1-1-2009

Geological data handling using Oracle 3D : a study on data services and management

Nedim Oren
Ryerson University

GEOLOGICAL DATA HANDLING USING ORACLE 3D: A STUDY ON DATA SERVICES AND MANAGEMENT

by

Nedim Oren

BSc, Geomatics Engineering, Istanbul Technical University, TURKEY, 1993

A thesis presented to Ryerson University

in partial fulfillment of the

requirements for the degree of

Master of Applied Science

in the Program of

Civil Engineering

Toronto, Ontario, Canada, 2009

Declaration

I hereby declare that I am the sole author of this thesis. I authorize Ryerson University to lend this thesis to other institutions or individuals for the purpose of scholarly research.

I further authorize Ryerson University to reproduce this thesis by photocopying or by other means, in total or in part, at the request of other institutions or individuals for the purpose of scholarly research.

GEOLOGICAL DATA HANDLING USING ORACLE 3D: A STUDY ON DATA SERVICES AND MANAGEMENT

Nedim Oren, MASc, Civil Engineering, 2009

Department of Civil Engineering

Ryerson University

Abstract

Efficient management of 3D geological and subsurface models require a robust 3D data modeling environment which can provide the necessary functions and flexibility to enable accessing 3D models in a collaborative work environment through the Internet. This allows geoscientists and geo-engineers to work collaboratively for better, informed decisions. Today, there is no data standard that satisfies the entire 3D geological modelling requirement in a collaborative work environment.

This thesis presents the result of a research project that focuses on identifying modelling and analytical requirements of geological models and the usability of existing technologies for both database management and applications that allow sharing 3D models in a collaborative modelling environment. Specifically, it examines current 3D data models and how they can fit into the requirements of the 3D geological modeling. Based on identified system requirements, an integrated solution prototype has been implemented that allows large-scale 3D data management and provides real-time Internet access to the underlying 3D models.

Acknowledgements

First of all, I would like to express my great appreciation and thanks to my supervisor, Dr. Songnian Li for his valuable advice and ideas as well as the guidance he has provided during the time of my thesis study. I also appreciate that he provided me with the great opportunity to continue my study in the field of Geographic Information Systems (GIS). I greatly appreciate his help in obtaining financial support for my studies. Without his support, I would not have been able to complete this work.

I also wish to thank Dr. Mike Chapman and other faculty and staff members in the Department of Civil Engineering at Ryerson University for their help and support. Thanks also to Desmond Rogan who patiently helped me with software installation and configuration.

The School of Graduate Studies at Ryerson University is acknowledged for granting me Ryerson Graduate Awards and the opportunity to conduct my research from 2007 to 2009. I would also like to thank Ms. Marry Neelands for her administrative support during my study.

Thanks also go to Eric (Zheng) Chang for providing the Collaboration Web Client (GeoLink) prototype and for his help. The advice and help provided by Baris Kazar of Oracle and Ken Bragg and other staff at Safe Software was priceless. Thanks also go to my friend Sandeep Prakash who offered his expertise and knowledge in geology and modeling.

Table of Contents

List of Tables

List of Figures

List of Acronyms and Abbreviations

API	Application Programming Interface
AEC	Architecture/Engineering/Construction
BIM	Building Information Models
CAD	Computer Aided Design
DBMS	Database Management System
DEM	Digital Elevation Model
ETL	Extract, Transform and Load
ESRI	Environmental Systems Research Institute
FME	Feature Manipulation Engine
GIS	Geographic Information Systems
GML	Geography Markup Language
GOCAD	Geological Object Computer Aided Design
HTML	Hypertext Markup Language
XML	Extensible Markup Language
KML	Keyhole Markup Language
KMZ	Compressed Version of the KML
LIDAR	Light Detection and Ranging
LOD	Level of Detail
NURBS	Non-uniform rational B-spline
OGC	Open Geospatial Consortium
OODBMS	Object-oriented DBMS
RDBMS	Relational Database Management System

SIMAL	Spatial Information Management and Applications Laboratory at Ryerson University
SQL	Structured Query Language
SQL/MM	SQL Multimedia and Application Packages
TIN	Triangulated Irregular Network
URL	Uniform Resource Locator
VTK	Visualization Toolkit
VRML	Virtual Reality Modeling Language
WFS	Web Feature Service
WMS	Web Map Service
PC	Point Cloud
PNG	Portable Network Graphics

Chapter 1. Introduction

With advanced data acquisition and computing technologies, a large amount of geospatial data has become readily available from many different sources. Different user requirements may be set by the different domain-specific application areas. GIS technology evolved rapidly from mainframe computing environments, to the desktop environment and finally Internet-based application architecture, which is currently used in many applications. Today, because of the recent improvements in the Internet technology and other enabling technologies, location-based solutions are incorporated in many daily tasks. A wide range of computing devices, such as personal computers, mobile computing devices and cell phones, can use these solutions.

Many well-established GIS tools can handle the common functionalities of GIS. Worboys and Duckham (2004) identify these common functions as capture, structuring, manipulation, analysis and presentation of the spatial data. Some advanced geoprocessing tools can also accomplish complicated analysis in 2D environment. However, in more complex situations such as geological modeling for the purpose of mineral exploration or groundwater modeling, 2D models may not characterize the phenomenon under consideration. In complex situations 3D models facilitates understanding of the reality through a 3D abstraction of complex geological and subsurface objects.

Despite the established use of GIS tools and functions for 2D space, challenges remain when modelling real-world objects in 3D space. These challenges exist within most of the common functionalities of 3D modeling, such as capturing the geological data, structuring it in a model allowing manipulation, and analysing and presenting data. Several commercial and open source applications are available for the visualization of 3D models. Beside visualization of 3D models, users also want to easily manipulate 3D data, perform GIS analysis in 3D space, and interpret the results for better management of subsurface resources. The next section describes the main problems of modelling subsurface objects in GIS environment.

1.1 Problems

GIS applications have long been a part of many resource management solutions, in both the private and public sectors. Most resource management activities benefit from analysis that would be impossible, or too expensive, without a GIS model built into a computing environment. In the last decade, specialized tools have been developed to perform specific 2D GIS tasks that help resource managers resolve domain-specific problems. However, certain more complex management activities in geological, hydrological, mineral exploration and petroleum domains cannot be efficiently performed without a proper 3D model of the complex subsurface phenomenon.

Today, well-established Computer Aided Design (CAD) and GIS systems can be used for visualization of complex 3D objects and visual inspection of the rendered 3D model, as well as the relations among included objects. Abdul-Rahman and Pilouk (2008) state that "a digital model must be capable of relating spatial and non-spatial aspect of reality and creating such a model as an artificial construction of reality in a computing environment requires a tool set to exploiting the technologies both of computer graphics and database management". Usually a group of collaborating geologists and scientists, located in dispersed geographic areas, generate geological models. To be able to make informed decisions, geologists and scientists need to work

collaboratively and share the same view of the 3D model. The Internet can provide the communication base for such collaboration and knowledge sharing.

One of the sub projects under the GEOIDE GeoTopo 3D project is to develop methods and tools for collaborative geomodeling and real-time sharing in a collaborative environment among a group of geographically distant geoscientists and geo-engineers. Such methods and tools can directly link geoscientists, exploration and data centers to allow faster analysis and decisions in situations where collaboration is a critical part of the process. Figure 1.1 shows the overall structure of the GEOIDE GeoTopo 3D project in relation to the other sub projects.

