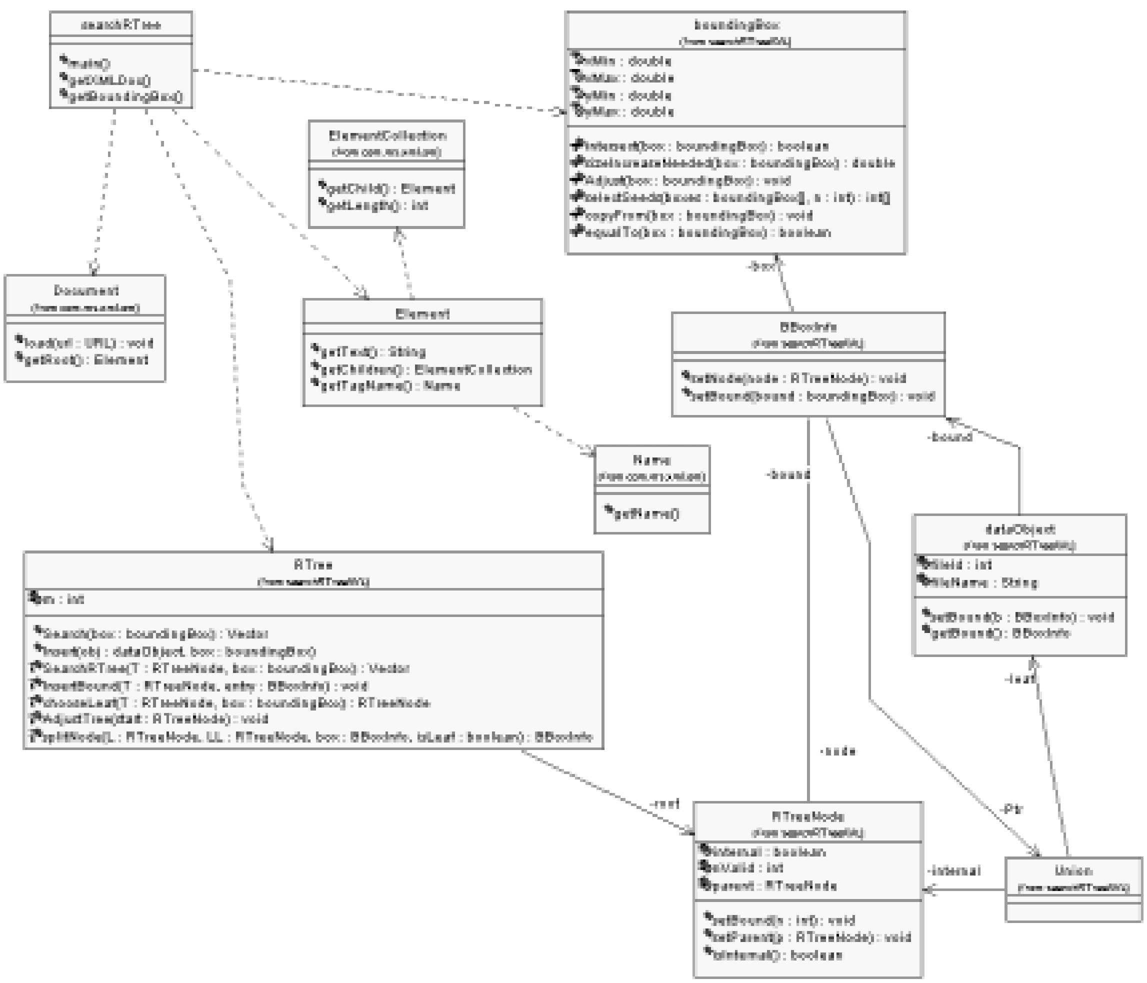


Figure 4.2 A UML diagram of the Java class structure for R-tree construction and search.

The `searchRTree` class imports `com.ms.xml.om` package and uses its `Document`, `Element`, `Name` and `ElementCollection` classes to parse XML files and extract the bounding box information in the XML files. The `RTree` class handles the Rtree's insert and search. The `RTreeNode` class represents the Rtree nodes in the Rtree. Class `BBoxinfo` represents each entry in an Rtree node. Class `boundingBox` and class `dataObject` represent the bounding box and the data object of the Rtree nodes. Class `Union` handles the different references to internal and leaf nodes.

An AVL tree is a binary search tree with the additional balance property that, for any node in the tree, the height of the left and right subtrees can differ by at most 1 [Weiss, 1996]. The AVL balance condition implies that the tree has only logarithmic depth. A consequence of this is that all searching operations in an AVL tree have logarithmic worst-case bounds.

Two AVL trees were built: one for the keywords and one for the dates of all the ISO standard XML metadata files obtained from the XML translator program illustrated in Figure 3.1. The construction of the AVL tree for the keywords needed a little bit more work than the building of the AVL tree for the dates. Since the keywords of the XML metadata files have long strings like "EARTH SCIENCE> LAND SURFACE> LAND USE/LAND COVER> LAND COVER", the long strings were broken down by using string tokenizer ">", and inserting the shorter strings (e.g. "EARTH SCIENCE") into the AVL tree.

4.3 R-Tree and AVL Tree Search Algorithms

The search algorithm of an R-tree descends the tree from the root in a manner similar to that of a B-tree. More than one subtree under a node visited may need to be searched, leading to a time of O(N) to find all rectangles (out of N) intersecting a query rectangle .

The pseudo code of the search algorithm used in this research is shown in Figure 4.3.

```
private Vector SearchRTree(RTreeNode T, boundingBox S){
   int i;
   Vector searchResult = new Vector();
   // check internal nodes
   if (T.isInternal()) {
       for (i = 0; i < T.Entry; i++)
         if (T.Data[i].Bound.Intersect(S))
            SearchRTree (T.Data[i].internal, S);
   }
   // else check leaf nodes
   else {
       for (i = 0; i < T.Entry; i++)
          if (T.Data[i].Bound.Intersect(S))
              searchResult.add((T.Data[i].leaf.fileId));
   }
   return searchResult;
  }
```

Figure 4.3 R-tree search algorithm.

The search algorithm of the AVL trees is the typical search algorithm of balanced binary search trees, and requires an exact match. A total of 500 lines of source code were written to implement the AVL tree part of the GSDindex.

Combined geospatial data query results were obtained by intersecting the three result sets of searching geospatial coordinate bounding boxes, strings and dates. The source code of the algorithm used for this intersection is shown in Figure 4.4.

```
private static Vector commonV(Vector v1, Vector v2) {
    AVL2 result = new AVL2();
    for(int i = 0; i < v1.size(); i++)
       result.Insert(((Integer)v1.elementAt(i)).intValue());
    for(int i = 0; i < v2.size(); i++)
       result.Insert(((Integer)v2.elementAt(i)).intValue());
    return result.commonV;
}
```

Figure 4.4. The algorithm for GSDindex intersection.

The AVL2's “Insert” method puts the duplicates in its commonV Vector. When getting the intersection of three sets was required, the main program called this function twice and used the result from the first call and the third Vector as the parameters in the second call. The time complexity of this algorithm is O(N), where N is the number of geospatial objects being searched.

Figure 4.5 shows the architecture of the Java classes used for range search with the GSDindex search engine. This program includes the program shown in Figure 4.2 and an `AVL` class. The `searchRTreeAVL` extract information about keywords and date besides that of the bounding box from the XML files. The `AVL` class handles the insert, search and the balance functions of the AVL tree.

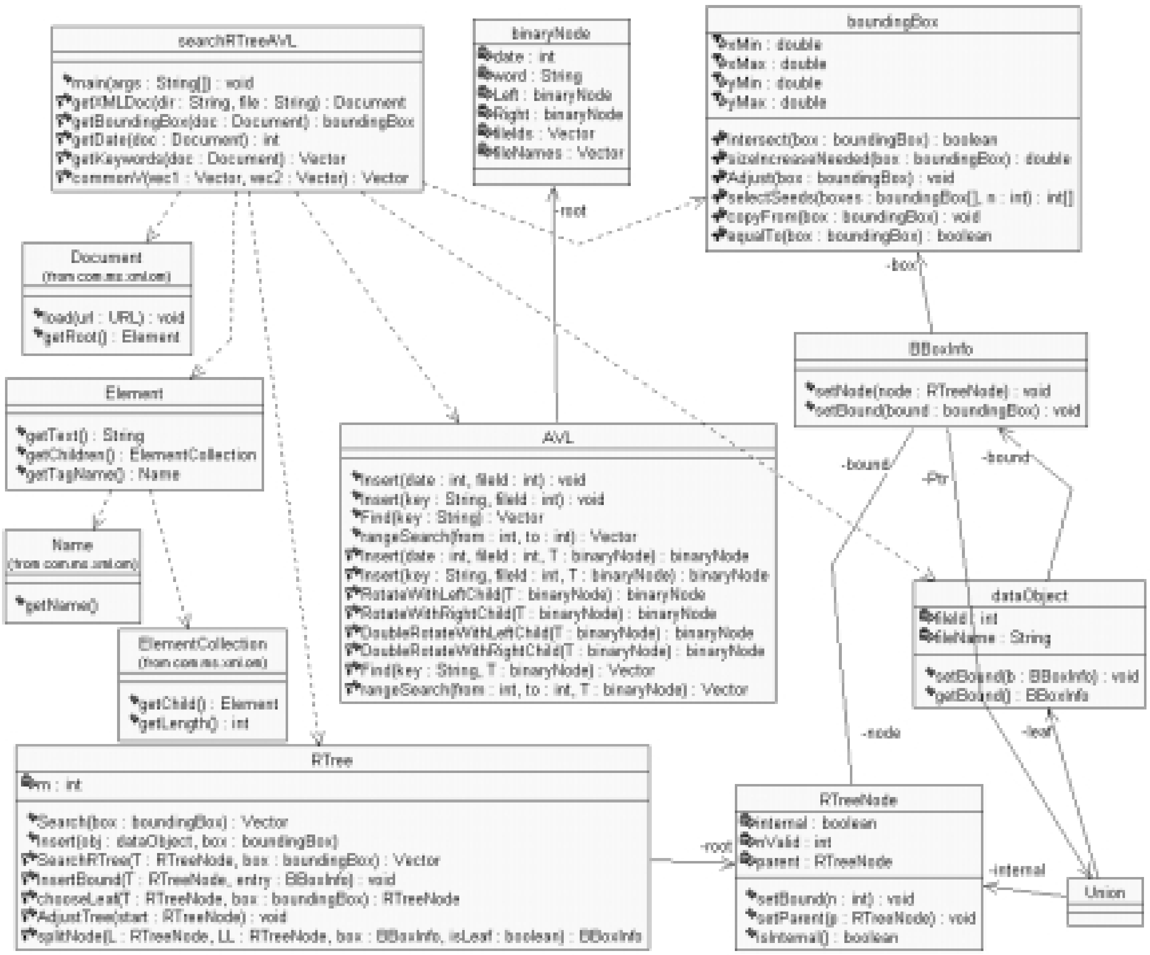


Figure 4.5 A UML diagram of the Java class structure for GSDindex range search.

4.4 Relational Database Implementation

A Relational Database Management System (RDBMS) is one of the best solutions for searching data when the amount of data grows [Hunter, 1998]. An RDBMS organizes data into tables. These tables are organized into rows and columns, much like a spreadsheet. Particular rows and columns in a table can be related to one or more rows and columns in another table.

An "insertTables" function was written to read data from ISO standard XML metadata files and insert all the information items into database tables created on the Oracle 8 NT server of the Faculty of Computer Science, University of New Brunswick. Table FileIdName stores the relation between the file ids and the file names of all the ISO standard XML metadata files in the directory; table Keywords has fileId and keyword columns; table Dates has file id and date value columns; table Coordinates has fileId, west, east, south and north columns. Tables 4.1-4.4 illustrate portions of the four tables created for this research.

Table 4.1 Relational FileIdName Table.

fileId	fileName
0	com==Aquariu...
1	com==Atlantis...
2	csr==Canada_...
3	csr==AAFC_@..

Table 4.2 Relational Keywords Table.

fileId	keyword
0	LAND SURFACE
1	BIOSPHERE
2	CANADA
3	CRYOSPHERE

Table 4.3 Relational Dates Table.

fileId	dateVal
0	19981010
1	19990410
2	19750512
3	19890901

Table 4.4 Relational Coordinates Table.

fileId	west	east	south	north
0	-102	-95	49	54
1	30	60	34	39
2	-76	-56	16	89
3	78	120	-48	-36

Figure 4.6 shows the relational schema defining the database design.

FileIdName

<u>fileId</u>	fileName

Keywords

<u>fileId</u>	<u>keyword</u>

Dates

<u>fileId</u>	dateVal

Coordinates

<u>fileId</u>	west	east	south	north

Figure 4.6 Relational schema of the database.

Data in the tables can be read, updated, appended and deleted using the Structured Query Language, or SQL. Figure 4.7 shows the SQL command used to query the Relational database for the three query types referenced above.

```
SELECT distinct FileIdName.fileName
FROM FileIdName, Coordinates
WHERE west <= Coordinates.east and
     east >= Coordinates.west and
     south <= Coordinates.north and
     north >= Coordinates.south and
     Coordinates.fileId = FileIdName.fileId
```

(a) The SQL command for querying only coordinates.

```
SELECT distinct FileIdName.fileName
FROM FileIdName, Coordinates
WHERE (west1 <= Coordinates.east or
       west <= Coordinates.east) and
      (east1 >= Coordinates.west or
       east >= Coordinates.west) and
       south <= Coordinates.north and
       north >= Coordinates.south and
      Coordinates.fileId = fileIdName.fileId
```

(b) The SQL command for querying only coordinates with longitude domain casting.

```
SELECT distinct FileIdName.fileName
FROM FileIdName, Coordinates, Dates
WHERE west <= Coordinates.east and
     east >= Coordinates.west and
     south <= Coordinates.north and
     north >= Coordinates.south and
     Dates.dateVal between date1 and date2 and
     Coordinates.fileId = Dates.fileId and
     Coordinates.fileId = FileIdName.fileId
```

(c) The SQL command for querying both coordinates and dates.

```
SELECT distinct FileIdName.fileName
FROM FileIdName, Coordinates, Keywords
WHERE west <= Coordinates.east and
     east >= Coordinates.west and
     south <= Coordinates.north and
     north >= Coordinates.south and
     Keywords.keyword like %key% and
     Coordinates.fileId = Keywords.fileId and
     Coordinates.fileId = FileIdName.fileId
```

(d) The SQL command for querying both coordinates and keywords.

```
SELECT distinct FileIdName.fileName
FROM FileIdName, Coordinates, Dates, Keywords
WHERE west <= Coordinates.east and
     east >= Coordinates.west and
     south <= Coordinates.north and
     north >= Coordinates.south and
     Dates.dateVal between date1 and date2 and
     Keywords.keyword like %key% and
     Coordinates.fileId = Dates.fileId and
     Coordinates.fileId = Keywords.fileId and
     Coordinates.fileId = FileIdName.fileId
```

(e) The SQL command for querying coordinates, dates and keywords.

Figure 4.7 SQL commands used to query the relational database.

After connecting to the relational database using the JDBC API, as shown in Figure 4.8, these SQL statements can be created and executed on the tables of the database.

```
public void JdbcConnection()
   {
       //init JDBC driver's parameters
     String dbUserName = "*****";
     String dbUserPassword = "*******";
     String dbUrl =
          "jdbc:oracle:thin:@alcuin.cs.unb.ca:1521:fcsg";

     //connect to the database
     try
     {
         DriverManager.registerDriver( new OracleDriver() );
         Connection con = DriverManager.getConnection(
          dbUrl, dbUserName,dbUserPassword);
     }
     catch (SQLException ex)
     {
          while (ex != null)
         {
            out.println("SQLState: " + ex.getSQLState() );
            out.println("Message:  " + ex.getMessage());
            out.println("Vendor:   " + ex.getErrorCode());
            ex = ex.getNextException();
         }
     }
   }  //end of function JdbcConnection
```

Figure 4.8 The connection function of JDBC Java Servlet.

The first step in using a JDBC driver to get a database connection involves loading the specific driver class into the application's JVM. This makes the driver available later, when we need it for opening the connection [Hunter, 1998]. An easy way to register the driver is to use `java.sql.DriverManager.registerDriver()` method as shown in Figure 4.8.

The next step is to ask the `DriverManager` class to open a connection to a given database, where the database is specified by a specially formatted URL. The method used to open the connection is the `DriverManager.getConnection()`. It returns a class that implements the `java.sql.Connection` interface, as shown in Figure 4.8.

A JDBC URL identifies an individual database in a driver-specific manner. Different drivers may need different information in the URL to specify the host database. JDBC URLs usually begin with *jdbc:subprotocol:subname*. For example, the Oracle JDBC-Thin driver uses a URL of the form of *jdbc:oracle:thin:@dbhost:port:sid*, as shown on line seven of Figure 4.8.

During the call to `getConnection()`, the `DriverManager` object asks each registered driver if it recognizes the URL. If a driver says yes, the driver manager uses that driver to create the Connection object.

To really use a database, we need to have some way to execute queries. Figure 4.9 shows how the test SQL statements are issued from the Java servlet program used to test the Oracle search engine. Other SQL commands are shown in Figure 4.7.

```
public Vector SelectTables(int date1, int date2, double
west, double east, double south, double north, String key)
                     throws SQLException
{     Vector result = new Vector();
      Statement stmt = con.createStatement();
      ResultSet rset;
      String select = "SELECT distinct fileIdName.fileName";
      String from = " FROM fileIdName, Coordinates";
      String where = " WHERE " + west + " <=
     Coordinates.east or " + east + ">= Coordinates.west or
     " + south + " <= Coordinates.north " + " or " + north
     + ">= Coordinates.south ) and
      Coordinates.fileId = fileIdName.fileId ";

      String queryString = select + from + where;
      rset = stmt.executeQuery(queryString);
      while (rset.next()){
          result.addElement(rset.getObject("fileName") );
      }
      stmt.close();
      return result;
    }
```

Figure 4.9 The query function of JDBC Java Servlet with example (a) of Figure 4.7.

The simplest way to execute a query is to use the `java.sql.Statement` class. Statement objects are never instantiated directly; instead, a program calls the `createStatement()` method of `Connection` to obtain a new `Statement` object. A query that returns data can be executed using the `executeQuery()` method of `Statement`. This method executes the statement and returns a `java.sql.ResultSet` that encapsulates the retrieved data. You can think of a `ResultSet` object as a representation of the query result returned one row at a time. You use the `next()` method of `ResultSet` to move from row to row. The `ResultSet` interface also boasts a multitude of methods designed for retrieving data from the current row. The `getString()` and `getObject()` methods are among the most often used for retrieving column values. The `ResultSet` is linked to its parent `Statement`. Therefore, if a `Statement` is closed or used for executing another query, any related `ResultSet` objects are closed automatically.

As mentioned in the beginning of this section, an “insertTables” function was written to insert data from ISO standard XML metadata files into database tables created on the relational database (Oracle 8) server. This was implemented using the `executeUpdate()` method of `Statement`. It returns a count that indicates the number of rows modified by the statement. It was used like the following when inserting data into the fileIdName table:

```
int count = stmt.executeUpdate(“insert into fileIdName
values (” + j + “, ‘” + filenames[j] + “’)” );
```

where j is a fileId, filenames[j] is a fileName with fileId j and fileIdName is the table listed in Table 4.1.

The Java source code used to build and search the relational database consists of a total of 550 lines. Figure 4.10 shows the UML diagram of the JDBC Java servlet program used to create and test the relational database search engine.

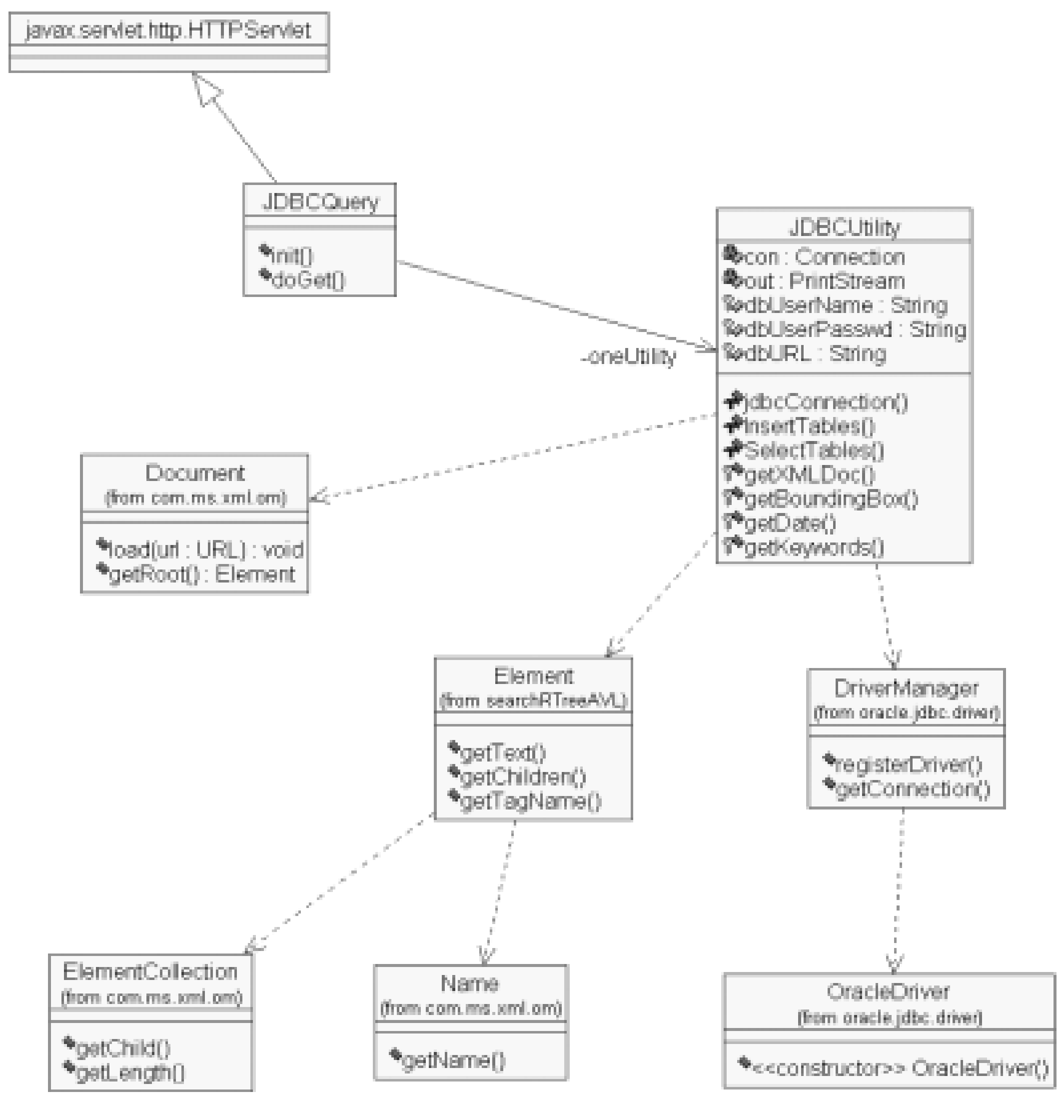


Figure 4.10 A UML diagram of the JDBC Java servlet program.

The `JDBCQuery` class is a subclass of javax.sevlet.http.HTTPServlet. Class `JDBCUtility` imports `com.ms.xml.om` package and uses its `Document`, `Element`, `Name` and `ElementCollection` classes to parse XML files and extract the bounding box, date and keywod information in the XML files. It also imports the `oracle.jdbc.driver` package and uses its `DriverManager` and `OracleDriver` classes to connect the Oracle database so as to insert and query tables.

Chapter 5 Multitier Client/Server Architecture Implementation

5.1 Introduction to Multitier Client/Server Architecture

The term client/server was first used in the 1980s in reference to personal computers (PCs) on a network. The actual client/server model started gaining acceptance in the late 1980s. The client/server software architecture is a versatile, message-based and modular infrastructure that is intended to improve usability, flexibility, interoperability, and scalability as compared to centralized, mainframe, time sharing computing [Sadoski, 1997].

The client/server model has become one of the central ideas of network computing. On the Internet, the Web browser is a client program that requests services (the sending of Web pages or files) from a Web server (e.g. a Hypertext Transport Protocol or HTTP server) in another computer somewhere on the Internet.

The two-tier client/server architecture has some deficiencies. For example, because database access functionality and business logic were often contained in the client component, any changes to the business logic, database access, or even the database itself, often required the deployment of a new client component to all the users of the application, resulting in a fragile application.

These deficiencies with the “two-tier" client/server approach were addressed by the multitier client/server architecture. Conceptually, an application can have any number of tiers, but the most popular multitier architecture is three-tier, which partitions the system into three logical tiers: the user interface layer, the business rules layer, and the data access layer. A three-tier client/server architecture is illustrated in Figure 5.1.

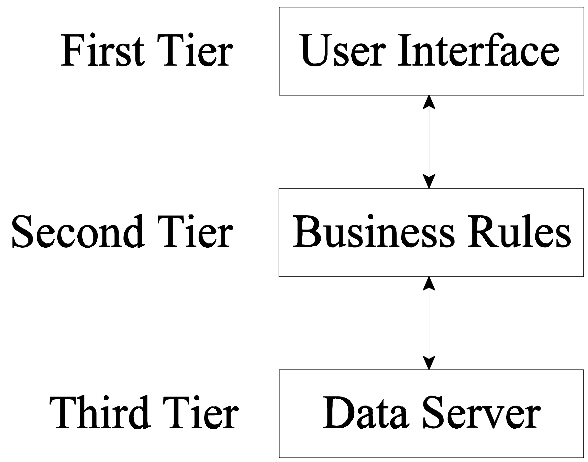


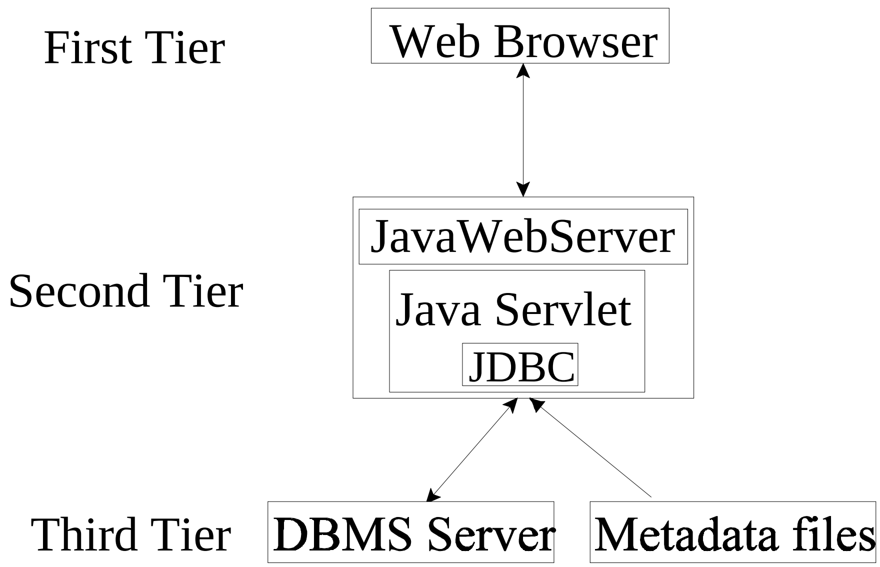
Figure 5.1. 3-tier client/server architecture.

A multitier client/server architecture enhances the two-tier client/server architecture in two ways. First, and perhaps most importantly, it makes the application less fragile by further insulating the client from changes in the rest of the application. Also, because the components (e.g. database, web server) are independent, it allows more flexibility in the deployment of an application.

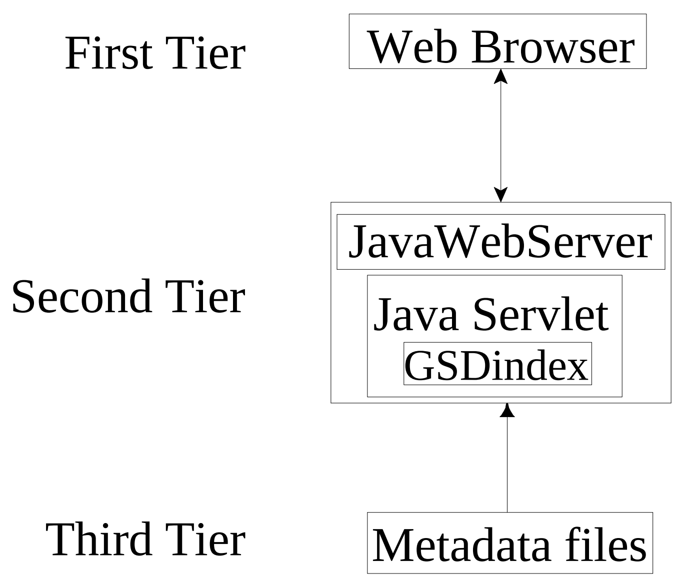
It is also possible to put each server component on a separate machine. Multiple business logic components (and multiple database access components, if multiple databases are being used) can be created for a single application, distributing the processing load and thus resulting in a more robust, scalable application.

5.2 Multitier Client/Server Architecture Implementation

Two different implementations of the three-tier architecture were carried out in this research for metadata search. Figure 5.2 illustrates the two implementations.



(a) Implementation 1 of a 3-tier architecture using Oracle database in the third tier.



(b) Implementation 2 of a 3-tier architecture using GSDindex persist data in the third tier.

Figure 5.2. Implementations of 3-tier client/server architecture for metadata query.

The user interface layer was implemented using a Java Applet program so that clients can use any Web browser supporting Java to run the client component. The business rules layer was implemented by using Java Servlet to achieve Java Applet-Java Servlet communication. In implementation (1), shown in Figure 5.2 (a), the author used JDBC to get Java Servlet and database communication and employed the Oracle 8 database server as the server layer. In implementation (2), shown in Figure 5.2 (b), the GSDindex persistent data described in chapter 4 was used as the third layer to provide data.

5.3 Client User Interface

Figure 5.3 shows the user interface extended in this research from the work of Xiao [2000]. Two sub systems were designed and implemented in this application. The first subsystem, implemented by Jun Xiao, was designed as an information system of the world map, which provides a user with context information about the political borders, rivers, lakes, and islands in a lake for the whole world [Xiao, 2000]. This subsystem also helps a user to locate and define a query of metadata in the second subsystem. The second subsystem, implemented in this research, enables a user to create and submit a metadata query combining latitude, longitude bounding box, date and keyword searching and then to list the metadata query results.

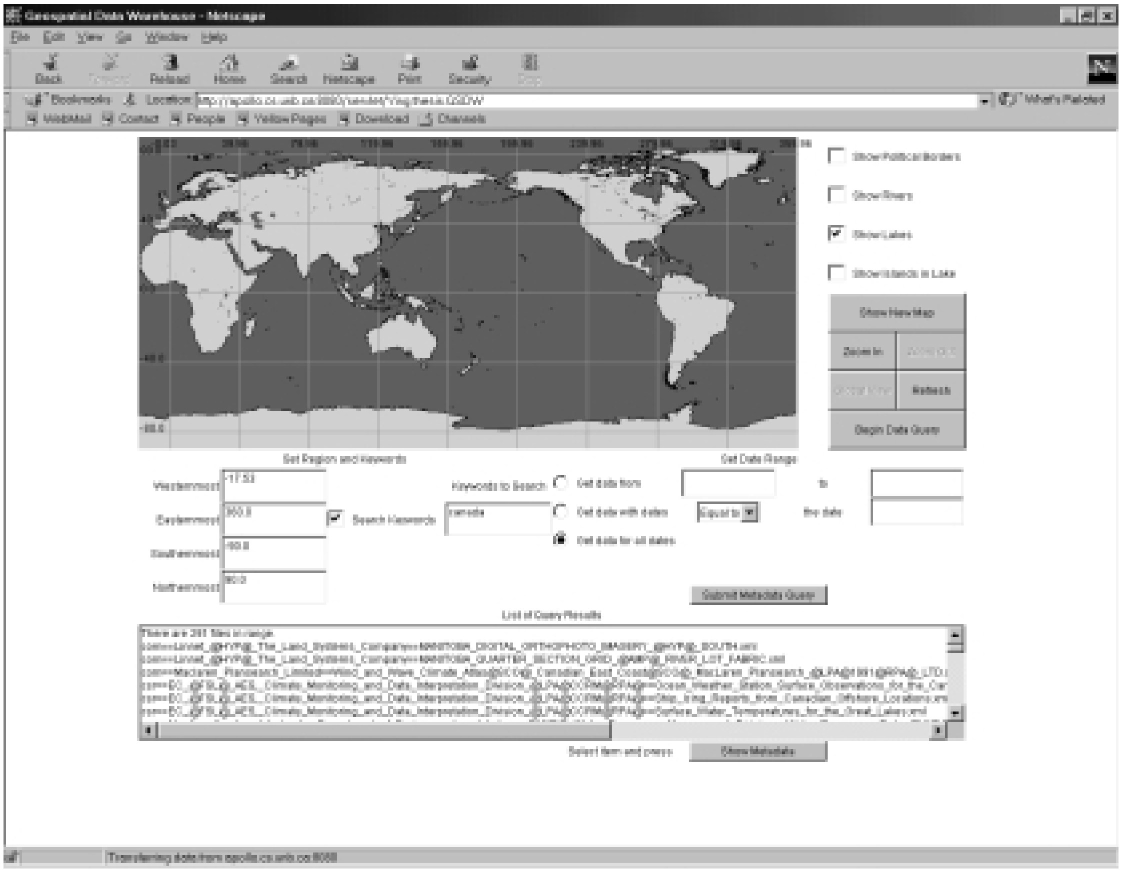


Figure 5.3 User interface of the 3-tier client/server architecture.

The first subsystem provides the following services [Xiao, 2000]:

(1) Shows a map in a view window.

(2) Allows a user to select different thematic layers of the map information, and to overlay any thematic layers (i.e. show a map with both political borders and rivers) if desired.

(3) The map shown in the view window can be expanded by choosing a rectangular region to “Zoom in” on. The region for zooming in is selected by a mouse click and drag, and the coordinate information of the region can also be shown in the text fields at the

same time.

(4) Allows “zoom out” operation.

(5) Provides a global view service to help a user find the location in the world map of the area which he/she selected to zoom in or zoom out (i.e. provides context information about where in the world the user is searching).

(6) Provides a refresh service to allow a user to refresh all the parameters of the system.

The following components displayed in the interface were implemented in this research.

(1) “Set keywords” defines the keywords used in the metadata query. When searching, the keywords searched are from the theme and place keywords of the ISO standard XML metadata files.

(2) “Set date range” defines the date range for the metadata query. There are three methods by which a date range can be set. The first method locates data within a certain date range. For example, get all data between January 1, 1998 and August 7, 1998. The second method locates data relative to a specific date. For example, “Get all metadata for all dates before, equal to or after August 7, 1998”. The last method (which is also the default method) is that XML metadata files with any dates will be returned.

(3) After a metadata query is defined (including latitude and longitude coordinates, keywords or dates) the user can click the “Submit Metadata Query” button to submit a metadata query.

(4) A list of XML metadata file names is returned after a metadata query is submitted. “No Match” is displayed in this list if there is no metadata satisfying the query criteria.

(5) After a list of metadata file names are shown in the query result list, if the user clicks

on any file name to select it, and then clicks the “Show Metadata” button, the XML metadata file is shown on the screen.

As mentioned above, the author’s client Java applet program extended previous work [Xiao, 2000]. This extension consists of adding one Java class `fileFrame` and adding a total of 500 lines of Java source code to Xiao’s existing applet of 1500 lines of Java source code. The UML diagram showing the class structure is given in Figure 5.4.

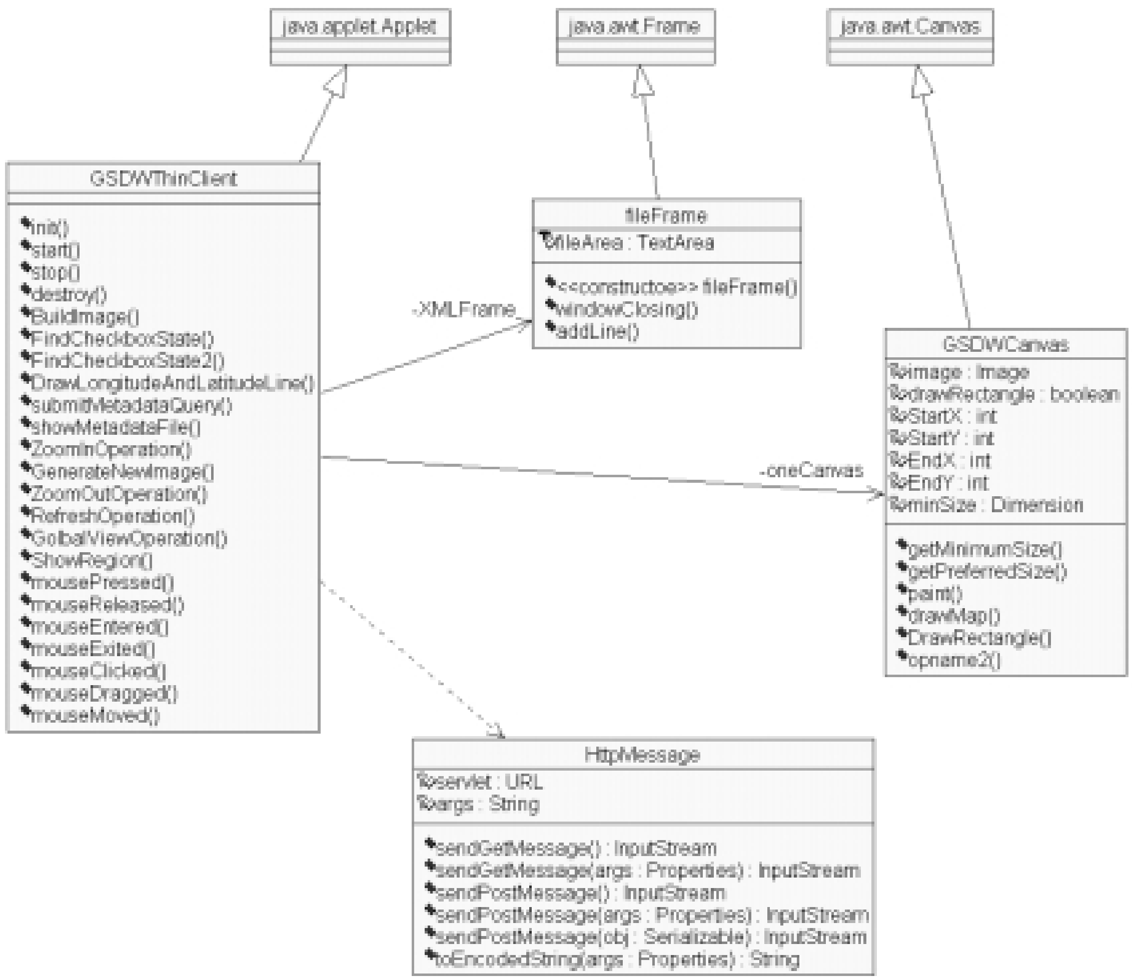


Figure 5.4 A UML diagram of the client applet program.

5.4 Middle Tier Design

One common place for servlets, especially servlets that access a database, is in what is called the “Middle Tier” [Hunter, 1998]. A middle tier connects the database to the client, and is responsible for routing all client requests for services to the proper server component.

On the Web, middle tiers are often implemented using servlets. Servlets provide a convenient way to connect clients built using HTML forms or applets to back-end servers. A client can communicate its requirements to the servlet using HTTP, and the business logic in the servlet handles the request by connecting to the back-end server.

5.4.1. Introduction to Java Servlet

Java Servlet is server-side Java technology. The rise of server-side Java applications is one of the latest and most exciting trends in Java programming. A servlet is a generic server extension— a Java class that can be loaded dynamically to expand the functionality of a server [Hunter, 1998]. Java Servlets provide web developers with a simple, consistent mechanism for extending the functionality of a web server and for accessing existing business systems. A servlet can be thought of as an applet that runs on the server side [Sun, 1999a]. Servlets are the Java platform technology of choice for extending and enhancing web servers. Servlets provide a component-based, platform-independent method for building web-based applications. Unlike proprietary server extension mechanisms (such as the Netscape Server API or Apache modules), Servlets are server- and platform-independent. This leaves you free to select a "best of breed" strategy for your servers, platforms, and tools.

Written in Java, servlets have access to the entire family of Java APIs, including the JDBC API to access enterprise databases. Servlets can also access a library of HTTP-specific calls, and all the benefits of the Java language, including portability, performance, reusability, and crash protection.

5.4.1.1 Java Servlet Lifecycle

Servlets are always dynamically loaded, although servers usually provide an administrative option to force servlet loading and initialization when the server starts up [Hunter, 1998]. Servlets are loaded using normal Java class loading facilities, which means that they may be loaded from remote directories as easily as from the local file system. This allows for increased flexibility in system architecture and easier distribution of services in a network.

Servers also vary in how they know when to load servlets. When a request comes in, the server knows how to map it to a servlet, which may first need to be loaded. That mapping is usually done in one of these common ways:

(1) Server administrators might specify that some kinds of client requests always map to a particular servlet. For example, one servlet might handle all requests for a particular database.

(2) Server administrators might specify that part of the client request is the name of the servlet, as found in an administered servlets directory. At many sites, that directory is shared between servers which share the processing load for the site's clients.

(3) Some servers may be able to automatically invoke servlets to filter the output of other servlets, based on their administrative configuration. For example, particular types of servlet output can trigger postprocessing by other servlets, perhaps to perform format conversions.

(4) Properly authorized clients might specify the servlet which is to be invoked, without administrative intervention.

After being loaded, three main methods are involved in the lifecycle of a servlet:

(1) Servlets are activated by the server through an *init* call. Servlet writers may, if they wish, provide their own implementation of this call, to perform potentially costly (usually I/O intensive) setup only once, rather than once per request. Examples of such setup include initializing sessions with other network services or getting access to their persistent data (stored in a database or file; e.g. the GSDindex data structures).

(2) After initialization, servlets handle many requests. Each client request generates one service upcall. These requests may be concurrent, which allows servlets to coordinate activities among many clients.

(3) Requests are processed until the servlet is explicitly shut down by the web server, by calling the *destroy* method. The servlet's class may then become eligible for garbage collection [JavaServer Product Group, 1999], which releases its memory back to the pool of available storage.

5.4.2. Java Servlet Implementation

Figure 5.4 combines the 3-tier implementations shown in Figure 5.2 with the showfile Java servlet and Jun Xiao's GSDW Java servlet [Xiao, 2000]:

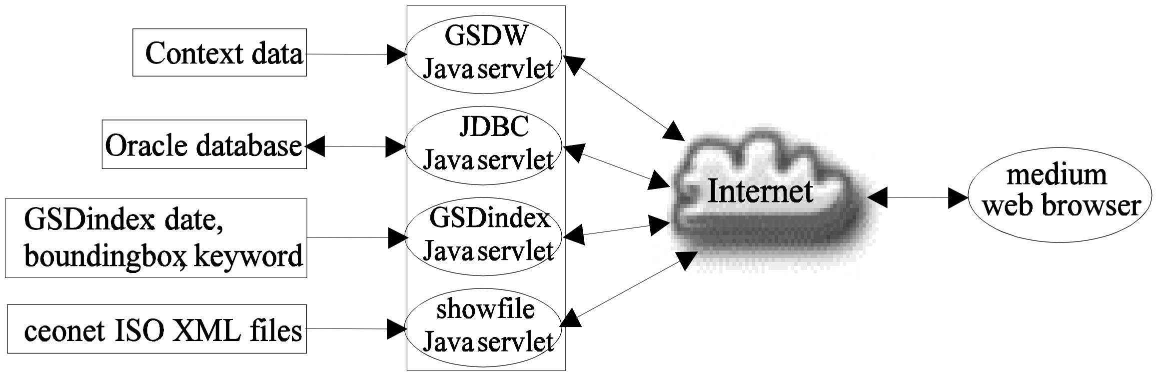


Figure 5.4. Architecture for testing combined geospatial data queries. A medium web browser supports Java applets.