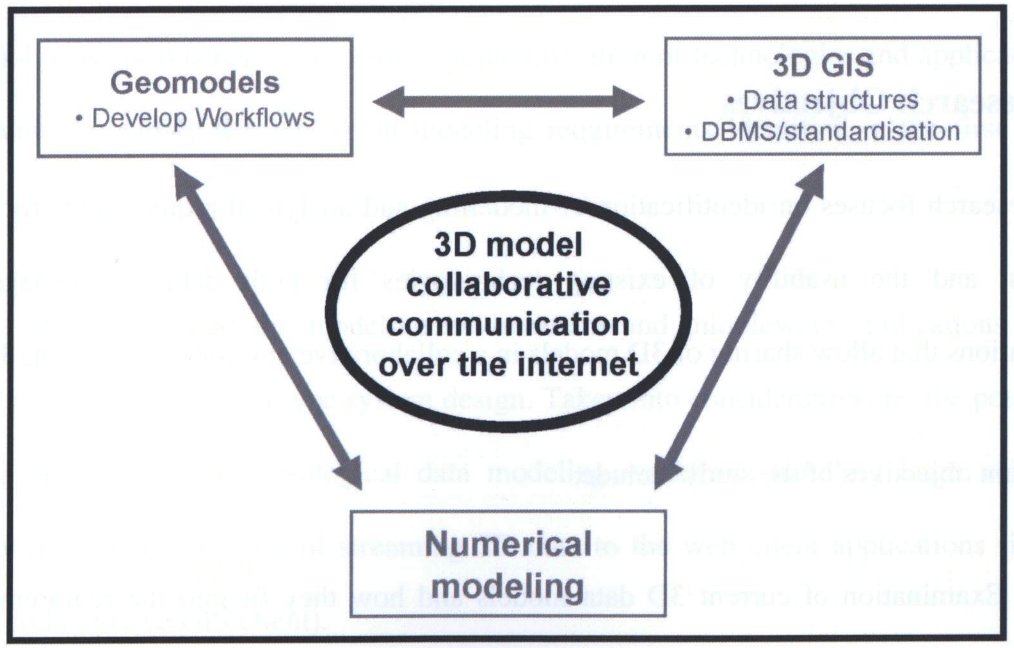

Figure 1.1 GEOIDE - GeoTopo 3D project structure [Blessent, 2006]

To realize such a system, a robust, efficient 3D data model is required to model 3D geological data which can, ideally, be stored in a Database Management System (DBMS). It is also important to develop methods and tools that allow different systems, either stand-alone or collaborative applications, to access 3D models, regardless of the data format of the underlying system.

These three aspects are important in this development:

1. Collaborative 3D system providing software support of model sharing;
2. DBMS allowing easy management of the large 3D models;
3. Tools and functions allowing manipulation of 3D models in either stand-alone or collaborative environments.

1.2 Research Objectives

This research focuses on identification of modelling and analytical requirements for subsurface models, and the usability of existing technologies for both database management and applications that allow sharing of 3D models in a collaborative environment over the Internet.

The main objectives of the study include:

1. Examination of current 3D data models and how they fit into the requirements of 3D subsurface and geological modeling, especially with the recent release of Oracle 3D;
2. Design of a framework for an integrated solution that can provide real-time 3D data management and access capacity over the Internet;

3. Prototyping of an integrated solution using Oracle 3D and other enabling tools to allow sharing of the 3D models in a collaborative environment over the Internet.

1.3 Research Methodology

The research began with a literature review that included relevant research papers and studies in 3D GIS, 3D data models, and geological and subsurface modeling concepts. The purpose of this phase was to build an understanding of the terminology and knowledge required to carry out this research, to summarize existing and proposed solutions for modelling 3D geological objects, and to provide a better understanding of geological modeling requirements.

The second phase of this research focused on identification of technologies and applications that may, potentially, satisfy the geological modeling requirements identified in the first phase of research.

The third phase identified the modeling environment and middleware applications that can potentially be used in the prototype system design. Taken into consideration are: the potential for managing the complete 3D geological data modeling workflow, handling large volume 3D geological data, and capability of streaming 3D data to the web client applications (including GeoLink collaborative GIS client).

The fourth phase installed and configured components of the prototype. Upon completion of the system implementation, a test dataset was processed and loaded into the Oracle database. The Feature Manipulation Engine (FME) Server services were created to stream the 3D geological models to the Virtual Reality Modeling Language (VRML) clients, including the GeoLink Prototype.

The prototype system was tested in the fifth phase. The main functionalities expected from the system were tested with several VRML web client applications. Figure 1.2 shows the overall procedure followed in this research.

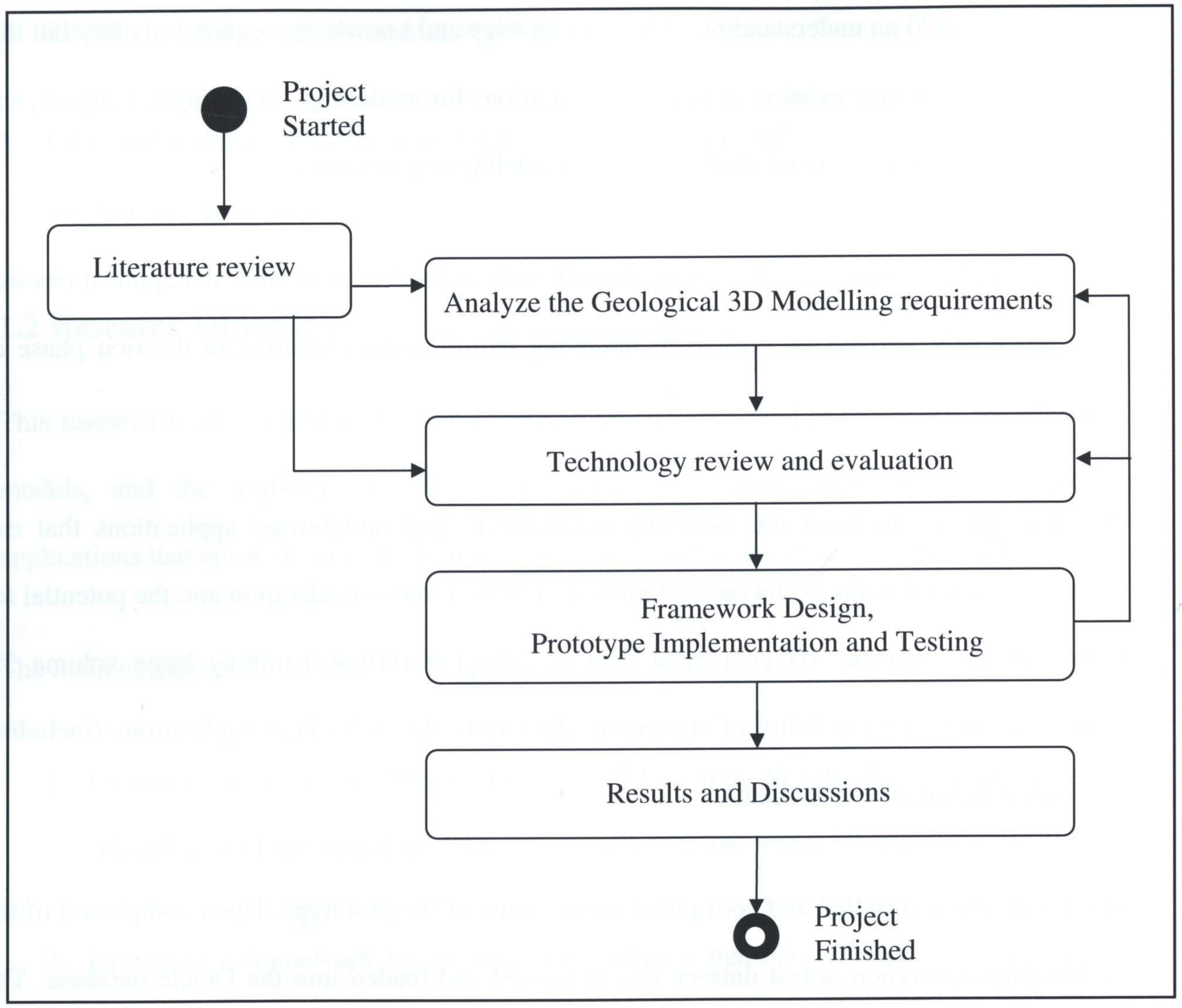

Figure 1.2 Research Methodology

1.4 Thesis Organization

The first chapter describes the problem of 3D modelling, collaboration, real-time 3D model sharing, research objectives and thesis research methodology.

The second chapter discusses concepts of 3D data modelling, a review of existing 3D systems, current implementations, and new developments and trends in 3D modelling.

Chapter 3 focuses on subsurface and geological 3D modelling processes. It discusses characteristics, modelling workflow and 3D geological data modeling requirements.

Oracle's 3D data management capacity and the detail of 3D data types, 3D spatial operations, existing framework for 3D data modelling that may fit into the modelling of the 3D geological objects are discussed in the fourth chapter.

The fifth chapter looks at the framework design and prototype system implemented for managing and sharing of 3D geological models using Oracle and FME solutions.

The sixth chapter includes concluding remarks and recommendations for further study.

Chapter 2. 3D Data Modeling

2.1 The Concept of 3D Modeling

The common definition of GIS underlines GIS' basic functionality, regardless of architecture or underlying data models. These are the common tasks of GIS functionality: capture, structuring, manipulation, data analysis, and data presentation. Because object relationships in 2D space are relatively simple compared with 3D space, most well-established and commonly-used 2D GIS tools can handle the functionality expected from a 2D GIS system. In contrast, these same tools are far from providing basic GIS functionality for objects modeled in 3D space. This is due to the complexity of real-world objects in 3D space. For example, existing 3D systems cannot perform commonly used spatial operators, such as intersection or overlay analysis.

Organization of 3D data and governing rules of modelled objects and their relationships are very important for the realization of a 3D system. A well-designed data model can provide a robust base for realizing complex spatial functions and operations. In the literature, both topological and geometric models have been reported as 3D data modeling options. Geometric data models store the geometry and require that all topological relationships are derived by computation from that geometric model. On the contrary, topological models implicitly store the relationships and geometry needed to be derived from the topological relationships for rendering and other metric operations.

In practice, both implementations have strong and weak areas. For example, deriving geometric model of 3D objects from the topological relationships is highly time consuming and needs extensive computations. However, because the rendering engines need the geometric model for

the visualization, visually-oriented 3D systems most likely prefer using the geometric models. In contrast, topological models allow running some of the spatial queries to the Structured Query Language (SQL) database without any computation, and provide faster results for these queries. For this reason, systems that deal with the spatial relationships of the model prefer using topological models as the underlying data structure. In certain other models, such as CityGML, storing both geometric and topological model is a suggested solution, but synchronization of both models remains a major problem.

It is inevitable that 3D GIS will be realized with integration of geomodeling tools and data management systems [Apel, 2006]. This type of integration can be achieved by splitting the intensive workload of the 3D system between DBMS and geomodeling tools. It is possible to implement some of the 3D GIS functions at the DBMS level to avoid costly data transfer processes between the database and the front-end-applications. Execution of some common computationally expensive functions in the DBMS will greatly help realize web-enabled 3D GIS solutions. In the literature, several research efforts have aimed to prove the possibility of implementing some spatial operators in DBMS, but so far none of the proposed solutions has been commonly accepted or implemented [Zlatanova et al., 2004].

Implementation of some GIS functions and operators at the database is related strictly to the underlying data model and data organization. Therefore, a standard and interoperable data model is needed to implement suggested functionality and operators. In the next section, the data models and options for organizing 3D data will be discussed.

2.2 Review of 3D Data Models

The historical experience gained from CAD systems provides the basis for 3D modeling in general. In the literature, several distinct groups of 3D modeling techniques have been reported for volume modeling. Based on distinct definitions, Latuada (2006) groups modeling techniques into the following major categories.

- **Sweep Representation:** objects are represented by sweeping a definition area or volume along a defined trajectory.

- **Primitive Instancing:** a technique that uses a set of predefined shapes or mathematical primitives positioned in space without intersecting.

- **Constructive Solid Geometry:** a combination of primitive shapes using set theory operators such as union, intersection and difference. Volumetric and parametric primitives compose object geometry. Constructive solid geometry is widely used in many applications and most CAD software provides extensive support for this representation and best suits the Architectural Design process. Figure 2.1 shows the basic volumetric and parametric primitives that are used to model 3D objects in constructive solid geometry models.

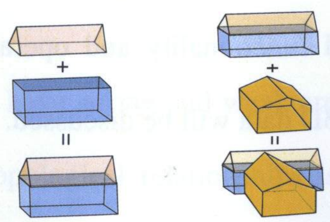

Figure 2.1 Constructive solid geometry representations

- **Boundary Representation:** In this option, objects are defined by their bounding surfaces where object's boundary is defined by contiguous simple or complex surface objects. This model can be described as accumulation of related surfaces to enclose the volume of the object. Because all surfaces are explicit in this representation, textures can be draped directly onto them and 3D polygons can be efficiently rendered using hardware acceleration. The topological relationships and the restrictions on the types of primitives can be applied based on the methodology and intended use of the model. Figure 2.2 shows boundaries that represent basic building structures where all surfaces are enclosing a 3D volume based on the restrictions and rules set for the model.

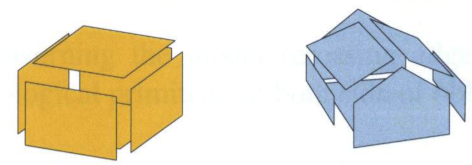

Figure 2.2 Boundary representation

- **Spatial Occupancy Enumeration:** This technique represents objects by uniting a set of cells. Each cell is a primitive, simple, regular or irregular shape. The adjacent cells are connected, but do not intersect.

- **Cell Decomposition:** The voxel approach is an example of the cell decomposition method. This volume modeling technique is broader in its scope, and applies to any shape and size of cell.

Although all these methods are used in different application areas, in theory a hybrid approach combining boundary representations and constructive solid geometry appears to be suitable in many domain-specific modeling areas of 3D GIS [Abdul-Rahman and Pilouk, 2008]. Boundary representation is commonly used for large volume 3D data collections and object reconstruction techniques employing Light Detection and Ranging (LIDAR), photogrammetry, and remote sensing technologies. Currently, most 3D systems rely on Boundary Representation and do not support Constructive Solid Geometry. Organizing 3D data as boundary representation appears to be the most flexible option for implementing a true 3D GIS in most GIS domain areas and so geological modeling.

Topological and geometric models are two common methods for modeling and storing geographic data and which can both be used for modeling 3D objects. The following sections will discuss both topological models and geometrical models in detail.

2.2.1 Topological Models

Objects can be modeled by using their topological properties, an approach that's best suited to complex spatial analysis. Most 2D GIS systems are realized on underlying topological models. Because 3D space introduces new issues, such as primitives, rules, and constraints representing objects and their relationships, existing well-defined 2D topological models may not be suitable for 3D modeling [Zlatanova et al., 2004].

The implicit or explicit description of many spatial relationships/operations such as inclusion, adjacency, equity, direction, intersection, and connectivity are expected within topological models. These relationships/operations must also be kept up-to-date in a sustainable form when the model is modified. The literature has reported several 3D abstractions for topological models, which have strong and weak points for representing spatial objects within the 3D space

[Zlatanova et al., 2002]. Some of these reported models are also examined in various experimental studies [Zlatanova, 2000; Heuel and Kolbe, 2001; Store and Zlatanova, 2003; Zlatanova et al., 2004; Zlatanova, 2005; Oosterom et al., 2002].

The following parameters characterize different topological structures, and can be used as comparison criteria for different models [Oosterom et al., 2002]:

- Dimension of embedded space (2D, 2.5D, 3D, 4D-Time added);
- Topological primitives used (node, edge, face, volume);
- Orientation of elements considered (directed or not);
- Explicit topological relationships stored (part-of, in, on.);
- Topological rules governing the model (crossing edge allowed, dangling elements allowed or same topological primitive on both side of boundary allowed).

Table 2.1 provides a short inventory of the most common topological structures reported in the literature [Oosterom et al., 2002]. As some 3D modelling domains may have specific requirements, no single topological model is satisfactory. Outstanding issues of 3D GIS include space partitioning, supported objects and primitives, and constructive rules in representing the objects and the relationships among them.

Where, and how, the model is to be physically stored is an important aspect that will effect the whole system design and needs to be carefully considered. The recent objective of Architecture/Engineering/Construction (AEC) is to bridge the gap between CAD and GIS. DBMS can provide a common ground, which different domain-specific applications can use to store and manage 3D models [Zlatanova, 2005]. This common ground can also provide a solid base for the development of interoperable 3D modeling standards. GIS systems use specialized DBMS systems to mange the underlying data. By using DBMS for data management, both GIS and CAD can integrate these systems with accepted data standards. Because of the size of 3D data involved in modeling geological objects, and the complexity of relationships in 3D space, DBMS is most likely the best option for storing 3D topological models for the geological domain.

If chosen, a topological model either can be mapped directly into relational tables or modeled as an object-relational DBMS. As a third option, the same model can be implanted as an object-oriented DBMS (OODBMS). This latter option is suitable for most 3D GIS domain applications, but requires extensive low-level programming in implementation.