(1) The GSDW Java sevlet was implemented by Jun Xiao [Xiao, 2000] and it handles the proper display of the map images.

(2) “GSDindex” is composed of three separate data structures: one for bounding boxes of the geospatial metadata (using an R-tree), a second one for geospatial metadata dates, and a third one for strings to hold metadata keywords. An AVL tree is used for both the date and keyword indexes. The GSDindex Java servlet handles the XML metadata file query by loading the date, boundingbox and keyword GSDindex persistent data (files storing serialized objects) and carrying out searching on the loaded R-tree and AVL trees, as explained in chapter 4.

(3) The JDBC Java servlet deals with the XML metadata query by using the JDBC API to generate and send SQL queries to the Oracle 8 database server.

(4) The showfile Java servlet accepts the XML file name as a parameter from the client and returns the corresponding Ceonet ISO standard XML metadata file to the client.

In summary, two different techniques were used to perform geospatial XML metadata searches, as shown in Figure 5.2: (1)our own hybrid geospatial index data structure called "GSDindex" and (2) connecting an Oracle database using JDBC.

The Java Applet-Java Servlet communication happens when all four different Java servlets in Figure 5.4 exchange data with the client. Java servlet-database communication techniques were used in the JDBC Java servlet programs. These two communication techniques are discussed in the following sections.

5.4.2.1 Applet-Servlet Communication

There are two options for applet-servlet communication [Hunter, 1998]:

(1) Have the applet establish an HTTP connection to a servlet program on the server machine. The applet acts like a browser and requests a page, parsing the response for its own use. The applet can provide information using a query string or POST data and can receive information from the returned page.

(2) Have the applet establish a raw socket connection to a non-HTTP servlet running on the server machine. The non-HTTP servlet can listen to a particular port and communicate with the applet.

Each of these approaches has advantages and disadvantages. Having an applet make an HTTP connection to a servlet program works well for the following reasons:

(1) It’s easy to write. The applet can take advantage of the java.net.URL and java.net.URLConnection classes to manage the communication channel.

(2) It works even for applets running behind a firewall. Most firewalls allow HTTP connections but disallow raw socket connections.

(3) It allows a Java applet to communicate with a program on the server side written in any language. The server side program doesn't have to be written in Java. It can be in Perl, C, C++, or any other language.

(4) It works with applets written using JDK 1.0, so it works with all Java-enabled browsers.

(5) It allows secure communication. An applet can communicate with a secure server.

The HTTP connection to a servlet program also has some problems, as follows:

(1) It's slow. Due to the HTTP request/response paradigm, the applet and the servlet program cannot communicate interactively.

(2) Only the applet can initiate communication. The servlet program has to wait passively for the applet to request something before it can respond.

An applet and servlet can also communicate by having the applet establish a socket connection to a non-HTTP server process. This provides the following advantage over the HTTP-based approach: it allows bidirectional, sustained communication. The applet and servlet can use the same socket to communicate interactively, sending messages back and forth. For security reasons, the applet must always initiate the connection by connecting to a server socket on the server machine, but after a socket connection has been established, either party can write to the socket at any time.

A socket connection also has disadvantages versus the HTTP-based approach. It fails for applets running behind firewalls. Most firewalls don't allow raw socket connections, and thus they disallow this sort of applet-servlet communication. Therefore, this mechanism should be used only when an applet is guaranteed to never run on the far side of a firewall, such as for an Intranet application.

Since the author can not guarantee that the applet user will not run the applet behind a firewall, and can not be sure if the user's web browser supports Java 1.1 or above, the HTTP data communication protocol was selected for this research.

Java object serialization has simplified the issues involved with formatting responses. With both applets and servlets written in Java, it is only natural that they should communicate by exchanging Java objects. In this research, the search result set's XML metadata file names were returned to the applets as a Vector object.

5.4.2.2 Servlet-Database Communication

The GSDindex Java servlet uses file storage on the local disk to store its persistent data. The use of a flat file is fine for a small amount of data, but it can quickly get out of control [Hunter, 1998]. As the amount of data grows, access times slow to a crawl. One solution to this problem is a Relational Database Management System (RDBMS). Java servlets, with their enduring life cycle, and JDBC (Java Database Connectivity), a well-defined database-independent database connectivity API (Application Program Interface), are an elegant and efficient solution for web masters who need to hook their web sites to back-end databases. In this research, the technology used in Java servlet and database communication is JDBC, an API that enables database access in Java. It consists of a set of classes and interfaces written in Java that allow the programmer to send SQL statements to a database server for execution and, in the case of an SQL query, to retrieve query results.

An individual database system is accessed via a specific JDBC driver that implements the java.sql.Driver interface. Drivers exists for nearly all popular RDBMS systems [Sun, 2000]. A JDBC technology-based driver ("JDBC driver") makes it possible to do three things [Sun, 1999b]:

1.Establish a connection with a data source.

2.Send queries and update statements to the data source.

3.Process the results.

5.4.3 Java Web Server

Although many web server vendors support Java Servlet [Hunter, 1998], Sun's Java Web Server 1.1 was chosen as the web server in this research. Java Web Server was written entirely in Java and fully supports Java Servlet. Starting the Java Web Server is easy. Look for the *httpd* script (*httpd.exe* program under Windows) in the *server_root/bin* directory (where *server_root* is the directory where the server is installed). This should start the server if running under Solaris or Windows. In the default configuration, the server listens on port 8080. There are several ways to access a servlet. In this research, the author used explicit URL access with */servet/* prepended to the servlet's class name. In the Web browser enter: *http:server:8080/servlet/classname.* If the servlet is part of a package, it would need to be placed in the *server_root/servlets/package/name* and referred to with the URL *http:server:8080/servlet/packagename.classname.* For this research discussed here, the URL was

http:apollo.cs.unb.ca:8080/servlet/Ying.thesis.GSDW.

5.5 Oracle 8 Database Server

The Oracle 8 database server running on a Windows NT server in the Faculty of Computer Science, University of New Brunswick was used as the server layer of the 3-tier architecture showed in Figure 5.2 (a). As described in section 4.3, a relational database with four simple tables was used.

5.6 Source Code Summary

Table 5.1 shows a summary of all source code written for this thesis.

Table 5.1 Summary of all source code written for this thesis.

Item	Program Name	Code language	No. of lines
1	NTDB translator	C++	630
2	XML translator	Java	2450
3	searchRTreeAVL	Java	1300
4	GSDindex servlet	Java	276
5	ShowFile servlet	Java	61
6	JDBC servlet	Java	550
7	GSDWThinClient	Java	2000 (1500 lines from Xiao [2000]

Chapter 6 Test Results

6.1 Test Data

Testing was carried out using 6979 geospatial FGDC CSDGM XML metadata files from CEONet [CEONet, 2000]. Fuqun Zhou of the CCRS Department of Natural Resources Canada kindly supplied us these metadata files. The metadata files require a space of about 100MB and they all look like the file in Appendix III. For example, file "com==Canadian_Seabed_Research_Ltd@DOT@==Canadian_Beaufort_Sea_Ice_Scour_Database" has the contents as shown in Appendix XI. The same file after transformed to ISO standard is listed in Appendix XII. These FGDC standard XML metadata files were transformed to ISO standard XML metadata files by the XML translator and XSLT. The XML translator 1862.16 seconds and and XSLT required 6 hours 14 minutes and 57 seconds, i.e. XSLT took about 12 times longer to do the translation.

Two different techniques were used to carry out geospatial XML metadata queries; our own hybrid geospatial index data structure called "GSDindex" and connecting to an Oracle database using JDBC.

6.2 Test Results of R-Tree, AVL Tree Index and Oracle 8 Database

Test results with 6979 CEONet ISO XML metadata files have been obtained for both the GSDindex and Oracle 8 database implementations. Testing was carried out using a Sun Ultra 5 workstation pituit with 128MB RAM running the Solaris 2.6 operating system. For testing purpose, both the client Web browser and the Java Web Server were running on the same workstation.

The times to build the R-tree and AVL trees from the 6979 CEONet ISO XML metadata files are listed in Table 6.1.

Table 6.1 Time to build GSDindex data structures.

Run Time (in seconds)		
	Build the Tree	Build and Write to a File
bounding box R-tree	602.37	621.40
date AVL tree	425.29	425.59
keywords AVL tree	1510.54	1663.29

The times to load the R-tree and AVL trees in the GSDindex Servlet are listed in Table 6.2.

Table 6.2 Time to load the GSDindex into the Java servlet.

Time to load the GSDindex in servlet (in seconds)	
bounding box R-tree	46.17
date AVL tree	0.62
keywords AVL tree	97.43

A relatively long time is required to initially load the GSDindex data structures (saved as serializable Java objects) into the GSDindex Java servlet. This may slow down the first time query when the GSDindex servlet is initialized by the Java Web Server. I encountered difficulties in loading bounding box the R-tree object. Due to the large amount of data (724KB), when loading, the readObject() method of ObjectInputStream class gave a “segmentation fault”. This was overcome by changing the R-tree implementation m (see section 4.2) value from 4 to 50.

The time to build the Oracle tables was 3989.12 seconds.

The search time is the time that it takes the client to get query results after the request is sent out. Both the web browser and the server programs were running on the same workstation for the tests. Search times for querying only bounding boxes of both approaches are listed in Table 6.3. The results in tables 6.3 to 6.9 are the average of 10 separate runs for each trial, and were timed using the getTime() method of java.util.Date class.

Table 6.3 Comparison of average search times for bounding box queries.

Average Search Time (in seconds)					
Trial NO.	Bounding Box Size (degrees)	Number of files returned	GSDindex	Oracle database	Ratio (Oracle/GSDindex)
1	360 x 180	6977	69.50	114.62	1.65
2	160 x 160	3324	27.43	53.46	1.95
3	120 x 80	2162	17.38	41.84	2.41
4	160 x 80	3406	29.05	65.84	2.27
5	80 x 80	3896	33.03	78.04	2.36
6	40 x 40	1780	14.82	34.57	2.33

The average ratio for the query time of the Oracle database approach to that of the GSDindex approach was 2.16.

Search times for querying only keywords of both approaches are listed in Table 6.4.

Table 6.4 Comparison of average search times for keyword queries.

Average Search Time (in seconds)					
Trial NO.	Keywords	Number of files returned	GSDindex	Oracle database	Ratio (Oracle/GSDindex)
1	earth science	6537	61.95	137.18	2.21
2	oceans	1523	9.41	33.56	3.57
3	Canada	718	5.14	13.69	2.66
4	transportation	119	0.85	3.00	3.53
5	new brunswick	108	0.89	2.88	3.24
6	China	74	0.50	2.42	4.84

The average ratio for the search time of the Oracle database approach to that of the GSDindex approach was 3.34.

Search times for querying only dates of both approaches are listed in Table 6.5. The dates are in YYYYMMDD format.

Table 6.5 Comparison of average search times for date queries.

Average Search Time (in seconds)					
Trial NO.	Dates	Number of files returned	GSDindex	Oracle database	Ratio (Oracle/GSDindex)
1	0-21001010	825	6.91	15.87	2.30
2	before 19991117	669	5.54	12.90	2.33
3	19981010-19991010	517	4.58	10.36	2.26
4	after 19991010	250	2.11	5.36	2.54
5	equal to 19991117	2	0.24	0.66	2.75

The average ratio for the search time of the Oracle database approach to that of the GSDindex approach was 2.44.

Search times for querying bounding boxes and keywords of both approaches are listed in Table 6.6. The keyword searched was “Canada”.

Table 6.6 Comparison of average search times for bounding box and keyword queries.

Average Search Time (in seconds)					
Trial NO.	Bounding Box Size	Number of files returned	GSDindex	Oracle database	Ratio (Oracle/GSDindex)
1	360 x 180	710	6.48	16.80	2.59
2	160 x 160	346	3.05	9.70	3.18
3	120 x 80	39	0.65	3.10	4.77
4	160 x 80	115	1.59	5.32	3.97
5	80 x 80	597	4.82	15.19	3.15
6	40 x 40	39	0.68	3.05	4.49

The average ratio for the search time of the Oracle database approach to that of the GSDindex approach was 3.69.

Search times for querying bounding boxes and dates of both approaches are listed in Table 6.7. The date range is from October 10^{th} , 1998 to October 10^{th}, 1999.

Table 6.7 Comparison of average search times for bounding box and date queries.

Average Search Time (in seconds)					
Trial NO.	Bounding Box Size	Number of files returned	GSDindex	Oracle database	Ratio (Oracle/GSDindex)
1	360 x 180	594	5.41	11.26	2.08
2	160 x 160	356	3.18	6.86	2.16
3	120 x 80	206	1.79	4.13	2.31
4	160 x 80	278	2.71	5.95	2.20
5	80 x 80	416	3.55	8.23	2.32
6	40 x 40	179	1.53	3.50	2.29

The average ratio for the search time of the Oracle database approach to that of the GSDindex approach was 2.23.

Search times for querying keywords and dates of both approaches are listed Table 6.8.The dates are in YYYYMMDD format.

Table 6.8 Comparison of average search times for keyword and date queries.

Average Search Time (in seconds)						
Trial NO.	keywords	dates	Number of files returned	GSDindex	Oracle database	Ratio (Oracle/ GSDindex)
1	earth science	19981010-19991010	517	4.42	10.11	2.29
2	Canada	19981010-19991010	246	1.88	5.63	2.99
3	oceans	before 19991212	180	1.38	5.24	3.80
4	new brunswick	after 19990000	104	0.93	3.16	3.40
5	transportation	after 19981010	26	0.36	1.89	5.25
6	China	0-21000000	4	0.26	1.08	4.15

The average ratio for the search time of the Oracle database approach to that of the GSDindex approach was 3.65.

Search times for querying bounding boxes, keywords and dates of both approaches are listed Table 6.9. The keyword searched was "Canada" and the date range is from October 10th ,1998 to October 10th, 1999.

Table 6.9 Comparison of average search times for bounding box, keywords and date queries.

Average Search Time (in seconds)					
Trial NO.	Bounding Box Size	Number of files returned	GSDindex	Oracle database	Ratio (Oracle/GSDindex)
1	360 x 180	245	3.16	6.55	2.07
2	160 x 160	141	1.73	4.32	2.50
3	120 x 80	16	0.53	1.85	3.49
4	160 x 80	29	1.03	2.09	2.03
5	80 x 80	173	1.61	5.10	3.17
6	40 x 40	16	0.48	1.76	3.67

The average ratio for the search time of the Oracle database approach to that of the GSDindex approach was 2.82.

Table 6.10 summarizes the ratios for the search time of the Oracle database approach to that of the GSDindex approach when the search domain changes.

Table 6.10 Summary of search time ratios.

Search Domain	Ratio (Oracle/GSDindex)
Bounding box	2.16
Keywords	3.34
Dates	2.44
Bounding box + keywords	3.69
Bounding box + dates	2.23
Keywords + dates	3.65
Bounding box + keywords + dates	2.82

From Table 6.10 we can see that GSDindex approach is about two and a half ((2.16 + 3.34 + 2.44)/3 = 2.65) times as fast as the Oracle database approach when only one type of data (e.g. bounding box) is queried. GSDindex approach is about three ((3.69 + 2.23 + 3.65)/3 = 3.19) times as fast as the Oracle database approach when two types of data (e.g. bounding box and date or bounding box and keywords) are queried. When all three types of data are searched, the GSDindex approach is also about three (2.82) times as fast as the Oracle database approach.

From the above observations, we can say that search using data structure indexing (e.g. R-tree) is more efficient than the Oracle database table search. When combined data queries are carried out, it appears that the GSDindex approach's set intersection procedure is also a little faster than that of the Oracle database. The test results agree with what was expected. The Oracle relational database was expected to be slower due to SQL queries against a database stored on disk. The GSDindex approach was expected to be faster because it carried out its queries on the data objects loaded in memory. Considering the ratio of memory access speed to disk access speed, which can be thousands of times greater, the Oracle database performance is better than expected.

Chapter 7 Conclusions and Future Work

7.1 Related Work

The proliferation of applications that retrieve geospatial data from multiple heterogeneous data sources has served as a catalyst for the research in areas of geospatial metadata integration and query processing. Much of the work on geospatial metadata search engines has focused on one or more local metadata representations. FGDC has a clearinghouse search site [FGDC, 1999] which allows users to search FGDC metadata files by defining search criteria (geospatial coordinates, date range and/or keywords) and servers. Other similar search site are the CEONet server sponsored by Natural Resources Canada [CEONet, 2000] and the Master Environmental Library sponsored by the U.S. Dept. of Defense [MEL, 1999]. No search engines has been built to search ISO geospatial metadata files.

This research provides the first systematic, scientific transformation of FGDC geospatial metadata standard files to ISO geospatial metadata standard files. This research has also built efficient search engines to search ISO geospatial metadata files on the Web so as to aid discovery and browsing of geospatial data.

7.2 Conclusions

We have defined an XML standard for geospatial metadata based on the ISO/TC 211 draft standard that appears to work effectively. In addition, we have made a preliminary pass at defining a formal grammar for geospatial metadata. This formal grammar uses XML tags (based on the ISO “short name”) to precisely define the limits of each metadata entity. We have also described an initial formal context free grammar representation that precisely defines how the geographic information metadata sentences are correctly constructed.

Building on previous research, a complete three-tier system for Web-based geospatial data search was designed, implemented using Java servlets in the middle tier and tested. Two different search engine approaches were used. The first, called GSDindex, is based on three independent data structures (i.e. an R-tree and two AVL trees). The second approach used a relational (Oracle 8) database.

Test results (carried out on 6979 geospatial metatada files) indicate that the GSDindex approach for web-based querying of geospatial metadata handles each query about 2.5 times as fast as the Oracle 8 database approach when only one type of data (e.g. bounding box) is searched. GSDindex approach is about 3 times as fast as the Oracle database approach when two types of data (e.g. bounding box and date or bounding box and keywords) are queried. When all three types of data are searched, the GSDindex approach is also about 3 times as fast as the Oracle database approach.. From the test results, we can conclude that search using data structure indexing (e.g. R-tree) is more efficient than the Oracle database table search. When combined data queries are carried out, it appears that the GSDindex approach's set intersection procedure is also faster than that of the Oracle database.

7.3 Future Work

We expect that the GSDindex performance can improve if we develop a data structure capable of supporting efficient search of combined spatial (e.g. bounding box), keyword and date queries. Such a data structure would be particularly valuable if it supported hierarchical organization of very large collections of spatial data. In addition, a formal investigation of how to specify XML for such combined geospatial data queries and geospatial data update operations would be worthwhile.

References

Aho, A.V., Sethi, R. and Ullman, J.D. *Compilers Principles, Techniques, and Tools*, Addison-Wesley, Reading, MA, 1986.

Appel, A.W. *Modern Compiler Implementation in Java*, Cambridge University Press, New York, NY, 1998.

Bédard, Y., Coleman, D., Han, J., Merrett, T., Moulin, B., Nickerson, B., Tao, C.T. *Designing the Technological Foundations of Geospatial Decision-Making Using the World Wide Web*. GEOID Strategic Plan, Proposed Research Program, University of Laval, Quebec City,1999.

Boumphrey F., Renzo O., Duckett J., Graf J., Hollander D., Houle P., Jenkins T., Jones P., Kingsley-Hughes A., Kingsley-Hughes K., McQueen C., Mohr S., *XML Applications,* Wrox Press Ltd., Olton, Birmingham, 1998.

CEONet, home page at http://ceonet.cgdi.gc.ca/cs/en/, 2000.

Chomsky, N. "Three models for the description of language", *IRE Trans. on Information Theory*, IT-2:3, 1956, pp.113-124.

Christian, E. *Application profile for the Government Information Locator* (GILS*).* Draft Version 2., 1996. Retrieved from World Wide Web: http://www.usgs.gov/public/gils/prof_v2.html

FGDC, Federal Geographic Data Committee, "Content Standard for Digital Geospatial Metadata", document FGDC-STD-001-1998, FGDC Secretariat, US Geological Survey, 590 National Center, Reston, VA, 1998, available from http://www.fgdc.gov/metadata/contstan.html

FGDC, Federal Geographic Data Committee, "Metadata Tools for Geospatial Data", 1999, available from http://badger.state.wi.us/agencies/wlib/sco/metatool/mtools.htm

FGDC, Federal Geographic Data Committee, "The Value of Metadata, Manager's Brochure", 1999, available from http://www.fgdc.gov/publications/documents/metadata/metabroc.html.

Gluck, M. Standards for electronic access to geographic and spatial information. In E. Logan, & M. Gluck. (Eds.). Electronic publishing: Applications and Implications. Contributed papers to the ASIS Midyear Meeting, Minneapolis, MN, May 1995.

Gosling, J., Joy, B. and Steele, G. *The Java Language Specification*, Addison-Wesley, Reading, MA, 1996, also available at http://java.sun.com/docs/books/jls/

Guttman, Antonin, *R-TREES: A DYNAMIC INDEX STRUCTURE FOR SPATIAL SEARCHING,* University of California, Berleley, *ACM SIGMOD Record*, vol.14, no.2, , 1984, pp.47-57.

Harold, Elliotte Rusty, Chapter 14 of the *XML Bible*, 1999, available at http://metalab.unc.edu/xml/books/bible/updates/14.html#Overview

Holzner, Steve, *XML Complete,* McGraw-Hill, New York, USA, 1998.

Hunter, Jason, William Crawford, *Java Servlet Programming,* O'Reilly & Associates, Inc., Sebastopol, CA, USA, 1998.