Table 2.1 Inventory of topological structures (source: [Oosterom et al., 2002])

		Primitives Used	**Topological Tables**	**Explicit Relationship**	**# of Tables**	**Rules**
TIN	2D	node, edge	node, edge	no	2	Planar Partition
Wing-edge	2D	node, face	edge, face	no	2	Planar Partition
Whell (Chain)	2D	node, face	edge, face	no	2	Planar Partition
3D FDS	3D	node, arc, edge, face	arc, edge, face	node-on-face, node-in-volume, arc-part of-line arc-on-face, arc-en-volume	8	Space Partition
TEN	3D	node, arc, triangle, tetrahedron	arc, triangle, tetrahedrons	tri-part of-surf, arc-part-of line	5	Space Partition
Cell-tuple	3D	0-cell, 1-cell 2-cell, 3-cell	cells	no	1	Space Partition
SSS	3D	node, face	face, line, surface, volume	node-in-volume, face-in-volume	6	Space Partition

2.2.2 Geometric Model

A 3D model can also be represented in DBMS using a geometric data type. The geometric data type and functions to operate on the model can be either created by the user, or provided by the DBMS as a native data type. User-defined geometric data types and functions are commonly written in C, C++ or Java in order to maintain an acceptable performance level [Chena at al., 2008]. Generally, DBMS vendor-provided geometric types are also written in one of these lower-level programming languages, but because of the lower-level integration, geometric types can potentially perform better.

Among other main DBMS vendors, Oracle provides a native 3D data model and several 3D data types. In general, a data type is defined as a class object with functions and properties. It may also allow spatial operations to be performed on them with SQL queries. With its well-defined properties, and compatibility with the Open Geospatial Consortium (OGC)'s simple feature specification, Oracle 11g's new data type can provide a solid platform for the geological 3D modelling process.

2.3 Geometric vs. Topological Data Models

To some degree, spatial operations and functions can be performed on both geometric and topological 3D data models. A comparison between these two types of data models is possible, based on the allowable 3D spatial operations. Chena at al. (2008) classifies the computational-geometry and metric operations that are only possible within the geometrical 3D model because of the need the mathematical computations for these types of functions. It is nearly impossible to implement these 3D computational-geometry and metric operations on any known 3D

topological data model. Figure 2.3 shows a possible comparison between geometric and topological data models.

On the other hand, once the topological model is constructed, performance of spatial relationship operations on that model is faster than performance of the same types of operations on geometric models. Moreover, topological models are likely more suitable for integrated 3D modeling approaches, and may not be suitable for modeling different geological entities as different data layers, similar to the 2D GIS layers.

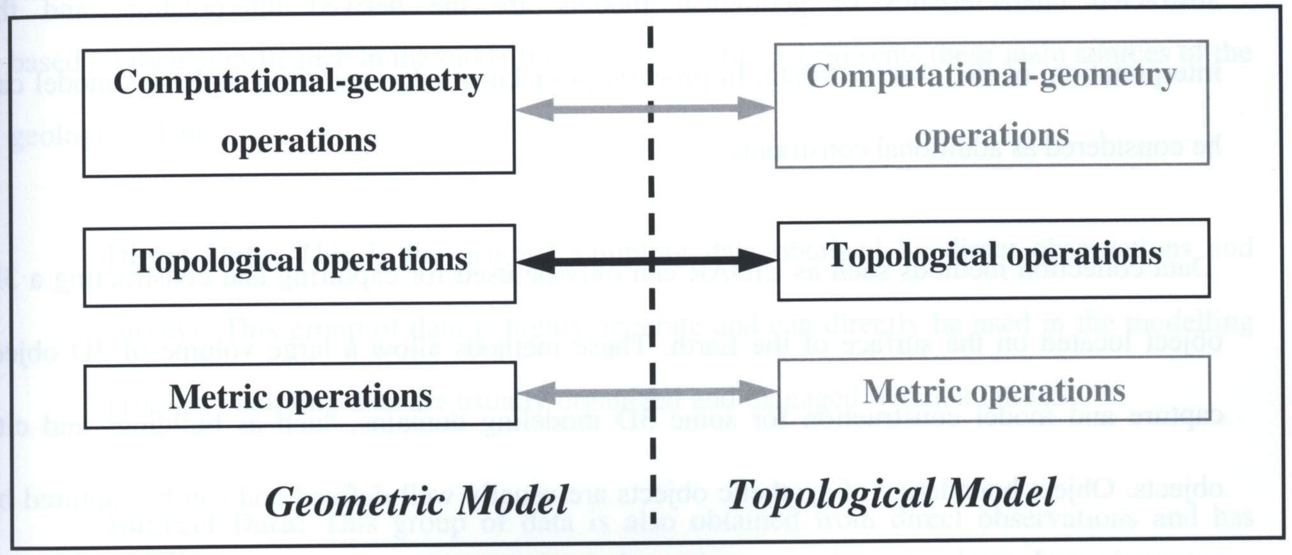

Figure 2.3 Comparison between geometrical and topological models (source: [Chena at al., 2008])

Chapter 3. Geological 3D Modeling Requirements

The geological modeling process uses mathematical methods to represent and integrate topology, geometry, and physical properties of geological objects [Pouliot et al., 2003]. Modeling subsurface geology is a most complex and time consuming process compared to any other domain-specific 3D modeling process. This difficulty is due to the inherited difficulties of collecting subsurface observations, and the intensive manual interaction with the observation data required to create 3D approximations of geological objects. Apel (2006) states that the abstracted characteristics of geological models are the derived interpolation, and the interpretation, of the observed data. In process, prior knowledge and the geophysics model can be considered as additional constraints.

Data collection methods such as LIDAR can only be used for capturing and constructing a 3D object located on the surface of the Earth. These methods allow a large volume of 3D object capture and model construction for some 3D modeling domains, such as buildings and city objects. Object boundaries of synthetic objects are usually well defined and can be captured by automatic and semi-automatic processing of the collected raw data. On the other hand, subsurface data collection is mostly based on well logs that are very expensive and time consuming to collect and to process in order to create the 3D approximations of geological objects.

3.1 Sources of Geological Data

In geology, observed data are usually defined as a finite set of points in 3D space. These observations are usually stored in field books and include qualitative descriptions of petrographic

compositions, indicators of stratigraphic faces and age like fossils, fabric, and structural descriptions. Measurements, drawings, images and information related sampling locations are also commonly collected data components of geology. Furthermore, another important source of geological data is age data and other quantitative data obtained by laboratory examination of the collected samples such as geochemical, petrochemical, and petrophysical data [Marcus, 2004].

The data for representing geology comes from different sources including boreholes, cross-sections, classic geology maps, and surface characterization data such as Digital Elevation Models (DEM). Wu et al. (2005) characterises geological data into three distinct, generic groups based on their significance in the modelling process. Table 3.1 presents these main sources of the geological data.

Direct Data: This is the original sampling data obtained by direct observations and surveys. This group of data is highly accurate and can directly be used in the modelling process. Sampling data are usually organized and managed within a DBMS.

Indirect Data: This group of data is also obtained from direct observations and has different precision with different resolution of graphs. It can be used as constraints in interpretation of direct data. This group of data is mostly collected from observations on the Earth's surface, such as boundaries, faults, folds and DEM derived from geological maps, topographic maps and structural geology maps, 2D and 3D seismic-reflection data. Exploring data can be categorized into this group. This type of data, usually collected by digitising hardcopy maps, is commonly stored in file formats.

Assistant Data: Assistant data is used in the process of 3D modelling and includes primitives that represent geological structure in models, texture maps, satellite or aerial imagery, and scanned maps.

Table 3.1 Sources of geological data (source: [Wu et al. 2005])

Category	Data	Origin/ Collection method	Common Storage
Direct Data	Qualitative descriptions of the petro-graphic compositions; indicators of stratigraphic facets and age like fossils; fabric and structural descriptions; measurements; drawings; images; sampling locations	Original sampling data obtained by direct observations and surveys	Usually organized and managed within a DBMS.
Indirect/derived Data	Age data, quantitative geochemical and petrochemical and petrophysical data, boundaries, faults, folds and DEM derived from geological maps, topographic and structural geology maps, 2D/3D seismic-reflection data, exploring data	By direct observations and laboratory examination of the collected data	These types of data usually need to be digitized from the hardcopy maps and commonly stored in file formats.
Assistant Data	Primitives that will represent geological structure in the model, texture maps, satellite or aerial imagery, scanned maps.	Not specific	Vector or image file formats

3.2 Characteristics of Geological Data and 3D Models

Geological data is collected for modeling geological characteristics of the subsurface. Its main characteristics can be summarized as follows:

- Geological data accumulates over a long period of time where technology and methodologies for collecting the data change. Because of change in collection methods and technology, not all existing data may be suitable for modeling. Depending on the level of interest, only some recent and older available data is adequately accessible or accurate to be included in the geological modelling process [Wu et al., 2005; Kaufmann and Martin, 2008].

- Data collection methods such as drilling and geophysics surveys are costly compared to technologies such as LIDAR and photogrammetry which allow the collection of large amounts of surface data for the characterization of 3D objects located on the surface of the Earth.

- Processing geological data for 3D modeling requires extensive human interaction to interpret and extend it in order to construct abstractions of subsurface objects in 3D space. Because interpretation is extensively used in 3D modelling of the subsurface, the original input data needs to be available along with the resulting model for future consideration and re-processing the model if new data becomes available.

- Geological models generated from data available at the time of building represent the best approximation of the geology given the data available at that time. The dynamic character of geological models needs to be considered in the system design process.

3.3 3D Geological Modeling Workflow

Because of the difficulties inherent in complex subsurface geometries, poor data density, irregular data distribution, and heterogeneous sources of geological data, the 3D geological

modeling process is time consuming and requires intensive human interaction. Construction of geological 3D models is complex because of the requirement to interpret low-density observation data and natural boundaries of geological objects, which are subjective to the interpreter. The resulting model is an approximation, only as good as the data and the interpretation skill used in its creation [Perrin at al, 2005; Pouliot et al., 2003].

Pouliot et al. (2003) divides the methodology of constructing 3D geological models into three generic phases:

Data gathering, setup and cleaning of objects in a 2D universe: This phase involves gathering the available data and converting it into usable forms. This phase includes format conversion, coordination of system transformation, and identification of the scale, quality, and precision of each distinct dataset. This step is not complicated and can be performed by certain software tools. However, the process may take a long time to complete and may involve manipulation of two-dimensional spatial object primitives (0D, 1D, 2D) for representing geological features such as drills, faults, rock type contacts, elevation points, and contour lines.

Projection of planar objects in the 3D universe: This phase involves interpolation/extrapolation of the gathered and cleaned data in order to construct the vertical and parallel sections that will be later be used to form the 3D volumetric objects in the third phase of the modelling process. The performing geologist's knowledge of structural geometry, and geological history of the study area, is very important in this phase. This phase is highly time consuming, complex, and requires manual interaction with a proper geomodeling tool.

Construction of continuous volumetric objects: In this final phase of the 3D geological modeling process, approximations of the geological 3D objects (volumetric) are formed by connecting vertical and parallel sections that were built in the previous phase. These approximations, done with the aid of a modeling tool, are highly dependent on decisions and interpretations of the processing geologist.

Table 3.2 summarises the three generic phases of the 3D geological modeling process.

Table 3.2 Phases of geological modeling

Phases of Geological Modeling	Action Performed	Time Intense	Process Complexity	Manual Interaction	Tools	Dependencies
Phase 1: Data setup and cleaning of objects	format conversion; coordinate transformation; scale, quality and precision identification	Y	M	L	computer software packages	none
Phase 2: Projection of planar objects in the 3D universe	vertical section construction; parallel cross-sections construction	Y	H	H	done manually; computer software packages	prior knowledge and conceptualization of the study area; available information; spatial orientation and geometry of geological objects;
Phase 3: Construction of continuous volumetric objects	connect each cross-section; build continuous 3D geological units	Y	M	H	done manually; computer software packages	prior of the study area; compatibility of the sections; conformity with the structural models

Y: Yes; **N**: No; **L**: Low; **M**: Medium; **H**: High

3.4 System and Data Modeling Requirements

In general, any 3D GIS system should provide functions such as the construction of a 3D data model from disparate input data, permit maintenance of existing models, facilitate effective 3D visualization, surface illumination, and texture mapping, and provide spatial analysis, for example to enable the calculation of volume, surface area, center of mass, optimal path, and spatial and non-spatial search capacity [Abdul-Rahman and Pilouk, 2008].

The following functionalities are identified as the basic requirements for geological models [Marcus, 2004; Apel, 2006; Kothuri et al., 2007; Kaufmann and Martin, 2008].

- Integration of various sources of 2D-3D spatial and descriptive data.
- Representation of topological relationships, geometry and material properties of geological structures.
- Capacity to build sophisticated structural models with distinct heterogeneous input data.
- Capacity to update 3D geological models by re-interpolation, when new input data is introduced.
- Visualization of 3D models.
- Capacity to model functionalities including advanced geostatistic methods.
- Extensive investigations of the model by the means of topological and metric queries.
- Integration of concepts such as fault intersection rules which can support the modeling process.
- Interoperability.

Chapter 4. Oracle 3D Data Management and FME Server

For the last couple of the years Oracle has provided extensive data management, retrieval, and analysis solutions in the 2D GIS domain. With its 11g release, Oracle moved into the 3D domain by implementing 3D data types, and providing some basic operators and functions at the database level. This development will change 3D GIS system considerations, and will contribute towards the realization of a fully functional 3D GIS. If the functions provided are further developed, the Oracle Database will be able to handle the entire geological modeling process.

Although it is impossible to create a system capable of providing all identified functionalities, nonetheless a functional 3D GIS system specialized in geology can be achieved through integration of DBMS and geomodeling tools which allow the user to interact with the underlying DBMS, create 3D geological objects, as well as perform analysis on the resulting 3D geological models.

As a second enabling technology, the FME Server centralizes spatial data transformation and distribution tasks. The FME Server facilitates the translation, transformation, and integration capabilities of spatial data flows [FME, 2008]. The following section will discuss Oracle's data management capacity, the FME Server architecture, as well as key capabilities supported by this technology.

4.1 Oracle 11g and 3D Data Management

Oracle, as a major DBMS vendor, provides 2D spatial data management capacity by supporting spatial data types of GIS and CAD applications. With the new 3D spatial data types offered by

Oracle 11g, a fully functional 3D data modeling system is ever more feasible. Table 4.1 summarises the spatial data geometries available in Oracle's SDO_GEOMETRY data type.

A 3D model can be implemented as an object-relational model in a DBMS. This implementation can be realized by extending the existing data types of the underlying DBMS [Zlatanova, 2000; Zlatanova et al., 2004; Stoter and Zlatanova, 2003; Zlatanova, 2005; Pu and Zlatanova, 2006; Zlatanova and Stoter, 2006]. A native 3D data type, supported by a mainstream database, can provide many benefits to implementing 3D GIS systems and data modelling activities. As summarized by Kothuri et al., (2007), this newly released capacity:Eliminates the need for dual architectures, because data can be stored in the same way in a unified data storage;

- Allows usage of SQL for accessing a relational database.
- Defines a spatial data tape, which is equivalent to the spatial types in the OGC and SQL Multimedia and Application Packages (SQL/MM standards).
- Standardizes access to spatial data by implementing well-known formats for spatial data specification, such as SQL/MM and Geography Markup Language (GML).
- Provides scalability, integrity, security, recoverability.
- Supports advanced user management features for handling spatial data.

Table 4.1 Oracle SDO_GEOMETRY 3D Geometries (source: [Kothuri et al., 2007])

Spatial Geometry Type	Explanations
UNKNOWN_GEOMETRY	Oracle Spatial ignores this geometry.
POINT	Geometry contains one point.
LINE or CURVE	Geometry contains one line string that can contain straight or circular arc segments, or both.
POLYGON or SURFACE	Geometry contains one polygon with or without holes, Foot 1 or one surface consisting of one or more polygons. In a three-dimensional polygon, all points must be on the same plane.
COLLECTION	Geometry is a heterogeneous collection of elements. COLLECTION is a superset that includes all other types.
MULTIPOINT	Geometry has one or more points. (superset of POINT.)
MULTILINE or MULTICURVE	Geometry has one or more line strings. (MULTILINE and MULTICURVE are synonymous in this context, and each is a superset of both LINE and CURVE.)
MULTIPOLYGON or MULTISURFACE	Geometry can have multiple, disjoint polygons (more than one exterior boundary) or surfaces (MULTIPOLYGON is a superset of POLYGON, and MULTISURFACE is a superset of SURFACE).
SOLID	Geometry consists of multiple surfaces and is completely enclosed in a three-dimensional space. It can be a *cuboid* or a *frustum*.
MULTISOLID	Geometry can have multiple, disjoint solids (more than one exterior boundary). (MULTISOLID is a superset of SOLID.)

4.1.1 3D Data Retrieval, Query and Analysis

In the first release of Oracle 11g, certain relationship functions such as *Distance*, *Closest Point* and *AnyInteract* were supported for 3D data types. The distance function is used for computing the distance between two three-dimensional geometries. The *ClosestPoints* function returns the closest pair of points on two geometries that were passed into the function. *AnyInteract* is the only relationship function that is supported in the current release of Oracle; it determines whether two three-dimensional geometries intersect [Kothuri et al., 2007].