IBM, *LotusXSL,* 2000, http://www.alphaworks.ibm.com/formula/LotusXSL

ISO/IEC 14882, *Programming Languages - C++*, International Organization for Standardization/International Electrotechnical Commission, Sept. 15, 1998, 732 pages, available for download from http://www.cssinfo.com/ncitsgate.html

ISO/TC 211 Secretariat, "ISO/TC 211 Geographic information/Geomatics CD 15046-15, Geographic information - Part 15: Metadata", Committee draft for comments, ISO/TC 211 N538, available from Norwegian Technology Standards Institution, Oslo, Norway, http://www.statkart.no/isotc211/, July 9th, 1998, 161 pages.

ISO/TC 211 Secretariat, "ISO/TC 211 Geographic information/Geomatics 2. CD 19115, Geographic information - Metadata", Committee draft for comments, ISO/TC 211 N831, available from Norwegian Technology Standards Institution, Oslo, Norway, http://www.statkart.no/isotc211/, November 30th, 1999, 118 pages.

JavaServer Product Group, "The Java Servlet API", 1999. http://jserv.javasoft.com:80/products/java-server/documentation/webserver1.0.2/servlets/api.html

Ken Sall, *XML: Structuring Data for the Web: An Introduction.* May 3rd 1998. http://wdvl.com/Authoring/Languages/XML/Intro/

MEL, Master Environmental Library, U.S. Dept. of Defense, 1999 http://mel.dmso.mil/

Microsoft, "Download Center", 2000, available at http://www.microsoft.com/downloads/

MSDN Online Web Workshop, "Why XML", 1999 http://msdn.microsoft.com/workshop/xml/toc.htm

Nebert, D. D, "Z39.50 Application Profile for Geospatial Metadata or "GEO" ", 2000 http://www.blueangeltech.com/Standards/GeoProfile/geo22.htm

NTDB, Natural Resources Canada, “National Topographic Data Base, Metadata Format, Edition 3", November, 1998, available from http://www.ccg.rncan.gc.ca/ext/html/english/products/ntdb/metantd3.pdf

NTDB, Natural Resources Canada, “National Topographic Data Base, 1:50 000 and 1:250 000",1999, available from http://www.ccg.rncan.gc.ca/ext/html/english/products/ntdb/ntdb.html

Nickerson, B.G. and Teng, Y. “A Formal Grammar for Geographic Information Metadata”, UNB Faculty of Computer Science Technical Report TR00-132, Feb. 2000, 25 pages.

Norman, Pontus “A study of eXtensible Markup Language (XML)”, Thesis for Master of Science degree in Computer Science at the Department of Teleinformatics, Royal Institute of Technology in Sweden, Decerno AB, Näsby Park, Sweden, February 25, 1999.

OCLC Online Computer Library Center. *Map formats*. (2d ed. and rev.). Dublin, OH: OCLC, 1988.

OCLC Online Computer Library Center. *Bibliographic formats and standards*. Dublin, OH: OCLC, 1995.

OGC, Open GIS Consortium, “The OpenGIS™ Abstract Specification Topic 11: Metadata” , Version 4, 1999.

OMB, Office of Management and Budget, *Establishment of Government Information Locator Service*. OMB 95-01. Federal Register, December 7, 1994, vol. 59234, pp. 63075—77.

OMG, Object Management Group, "The Common Object Request Broker: Architecture and Specification", minor revision 2.3.1, October, 1999, refer specifically to Chapter 3-OMG IDL Syntax and Semantics, http://www.omg.org/corba/cichpter.html, 58 pages.

Oracle, *Download Oracle Products, Drivers, and Utilities,* 2000, available at http://technet.oracle.com/software/

Phillips, H. "Metadata Tools for Geospatial Data", web page available from http://badger.state.wi.us/agencies/wlib/sco/metatool/mtools.htm

Sadoski, Darleen ,“Client/Server Software Architectures--An Overview”, Carnegie Mellon University, 1997. http://www.sei.cmu.edu/str/descriptions/clientserver_body.html

Sall, Ken, “XML: Structuring Data for the Web: An Introduction”. May 3rd 1998.

http://wdvl.com/Authoring/Languages/XML/Intro/

Sebasta, R.W. *Concepts of Programming Languages*, third edition, Addison-Wesley, Reading, MA, 1996.

St. Pierre, M and LaPlant, W. P., "Issues in Crosswalking Content Metadata Standards", 1998, available at http://www.niso.org/crsswalk.html

Stroustrup, B. *The C++ Programming Language*, third edition, Addison-Wesley, Reading, MA, 1997.

Sun Microsystems Inc., *JAVATM SERVLET API: The Power Behind the Server*, 1999a.
http://java.sun.com/products/servlet/index.html

Sun Microsystems Inc., *JDBC Technology Guide: Getting Started*, 1999b.
http://java.sun.com:80/products/jdk/1.3/docs/guide/jdbc/getstart/GettingStartedTOC.fm.html

Sun Microsystems Inc., *JDBC Data Access API*, 1999c.
http://java.sun.com:80/products/jdbc/index.html

Sun Microsystems Inc. 'JDBC Data Access API:DRIVERS", 2000.
http://industry.java.sun.com/products/jdbc/drivers

Teng, Y. and Nickerson, B.G. "XML (eXtensible Markup Language) for Geospatial Metadata", UNB Faculty of Computer Science Technical Report TR00-131, Jan. 2000, 34 pages.

W3C,World Wide Web Consortium, *Extensible Markup Language (XML) Version 1.0*, Recommendation 10-February-1998,
http://www.w3.org/TR/1998/REC-xml-19980210.html

Weiss, Mark Allen, *Algorithms, Data Structures, and Problem Solving with C++,* Addison-Wesley Publishing Company, Inc. 1996.

Xiao, Jun, "WWW Access to Geospatial Data Warehouses", Master of Computer Science Thesis, Faculty of Computer Science, University of New Brunswick, Fredericton, NB, Canada, May 2000.

APPENDIX I Sample NTDB Metadata

The demonstration data set 063c03.met [NTDB, 1999] metadata in NTDB (National Topographic Data Base) format.

```
 BEGIN          FILE
 BEGIN          TERRITORY_SECTION
 NTS            063C03
 DATA_SET_NAME  SWAN RIVER
 PROVINCE       MB (Manitoba)
 ZONE_NUMBER_1  14
 ZONE_NUMBER_2  -1
 PCT_OF_LAND    100
 SPECIAL_LIMITS N (Normal)
 END            TERRITORY_SECTION
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
 BEGIN          DATA_SET_SECTION
 EDITION_VERSIO 2.02
 NTDB_SPEC      3.1
 DATE_AVAILABLE 23-SEP-98
 FORMAT         IFF-BNDT-3.0
 FORMAT         CCOGIF-P3.0
 UNIT_CONTOURS  P (Foot(feet))
 CONTOUR_INTERV 25
 CONT_AUXILIARY -1
 DIMENSION      2D
 MAP_EDITION    2
 COMMENT
 END            DATA_SET_SECTION
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
 BEGIN          INTEGRATION_SECTION
 NORTH_EDGE     O (Yes)
 SOUTH_EDGE     I (Imperfect)
 EAST_EDGE      O (Yes)
 WEST_EDGE      O (Yes)
 END            INTEGRATION_SECTION
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!!!!!!
 BEGIN          POLYGON_SECTION
 NB_POLYGONS    2
 BEGIN          POLYGON
 ID_POLYGON     0001
 COORDINATES    362705 5762926 345545 5763428 328386 5763989 329344
5791789
 COORDINATES    346408 5791229 363471 5790728 362705 5762926
 ENTITIES       P 1-1849,1856-2047
 ENTITIES       L 1-1809, 1825-2047
 ENTITIES       S 1-1829, 1834-2047
 SOURCE_TYPE    REPRO (Reprographic material)
 SOURCE_NAME    063C03-ED2
```

```
 VALID_DATE     1976/00
 PLAN_ACCU_QUAL E (Estimated)
 PLAN_ACCURACY  50
 ALTI_ACCU_QUAL E (Estimated)
 ALTI_ACCURACY  5
 RES_PLAN_ACC   50
 ACTION         ACQ.COMP.SCAN (Complete acquisition of entities by
scanning)
 IMPACT_ACT_C   S (Systematic)
 IMPACT_ACT_P   N (No)
 POL_ED_VER     2.00
 COMMENT
 END            POLYGON
!
 BEGIN          POLYGON
 ID_POLYGON     0002
 COORDINATES    362705 5762926 345545 5763428 328386 5763989 329344
          5791789
 COORDINATES    346408 5791229 363471 5790728 362705 5762926
 ENTITIES       P 1850-1855
 SOURCE_TYPE    BDN+CARTE (Digital data base and paper map)
 SOURCE_NAME    BDTC-MB-1995
 VALID_DATE     1976/00
 PLAN_ACCU_QUAL I (Unknown / Non-applicable)
 PLAN_ACCURACY  -1
 ALTI_ACCU_QUAL I (Unknown / Non-applicable)
 ALTI_ACCURACY  -1
 RES_PLAN_ACC   -1
 ACTION         ACQ.TO (Acquisition of toponymy)
 IMPACT_ACT_C   S (Systematic)
 IMPACT_ACT_P   N (No)
 POL_ED_VER     2.00
 COMMENT
 END            POLYGON
!
 END            POLYGON_SECTION
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
 BEGIN          THEMES_SECTION
 NB_THEMES      14
 BEGIN          THEME
 NAME           AD (Designated Area)
 AVAIL_THEME    O (Yes)
 RESOLUTION     50000
 NB_KM          17
 NB_POINTS      2
 END            THEME
!
 BEGIN          THEME
 NAME           CH (Roads)
 AVAIL_THEME    O (Yes)
 RESOLUTION     50000
 NB_KM          207
 NB_POINTS      0
 END            THEME
```

```
!
 BEGIN          THEME
 NAME           CO (Manmade Features)
 AVAIL_THEME    O (Yes)
 RESOLUTION     50000
 NB_KM          36
 NB_POINTS      1737
 END            THEME
!
 BEGIN          THEME
 NAME           FO (Relief and Landforms)
 AVAIL_THEME    O (Yes)
 RESOLUTION     50000
 NB_KM          0
 NB_POINTS      1
 END            THEME
!
 BEGIN          THEME
 NAME           GE (General)
 AVAIL_THEME    O (Yes)
 RESOLUTION     50000
 NB_KM          124
 NB_POINTS      0
 END            THEME
!
 BEGIN          THEME
 NAME           HD (Hydrography)
 AVAIL_THEME    O (Yes)
 RESOLUTION     50000
 NB_KM          923
 NB_POINTS      1
 END            THEME
!
 BEGIN          THEME
 NAME           HP (Hypsography)
 AVAIL_THEME    O (Yes)
 RESOLUTION     50000
 NB_KM          1769
 NB_POINTS      0
 END            THEME
!
 BEGIN          THEME
 NAME           LA (Administrative Boundaries)
 AVAIL_THEME    N (No)
 RESOLUTION     -1
 NB_KM          0
 NB_POINTS      0
 END            THEME
!
 BEGIN          THEME
 NAME           RE (Power Network)
 AVAIL_THEME    O (Yes)
 RESOLUTION     50000
 NB_KM          38
```

```
 NB_POINTS      0
 END            THEME
!
 BEGIN          THEME
 NAME           RF (Rail Network)
 AVAIL_THEME    O (Yes)
 RESOLUTION     50000
 NB_KM          51
 NB_POINTS      0
 END            THEME
!
 BEGIN          THEME
 NAME           RR (Road Network)
 AVAIL_THEME    O (Yes)
 RESOLUTION     50000
 NB_KM          670
 NB_POINTS      0
 END            THEME
!
 BEGIN          THEME
 NAME           SS (Water saturated soils)
 AVAIL_THEME    O (Yes)
 RESOLUTION     50000
 NB_KM          34
 NB_POINTS      0
 END            THEME
!
 BEGIN          THEME
 NAME           TO (Toponymy)
 AVAIL_THEME    O (Yes)
 RESOLUTION     50000
 NB_KM          0
 NB_POINTS      43
 END            THEME
!
 BEGIN          THEME
 NAME           VE (Vegetation)
 AVAIL_THEME    O (Yes)
 RESOLUTION     50000
 NB_KM          1381
 NB_POINTS      0
 END            THEME
!
 END            THEMES_SECTION
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
 END            FILE
```

APPENDIX II Sample NTDB FGDC Metadata

The 063c03.txt metadata file in FGDC format obtained from the software developed by Ying Teng in the Faculty of Computer Science, University of New Brunswick using 063c03.met as input file.

```
Identification_Information
   Citation:
      Citation_Information:
         Originator: Canadian Government Agencies: NRCan
         Publication_Date: 1998-09-23
         Title: SWAN RIVER
         Edition:2.02
         Online_Linkage:

http://www.ccg.rncan.gc.ca/ext/html/english/products/ntdb/demontd5.html
   Description:
      Abstract:
       The National Topographic Data Base (NTDB) is a digital data base
      developed by Geomatics Canada. It covers the entire Canadian
      landmass and contains the features normally found on topographic
      maps at the scales of 1:50 000 and 1:250 000: hydrography,
      hypsography (contours), vegetation, the road network, roads, the
      rail network, the electric power network, designated area, land
      form, wetlands, and manmade features.

       In addition to allowing map production, the NTDB is highly useful
      to users of geographic information systems (GIS).

       This data set has been extracted from a complete NTDB
      Demonstration Data File at the scale of 1:50 000. It covers the
      area of SWAN RIVER ,MB Manitoba), Canada and bears the number
      063C03 based on the National Topographic System (NTS).

      Purpose:
```

```
    To provide Canadians with topographic and toponymic information
   to support the sustainable development of natural resources,
   environmental protection and the management of Canadian territory.
  Spatial_Domain:
    Bounding_Coordinates:
      West_Bounding_Coordinate: -101.50
      East_Bounding_Coordinate: -101.00
      North_Bounding_Coordinate: 52.25
      South_Bounding_Coordinate: 52.00
  Keywords:
   Theme:
    Theme_Keyword: AD(Designated Area)
    Theme_Keyword: CH(Roads)
    Theme_Keyword: CO(Manmade Features)
    Theme_Keyword: FO(Relief adn Landforms)
    Theme_Keyword: GE(General)
    Theme_Keyword: HD(Hydrography)
    Theme_Keyword: HP(Hypsography)
    Theme_Keyword: LA(Administrative Boundaries)
    Theme_Keyword: RE(Rail Network)
    Theme_Keyword: RR(Road Network)
    Theme_Keyword: SS(Water Saturated Soils)
    Theme_Keyword: TO(Toponymy)
    Theme_Keyword: VE(Vegetation)
   Place:
    Place_Keyword:SWAN RIVER
    Place_Keyword:MB (Manitoba)
    Place_Keyword: Canada
    Place_Keyword: North America
  Access_Constraints: None
  Use_Constraints: None
Data_Quality_Information:
  Positional_Accuracy:
   Horizontal_Positional_Accuracy:
    Horizontal_Positional_Accuracy_Report: 50
   Vertical_Positional_Accuracy:
    Vertical_Positional_Accuracy_Report: 5
  Spatial_Data_Organization_Information:
   Direct_Spatial_Reference_Method: Vector
  Spatial_Reference_Information:
   Horizontal_Coordinate_System_Definition:
    Planar:
     Map_Projection:
       Map_Projection_Name: UTM projection
    Geodetic_Model:
     Horizontal_Datum_Name: North American Datum of 1927
   Vertical_Coordinate_System_Definition:
    Altitude_System_Definition:
     Altitude_Datum_Name: Mean Sea Level
     Altitude_Distance_Units: 25.0
     Altitude_Encoding_Method: feet
Entity_and_Attribute_Information:
  Detailed_Description:
   Entity_Type:
```

```
          Entity_Type_Label: Polygon
Metadata_Reference_Information:
   Metadata_Standard_Name: FGDC Content Standards for Digital Geospatial
   Metadata
   Metadata_Standard_Version: CSDGM Version 2 - FGDC-STD-001-1998
```

APPENDIX III Sample NTDB FGDC XML Metadata

The 063c03.xml obtained from mp using 063c03.txt as input file.


```
<?xml version="1.0" encoding="ISO-8859-1"?>
<!DOCTYPE metadata
SYSTEM"http://www.fgdc.gov/metadata/fgdc-std-001-1998.dtd">
<metadata>
<idinfo>
<citation>
<citeinfo>
<origin>Canadian Government Agencies: NRCan</origin>
<pubdate>1998-09-23</pubdate>
<title>SWAN RIVER</title>
<edition>2.02</edition>
<onlink>http://www.ccg.rncan.gc.ca/ext/html/english/products/ntdb/demont
d5.htm
l</onlink>
</citeinfo>
</citation>
<descript>
<abstract>
The National Topographic Data Base (NTDB) is a digital data base
developed by
GeomaticsCanada. It covers the entire Canadian landmass and contains the
features normally found on topographic maps at the scales of 1:50 000
and
1:250 000: hydrography, hypsography(contours), vegetation, the road
network,
roads, the rail network, the electric powernetwork, designated area,
land
form, wetlands, and manmade features.

In addition to allowing map production, the NTDB is highly useful to
users of
geographic information systems (GIS).

This data set has been extracted from a complete NTDB Demonstration Data
File
at the scaleof 1:50 000. It covers the area of SWAN RIVER ,MB
(Manitoba),
Canada and bears the number 063C03 based on the National Topographic
System
(NTS).
</abstract>
<purpose>
To provide Canadians with topographic and toponymic information to
support the
sustainabledevelopment of natural resources, environmental protection
and the
management of Canadian territory.
</purpose>
```

</descript>
<spdom>
<bounding>
<westbc>-101.50</westbc>
<eastbc>-101.00</eastbc>
<northbc>52.25</northbc>
<southbc>52.00</southbc>
</bounding>
</spdom>
<keywords>
<theme>
<themekey>AD(Designated Area)</themekey>
<themekey>CH(Roads)</themekey>
<themekey>CO(Manmade Features)</themekey>
<themekey>FO(Relief adn Landforms)</themekey>
<themekey>GE(General)</themekey>
<themekey>HD(Hydrography)</themekey>
<themekey>HP(Hypsography)</themekey>
<themekey>LA(Administrative Boundaries)</themekey>
<themekey>RE(Rail Network)</themekey>
<themekey>RR(Road Network)</themekey>
<themekey>SS(Water Saturated Soils)</themekey>
<themekey>TO(Toponymy)</themekey>
<themekey>VE(Vegetation)</themekey>
</theme>
<place>
<placekey>SWAN RIVER</placekey>
<placekey>MB (Manitoba)</placekey>
<placekey>Canada</placekey>
<placekey>North America</placekey>
</place>
</keywords>
<accconst>None</accconst>
<useconst>None</useconst>
</idinfo>
<dataqual>
<posacc>
<horizpa>
<horizpar>50</horizpar>
</horizpa>
<vertacc>
<vertaccr>5</vertaccr>
</vertacc>
</posacc>
<spdoinfo>
<direct>Vector</direct>
</spdoinfo>
<spref>
<horizsys>
<planar>
<mapproj>
<mapprojn>UTM projection</mapprojn>
</mapproj>
</planar>

<geodetic>
<horizdn>North American Datum of 1927</horizdn>
</geodetic>
</horizsys>
<vertdef>
<altsys>
<altdatum>Mean Sea Level</altdatum>
<altunits>25.0</altunits>
<altenc>feet</altenc>
</altsys>
</vertdef>
</spref>
</dataqual>
<eainfo>
<detailed>
<enttyp>
<enttypl>Polygon</enttypl>
</enttyp>
</detailed>
</eainfo>
<metainfo>
<metstdn>FGDC Content Standards for Digital Geospatial
Metadata</metstdn>
<metstdv>FGDC-STD-001-1998</metstdv>
</metainfo>
</metadata>

APPENDIX IV Comparison of FGDC and ISO elements

ISO 19115 standard			FGDC CSDGM		
Name / Role name	**Short Name**	**Definition**	**Name / Role name**	**Short Name**	**Definition**
MD_Metadata	Metadata	Information about the metadata	Metadata	metadata	data about the content, quality, condition, and other characteristics of data
fileIdentifier	fileID	Unique identifier for this metadata file			
language	lang	Language used for documenting metadata			
characterSet	charSet	Full name of the ISO character coding standard used for the metadata set			
parentIdentifier	parID	Unique identifier of the parent metadata file			
hierarchyLevel	hierLev	Scope to which the metadata applies			
hierarchyLevelName	hierLevName	Name of the hierarchy level			
contact	contact	Party responsible for the metadata information	Metadata Contact	metc	the party responsible for the metadata information
date	date	Date that the metadata were created or last updated	Metadata Date	metd	the date that the metadata were created or last updated
metadataStandardName	mdStanName	Name of the metadata standard used	Metadata Standard Name	metstdn	the name of the metadata standard used to document the data set
metadataStandardVersion	mdStanVer	Version of the metadata standard used	Metadata Standard Version	metstdv	identification of the version of the metadata standard used to document the data set
Role name: spatialRepresentationInfo	spatRepInfo	Digital mechanism used to represent spatial information in the dataset	Spatial Data Organization Information	spdoinfo	the mechanism used to represent spatial information in the data set
Role name: referenceSystemInfo	refSysInfo	Description of the spatial and temporal reference systems used in the dataset	Spatial Reference Information	spref	the description of the reference frame for, and the means to encode, coordinates in the data set

ISO 19115 standard			FGDC CSDGM		
Name / Role name	**Short Name**	**Definition**	**Name / Role name**	**Short Name**	**Definition**
Role name: metadataExtensionInfo	metExtensInf	Information describing metadata extensions	Metadata Extensions	metextns	a reference to extended elements to the standard which may be defined by a metadata producer or a user community
Role name: identificationInfo	idInfo	Basic information about the resource for which the matadata is about	Identification Information:	idinfo	basic information about the data set
Role name: featureCollection	featColl	A collection of geographic data to which metadata applices			
Role name: featureCatalogueInfo	featCatInfo	Information about a catalogue which defines and describes the feature types, functions, attributes, and relationships, occurring in a set of geographic data			
Role name: distributionInfo	distInfo	Provides information about the distributor of and options for obtaining the dataset	Distribution Information	distinfo	-- information about the distributor of and options for obtaining the data set
Role name: dataQualityInfo	dataQualInfo	Provides overall assessment of quality of data	Data Quality Information	dataqual	a general assessment of the quality of the data set
Role name: portrayalCatalogueInfo	portCatInfo	Provides information about the catalogue of rules defined for the portrayal of data			
Role name: metadataConstraints	metConst	Provides restrictions on the acess and use of data			
Role name: applicationSchemaInfo	appSchInf	Provides informaiton about the conceptual schema of a dataset			
Role name: metadataMaintenance	metaMaint	Provides information about the frequency of metadata updates, and the scope of those updates			
Role name: propertyType	propTyp	Metadata is associated with the property of a feature			
Role name: featureType	featTyp	Metadata is associated with feature types			