The first release of the Oracle 11g also has some geometric analysis functions – *length*, *area* and *volume*. The *length* function computes the length of a three-dimensional shape; the *area* function computes the area of a surface or the sides of a solid, and the *volume* function computes the volume of a three-dimensional solid [Kothuri et al., 2007].

Even though the functions provided by Oracle 11g are fewer that those needed for implementing an ideal geological 3D system, they can nonetheless be used as a basis for development of advanced functions.

The J3D_Geometry class, a subclass of JGeometry, allows manipulation through several methods of 3D objects such as surfaces and solids. Table 4.2 summarises 3D processing functions provided by Oracle 11g [Kothuri et al., 2007].

Table 4.2 3D Geometry processing functions (source: [Kothuri et al., 2007])

Method	Purpose
AnyInteract	Determines whether two three-dimensional geometries interact in any way
Extrusion	Returns a three-dimensional geometry extruded from a two dimensional polygon
ClosestPoints	Computes the closest points of approach between two three-dimensional geometries
getMBH	Returns the three-dimensional bounding box of a three dimensional geometry
Validate	Verifies the validity of a three-dimensional geometry
Area	Computes the area of a surface or of the sides of a solid
Length	Computes the length of a three-dimensional shape
Volume	Computes the volume of a three-dimensional solid
Distance	Computes the distance between two three-dimensional geometries

4.2 FME Server: Key Capabilities and Services

4.2.1 Key Capabilities

FME Sever allows data to be remodelled on-the-fly and eliminates the need for human interaction. After setup, the data is first dynamically transformed based on user requests, and then presented to users via a web interface. This on-the-fly data transformation capacity of the FME Server supports a wide range of domain-specific data models, such as GIS, CAD, Building Information Models (BIM) and 3D. It allows on-the-fly access to custom views of the underlying

2D, as well as 3D, data. FME Server services can be accessed either through an online spatial data downloading service, or through a web streaming service. With downloading services, clients use a web browser to download the requested spatial data in the format and projection of their choice. Clients can use this option as an offline data resource of up-to-date data to which the FME server is connecting. More importantly, FME Server can dynamically transform spatial data into web-friendly formats such as Keyhole Markup Language (KML), GeoRSS, and GeoJSON, VRML. This capability of FME Server allows streaming 3D datasets directly into web applications. ESRI ArcGIS® Server, Google™ Earth, Google™ Maps, Microsoft® Virtual Earth™, OpenLayers, JavaX3D Clients, and VRML viewers are some of the client applications that can use data streamed by FME Server.

FME Server fully supports OGC data standards with the option for data owner to keep propriety data models and while automatically transforming the data into a different 3D data format. FME Server provides Java and C++ Application Programming Interface (API)'s which allow components to be integrated into any current web environment, including 3rd party web mapping applications, such as: ESRI ArcGIS® Server, ESRI ArcIMS®, Intergraph GeoMedia® WebMap, MapInfo MapXtreme®, and Autodesk MapGuide®.

The following outlines the key capabilities and functions of the FME Server [FME, 2008];

— Enable on-the-fly data transformation via the web;

— Create spatial data download services;

— Deliver spatial data as a web streaming service;

— Share common spatial Extract, Transform and Load (ETL) tasks.

Figure 4.1 shows the operational concept of the FME Server.

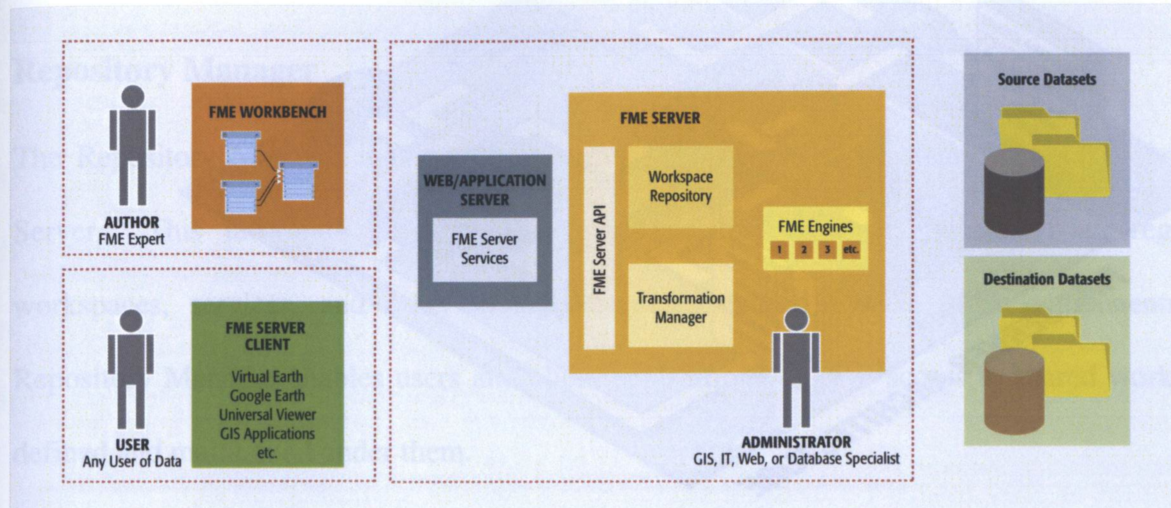

Figure 4.1 Operational concept (source: [Safe Software, 2008a])

4.2.2 FME Server Architecture

FME Server architecture harmonizes multiple components such as Transformation Manager, FME Engines, Repository Manager, FME Server Services, FME Server Console, and FME Server Services. Each component performs a dedicated task to accomplish spatial data transformation and distribution tasks. Depending upon the task, object-oriented architecture allows maximum scalability of an application to deliver the intended service. Figure 4.2 illustrates FME Server architecture and the role of each component. These will be further discussed below.

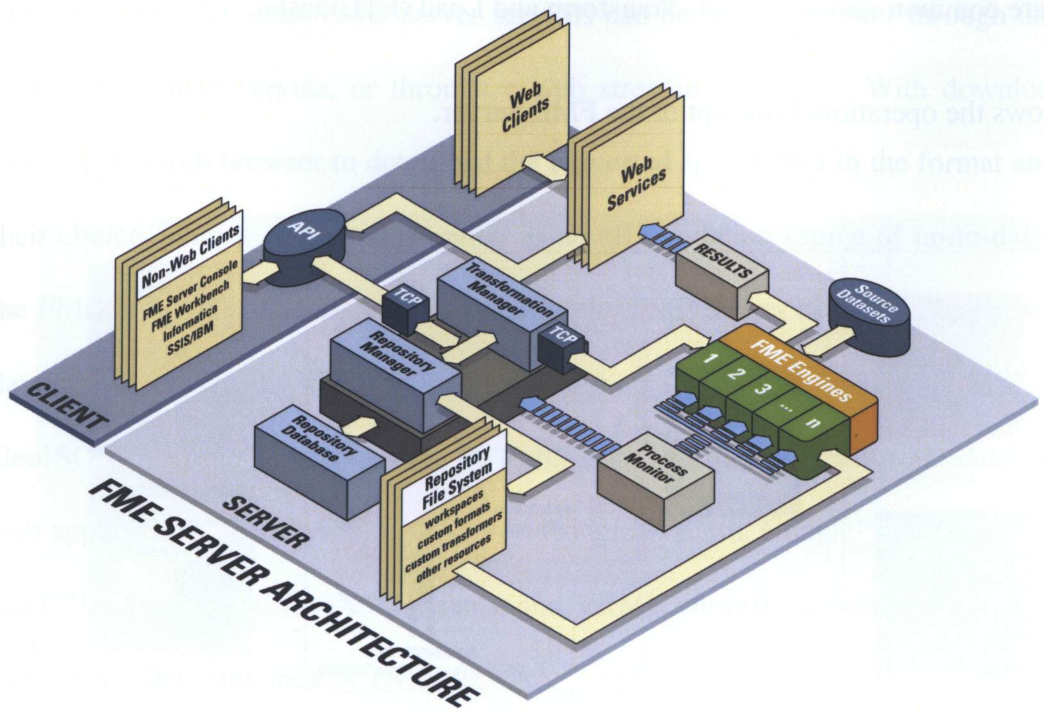

Figure 4.2 Role of FME Server components (source: [Safe Software, 2008c])

Transformation Manager

The Transformation Manager acts as a central mechanism that distributes the workload assigned to the FME Server at a given time. Every web, or non-web, client request eventually passes through the Transformation Manager. An incoming request goes to the first available FME Engine capable of performing the requested task. After the task completed, the Transformation Manager ensures that the FME Server client receives a response from the FME Engine.

FME Engines

Based on system requirements and configuration, an FME Server deployment contains one or more FME Engines. FME Engines perform the requested transformation task by running the

workspace specified by the client. Increasing the number of FME Engines increases the spatial ETL processing resources available to satisfy incoming client requests. FME Engines perform every spatial ETL request that is sent to the FME Server.

Repository Manager

The Repository Manager maintains and stores all information necessary to operate the FME Server. This information, stored for each repository, includes details of all registered workspaces, services, and other system resources needed by the other components. The Repository Manager enables users and client applications to have access to shared workspaces defined and maintained under them.

FME Server Console

The FME Server Console is a common-line interface that provides access to FME Server operations, for example sending transformation requests to the FME Server, and viewing available services and workspaces registered with the FME Server. It can be used effectively for managing workspaces within the FME Server repository, and for running workspaces on the FME Server.

FME Server Services

FME Server Services are components that add extra capabilities to workspace translation by allowing a workspace translation request and response to be generated in a specialized way. In FME Server architecture, the services provided by the FME Server carry out common tasks.

For example, a Web Map Service (WMS) Uniform Resource Locator (URL) Query is a specialized request that can be generated by the client and sent to an FME Server Service to be processed. In response, an image based on the request will be generated and sent via HTTP to the client by the service. In the FME Server, the context of the specialized request (WMS URL query) is converted to a workspace translation request; the result of the translation is transformed into a specialized response (streamed image) that is sent to the requesting client. With the FME Server, the client is generally a web browser which passes requests to the FME Server (the FME Server API) using a service; however, in certain cases the FME server client is a desktop application or a thick web client. Figure 4.3 illustrates the relationship of the Web Browser, FME Service and FME Server.

Several services provided by the FME Server can be used for general tasks. For specific tasks, the extendible service capability of FME allows the addition of new customized services to perform specific tasks.

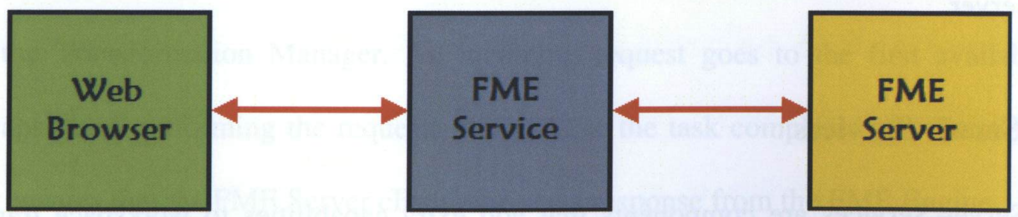

Figure 4.3 FME Server, Service and Web Browser relationships (source: [Safe Software, 2008a])

4.2.3 FME Server Services

Data Downloading

A data downloading service request can be either a URL, or a Form request with no restrictions. For example: *http://<HostName>/fmedatadownload/<Repository>/<workspace>.fmw?<pa-rameters>* is a valid FME Data Download request that can be generated on any client application. When this request is passed to the FME server, a web page with a link to a zip file containing the results of the workspace translation of the request will be generated. The FME Data Download Service works with any workspace that writes feature types without published parameters (options to run the workspace). However, in certain cases publishing some parameters from the workspace in order to control the translation from the URL or form request is a useful option. For example, commonly published-parameters when using this service are feature types to read, output coordinate system, and format.

Data Streaming

A Data Streaming service request can be a URL or form request with no restriction. For example *http://<HostName>/fmedatastreaming/<Repository>/<workspace>.fmw?<pa-rameters>* is a valid request which a client can send to the FME server. The FME Server will process the request and generate a dataset that is based on the parameters passed with the request. After the translation is complete, the resulting dataset (if it is one file only) is streamed with appropriate content-type (mime-type) over HTTP back to the client (e.g., GeoLink, web browsers, Google Earth) file. The FME Data Streaming service can generate streams in several formats including Hypertext Markup Language (HTML), Portable Network Graphics (PNG), KML, VRML among others.

KML Network Link

KML Network Request is also a URL or form request without restrictions. The following represents a valid KML Network Link Request: *http://<Host>/fmekmllink/<Repository>/<workspace>.fmw?<parame-ters>*. Based on the request and the parameters a response is generated and returned to the client; the response is in a compressed version of the KML (KMZ) file and contains a KML Network Link with a URL pointing to the Data Streaming service, as well references to the requested work-space. The FME Server performs no transition for the KMZ files, but rather streams the generated context on-the-fly.

Open Geospatial Consortium (OGC) WebFeature Service (WFS)

The FME Server WFS service accepts standard WFS URL request defined on OGC standards. As a Base URL Example: http://<Host>/fmeogc/<Repository>/<workspaceName>.fmw is a valid FME Server WFS service request, while http://<Host>/fmeogc/<Repository>/<workspaceName>.fmw?SER-VICE=WFS&REQUEST =GetCapabilities&VERSION=1.1.0 is a valid GetCapabilities URL example for this type of request. The results of a valid WFS request is a WFS formatted GML file that contains the requested spatial dataset returned as a response stream.

OGC WMS (Web Map Service)

The FME Server WFS service accepts standard Web Map Service (WMS) URL requests defined on the OGC standard. An example of a valid Base URL can be similar to: *http://<HostName>/fmeogc/<Repository>/<workspaceName>.fmw* while a valid GetCapabilities URL Example can be http://<Host>/fmeogc/<Repository>/<workspaceName>.fmw?SER-VICE=WMS&REQUEST =GetCapabilities&VERSION=1.1.1

Chapter 5. Framework Design and Prototype Implementation

5.1 Framework Design

An integrated 3D system framework was designed to allow managing a 3D geological data modeling workflow, handling a large volume 3D geological data, and streaming the underlying data to the GeoLink collaborative GIS prototype over the Internet. Because of data format and functional constrains of the GeoLink prototype not all identified geological modeling system requirements were considered in the framework design. The details of the framework design are explained in the following sections.

5.1.1 Existing 3D Data Systems and Design Approaches

All 3D GIS functionalities are performed within a GIS System in order to capture, model, store, retrieve, share, manipulate, analyze and present 3D models. Spatial functionalities are performed on spatial objects and can be implemented either in a DBMS or in front-end application. Implementation of these spatial operators and functions on a front-end application is common for 2D GIS systems, some of which arehighly-specialized GIS systems which can perform complicated spatial operations [Zlatanova and Stoter, 2006].

Traditionally, many 3D systems utilize file-based options to store and manage 3D models. In this structure all data is retrieved and constructed to reflect the logical model components at run time. This file-based storage option does not allow access to a specific section of the stored model without retrieving the entire model. Due to the immense size of geological raw data and constructed models, a better data storage option is needed for an operable 3D geological modeling system.

Some major DBMS such as Oracle, SQL, DB2, and PostgreSQL offer spatial data types to support spatial objects in 2D space. With the spatial data type offered by major DBMS vendors certain spatial operations, traditionally performed in a costly manner by the front-end-application, can be implemented more cost effectively. The possibility of performing computationally demanding GIS operations at the database level opens up the possibility of developing Web-enabled 3D systems for 3D GIS domains, including geological and subsurface modeling.

Because of the large volume of 3D data, effective data management capacity is necessary for well-designed 3D solutions. DBMS systems are ideal for storing the resulting 3D model as well as the raw data used to create it. Geomodeling tools allow end users to construct, visualize, interact and analyze the constructed models as well as the input data. In the domain of geology, a successful integration of geomodeling tools and DBMS will produce a 3D system capable of capturing, modeling, storing, retrieving, sharing, manipulating, analyzing and presenting 3D geological models.

Because the integration is the key for a successful realization of 3D systems, it is important to correctly identify which group of the functionalities can ideally be implemented in frond-end-application and which in DBMS. A balanced approach for implementing the 3D GIS functions is critical

Front-end Application Centric 3D Systems

This approach includes most GIS functions and operators at the front-end application level and is used by most traditional CAD and geomodeling tools; file-based data storage is commonly used

for storing 3D models in this type of implementation. With the significant increase in data for 3D modelling, some CAD and geomodeling tools have started to take advantage of DBMS large volume data management capacity in which either a topological or a geometric model can be mapped into a relational database to create the 3D model. The data is retrieved from the database tables and the model is constructed at the front-end application runtime. If the selected model is topological, certain topological relationship operations can be performed as a standard SQL query at the database level. The level of SQL interaction usually depends on the topological model and the accuracy of relationships mapped within the DBMS. Although DBMS is used to store the model in this type of system, the underlying DBMS is unaware of spatial characteristics of the modeled data; spatial functions and operations need to be implemented in front-end applications.