ISO 19115 standard			FGDC CSDGM		
Name / Role name	**Short Name**	**Definition**	**Name / Role name**	**Short Name**	**Definition**
Role name: featureAttribute	featAtt	Metadata is associated with the characteristic(s) of a feature			
Role name: feature	feat	Metadata is associated with an abstraction of real world phenomena			
Role name: aggregateDataset	aggDS	Metadata is associated with multiple datasets			
MD_Identification	idInfo	Basic information about data	Identification Information:	idinfo	basic information about the data set.
language	lang	Language(s) used within the dataset			
characterSet	charSet	Full name of the ISO character coding standard used for the data			
abstract	abstract	Brief narrative summary of the content of the dataset	Abstract	abstract	a brief narrative summary of the data set.
purpose	purpose	Summary of the intentions with which the dataset was developed	Purpose	purpose	a summary of the intentions with which the data set was developed
supplementalInformation	suppInfo	Other descriptive information about the dataset. Example; Data Model	Supplemental Information	supplinf	other descriptive information about the data set
credit	credit	Recognition of those who contributed to the dataset	Data Set Credit	datacred	recognition of those who contributed to the data set.
status	status	Status of dataset	Status	status	the state of and maintenance information for the data set.
environment	envir	Description of the dataset in the producer's processing environment.	Native Data Set Environment	native	a description of the data set in the producer's processing environment
geographicBox	geoBox	Geographic areal domain of the dataset	Bounding Coordinates	bounding	the limits of coverage of a data set expressed by latitude and longitude
geographicDescription	geoDesc	Commonly used or well known name of a place, area or region which describes a spatial domain of the dataset			

ISO 19115 standard			FGDC CSDGM		
Name / Role name	**Short Name**	**Definition**	**Name / Role name**	**Short Name**	**Definition**
spatialResolution	spatRes	Factor which provides a general understanding of the density of spatial data in the dataset.			
category	category	Keywords, describing a subject of a dataset			
datasetCitation	dsCitation	Recommended reference to be used for the dataset	Citation Information	citeinfo	the recommended reference to be used for the data set.
datasetExtent	dsExt	Additional information about the bounding polygon, vertical, and temporal extent of the dataset			
datasetPointOfContact	dsPOC	Identification of, and means of communication with, person(s) and organisations(s) associated with the dataset	Point of Contact	ptcontac	contact information for an individual or organization that is knowledgeable about the data set
Role name: datasetMaintenance	dsMaint	Provides information about the scope and frequency of updating			
Role name: graphicOverview	graphOver	**x** Provides a graphic that illustrates the dataset			
Role name: datasetFormat	dsFormat	Provides a description of the form of the data to be distributed			
Role name: descriptiveKeywords	descKey	Provides keywords, their type, and reference source			
Role name: datasetSpecificUse	dsSpecUse	Provides basic information about specific application(s) for which the dataset has been or is being used by different users			

ISO 19115 standard			FGDC CSDGM		
Name / Role name	**Short Name**	**Definition**	**Name / Role name**	**Short Name**	**Definition**
Role name: datasetConstraints	dsConst	Provides information about constraints which the dataset must fall under			
MD_ImageIdentification	ImageID	Information required identifying a series of images.			
passSequenceIdentifier	passSeqID	Number that uniquely identifies the pass performed by a platform			
imageOrbitalIdentifier	imagOrbID	identifier for the orbital path of a platform and the row along an orbital path of a platform			
orbitNumber	orbNum	Numer of the orbit in which the image was taken			
MD_Keywords	Keywords	Keywords, their type and reference source	Keywords	keywords	words or phrases summarizing an aspect of the data set.
keyword	keyword	Common-use word(s) or phrase(s) used to describe the subject			
type	type	Method used to group similar keywords			
thesaurusName	thesaName	Name of the formally registered thesaurus or a similar authoritative source of keywords			
MD_DataConstraints	DataConst	Restrictions on the access and use of a dataset or metadata			
useLimitation	useLimit	Any limitation affecting the fitness for use of the dataset			
MD_LegalConstraints	LegalConst	Restrictions and legal prerequisites for accessing and using the dataset			
accessConstraints	accConst	Access constraints or limitations on obtaining the dataset	Access Constraints	accconst	restrictions and legal prerequisites for accessing the data set.
useConstraints	useConst	limitationson using the dataset	Use Constraints	useconst	restrictions and legal prerequisites for using the data set after access is granted

ISO 19115 standard			FGDC CSDGM		
Name / Role name	**Short Name**	**Definition**	**Name / Role name**	**Short Name**	**Definition**
otherConstraints	othConst	Other restrictions and legal prerequisites			
MD_MaintenanceInformation	MaintInfo	Information about the scope and frequency of updating			
maintenanceAndUpdateFreq	maintUpFreq	Frequency with which changes and additions are made to the dataset after the initial dataset is completed.	Maintenance and Update Frequency	update	the frequency with which changes and additions are made to the data set after the initial data set is completed.**v**
otherMaintenancePeriod	othMaintPer	Maintenance period other than those defined			
updateScope	upScp	Scope at which changes are applied			
updateScopeDescription	upScpDesc	Additional information about the range or extent of the dtataset			
MD_SpatialRepresentation	SpatRep	Digital mechanism used to represent spatial information	Spatial Data Organization Information	spdoinfo	-- the mechanism used to represent spatial information in the data set.
representationType	repType	Method used to represent geographic information			
MD_ImageSpatialRepresentation	ImgSpatRes	Relevant data about the image used to represent geographic information			
imageIdentifier	imageID	Unique discriptor for an image within a dataset series			
imageType	imageType	Identifies the general kind of image represented by the data			
meanGroundSampleDistance	meanGrSampDst	Geometric mean of the across and along scan centre-to-centre distance between continuous ground samples in metres			

ISO 19115 standard			FGDC CSDGM		
Name / Role name	**Short Name**	**Definition**	**Name / Role name**	**Short Name**	**Definition**
groundToImageCoefficientAvailability	grToImgCoAvl	Code which indicates whether or not Ground-to-Image coefficients are available and contained within the product data			
MD_RasterSpatialRepresentation	RastSpatRep	Types and numbers of raster spatial objects in the dataset	Raster Object Information	rastinfo	the types and numbers of raster spatial objects in the data set
cellType	cellType	Raster spatial objects used to locate zero-, two-, or three-dimensional locations in the dataset	Raster Object Type	rasttype	raster spatial objects used to locate zero-, two-, or three-dimensional locations in the data set
cellOrigin	cellOrig	Location of pixel 1,1 (example NW corner)			
rows	rows	Maximum number of raster objects along the ordinate (y) axis	Row Count	rowcount	the maximum number of raster objects along the ordinate (y) axis
columns	cols	Maximum number of raster objects along the abscissa (x) axis	Column Count	colcount	the maximum number of raster objects along the abscissa (x) axis.
verticals	verts	Maximum number of raster objects along the vertical (z) axis	Vertical Count	vrtcount	the maximum number of raster objects along the vertical (z) axis
ScanResolution	scanRes	Units used to express data density along the axes			
groundSpacing	grSpac	Unit of measurement used to describe the distance			
MD_GroundSpacingResolution					
Role name: cellDomain	cellDom	Provides information about the domain of a raster cell			
MD_VectorSpatialRepresentation	VectSpatRep	Information about the vector spatial objects in the dataset	Point and Vector Object Information	ptvctinf	the types and numbers of vector or nongridded point spatial objects in the data set.
geometricObjectType	geometObjTyp	Name of point and vector spatial objects used to locate zero-, one-, and two-dimensional spatial locations in the dataset	SDTS Point and Vector Object Type	sdtstype	name of point and vector spatial objects used to locate zero-, one-, and two-dimensional spatial locations in the data set
geometricObjectCount	geometObjCnt	Total number of the point or vector object type occurring in the dataset	Point and Vector Object Count	ptvctcnt	the total number of the point or vector object type occurring in the data set

ISO 19115 standard			FGDC CSDGM		
Name / Role name	**Short Name**	**Definition**	**Name / Role name**	**Short Name**	**Definition**
topologyLevel	topLevel	Code which identifies the degree of complexity of the spatial relationships			
MD_CellValue Domain	CellValDom	Information about the domain of the raster cell			
toneGradation	toneGrad	Number of colours present in the image			
bitsPerBand	bitsPB	Maximum number of significant bits for the value in each band of each pixel without compression			
cellAttributeDescription	cellAttDesc	Description of the attribute described by the measurement value			
cellUnit	cellUnit	Units of the cell attribute			
MD_GroundSpacing	GrSpac	Geometric mean of the distance between continuous ground samples			
spacing	spac	Center to center distance between continuous samples			
unit	unit	Unit of measurement used to depict ground spacing			
MD_GroundSpacingResolution	GrSpacRes	The distance represented by a pixel in ground space units in up to 3 dimensions			
xSpacing	xSpac	The distance represented by a pixel in the x direction on the ground			
ySpacing	ySpac	The distance represented by a pixel in the y direction on the ground			
zSpacing	zSpac	The distance represented by a pixel in a direction perpindicular to the x-y plane			
MD_ImageSuitabilityDescription	ImagSuitDsc	Information about an image's suitability for use			

ISO 19115 standard			FGDC CSDGM		
Name / Role name	**Short Name**	**Definition**	**Name / Role name**	**Short Name**	**Definition**
illuminationElevationAngle	illElevAng	Illumination elevation measured in degrees clockwise from the target plane at intersection of the optical line of sight with the earth's surface			
illuminationAzimuthAngle	illAziAng	Illumination azimuth measured in degrees clockwise from true north at the time the image is taken			
imageOrientationAngle	imagOrieAng	Angle from the first row of the image to true North in degrees, clockwise			
imagingCondition	imagCond	Code which indicates conditions which affect the quality of the image			
imageQualityRatingSystem	imagQualRatSys	Rating system on which the Image Quality Code is based			
imageQualityCode	imagQualcode	Specifies the image quality			
cloudCoverPercentage	cloudCovPer	Area of the dataset obscured by clouds, expressed as a percentage of the spatial extent			
preProcessingTypeCode	preprcTypCde	Image distributor's code that identifies the level of radiometric and geometric processing applied against the image			

ISO 19115 standard			FGDC CSDGM		
Name / Role name	**Short Name**	**Definition**	**Name / Role name**	**Short Name**	**Definition**
compressionGenerationQuantity	compGenQuan	Counts the number of lossy compression cycles performed on the image			
triangulationIndicator	triID	Code which indicates whether or not triangulation has been performed upon the image			
radiometricDataAvailability	radDatAvail	Code which indicates whether or not Standard Radiometric Product data is available			
ESDAvailability	ESDAvail	Indicates whether or not Image Exploitation Support Data (ESD) is available such as position and attitude information			
MD_PixelResolution	PixRes	Average unit of information in a grid cell			
pixelsPerUnit	pixPerUnit	Number of pixels contained in one unit of measurement			
unit	unit	Units of measure used to describe pixels \per unit			
MD_ScanResolution	ScanRes	Units used to express data density along the axes			
xResolution	xRes	Units used to express data density along the x axis			
yResolution	yRes	Units used to express data density along the y axis			
zResolution	zRes	Units used to express data density along the z axis			
MD_SensorParameters	SenPara	Identifies the parameters defining the sensor			
focalLength	focLen	Focal length of the lens in millimetres			

ISO 19115 standard			FGDC CSDGM		
Name / Role name	**Short Name**	**Definition**	**Name / Role name**	**Short Name**	**Definition**
obliquityAngle	oblAng	Angle off vertical of image in degrees			
imageSensorTime	imgSenTime	The precise time at which the image was captured in the sensor's time system			
sensorCategory	senCat	Identifies the specific category of imagery			
sensorMode	senMode	Identifies the sensor mode used in capturing the image			
spectralProperties	spectProp	Electromagnetic spectrum sensitivity of sensor			
fieldOfView	fieldOView	Area of measurement of sensor			
orientationOnPlatform	orieOnPlat	Orientation of instrument relative to platform			
operationMode	opMode	Sensor status Examples: launch, survival, initialization, safe, diagnostic, standby, crosstrack, biaxial, solar calibration			
Role name: band	band	Set of wavelengths that the sensor operates in			
MD_SensorBand	SenBand	Set of adjacent wavelengths in the electro-magnetic spectrum with a common characteristic, such as the visible band			
sequenceIdentifier	seqId	Number that uniquely identifies instances of bands of wavelengths on which a sensor operates			
highWavelength	hiWavelen	Highest wavelength that the sensor is capable of collecting within a designated band in metres			

ISO 19115 standard			FGDC CSDGM		
Name / Role name	**Short Name**	**Definition**	**Name / Role name**	**Short Name**	**Definition**
lowWavelength	lowWavelen	Lowest wavelength that the sensor is capable of collecting within a designated band in metres			
cameraCalibrationInfoAvailability	camCalInfAvl	Code which ndicates whether or not constants are available which allow for camera calibration corrections			
filmDistortionInfoAvailability	filmDistrtInfAvl	Code which indicates whether or not Calibration Reseau information is available			
lensDistortionInfoAvailability	lensDistrtInfAvl	Code which indicates whether or not lens aberration correction information is available			
MD_Distribution	Dist	Information about the distributor of and options for obtaining the dataset Use obligation/condition from referencing object	Distribution Information	distinfo	information about the distributor of and options for obtaining the data set
Role name: distributionFormat	distFormat	Provides a description of the form of the data to be distributed			
Role name: distributor	distributor	Provides information about the distributor	Distributor	distrib	the party from whom the data set may be obtained
Role name: transferOptions	distribTrnsOps	Provides information about technical means and media by which a dataset is obtained from the distributor			
MD_DigitalTransferOptions	DigTransOpts	Technical means and media by which a dataset is obtained from the distributor	Digital Form	digform	the description of options for obtaining the data set on computer-compatible media
unitsOfDistribution	unitsODist	Tiles, layers, geographic areas, etc., in which data is available			
transferSize	transSize	Estimated size of the transferred dataset in megabytes. The transfer size is > 0.0	Transfer Size	transize	the size, or estimated size, of the transferred data set in megabytes
onLine	onLine	Information about online sources from which the dataset can be obtained	Online Option	onlinopt	information required to directly obtain the data set electronically.

ISO 19115 standard			FGDC CSDGM		
Name / Role name	**Short Name**	**Definition**	**Name / Role name**	**Short Name**	**Definition**
Role name: offLine	offLine	Information about offline sources from which the dataset can be obtained	Offline Option	offoptn	information about media-specific options for receiving the data set.
MD_Distributor	Distributor	Information about the distributorUse obligation/condition from referencing object	Distributor	distrib	the party from whom the data set may be obtained
Role name: distributionOrderProcess	distOrdProc	Provides information about how the dataset may be obtained, and related instructions and fee information			
distributorContact	distCont	Party from whom the dataset may be obtained			
Role name: distributorFormat	distFormat	Provides information about the Format in which the dataset may be obtained			
MD_Format	Format	Description of the form of the data to be distributed			
name	name	Name of the data transfer format(s) offered by the distributor for an available dataset.	Format Name	formname	the name of the data transfer format
version	verNum	Version number of the format	Format Version Number	formvern	version number of the format
amendmentNumber	amendNum	Amendment number of the format version			
specification	spec	Name of a subset, profile, or product specification of the format	Format Specification	formspec	name of a subset, profile, or product specification of the format.
fileDecompressionTechnique	fileDecmTech	Recommendations of algorithms or processes that can be applied to read or expand datasets to which data compression techniques have been applied	File Decompression Technique	filedec	recommendations of algorithms or processes (including means of obtaining these algorithms or processes) that can be applied to read or expand data sets to which data compression techniques have been applied.
Role name: distributorFormat	distFormat	Provides information about the distributor's Format			
MD_Medium	Medium	Information about the media on which the data can be distributed			

ISO 19115 standard			FGDC CSDGM		
Name / Role name	**Short Name**	**Definition**	**Name / Role name**	**Short Name**	**Definition**
name	name	Name of the media on which the dataset can be received	Offline Media	offmedia	name of the media on which the data set can be received.
density	density	Density in which the dataset can be recorded	Recording Density	recden	the density in which the data set can be recorded
densityUnits	densityUn	Units of measure for the recording density	Recording Density Units	recdenu	the units of measure for the recording density
volumes	vols	Number of items in the media identified			
mediaFormat	medFormat	Options available or method used to write the dataset to the medium	Recording Format	recfmt	the options available or method used to write the data set to the medium.
compatibility	compat	Description of other limitations or requirements for using the medium	Compatibility Information	compat	description of other limitations or requirements for using the medium.
MD_StandardOrderProcess	StanOrdPrc	Common ways in which the dataset may be obtained or received, and related instructions and fee information	Standard Order Process.	stdorder	the common ways in which the data set may be obtained or received, and related instructions and fee information
fees	fees	Fees and terms for retrieving the dataset. Include monetary units	Fees	fees	the fees and terms for retrieving the data set
plannedAvailableDateTime	plnAvlDatTim	Date and time when the dataset will be available	Available Time Period	availabl	the time period when the data set will be available from the distributor
orderingInstructions	ordInstr	General instructions , terms and services provided by the distributor when ordering the dataset	Ordering Instructions	ordering	general instructions and advice about, and special terms and services provided for, the data set by the distributor
turnaround	turnaround	Typical turnaround time for the filling of an order	Turnaround	turnarnd	typical turnaround time for the filling of an order
EX_GeographicExtent	GeoExt	Geographic area of the dataset			
extentType	extType	Identifies whether the bounding polygon encompasses an area covered by the data or an area where data is not present			
EX_BoundingPolygon	BoundPoly	Boundary enclosing the dataset expressed as the closed set of (x,y) coordinates of the polygon			
polygon	poly	Sets of points in a particular coordinate reference system.			

ISO 19115 standard			FGDC CSDGM		
Name / Role name	**Short Name**	**Definition**	**Name / Role name**	**Short Name**	**Definition**
EX_GeographicBoundingBox	GeoBndBox	Geographic area of the entire dataset referenced to WGS 84	Bounding Coordinates	bounding	the limits of coverage of a data set expressed by latitude and longitude values
westBoundLongitude	westBL	Western-most coordinate of the limit of the dataset extent expressed in longitude, in decimal degrees	West Bounding Coordinate	westbc	western-most coordinate of the limit of coverage expressed in longitude
eastBoundLongitude	eastBL	Eastern-most coordinate of the limit of the dataset extent expressed in longitude, in decimal degrees	East Bounding Coordinate	eastbc	eastern-most coordinate of the limit of coverage expressed in longitude
southBoundLatitude	southBL	Southern-most coordinate of the limit of the dataset extent expressed in latitude, in decimal degrees	South Bounding Coordinate	southbc	southern-most coordinate of the limit of coverage expressed in latitude
northBoundLatitude	northBL	Northern-most coordinate of the limit of the dataset extent expressed in latitude, in decimal degrees	North Bounding Coordinate	northbc	northern-most coordinate of the limit of coverage expressed in latitude
CI_Citation	Citation	Standardized resource reference	Citation	citation	information to be used to reference the data set.
title	title	Name by which the cited information is known	Title	title	the name by which the data set is known
alternateTitle	altTitle	Short name or other language name by which the cited information is known.			
date	date	Reference date for the cited information	Publication Date	pubdate	the date when the data set is published or otherwise made available for release
dateType	dateType	Event used for reference data			
edition	edition	Version of the dataset	Edition	edition	the version of the title
editionDate	edDate	Date of the edition			
identifier	citID	Unique identifier for the data referenced by the metadata			
identifierType	idType	Reference form of the unique identifier			
presentationForm	presForm	Mode in which the data is represented			

ISO 19115 standard			FGDC CSDGM		
Name / Role name	**Short Name**	**Definition**	**Name / Role name**	**Short Name**	**Definition**
seriesName	serName	Name of the series of which the dataset is a part	Series Name	sername	the name of the series publication of which the data set is a part
issueIdentification	issID	Information identifying the issue of the series publication of which the dataset is a part			
otherCitationDetails	otherCitDet	Other information required to complete the citation	Other Citation Details	othercit	other information required to complete the citation
citedResponsibleParty	citRespParty	Name and position information for an individual or organisation that is responsible for the resource			
collectiveTitle	collTitle	Common title with holdings note			
page	page	Details on which pages of the periodical the article was published			
ISBN	ISBN	International Standard Book Number			
ISSN	ISSN	International Standard Serial Number			
CI_ResponsibleParty	RespParty	Identification of, and means of communication with, person(s) and organisations associated with the dataset	Contact Person Primary	cntperp	the person, and the affiliation of the person, associated with the data
mandatoryPartyInfo	mandPartyInfo	Individual, organisation, or position that is knowledgable about the dataset			
individualName	rpIndName	Name of the responsible person- SURNAME given name, title separated by a delimiters	Contact Person	cntper	the name of the individual to which the contact type applies
organisationName	rpOrgName	Name of the responsible organisation	Contact Organization	cntorg	the name of the organization to which the contact type applies
positionName	rpPosName	Role or position of the responsible person	Contact Position	cntpos	the title of individual
responsibility	resp	Function performed by the responsible party			

ISO 19115 standard			FGDC CSDGM		
Name / Role name	**Short Name**	**Definition**	**Name / Role name**	**Short Name**	**Definition**
CI_Address	Address	Location of the responsible individual or organisation	Contact Address	cntaddr	the address for the organization or individual
deliveryPoint	postAdd	Address line for the physical address	Address	address	an address line for the address
city	city	City of the physical address	City	city	the city of the address
administrativeArea	adminArea	State, province of the physical address	State or Province	state	the state or province of the address
postalCode	postCode	ZIP or other postal code	Postal Code	postal	the ZIP or other postal code of the address
country	country	Country of the physical address	Country	country	the country of the address
electronicMailAddress	electMailAdd	Address of the electronic mailbox of the responsible organisation or individual	Contact Electronic Mail Address	cntemail	the address of the electronic mailbox of the organization or individual.
CI_Contact	Contact	Information required enabling contact with the responsible person and/or organisation	Contact Information	cntinfo	Identity of, and means to communicate with, person(s) and organization(s) associated with the data set.
hoursOfService	hrsOfServ	Time period (including time zone) when individuals can contact the organisation or individual	Hours of Service	hours	time period when individuals can speak to the organization or individual
contactInstructions	contInstr	Supplemental instructions on how or when to contact the individual or organisation	Contact Instructions	cntinst	supplemental instructions on how or when to contact the individual or organization
phone	phone	Telephone numbers at which the organisation or individual may be contacted	Contact Voice Telephone.	cntvoice	the telephone number by which individuals can speak to the organization or individual
address	address	Physical and email address at which the organisation or individual may be contacted	Contact Address	cntaddr	the address for the organization or individual
onlineResource	onlineRes	Online information that can be used to contact the individual or organisation			
CI_Mandatory Party	MandParty	Individual, organisation, or position that is knowledgeable about the resource			

ISO 19115 standard			FGDC CSDGM		
Name / Role name	**Short Name**	**Definition**	**Name / Role name**	**Short Name**	**Definition**
individualName	mpIndName	Name of the responsible person- SURNAME given name, title separated by a delimiters	Contact Person	cntper	the name of the individual to which the contact type applies
organisationName	mpOrgName	Name of the responsible organisation	Contact Organization	cntorg	the name of the organization to which the contact type applies
positionName	mpPosName	Role or position of the responsible person	Contact Position	cntpos	the title of individual
CI_OnlineResource	OnlinRes	Information about online sources from which the dataset, can be obtained.			
linkage	linkage	Method, source, or location for online access	Online Linkage	onlink	the name of an online computer resource that contains the data set
functionCode	functCode	Function performed by the resource			
protocol	protocol	Connection protocol to be used			
applicationProfile	appProfile	Name of an application profile that can be used with the resource			
name	name	Name of the resource			
description	desc	Description of what the resource is/does			
CI_Telephone	Telephone	Telephone numbers for contacting the responsible individual or organisation			
voice	voice	Telephone number by which individuals can speak to the responsible organisation or individual	Contact Voice Telephone	cntvoice	the telephone number by which individuals can speak to the organization or individual
facsimile	fax	Telephone number of a facsimile machine for the responsible organisation or individual	Contact Facsimile Telephone	cntfax	the telephone number of a facsimile machine of the organization or individual
other	other	Telephone number for contacting the responsible individual or organisation			
otherType	othType	Description of telephone number provided in "other" phone element			

Appendix V Suggested formal grammar representation for B.2 Metadata entity set information from [ISO/TC 211,1999]

```
(B.2)  <MD_Metadata> ::= “<Metadata>”
            [ “<fileID>” <string_literal> “</fileID>” ]
            [ “<lang>” <LanguageCode> “</lang>” ]
            [ “<charSet>” <CharacterSetCode> “</charSet>” ]
            [ “<parID>” <string_literal> “</parID>” ]
            [ {“<hierLev>” “dataset” <MD_Scope> “</hierLev>”} |
                  {“<hierLev>” “notdataset”
                        “<hierLevName>” <string_literal> “</hierLevName>”
                  “</hierLev>”} ]
            [ “<contact>” <CI_ResponsibleParty> “</contact>” ]
            [ “<date>” <Date> “</date>” ]
            [ “<mdStanName>” <string_literal> “</mdStanName>” ]
            [ “<mdStanVer>” <string_literal> “</mdStanVer>” ]
            { “<spatRepInfo>” <MD_SpatialRepresentation> “</spatRepInfo>” }*
            { “<refSysInfo>” <RS_ReferenceSystem> “</refSysInfo>” }*
            { “<metExtensInf>” <MD_MetadataExtensionInformation>
                              “</metExtensInf>”  }*
            { “<idInfo>” <MD_Identification> “</idInfo>” }+
            { “<featColl>” <FT_FeatureCollection> “<featColl>” }+
            { “<featCatInfo>” <FC_FeatureCatalogueDescription> “</featCatInfo>” }*
            [ “<distInfo>” <MD_Distribution> “</distInfo>” ]
            { “<dataQualInfo>” <DQ_DataQualityInformation> “</dataQualInfo>” }*
            { “<portCatInfo>” <MD_PortrayalcatalogueRef> “</portCatInfo>” }*
            { “<metConst>” <MD_DataConstraints> “</metConst>” }*
            { “<appSchInf>” <MD_ApplicationSchemaInfo> “<appSchInf>” }*
            [ “<metaMaint>” <MD_MaintenanceInformation> “</metaMaint>” ]
            { “<PropTyp>” <GF_PropertyType> “</PropTyp>” }*
            { “<featTyp>” <GF_FeatureType> “</featTyp>” }*
            { “<featAtt>” <FT_FeatureAttribute> “</featAtt>” }*
            { “<feat>” <FT_Feature> “</feat>” }*
            { “<aggDS>” <DS_Aggregate> “</aggDS>” }+
            “</Metadata>”
```

Appendix VI Suggested formal grammar representation for B.3

Identification information from [ISO/TC 211,1999]

```
(B.3)  <MD_Identification>  ::= "<ID>"
        { "<lang>" <LanguageCode> "</lang>" }+
        [ "<charSet>" <CharacterCodeSet> "</charSet>" ]
        "<abstract>" <string_literal> "</abstract>"
        [ "<purpose>" <string_literal> "</purpose>" ]
        [ "<suppInfo>" <string_literal> "</suppInfo>" ]
        [ "<credit>" <string_literal> "</credit>" ]
        [ "<status>" <MD_ProgressCode> "</status>" ]
        [ "<envir>" <string_literal> "</envir>" ]
        { { "<geoBox>" <EX_GeographicBoundingBox> "</geoBox>" } |
                { "<geoDesc>" <SI_LocationInstance> "</geoDesc>" } }
        { "<spatRes>" <string_literal> "</spatRes>" }*
        { "<category>" <MD_Category> "</category>" }+
        { "<dsCitation>" <CI_Citation> "</dsCitation>" }+
        { "<dsExt>" <EX_Extent> "</dsExt>" }*
        { "<dsPOC>" <CI_ResponsibleParty> "</dsPOC>" }*
        { "<dsMaint>" <MD_MaintenanceInformation> "</dsMaint>" }*
        { "<graphOver>" <MD_BrowseGraphic> "</graphOver>" }*
        { "<dsFormat>" <MD_Format> "</dsFormat>" }*
        { "<descKey>" <MD_Keywords> "</descKey>" }*
        { "<dsSpecUse>" <MD_Use> "</dsSpecUse>" }*
        [ "<dsConst>" <MD_DataConstraints> "</dsConst>" ]
        [ "<ImageID>" <MD_ImageIdentification>
                "<passSeqID>" <integer_literal> "</passSeqID>"
                "<imagOrbID>" <string_Literal> "</imagOrbID>"
                "<orbNum>" <integer_literal> "</orbNum>"
                "</ImageID>" ]
        "</ID>"
```

```
(B.3.1) <MD_BrowseGraphic> ::= "<BrowGraph>"
        "<fileName>" <string_literal> "</fileName>"
        [ "<fileDesc>" <string_literal> "</fileDesc>" ]
        [ "<fileType>" <string_literal> "</fileType>" ]
        "</BrowGraph>"
```

```
(B.3.2) <MD_Keywords> ::= "<Keywords>"
        { "<keyword>" <string_literal> "</keyword>" }+
        [ "<type>" <MD_KeywordType> "</type>"]
        [ "<thesaName>" <string_literal> "</thesaName>" ]
        "</Keywords>"
```

```
(B.3.4) <MD_Use> ::= "<Use>"
        "<specUse>" <string_literal> "</specUse>"
        [ "<useDatTim>" <DateTime> "</useDatTim>" ]
        [ "<usrDefLims>" <string_literal> "</usrDefLims>" ]
        { "<usrContInfo>" <CI_ResponsibleParty> "</usrContInfo>" }*
        "</Use>"
```

APPENDIX VII XML DTD for ISO metadata files

```
<!-- edited with XML Spy v3.0 (http://www.xmlspy.com) by Ying Teng
(University of New Brunswick) -->
<!-- ISO Metadata DTD 3.0.1 1999-11-30  -->
<!-- This is the Document Type Declaration for formal metadata  -->
<!-- conforming to the ISO CD 19115. This DTD corresponds to   -->
<!-- the November, 1999 version of the standard, ISO CD 19115. -->
<!-- This file is the XML DTD.       -->
<!-- Entity sets:               -->
<!--     Scalar values (meaning the values of elements that are not  -->
<!--     compound) are here declared #PCDATA to allow parsers to  -->
<!--     recognize and support entities representing special characters
-->
<!--     such as the degree symbol, less, and greater.      -->
<!-- Element ordering:          -->
<!--     Generally the order of elements is now significant. XML makes
-->
<!--     it difficult to write a DTD that allows elements to be in any
-->
<!--     order.  -->
<!-- Authors:  -->
<!--   Ying Teng                -->
<!--=================================================================-->

<!ELEMENT metadata (fileID?, parID?, hierLev?, HierLevName?, contact?,
date?, mdStanName?, mdStanVer?, spatRepInfo*, refSysInfo*,
metExtensInfo*, idInfo+, featColl+, featCatInfo*, distInfo?,
dataQualInfo*, portCatInfo*, metConst*, appSchInf*, metaMaint?,
propTyp*, featTyp*, featAtt*, feat*, aggDS+)>
<!ELEMENT portCatInfo (PortCatRef)*>
<!ELEMENT idInfo (ID)>
<!ELEMENT ID (abstract, purpose?, suppInfo?, credit?, status?, envir?,
geoBox*, geoDesc*, spatRes*, category+, dsCitation, dsExt*, dsPOC*,
dsMaint*, graphOver*, dsFormat*, descKey*, dsSpecUse*, dsConst*,
ImageID?)>
<!ELEMENT geoBox (GeoBndBox)*>
<!ELEMENT dsFormat (Format)*>
<!ELEMENT ImageID (passSeqID, imagOrbID, orbNum)>
<!ELEMENT graphOver (BrowGraph)>
<!ELEMENT BrowGraph (fileName, fileDesc?, fileType?)>
<!ELEMENT descKey (Keywords)>
<!ELEMENT Keywords (keyword+, type?, thesaName?)>
<!ELEMENT dsSpecUse (Use)>
<!ELEMENT Use (specUse+, useDatTim?, usrDefLims?, usrContInfo*)>
<!ELEMENT dsConst (DataConst*)>
<!ELEMENT metConst (DataConst)*>
<!ELEMENT DataConst (useLimit*, LegalConst*, accConst?, useConst?,
othConst*, SecInfo*, class, userNote?, classSys, handDesc?,
otherUserDef?)>
<!ELEMENT accConst (DataConst)?>
<!ELEMENT useConst (DataConst)?>
<!ELEMENT metaMaint (MaintInfo)?>
<!ELEMENT dsMaint (MaintInfo)>
<!ELEMENT MaintInfo (maintUpFreq, othMaintPer?, upScp?, upScpDesc?)>
<!ELEMENT ScpDesc (attribs, feats, featInsts, attribInsts, featColl,
```

```
other)>
<!ELEMENT spatRepInfo (SpatRep)>
<!ELEMENT SpatRep (repType*, ImgSpatRes*, grToImgCoAvl?, params?,
imgSuitDesc?, RasterSpatRep*, VectSpatRep*)>
<!ELEMENT ImgSpatRes (imageID?, imageType, meanGrSampDst?)*>
<!ELEMENT RasterSpatRep (cellType, cellOrig?, rows?, cols?, verts?,
scanRes?, grSpac?, cellDom)*>
<!ELEMENT VectSpatRep (geometObjTyp, geometObjCnt?, topLevel?)*>
<!ELEMENT cellDom (CellValDom)>
<!ELEMENT CellValDom (toneGrad?, bitsPB?, cellAttDesc, cellUnit)>
<!ELEMENT GrSpac (spac, unit)>
<!ELEMENT grSpac (GrSpac)>
<!ELEMENT meanGrSampDst (Grspac)>
<!ELEMENT Grspac (xSpac, ySpac, zSpac)>
<!ELEMENT xSpac (GrSpac)>
<!ELEMENT ySpac (GrSpac)>
<!ELEMENT zSpac (GrSpac)>
<!ELEMENT imgSuitDesc (ImagSuitDsc)>
<!ELEMENT ImagSuitDsc (illElevang?, illAziAng?, imgOrieAng?, imagCond?,
imgQualRatsys?, imagQualcode?, cloudCovPer?, prePrcTypCde?,
compGenQuan?, trilD?, radDatAvail?, ESDAvail?)>
<!ELEMENT scanRes (xRes, yRes, xRes)>
<!ELEMENT xRes (PixRes)>
<!ELEMENT yRes (PixRes)>
<!ELEMENT zRes (PixRes*)>
<!ELEMENT PixRes (pixPerUnit, unit)>
<!ELEMENT params (SenPara)>
<!ELEMENT SenPara (focLen?, oblAng?, imgSenTime?, senCat?, senMode?,
spectProp?, fieldOview?, orieOnPlat?, opMode?, band*)>
<!ELEMENT band (SenBand*)>
<!ELEMENT SenBand (seqId?, hiWavelen?, lowWavelen?, camCalInfAvl?,
filmDistrtInfAvl?, lensDistrtInfAvl)>
<!ELEMENT refSysInfo (Refsys)>
<!ELEMENT Refsys (Refname, domOValid*, TMRefSys, SISPatRefSysGeoID,
CRS)>
<!ELEMENT Refname (Identifier)>
<!ELEMENT Identifier (identifier, citation)>
<!ELEMENT citation (Citation)>
<!ELEMENT SISPatRefSysGeoID (theme, overOwner)>
<!ELEMENT CRS (kindCode, remarks?)>
<!ELEMENT featCatInfo (FeatCatDesc*)>
<!ELEMENT FeatCatDesc (compcode, langCode+, incWithDS, featType?,
featcatCit+)>
<!ELEMENT featcatCit (Citation)+>
<!ELEMENT PortCatRef (portCatcit+)>
<!ELEMENT portCatcit (Citation)+>
<!ELEMENT distInfo (Dist)>
<!ELEMENT Dist (distFormat+, distributor*, distribTrnOps*)>
<!ELEMENT distribTrnOps (DigTransOpts)>
<!ELEMENT DigTransOpts (unitsODist?, transSize?, onLine*, offline?)>
<!ELEMENT distributor (Distributor)>
<!ELEMENT Distributor (distOrdProc*, distCont, distFormat+)>
<!ELEMENT distFormat (Format)>
<!ELEMENT Format (name, verNum, amendNum?, spec?, filDecmTech?,
```

```
distFormat*)>
<!ELEMENT offLine (Medium)>
<!ELEMENT Medium (name?, density*, densityUn?, vols?, medFormat*,
compat?)>
<!ELEMENT distOrdProc (StanOrdPrc)>
<!ELEMENT StanOrdPrc (fees?, plnAvldatTim?, ordInstr?, turnaround)>
<!ELEMENT metExtensInfo (MetExtnsInf)>
<!ELEMENT MetExtnsInf (extnsEleInf*, extnsOnliRes)>
<!ELEMENT extnsEleInf (ExtendEleInf)>
<!ELEMENT ExtendEleInf (name, identifier, defin, oblig, datType, domVal,
maxOcc, parEnt+, rule*, rationale*, source*)>
<!ELEMENT appSchInf (AppSchInfo)>
<!ELEMENT AppSchInfo (Appname, schLang, constrlang, schAsc, graFilTyp,
swDevFile, swDevFormat, featCatSup)>
<!ELEMENT Appname (Citation)>
<!ELEMENT featCatSup (SpatAttSup)>
<!ELEMENT SpatAttSup (featTypList)>
<!ELEMENT featTypList (FeatTypList)>
<!ELEMENT FeatTypList (spatObj, sparSchName)>
<!ELEMENT dsExt (Extent)>
<!ELEMENT domOValid (Extent)>
<!ELEMENT Extent (desc?, ele*)>
<!ELEMENT GeoExt (extType?, BoundPoly*, GeoBndBox*, SetLocInst)>
<!ELEMENT BoundPoly (poly+)*>
<!ELEMENT GeoBndBox (westBL, eastBL, southBL, northBL)*>
<!ELEMENT SetLocInst (elements)>
<!ELEMENT ele (GeoExt, TempExt, VertExt)>
<!ELEMENT TempExt (extent, SpatTempExt)>
<!ELEMENT SpatTempExt (TempExt | GeoExt)>
<!ELEMENT VertExt (minVal, maxVal, uOfMeas, vetDat)>
<!ELEMENT Citation (title, altTitle*, date, dateType?, edition?,
edDate?, citID*, idType*, presForm*, serName?, issID?, otherCitDet?,
citRespParty*, collTitle?, page?, ISBN, ISSN)>
<!ELEMENT contact (RespParty)>
<!ELEMENT dsCitation (Citation)>
<!ELEMENT dsPOC (RespParty)>
<!ELEMENT usrContInfo (RespParty)>
<!ELEMENT overOwner (RespParty)>
<!ELEMENT featCatCit (Citation)>
<!ELEMENT portcatcit (Citation)>
<!ELEMENT distCont (RespParty)>
<!ELEMENT RespParty (mandPartyInfo, rpIndName?, rpOrgName?, rpPosName?,
resp*, contactInfo+)>
<!ELEMENT contactInfo (Contact)>
<!ELEMENT citRespParty (RespParty)>
<!ELEMENT Contact (hrsOfServ?, contInstr?, phone?, address?,
onlineRes?)>
<!ELEMENT address (Address)>
<!ELEMENT Address (postAdd*, city?, adminArea?, postCode?, country?,
electMailAdd*)>
<!ELEMENT mandPartyInfo (MandParty)>
<!ELEMENT MandParty (mpIndName?, mpOrgName?, mpPosName?)>
<!ELEMENT onlineRes (OnlinRes)>
<!ELEMENT OnlinRes (linkage, functCode?, protocol?, appProfile?, name?,
```

```
desc?)>
<!ELEMENT onLine (OnlinRes)>
<!ELEMENT extnsOnliRes (OnlinRes)>
<!ELEMENT phone (Telephone)>
<!ELEMENT Telephone (voice*, fax*, other*, othType*)>
<!ELEMENT aggDS (DSAgg)>
<!ELEMENT DSAgg (aggDSMet+, agg+, super*, sub*, DSDataset, DSInit)>
<!ELEMENT aggDSMet (metadata)>
<!ELEMENT agg (DSDataset+)>
<!ELEMENT DSDataset (dataset+)>
<!ELEMENT DSInit (initType)>
<!ELEMENT dataset (DSAgg)>
<!ELEMENT super (DSAgg)>
<!ELEMENT sub (DSAgg)>
<!ATTLIST functCode
      OnLineFunction (access | additionalInformation | download | order
| search) #REQUIRED>
<!ATTLIST repType
      SpatRepType (matrix | raster | text | vector) #REQUIRED>
<!ATTLIST imagCond
      ImageCondCode (blurredImage | cloud | degradingObliquity | fog |
heavySmokeOrDust | night | rain | semiDarkness | shadow | snow |
terrainMasking) #REQUIRED>
<!ATTLIST cellType
      RasterCellType (matrixCoded | matrixValues | pixelCodes | pixelHSI
| pixelHLS | pixelRGB | TekHVC) #REQUIRED>
<!ATTLIST geometObjTyp
      GeoObjTypes (complexes | composites | curvs | points | solids |
surfaces) #REQUIRED>
<!ATTLIST topLevel
      TopologyLevel (fullTopology3D | geometryOnly | nonPlanarGraph1D |
planarGraph1D | planarGraph2D) #REQUIRED>
<!ATTLIST type
      KeywordType (discipline | place | stratum | temporal | theme)
#REQUIRED>
<!ATTLIST class
      classification (codeWord | confidential | secrete | topsecrete |
unclassified | otherUserDefined) #REQUIRED>
<!ATTLIST maintUpFreq
      MaintenanceFrequency (continual | daily | weekly | monthly |
biannual | annual | asNeeded | irregular | notPlanned | unknown |
otherMaintenancePeriod) #REQUIRED>
<!ATTLIST upScp
      Scope (dataset | series | nonGeographicdataset | propertyType |
featureAttribut | attribute | featureType | featureCollection |
dimensiongroup | collectionHardware | fieldSession | collectionSession)
#REQUIRED>
<!ATTLIST upScpDesc
      ScopeDescription (attributes | features | featureInstances |
attributeInstances | featureCollection | other) #REQUIRED>
<!ATTLIST unit
      LengthUnitcode (millimetre | centimetre | metre | kilometre |
internationalInch | internationalFoot | internationalMile | degree |
arcMinute | arcSecond) #REQUIRED>
```

```
<!ATTLIST presForm
      PresentationFormCode (document | hardcopyMap | image | model |
profile | rsterMap | table | VectorMap | view) #REQUIRED>
<!ATTLIST resp
      ResponsibilityCode (contentProvider | custodian | Steward | owner)
#REQUIRED>
<!ATTLIST category
      Category (Agriculture | Aquaculture| Biota | Communications |
Climatology | Economy | Environment | GeoscientificInfo | Health |
Imagery | Infrastructure | InlandWaters | Military | Oceans | Planning |
PoliticalBoundaries | Society | Utilities) #REQUIRED>
<!ATTLIST extType
      ExtentType (inclusion | exclusion) #REQUIRED>
<!ATTLIST fileID
      id ID #REQUIRED>
<!ATTLIST parID
      id ID #REQUIRED>
<!ATTLIST passSeqID
      id ID #REQUIRED>
<!ATTLIST imageID
      id ID #REQUIRED>
<!ATTLIST seqId
      id ID #REQUIRED>
<!ATTLIST citID
      id ID #REQUIRED>
<!NOTATION ISBN PUBLIC "http://www.isbn.spk-berlin.de/html/userman.htm">
<!ATTLIST ISBN
      no ID #REQUIRED
      format NOTATION (ISBN) #IMPLIED>
<!NOTATION ISSN PUBLIC
"http://issnic.issn.org:591/issn_check/search.htm">
<!ATTLIST ISSN
      no ID #REQUIRED
      format NOTATION (ISSN) #IMPLIED>
<!NOTATION ISO19108 PUBLIC
"http://www.statkart.no/isotc211/scope.htm#19108">
<!ATTLIST date
      format NOTATION (ISO19108) #IMPLIED>
<!ATTLIST useDatTim
      format NOTATION (ISO19108) #IMPLIED>
<!ATTLIST plnAvlDatTim
      format NOTATION (ISO19108) #IMPLIED>
<!ATTLIST edDate
      format NOTATION (ISO19108) #IMPLIED>
<!ATTLIST dataQualInfo
      format NOTATION (ISO19113) #IMPLIED>
<!ELEMENT fileID EMPTY>
<!ELEMENT parID EMPTY>
<!ELEMENT HierLevName (#PCDATA)>
<!ELEMENT date (#PCDATA)>
<!ELEMENT mdStanName (#PCDATA)>
<!ELEMENT mdStanVer (#PCDATA)>
<!ELEMENT abstract (#PCDATA)>
<!ELEMENT purpose (#PCDATA)>
```

```
<!ELEMENT suppInfo (#PCDATA)>
<!ELEMENT credit (#PCDATA)>
<!ELEMENT envir (#PCDATA)>
<!ELEMENT spatRes (#PCDATA)>
<!ELEMENT passSeqID EMPTY>
<!ELEMENT imagOrbID (#PCDATA)>
<!ELEMENT orbNum (#PCDATA)>
<!ELEMENT fileName (#PCDATA)>
<!ELEMENT fileDesc (#PCDATA)>
<!ELEMENT fileType (#PCDATA)>
<!ELEMENT keyword (#PCDATA)>
<!ELEMENT type EMPTY>
<!ELEMENT thesaName (#PCDATA)>
<!ELEMENT specUse (#PCDATA)>
<!ELEMENT useDatTim (#PCDATA)>
<!ELEMENT usrDefLims (#PCDATA)>
<!ELEMENT useLimit (#PCDATA)>
<!ELEMENT LegalConst (#PCDATA)>
<!ELEMENT othConst (#PCDATA)>
<!ELEMENT SecInfo (#PCDATA)>
<!ELEMENT userNote (#PCDATA)>
<!ELEMENT classSys (#PCDATA)>
<!ELEMENT handDesc (#PCDATA)>
<!ELEMENT otherUserDef (#PCDATA)>
<!ELEMENT othMaintPer (#PCDATA)>
<!ELEMENT other (#PCDATA)>
<!ELEMENT imageID EMPTY>
<!ELEMENT imageType (#PCDATA)>
<!ELEMENT grToImgCoAvl (#PCDATA)>
<!ELEMENT cellOrigin (#PCDATA)>
<!ELEMENT rows (#PCDATA)>
<!ELEMENT cols (#PCDATA)>
<!ELEMENT verts (#PCDATA)>
<!ELEMENT geometObjCnt (#PCDATA)>
<!ELEMENT toneGrad (#PCDATA)>
<!ELEMENT bitsPB (#PCDATA)>
<!ELEMENT cellAttDesc (#PCDATA)>
<!ELEMENT cellUnit (#PCDATA)>
<!ELEMENT spac (#PCDATA)>
<!ELEMENT illElevAng (#PCDATA)>
<!ELEMENT illAziAng (#PCDATA)>
<!ELEMENT imgOrieAng (#PCDATA)>
<!ELEMENT imgQualRatSys (#PCDATA)>
<!ELEMENT imagQualCode (#PCDATA)>
<!ELEMENT triID (#PCDATA)>
<!ELEMENT radDatAvail (#PCDATA)>
<!ELEMENT compenQuan (#PCDATA)>
<!ELEMENT prePrcTypcde (#PCDATA)>
<!ELEMENT cloudCovPer (#PCDATA)>
<!ELEMENT ESDAvail (#PCDATA)>
<!ELEMENT pixPerUnit (#PCDATA)>
<!ELEMENT focLen (#PCDATA)>
<!ELEMENT oblAng (#PCDATA)>
<!ELEMENT imgSenTime (#PCDATA)>
```

```
<!ELEMENT senCat (#PCDATA)>
<!ELEMENT senMode (#PCDATA)>
<!ELEMENT spectProp (#PCDATA)>
<!ELEMENT fieldOView (#PCDATA)>
<!ELEMENT orieOnPlat (#PCDATA)>
<!ELEMENT opMode (#PCDATA)>
<!ELEMENT seqId EMPTY>
<!ELEMENT hiWavelen (#PCDATA)>
<!ELEMENT lowWavelen (#PCDATA)>
<!ELEMENT camCalInfAvl (#PCDATA)>
<!ELEMENT filmDistrtInfAvl (#PCDATA)>
<!ELEMENT lensSidtrtInfAvl (#PCDATA)>
<!ELEMENT identifier (#PCDATA)>
<!ELEMENT theme (#PCDATA)>
<!ELEMENT remarks (#PCDATA)>
<!ELEMENT compCode (#PCDATA)>
<!ELEMENT lanfCode (#PCDATA)>
<!ELEMENT incWithDS (#PCDATA)>
<!ELEMENT featType (#PCDATA)>
<!ELEMENT unitsODist (#PCDATA)>
<!ELEMENT transSize (#PCDATA)>
<!ELEMENT name (#PCDATA)>
<!ELEMENT verNum (#PCDATA)>
<!ELEMENT amendNum (#PCDATA)>
<!ELEMENT spec (#PCDATA)>
<!ELEMENT filDecmTech (#PCDATA)>
<!ELEMENT density (#PCDATA)>
<!ELEMENT densityUn (#PCDATA)>
<!ELEMENT vols (#PCDATA)>
<!ELEMENT medFormat (#PCDATA)>
<!ELEMENT compat (#PCDATA)>
<!ELEMENT fees (#PCDATA)>
<!ELEMENT plnAvlDatTim (#PCDATA)>
<!ELEMENT ordInstr (#PCDATA)>
<!ELEMENT turnaround (#PCDATA)>
<!ELEMENT defin (#PCDATA)>
<!ELEMENT oblig (#PCDATA)>
<!ELEMENT datType (#PCDATA)>
<!ELEMENT domVal (#PCDATA)>
<!ELEMENT maxOcc (#PCDATA)>
<!ELEMENT parEnt (#PCDATA)>
<!ELEMENT rule (#PCDATA)>
<!ELEMENT rationale (#PCDATA)>
<!ELEMENT source (#PCDATA)>
<!ELEMENT schLang (#PCDATA)>
<!ELEMENT constrLang (#PCDATA)>
<!ELEMENT schAsc (#PCDATA)>
<!ELEMENT graFilTyp (#PCDATA)>
<!ELEMENT swDevFile (#PCDATA)>
<!ELEMENT swDevFormat (#PCDATA)>
<!ELEMENT spatObj (#PCDATA)>
<!ELEMENT sparSchName (#PCDATA)>
<!ELEMENT desc (#PCDATA)>
<!ELEMENT poly (#PCDATA)>
```

```
<!ELEMENT westBL (#PCDATA)>
<!ELEMENT eastBL (#PCDATA)>
<!ELEMENT southBL (#PCDATA)>
<!ELEMENT northBL (#PCDATA)>
<!ELEMENT extent (#PCDATA)>
<!ELEMENT minVal (#PCDATA)>
<!ELEMENT maxVal (#PCDATA)>
<!ELEMENT uOfMeas (#PCDATA)>
<!ELEMENT vetDat (#PCDATA)>
<!ELEMENT title (#PCDATA)>
<!ELEMENT altTilte (#PCDATA)>
<!ELEMENT dateType (#PCDATA)>
<!ELEMENT edition (#PCDATA)>
<!ELEMENT edDate (#PCDATA)>
<!ELEMENT citID EMPTY>
<!ELEMENT idType (#PCDATA)>
<!ELEMENT serName (#PCDATA)>
<!ELEMENT issID (#PCDATA)>
<!ELEMENT otherCitDet (#PCDATA)>
<!ELEMENT collTitle (#PCDATA)>
<!ELEMENT page (#PCDATA)>
<!ELEMENT ISBN EMPTY>
<!ELEMENT ISSN EMPTY>
<!ELEMENT rpIndName (#PCDATA)>
<!ELEMENT rpOrgName (#PCDATA)>
<!ELEMENT rpPosName (#PCDATA)>
<!ELEMENT postAdd (#PCDATA)>
<!ELEMENT city (#PCDATA)>
<!ELEMENT adminArea (#PCDATA)>
<!ELEMENT postCode (#PCDATA)>
<!ELEMENT country (#PCDATA)>
<!ELEMENT electMailAdd (#PCDATA)>
<!ELEMENT hrsOfServ (#PCDATA)>
<!ELEMENT contInstr (#PCDATA)>
<!ELEMENT mpIndName (#PCDATA)>
<!ELEMENT mpOrgName (#PCDATA)>
<!ELEMENT mpPosName (#PCDATA)>
<!ELEMENT linkage (#PCDATA)>
<!ELEMENT protocol (#PCDATA)>
<!ELEMENT appProfile (#PCDATA)>
<!ELEMENT voice (#PCDATA)>
<!ELEMENT fax (#PCDATA)>
<!ELEMENT othType (#PCDATA)>
<!ELEMENT TMRefSys (#PCDATA)>
<!ELEMENT altTitle (#PCDATA)>
<!ELEMENT attribInsts (#PCDATA)>
<!ELEMENT featColl (#PCDATA)>
<!ELEMENT attribs (#PCDATA)>
<!ELEMENT category EMPTY>
<!ELEMENT cellOrig (#PCDATA)>
<!ELEMENT cellType EMPTY>
<!ELEMENT class EMPTY>
<!ELEMENT compGenQuan (#PCDATA)>
<!ELEMENT compcode (#PCDATA)>
```

```
<!ELEMENT constrlang (#PCDATA)>
<!ELEMENT dataQualInfo (#PCDATA)>
<!ELEMENT elements (#PCDATA)>
<!ELEMENT extType EMPTY>
<!ELEMENT featAtt (#PCDATA)>
<!ELEMENT feat (#PCDATA)>
<!ELEMENT featInsts (#PCDATA)>
<!ELEMENT featTyp (#PCDATA)>
<!ELEMENT feats (#PCDATA)>
<!ELEMENT fieldOview (#PCDATA)>
<!ELEMENT functCode EMPTY>
<!ELEMENT geoDesc (#PCDATA)>
<!ELEMENT geometObjTyp EMPTY>
<!ELEMENT hierLev (#PCDATA)>
<!ELEMENT illElevang (#PCDATA)>
<!ELEMENT imagQualcode (#PCDATA)>
<!ELEMENT imagCond EMPTY>
<!ELEMENT imgQualRatsys (#PCDATA)>
<!ELEMENT kindCode (#PCDATA)>
<!ELEMENT langCode (#PCDATA)>
<!ELEMENT lensDistrtInfAvl (#PCDATA)>
<!ELEMENT maintUpFreq EMPTY>
<!ELEMENT offline (#PCDATA)>
<!ELEMENT plnAvldatTim (#PCDATA)>
<!ELEMENT prePrcTypCde (#PCDATA)>
<!ELEMENT presForm EMPTY>
<!ELEMENT propTyp (#PCDATA)>
<!ELEMENT repType EMPTY>
<!ELEMENT status (#PCDATA)>
<!ELEMENT resp EMPTY>
<!ELEMENT topLevel EMPTY>
<!ELEMENT trilD (#PCDATA)>
<!ELEMENT unit EMPTY>
<!ELEMENT upScp EMPTY>
<!ELEMENT upScpDesc EMPTY>
<!ELEMENT initType (#PCDATA)>
```

APPENDIX VIII Selected implementation details

1. Microsoft XML parser

The Microsoft XML parser comes as prewritten Java code that can be placed right into my own Java applications. I made use of that code in my programs to read in XML documents and determine their structure. The Microsoft XML parser can be obtained from Microsoft [2000]. After downloading and extracting, I got the Microsoft com Java package. I placed this package into a library directory so the XML translator program could find the parser class files easily. The major MSXML classes and their functionality are as shown in Table 3.1.

Table VI.1. Major Microsoft XML Parser Classes.

Microsoft Parser Class	*Function*
com.ms.xml.om.Element	Handles XML elements
com.ms.xml.om.Document	Loads and handles XML documents
com.ms.xml.parser.ParseException	Handles Parsing exceptions

For example, to load in an XML document, I used the Document class load() method.

2. Oracle JDBC Driver

In this research, I downloaded the Oracle JDBC driver from Oracle [2000]. After extracting, I got the Oracle package. I placed this package into the JDK(Java Development Kit) library directory so the JDBC servlet program could find the driver class files easily.

3. XSL Processor

After download the LotusXSL from IBM[2000], set the Java interpreter's class path to include the files: xerces.jar, xalan.jar and lotusxsl.jar and run com.lotus.xsl.Process Java program with command line arguments, (i.e. -in fgdc.xml -xsl FGDC_to_ISO19115.xsl -out iso.xml, where fgdc.xml is the input XML file, FGDC_to_ISO19115.xsl is the XSL style sheet file and iso.xml is the output XML file).

Appendix IX XSL stylesheet

```
<?xml version="1.0" encoding="ISO-8859-1"?>
```

```
<xsl:stylesheet
xmlns:xsl="http://www.w3.org/1999/XSL/Transform">

<xsl:output method="xml" indent="yes"/>

<xsl:param name="fileid" select="$fileid"/>

<xsl:template match="/">
      <xsl:apply-templates select="metadata"/>
</xsl:template>

<xsl:template match="metadata">
      <xsl:element name="metadata">
            <xsl:element name="fileID">
                  <xsl:value-of select="$fileid"/>
            </xsl:element>
            <xsl:element name="contact">
                  <xsl:apply-templates select="metainfo/metc/cntinfo"/>
            </xsl:element>
            <xsl:element name="date">
                  <xsl:value-of select="metainfo/metd"/>
            </xsl:element>
            <xsl:element name="mdStanName">
                  ISO CD 19115
            </xsl:element>
            <xsl:element name="mdStanVer">
                  CD 2.0
            </xsl:element>


            <xsl:apply-templates select="idinfo"/>

            <xsl:element name="distInfo">
                  <xsl:element name="disFormat">
                        <xsl:for-each
select="distinfo/stdorder/digform/digtinfo">
                              <xsl:element name="Format">
                                    <xsl:element name="name"><xsl:value-
of select="formname"/></xsl:element>
                              </xsl:element>
                        </xsl:for-each>
                  </xsl:element>

                  <xsl:for-each select="distinfo/distrib">
                        <xsl:element name="distributor">
                              <xsl:element name="distCont">
                                    <xsl:apply-templates
select="cntinfo"/>
                              </xsl:element>
                        </xsl:element>
                  </xsl:for-each>

                  <xsl:element name="distribTrnsOps">
                        <xsl:apply-templates select="distinfo"/>
```

```
                        </xsl:element>
                        <xsl:element name="transSize">
                              <xsl:value-of select="distinfo/transize"/>
                        </xsl:element>
                        <xsl:element name="onLine">
                              <xsl:value-of select="distinfo/onlineopt"/>
                        </xsl:element>
                  </xsl:element>
            </xsl:element>
      </xsl:template>

      <xsl:template match="idinfo">
            <xsl:element name="idinfo">
                  <xsl:element name="abstract"><xsl:value-of
select="descript/abstract"/></xsl:element>
                  <xsl:element name="purpose"><xsl:value-of
select="descript/purpose"/></xsl:element>
                  <xsl:element name="supplinfo"><xsl:value-of
select="descript/supplinf"/></xsl:element>
                  <xsl:element name="credit"><xsl:value-of
select="datacred"/></xsl:element>
                  <xsl:element name="envir"><xsl:value-of
select="native"/></xsl:element>

                  <xsl:for-each select="spdom/bounding">
                        <xsl:element name="GeoBndBox">
                              <xsl:element name="westBL"><xsl:value-of
select="westbc"/></xsl:element>
                              <xsl:element name="eastBL"><xsl:value-of
select="eastbc"/></xsl:element>
                              <xsl:element name="southBL"><xsl:value-of
select="southbc"/></xsl:element>
                              <xsl:element name="northBL"><xsl:value-of
select="northbc"/></xsl:element>
                        </xsl:element>
                  </xsl:for-each>

                  <xsl:element name="dsCitation">
                        <xsl:apply-templates select="citation/citeinfo"/>
                  </xsl:element>
                  <xsl:element name="dsPOC">
                        <xsl:apply-templates select="ptcontac/cntinfo"/>
                  </xsl:element>

                  <xsl:element name="MaintInfo">
                        <xsl:element name="maintUpFreq"><xsl:value-of
select="status/update"/></xsl:element>
                  </xsl:element>
                  <xsl:element name="descKey">
                        <xsl:apply-templates
select="keywords/theme|keywords/place"/>
                  </xsl:element>

                  <xsl:element name="LegalConst">
```

```
                  <xsl:element name="accConst">
                        <xsl:value-of select="accconst"/>
                  </xsl:element>
                  <xsl:element name="useConst">
                        <xsl:value-of select="useconst"/>
                  </xsl:element>
            </xsl:element>
      </xsl:element>
</xsl:template>

<xsl:template match="place|theme">
      <xsl:element name="keywords">
            <xsl:element name="thesaName">
                  <xsl:if test="name(.) = 'place'">place</xsl:if>
                  <xsl:if test="name(.) = 'theme'">theme</xsl:if>
            </xsl:element>
            <xsl:element name="type"><xsl:value-of
select="placekt|themekt"/></xsl:element>
            <xsl:for-each select="placekey|themekey">
                  <xsl:element name="keyword"><xsl:value-of
select="."/></xsl:element>
            </xsl:for-each>
      </xsl:element>
</xsl:template>

<xsl:template match="citeinfo">
      <xsl:element name="Citation">
            <xsl:element name="title"><xsl:value-of
select="title"/></xsl:element>
            <xsl:element name="date"><xsl:value-of
select="pubdate"/></xsl:element>
            <xsl:element name="edition"><xsl:value-of
select="edition"/></xsl:element>
            <xsl:element name="serName"><xsl:value-of
select="sername"/></xsl:element>
            <xsl:element name="otherCitDet"><xsl:value-of
select="othercit"/></xsl:element>
      </xsl:element>
</xsl:template>

<xsl:template match="cntinfo">
      <xsl:element name="RespParty">
            <xsl:element name="MandParty">
                  <xsl:element name="mpIndName"><xsl:value-of
select="cntperp/cntper"/></xsl:element>
                  <xsl:element name="mpOrgName"><xsl:value-of
select="cntperp/cntorg"/></xsl:element>
                  <xsl:element name="mpPosName"><xsl:value-of
select="cntperp/cntpos"/></xsl:element>
            </xsl:element>
            <xsl:element name="contact">
                  <xsl:element name="telephone">
                        <xsl:for-each select="cntvoice">
                              <xsl:element name="voice"><xsl:value-of
```

```
select="."/></xsl:element>
                        </xsl:for-each>
                        <xsl:for-each select="cntfax">
                              <xsl:element name="fax"><xsl:value-of
select="."/></xsl:element>
                        </xsl:for-each>
                  </xsl:element>

                  <xsl:element name="Address">
                        <xsl:element name="postAdd">
                              <xsl:value-of select="cntaddr/address"/>
                        </xsl:element>
                        <xsl:element name="city">
                              <xsl:value-of select="cntaddr/city"/>
                        </xsl:element>
                        <xsl:element name="adminArea">
                              <xsl:value-of select="cntaddr/state"/>
                        </xsl:element>
                        <xsl:element name="postCode">
                              <xsl:value-of select="cntaddr/postal"/>
                        </xsl:element>
                        <xsl:element name="country">
                              <xsl:value-of select="cntaddr/country"/>
                        </xsl:element>
                        <xsl:element name="adminArea">
                              <xsl:apply-templates
select="cntaddr/state"/>
                        </xsl:element>
                        <xsl:for-each select="cntemail">
                              <xsl:element
name="electMailAdd"><xsl:value-of select="."/></xsl:element>
                        </xsl:for-each>
                  </xsl:element>
            </xsl:element>
            <xsl:element name="resp">
                  <xsl:if test="name(..) = 'metc'">originator</xsl:if>
                  <xsl:if test="name(..) =
'ptcontac'">pointOfContact</xsl:if>
                  <xsl:if test="name(..) =
'distrib'">distributor</xsl:if>
            </xsl:element>
      </xsl:element>
</xsl:template>

<xsl:template match="metadata/cntinfo">
      <xsl:element name="contact">
            <xsl:element name="hrsOfServ"><xsl:apply-templates
select="hours"/></xsl:element>
            <xsl:element name="contInstr"><xsl:apply-templates
select="cntinst"/></xsl:element>
            <xsl:element name="phone"><xsl:apply-templates
select="cntvoice"/></xsl:element>
            <xsl:element name="address"><xsl:apply-templates
select="cntaddr"/></xsl:element>
```

```
            <xsl:element name="telephone"><xsl:apply-templates
select="cntvoice|cntfax"/></xsl:element>

            <xsl:apply-templates select="/metainfo/metaextns"/>
      </xsl:element>
</xsl:template>

<xsl:template match="distinfo">
      <xsl:element name="digTransOpts">
            <xsl:for-each select="stdorder/digform/digtopt/offoptn">
                  <xsl:element name="Medium">
                        <xsl:element name="name">
                              <xsl:value-of select="offmedia"/>
                        </xsl:element>
                        <xsl:element name="density">
                              <xsl:value-of select="reccap/recden"/>
                        </xsl:element>
                        <xsl:element name="densityUn">
                              <xsl:value-of select="reccap/recdenu"/>
                        </xsl:element>
                        <xsl:element name="medFormat">
                              <xsl:value-of select="recfmt"/>
                        </xsl:element>
                        <xsl:element name="compat">
                              <xsl:value-of select="compat"/>
                        </xsl:element>
                        <xsl:element name="fees">
                              <xsl:value-of select="../../../fees"/>
                        </xsl:element>
                        <xsl:element name="ordInstr">
                              <xsl:value-of select="../../../ordering"/>
                        </xsl:element>
                        <xsl:element name="turnaround">
                              <xsl:value-of
select="../../../turnaround"/>
                        </xsl:element>
                  </xsl:element>
            </xsl:for-each>
      </xsl:element>
</xsl:template>

<xsl:template match="digtinfo">
      <xsl:element name="Format">
            <xsl:element name="name">
                  <xsl:apply-templates select="formname"/>
            </xsl:element>
            <xsl:element name="verNum">
                  <xsl:apply-templates select="formvern"/>
            </xsl:element>
            <xsl:element name="spec">
                  <xsl:value-of select="formspec"/>
            </xsl:element>
            <xsl:element name="filDecmTech">
                  <xsl:value-of select="filedesc"/>
```

```
            </xsl:element>
      </xsl:element>
</xsl:template>

<xsl:template match="citation">
      <xsl:element name="citation">
            <xsl:element name="title">
                  <xsl:value-of select="title"/>
            </xsl:element>
            <xsl:element name="date">
                  <xsl:value-of select="date"/>
            </xsl:element>
            <xsl:element name="edition">
                  <xsl:value-of select="edition"/>
            </xsl:element>
            <xsl:element name="serName">
                  <xsl:value-of select="sername"/>
            </xsl:element>
            <xsl:element name="otherCitDet">
                  <xsl:value-of select="othercit"/>
            </xsl:element>
            <xsl:element name="citRespParty">
                  <xsl:apply-templates select="cntperp"/>
            </xsl:element>
      </xsl:element>
</xsl:template>

<xsl:template match="cntaddr">
      <xsl:element name="Address">
            <xsl:element name="city">
                  <xsl:value-of select="city"/>
            </xsl:element>
            <xsl:element name="adminArea">
                  <xsl:value-of select="state"/>
            </xsl:element>
            <xsl:element name="postCode">
                  <xsl:value-of select="postal"/>
            </xsl:element>
            <xsl:element name="country">
                  <xsl:value-of select="country"/>
            </xsl:element>
            <xsl:element name="adminArea">
                  <xsl:apply-templates select="state"/>
            </xsl:element>

            <xsl:for-each select="address">
                  <xsl:element name="postAdd">
                        <xsl:value-of select="."/>
                  </xsl:element>
            </xsl:for-each>
            <xsl:for-each select="cntemail">
                  <xsl:element name="elecMailAdd">
                        <xsl:value-of select="."/>
                  </xsl:element>
```

```
            </xsl:for-each>
      </xsl:element>
</xsl:template>

</xsl:stylesheet>
```

Appendix X Different longitude domain casting

Longitude is measured from the Prime Meridian (which is the longitude that runs through Greenwich, England), with positive values going east and negative values going west. So, both the FGDC CSDGM and the ISO 19115 have [-180.0, 180.0] as the domain for longitude. Since the user interface uses [0.0, 360.0] as the domain for longitude, the correct casting between these two different domain representations is very important.

The difference between the two domain representations is that the negative half [-180.0, 0.0] of the metadata files is represented by [180.0, 360.0] in the user interface. So, when constructing the R-tree or the database table, if the dataset's western-most coordinate is in [0.0, 180.0) and the eastern-most coordinate is in [-180.0, 0.0), we need to add 360 to the eastern-most coordinate so that the dataset has a longitude range falling in the [0.0, 360.0] domain, i.e. if 8_{west} 0 [0.0, 180.0) and 8_{east} 0 [-180.0, 0.0) then

$$8_{east} = 8_{east} + 360.0. \qquad (1)$$

For searching to match, two cases arise. Case 1: if the query window's western-most coordinate is in [0.0, 180.0] and the eastern-most coordinate is in (180.0, 360.0], we need to combine two separate searches. One search is to use the original western-most and eastern-most coordinates of the query window and the second search is to use the query window with -180 as the western-most coordinate and an eastern-most coordinate resulting from subtracting 360 from the original one, i.e.

$$S = S_{QW} \text{ c } S_{QW1} \qquad (2)$$

where S = result set of objects in range, S_{QW} = set of objects resulting from a range search with the original query window, and S_{QW1} = set of objects resulting from a range search with QW1, where

$$l_{west}^{QW1} = -180.0, \quad l_{east}^{QW1} = l_{east}^{QW} - 360.0 \qquad (3)$$

Case 2: The query window's western-most coordinate and the eastern-most coordinate are both in [180.0, 360.0]. Again, two separate searches are combined. One search is to use the original western-most and eastern-most coordinates of the query window and the second search is to use the query window with western-most and eastern-most coordinates resulting from subtracting 360 from the original respective ones, i.e.

$$S = S_{QW} \text{ c } S_{QW2} \qquad (4)$$

where S and S_{QW} are as before, and S_{QW2} = set of objects resulting from a range search with QW2, where

$$l_{west}^{QW2} = l_{west}^{QW} -360.0, \quad l_{east}^{QW2} = l_{east}^{QW} - 360.0 \qquad (5)$$

APPENDIX XI Sample CEONet FGDC XML Metadata

A sample CEONet FGDC XML metadata file with file name "com==Canadian_Seabed_Research_Ltd@DOT@==Canadian_Beaufort_Sea_Ice_Scour_Database" is listed below.


```xml
<?xml version="1.0" encoding="ISO-8859-1"?>
<metadata>
  <idinfo>
    <citation>
      <citeinfo>
        <origin>Geological Survey of Canada (Atlantic)</origin>
        <title>Canadian Beaufort Sea Ice Scour Database</title>
      </citeinfo>
    </citation>
    <descript>
      <abstract>
        This database consists of quantitative ice scour parameters
based on analysis of side scan sonar and high resolution echosounder
data. Descriptive report available. Coverage is of the continental shelf
of the Yukon and Northwest Territories.
      </abstract>
      <purpose>
        This data is currently applied in spatial comparison of ice
scour characteristics (width, depth, shape), their distribution,
frequency and effect on the sea floor.

        Potential applications include: risk assessment for subsea
installations, correlation of ice scour record to oceanographic,
geologic and meteorological factors; determination of scour rates.

        Data plans are the implementation of GIS techniques using SPANS
to analyze and model ice scour processes and attributes.
      </purpose>
      <supplinf>
        DIF Entry_ID:Canada_CdnSeaRes_icescour
        IDN_Node:CANADA/CCRS
      </supplinf>
    </descript>
    <timeperd>
      <timeinfo>
        <rngdates>
          <begdate>19840101</begdate>
        </rngdates>
      </timeinfo>
    </timeperd>
    <status>
      <update>As Needed</update>
    </status>
```

```
    <spdom>
      <bounding>
        <westbc>-141</westbc>
        <eastbc>-129</eastbc>
        <northbc>+71</northbc>
        <southbc>+69</southbc>
      </bounding>
    </spdom>
    <keywords>
      <theme>
        <themekt>GCMD</themekt>
        <themekey>Earth Science > Oceans > Bathymetry > Seafloor
Topography > Sea Floor Scouring</themekey>
        <themekey>Earth Science > Oceans > Bathymetry > Water
Depth</themekey>
        <themekey>Earth Science > Oceans > Marine Geophysics >
Continental Shelves</themekey>
        <themekey>COASTAL ZONE</themekey>
        <themekey>ENGINEERING GEOLOGY</themekey>
        <themekey>GEOMORPHOLOGY</themekey>
        <themekey>SEA FLOOR SCOURING</themekey>
        <themekey>MARINE GEOLOGY</themekey>
        <themekey>MARINE GEOPHYSICS</themekey>
        <themekey>SEA BED</themekey>
      </theme>
      <place>
        <placekt>GCMD</placekt>
        <placekey>NORTH AMERICA</placekey>
        <placekey>CANADA</placekey>
        <placekey>ARCTIC OCEAN</placekey>
        <placekey>BEAUFORT SEA</placekey>
        <placekey>SEA FLOOR</placekey>
        <placekey>Offshore North</placekey>
        <placekey>Arctic</placekey>
        <placekey>Canada > Yukon</placekey>
        <placekey>Canada > Northwest Territories</placekey>
        <placekey>Canadian Beaufort Continental Shelf</placekey>
        <placekey>Canada > Nunavut</placekey>
      </place>
    </keywords>
    <accconst>none</accconst>
    <useconst>none</useconst>
    <ptcontac>
      <cntinfo>
        <cntperp>
          <cntper>Bob Harmes</cntper>
          <cntorg>
            Geological Survey of Canada (Atlantic) Marine Regional
Geoscience
          </cntorg>
        </cntperp>
        <cntpos>Technical Contact</cntpos>
        <cntaddr>
          <addrtype>mailing and physical address</addrtype>
```

```
          <address>
            1 Challenger Drive,
            P.O. Box 1006
          </address>
          <city>Dartmouth</city>
          <state>Nova Scotia</state>
          <postal>B2Y 4A2</postal>
          <country>Canada</country>
        </cntaddr>
        <cntvoice>(902) 426-3933</cntvoice>
        <cntfax>(902) 426-4104</cntfax>
        <cntemail>blasco@agc.bio.ns.ca</cntemail>
      </cntinfo>
    </ptcontac>
  </idinfo>
  <distinfo>
    <distrib>
      <cntinfo>
        <cntperp>
          <cntper>Steve Blasco</cntper>
          <cntorg>
            Geological Survey of Canada (Atlantic) Marine Regional
Geoscience
          </cntorg>
        </cntperp>
        <cntaddr>
          <addrtype>mailing and physical address</addrtype>
          <address>
            1 Challenger Drive,
            P.O. Box 1006
          </address>
          <city>Dartmouth</city>
          <state>Nova Scotia</state>
          <postal>B2Y 4A2</postal>
          <country>Canada</country>
        </cntaddr>
        <cntvoice>(902) 426-3932</cntvoice>
        <cntfax>(902) 426-4104</cntfax>
        <cntemail>blasco@agc.bio.ns.ca</cntemail>
      </cntinfo>
    </distrib>
    <distrib>
      <cntinfo>
        <cntorgp>
          <cntorg>Canadian Seabed Research Ltd.</cntorg>
          <cntper>Mr. Glen  Gilbert</cntper>
        </cntorgp>
        <cntaddr>
          <addrtype>Mailing and physical address</addrtype>
          <address>341 Myra Road, P.O. Box 299</address>
          <city>Porter's Lake</city>
          <state>Nova Scotia</state>
          <postal>B0J 2S0</postal>
          <country>Canada</country>
```

```
        </cntaddr>
        <cntvoice>902-827-4200</cntvoice>
        <cntfax>902-827-2002</cntfax>
        <cntemail>csr@atcon.com</cntemail>
      </cntinfo>
    </distrib>
    <distliab>Not Available</distliab>
    <stdorder>
      <digform>
        <digtinfo>
          <formname>AUTOCAD DXF; Point Data Flat File; ASCII Text
file</formname>
        </digtinfo>
        <digtopt>
          <offoptn>
            <offmedia>Magnetic Tape/disk.</offmedia>
          </offoptn>
        </digtopt>
      </digform>
      <fees>Yes.</fees>
    </stdorder>
  </distinfo>
  <metainfo>
    <metd>19990603</metd>
    <metrd>19990603</metrd>
    <metfrd>20000528</metfrd>
    <metc>
      <cntinfo>
        <cntperp>
          <cntper>Susan Merchant</cntper>
          <cntorg>
            Geological Survey of Canada (Atlantic) Marine Regional
Geoscience
          </cntorg>
        </cntperp>
        <cntaddr>
          <addrtype>mailing and physical address</addrtype>
          <address>
            1 Challenger Drive,
            P.O. Box 1006
          </address>
          <city>Dartmouth</city>
          <state>Nova Scotia</state>
          <postal>B2Y 4A2</postal>
          <country>Canada</country>
        </cntaddr>
        <cntvoice>(902) 426-3410</cntvoice>
        <cntfax>(902) 426-4465</cntfax>
        <cntemail>merchant@agc.bio.ns.ca</cntemail>
      </cntinfo>
    </metc>
    <metstdn>FGDC Content Standards for Digital Geospatial
Metadata</metstdn>
    <metstdv>19940608</metstdv>
```

    <mettc>local time</mettc>
  </metainfo>
</metadata>

Appendix XII Sample CEONet ISO XML Metadata

The CEONet ISO XML metadata file obtained transforming the CEONet FGDC XML metadata file listed in Appendix IV using the XML translator program is show as follows:


```
<?xml version="1.0" encoding="UTF-8"?>
<metadata>
   <fileID
id=”com==Canadian_Seabed_Research_Ltd@DOT@==Canadian_Beaufort_Sea_Ice_Sc
our_Database”/>
   <hierLev> dataset </hierLev>
   <contact>
      <RespParty>
         <mandParty>
            <mpIndName> Susan Merchant </mpIndName>
            <mpOrgName>  Geological Survey of Canada (Atlantic) Marine
Regional Geoscience  </mpOrgName>
         </mandParty>
         <contact>
            <telephone>
               <fax>(902) 426-4465</fax>
            </telephone>
            <address>
               <postAdd> 1 Challenger Drive, P.O. Box 1006 </postAdd>
               <city> Dartmouth </city>
               <adminArea> Nova Scotia </adminArea>
               <postCode> B2Y 4A2 </postCode>
               <country> Canada </country>
               <electMailAdd>merchant@agc.bio.ns.ca</electMailAdd>
            </address>
         </contact>
         <resp ResponsibilityCode=”metadataProvider”/>
      </RespParty>
   </contact>
   <date format=”ISO 19108"> 19990603 </date>
   <mdStanName> ISO 19115 </mdStanName>
   <mdStanVer> CD 2.0 </mdStanVer>
   <idInfo>
      <abstract>  This database consists of quantitative ice scour
parameters based on analysis of side scan sonar and high resolution
echosounder data. Descriptive report available. Coverage is of the
continental shelf of the Yukon and Northwest Territories.  </abstract>
      <purpose>  This data is currently applied in spatial comparison of
ice scour characteristics (width, depth, shape), their distribution,
frequency and effect on the sea floor. Potential applications include:
risk assessment for subsea installations, correlation of ice scour
record to oceanographic, geologic and meteorological factors;
determination of scour rates. Data plans are the implementation of GIS
techniques using SPANS to analyze and model ice scour processes and
attributes.  </purpose>
```

```
      <suppInfo>  DIF Entry_ID:Canada_CdnSeaRes_icescour
IDN_Node:CANADA/CCRS  </suppInfo>
      <GeoBndBox>
         <westBL> -141 </westBL>
         <eastBL> -129 </eastBL>
         <northBL> +71 </northBL>
         <southBL> +69 </southBL>
      </GeoBndBox>
      <dsCitation>
         <Citation>
            <title> Canadian Beaufort Sea Ice Scour Database </title>
            <citRespParty>
               <RespParty>
                  <mandParty>
                        <mpOrgName> Geological Survey of Canada
                        (Atlantic) </mpOrgName>
                  </mandParty>
                  <resp ResponsibilityCode=”originator”/>
               </RespParty>
            </citRespParty>
         </Citation>
      </dsCitation>
      <dsPOC>
         <RespParty>
            <mandParty>
               <mpIndName> Bob Harmes </mpIndName>
               <mpOrgName>  Geological Survey of Canada (Atlantic)
Marine Regional Geoscience  </mpOrgName>
               <mpPosName> Technical Contact </mpPosName>
            </mandParty>
            <contact>
               <telephone>
                  <fax>(902) 426-4104</fax>
               </telephone>
               <address>
                  <postAdd> 1 Challenger Drive, P.O. Box 1006 </postAdd>
                  <city> Dartmouth </city>
                  <adminArea> Nova Scotia </adminArea>
                  <postCode> B2Y 4A2 </postCode>
                  <country> Canada </country>
                  <electMailAdd>blasco@agc.bio.ns.ca</electMailAdd>
               </address>
            </contact>
            <resp ResponsibilityCode=”pointOfContact”/>
         </RespParty>
      </dsPOC>
      <dsMaint>
         <MaintInfo>
            <maintUpFreq MaintenanceFrequency=”asNeeded”/>
         </MaintInfo>
      </dsMaint>
      <descKey>
         <Keywords>
            <keyword>Earth Science > Oceans > Bathymetry > Seafloor
```

```
Topography > Sea Floor Scouring</keyword>
            <keyword>Earth Science > Oceans > Bathymetry > Water
Depth</keyword>
            <keyword>Earth Science > Oceans > Marine Geophysics >
Continental Shelves</keyword>
            <keyword>COASTAL ZONE</keyword>
            <keyword>ENGINEERING GEOLOGY</keyword>
            <keyword>GEOMORPHOLOGY</keyword>
            <keyword>SEA FLOOR SCOURING</keyword>
            <keyword>MARINE GEOLOGY</keyword>
            <keyword>MARINE GEOPHYSICS</keyword>
            <keyword>SEA BED</keyword>
            <type KeywordType=”theme”/>
            <thesaName> themekt </thesaName>
         </Keywords>
         <Keywords>
            <keyword>NORTH AMERICA</keyword>
            <keyword>CANADA</keyword>
            <keyword>ARCTIC OCEAN</keyword>
            <keyword>BEAUFORT SEA</keyword>
            <keyword>SEA FLOOR</keyword>
            <keyword>Offshore North</keyword>
            <keyword>Arctic</keyword>
            <keyword>Canada > Yukon</keyword>
            <keyword>Canada > Northwest Territories</keyword>
            <keyword>Canadian Beaufort Continental Shelf</keyword>
            <keyword>Canada > Nunavut</keyword>
            <type KeywordType=”place”/>
            <thesaName> placekt </thesaName>
         </Keywords>
      </descKey>
      <dsConst>
         <LegalConst>
            <accConst> none </accConst>
            <useConst> none </useConst>
         </LegalConst>
      </dsConst>
   </idInfo>
   <distInfo>
      <distFormat>
            <Format>
            <name> AUTOCAD DXF; Point Data Flat File; ASCII Text file
</name>
            </Format>
      </distFormat>
      <distributor>
         <distCont>
            <RespParty>
               <mandParty>
                  <mpIndName> Steve Blasco </mpIndName>
                  <mpOrgName>  Geological Survey of Canada (Atlantic)
Marine Regional Geoscience  </mpOrgName>
               </mandParty>
               <contact>
```

```
                  <telephone>
                     <fax>(902) 426-4104</fax>
                  </telephone>
                  <address>
                     <postAdd> 1 Challenger Drive, P.O. Box 1006
</postAdd>
                     <city> Dartmouth </city>
                     <adminArea> Nova Scotia </adminArea>
                     <postCode> B2Y 4A2 </postCode>
                     <country> Canada </country>
                     <electMailAdd>blasco@agc.bio.ns.ca</electMailAdd>
                  </address>
               </contact>
               <resp ResponsibilityCode="distributor"/>
            </RespParty>
         </distCont>
      </distributor>
      <distributor>
         <distCont>
            <RespParty>
               <rpIndName> Mr. Glen Gilbert </rpIndName>
               <rpOrgName> Canadian Seabed Research Ltd. </rpOrgName>
               <contact>
                  <telephone>
                     <fax>902-827-2002</fax>
                  </telephone>
                  <address>
                     <postAdd>341 Myra Road, P.O. Box 299</postAdd>
                     <city> Porter's Lake </city>
                     <adminArea> Nova Scotia </adminArea>
                     <postCode> B0J 2S0 </postCode>
                     <country> Canada </country>
                     <electMailAdd>csr@atcon.com</electMailAdd>
                  </address>
               </contact>
               <resp ResponsibilityCode="distributor"/>
            </RespParty>
         </distCont>
      </distributor>
      <distribTrnsOps>
         <DigTransOpts>
            <medium>
               <name> Magnetic Tape/disk. </name>
            </medium>
         </DigTransOpts>
      </distribTrnsOps>
   </distInfo>
</metadata>
```

VITA

Candidates's full name : Ying Teng

University attended : BSc (Chemistry), 1993
Zhejiang University
Hangzhou, P.R.China

MSc (Chemistry), 1998
University of New Brunswick
Fredericton, Canada

Publications:

1. Nickerson, B.G. and Teng, Y. "A Formal Grammar for Geographic Information Metadata", UNB Faculty of Computer Science Technical Report TR00-132, Feb. 2000, 25 pages.

2. Teng, Y. and Nickerson, B.G. "XML (eXtensible Markup Language) for Geospatial Metadata", UNB Faculty of Computer Science Technical Report TR00-131, Jan. 2000, 34 pages.

3. "High resolution Laser Ablation and Molecular Beam Spectroscopy of CoCl" (in publishing)

4. R. S. Ram, J. R. D. Peers, Ying Teng, A. G. Adam, A. Muntianu, P. F. Bernath and S. P. Davis, "Laser and Fourier Transform Emission Spectroscopy of the G^4M-X^4M System of TiF" *J.of Mol. Spectrosc.* **184**, 186-201, 1997

5. Haoran Li, Ying Teng and Shijun Han, "Bubble Point Pressure Measurement for Isothermal Quaternary System and Prediction of Vapour-Liquid Equilibrium" *Chinese J. of Chem. Eng.*, **3(1)** 51-56, 1995

6. Haoran Li, Ying Teng and Shijun Han "Bubble Point Measurement for System Chloroform-Ethanol-Benzene by inclined ebulliometer" *Fluid Phase Equilibria* **113**, 185-195, 1995

Conference Presentations:

1. "XML to Support Query Processing for Geospatial Metadata", GEOIDE Second Annual Conference, Calgary, May 24-25, 2000.

2. "Use of XML for Query Processing in Web-Oriented Geospatial Data Warehouse", Atlantic Institute Graduate Student Annual Seminar, Quebec City, June 8-9, 1999.