DBMS Centric 3D Systems

Some major DBMS vendors have implemented spatial data types and tools that allow certain GIS tasks, using simple SQL query operations. Vendors include Oracle Spatial, IBM DB2 Spatial Extender, Informix, Spatial Datablade, PostGIS (PostgreSQL) and MySQL.

Implementing common GIS functions at DBMS, ensures consistency of the models and performance optimization of the system. Avoiding unnecessary transport and conversion of data between DBMS and GIS application provides for better data maintenance. For instance, operations that organize data according to topological rules set by the model may be better implemented in DBMS. Select, navigate through spatial objects, and create an object based on an existing one are some of the operations that can be implemented in a database [Zlatanova and Stoter, 2006]. Furthermore, some selection operators, namely metric, proximity, and relationship operators might also be possible to implement in DBMS; however, some domain-specific functions must still be implemented in the front-end-application.

5.1.2 Problem Areas of 3D System Development

In general, a 3D GIS system is expected to provide all basic GIS functions. Specifically, a 3D system that can be used for complex geological modeling may need to provide additional functions to support specific requirements of the geological domain.

Visualization refers to the process of extracting data from a model and representing the extracted object on screen. In the 3D GIS, context visualization denotes the 3D graphic on screen, detection of user interaction, and re-computation of the model's parameters to produce a new screen graph [Zlatanova, 2000]. Visualization of 3D data is in high demand by any GIS system because it is the only way an end user can see, and interact with, the 3D model, regardless of the underlying complexity of the system architecture. Traditional CAD systems employ capabilities for visualization of complex 3D models that are higher then traditional GIS systems, but neither provides all the desired 3D system functions of the geological domain.

Rendering Engines, Constraints and Interoperability

Rendering engines require geometric objects to be organized in a manner that allows the interpreter to understand the structure. At the system architecture level, complex data models and specialized viewers can be implemented when there are no interoperability concerns. Another rendering engine constraint relates to the type of surfaces supported. For example, most man made objects can be modeled more precisely with free-form shapes such as Bezier curve, B-

spline curve, or Non-uniform rational B-spline (NURBS) curve/surface; however, because these surfaces are not supported by rendering engines, they cannot be considered in 3D systems.

Other Visualization Related Issues

3D visualization differs from 2D map visualizations in term of user interaction and delivery of information. In traditional 2D GIS applications, interaction means the request and response for each scene. In 3D GIS interaction, the user can move and investigate the model by performing operations such as zooming, panning, walk-through, and fly-over; as well, the user can examine the objects without changing the original parameters of the scene. Accessing elements of a scene and changing the scene entirely, or in part, by performing operations such as edit, add, delete, scale, rotate and translate [Zlatanova, 2000] are among the user interactions desirable in both 2D and 3D GIS systems. 2D GIS systems have already achieved all desirable user interaction operations, but 3D GIS systems still face the challenge of accessing 3D elements from the scene.

5.1.3 Options for Storing and Managing Geological Data and Models

To create a 3D geological model using the boundary representation technique, surface space must be delimited into distinct homogeneous regions based on either qualitative or quantitative parameters obtained with direct observations, or on information derived by laboratory examination. Kothuri et al. (2007) specify commonly used geological objects in a 3D model as geological units (lithological or chronostratigraphic), structural elements (faults and fractures), hydrocarbon reservoirs and ore deposits. Geometric objects such as point sets, lines, curves, surfaces, and solids can be used to represent geological objects and structures.

Either these constructed geological objects can be stored as an integrated model or as disperse layers. In an integrated model, all geological objects must be maintained within a single model where maintenance may be difficult. In the disperse layers option, geological objects are maintained in separate layers, similar to data layers in 2D GIS, where each distinct volume group constitutes a distinct parameter, and the model permits overlapping volumes among separate layers.

In both integrated or disperse layers options, the volume of data to be managed makes DBMS the best option. Geological 3D model requirements suggest that original sampling data obtained by direct observations and surveys, laboratory examination results, and the resulting 3D model must be stored in a system that can support and manage all data types. Although some GIS applications still use file based solutions for managing 2D/3D modeling data for small scale operations, DBMS is more appropriate for a geological 3D system. The next section discusses three options for managing 3D geological models in a DBMS based solution.

Option 1: Relational Database Management Systems

A 3D topological data structure can be directly mapped into a relational database to create a 3D geological model. This implementation translates the conceptual model's topological primitives into separate relational database tables [Zlatanova and Stoter, 2006]. For example, a conceptual model consisting of three topological primitives (node, face, body) will be converted into three tables where each table presents one topological primitive. The node table will contain an identifier and three columns for node coordinates. The face table will contain a unique identifier for the face and node IDs. These tables can optionally denote face orientation. The third table

will contain reference to the faces that form volume. In principle, each primitive consists of lower level primitives and primitive tables that store links to lower level primitives.

This option strictly uses a selected topological model and all metric operations needed to be implemented in the front-end-applications. The Relational Database Management System (RDBMS) implementation of a 3D model can integrate various sources of 2D and 3D spatial data, as well as descriptive data. However, data interpretation must be performed outside the DBMS, and the resulting 3D geological model must be converted and mapped into the primitive tables of the topological model. This implementation allows representation of topological relationships and material properties of geological objects, but the geometry required for visualisation must be derived from topological relationships.

Because the underlying database is unaware of the distinct characteristics of 3D model components, it does not provide capacity to update or analyze a 3D geological model.

Option 2: Object-Relational Model in a DBMS

Alternatively, a geological model can be implemented as an object-relational model in a DBMS. This can be accomplished by extending existing data types of an underlying DBMS. User-defined object type data will be based on a variable array or nested table principle, where references to lower dimension primitives are stored as arrays. Examples of this alternative implementation are discussed in the literature [Zlatanova, 2000; Zlatanova et al., 2004; Store and Zlatanova, 2003; Gröger et al., 2004; Zlatanova, 2005; Pu and Zlatanova, 2006].

The object-relational model also requires that geological observations be processed with separate modeling applications for 3D geological model construction. The resulting model can then be organized according to the user-defined data type in the RDBMS. Integration of various sources of 2D and 3D spatial data, as well as the descriptive data requirement of geological modelling can be satisfied with object-relational models.

Option 3: Object-Oriented DBMS

The third alternative reported in the literature is object-oriented DBMS (OODBMS). Several examples of implementations for modeling complex 3D object exist. This alternative is suitable for certain applications that require very specific modeling and computational capabilities [Shi et al., 2003].

Although OODBMS implementations are specialized, a well-designed model can meet all user requirements; however, in the domain of geology, there has been no major accomplishment using OODBMS for modeling 3D space.

5.1.4 Oracle 3D for Subsurface and Geological Modeling

Geology usually defines observed data as a finite set of points in 3D space. An approximation of a geological object is generated by interpreting and extending this point dataset. This process requires extensive human interaction and geological knowledge. One geological data-modeling requirement is that observation data must be available for further consideration, or re-interpolation of the model.

Oracle's PointClouds (PC) data type, suitable for storing a large amount of geological 3D data points, allows the Triangulated Irregular Network (TIN) surface generation function (provided

by Oracle) to operate on selected sets of PC subset data. Generated surfaces can also be stored in Oracle as TIN data type as well as large TIN datasets. The existing procedure to create TIN objects uses Delany Triangulation from selected point sets [Oracle and Murray, 2007]. A geologist can then process these TIN surfaces with the help of geomodeling software in order to construct a 3D geological object as either 3D surfaces or volumes. A 3D geological object can be stored as Oracle 3D data type in the same DBMS.

Oracle and Murray (2007) outline the typical steps to create a 3D model. The following modifies these steps to suit the domain of geology:

1. Direct observation, or information derived by laboratory examination, is organized in a structure, and a point cloud representation of the distinct geological objects is created. This point cloud representation is a set of three-dimensional point values.

2. Typically, Delaunay triangulation is used to create TIN surfaces for data points sharing common attributes representing the vertical and parallel surfaces, which are the faces of 3D geological objects. Oracle provides a utility that can create TIN surfaces from a selected set of Point Cloud data by using Delaunay Triangulation. (Delaunay triangulation is the method provided by Oracle as a utility package.)

3. Triangulated surfaces are further refined in order to create a smooth 3D geological object.

Figure 5.1.a shows the typical workflow of 3D modeling in Oracle. Figure 5.1.b shows the typical workflow of 3D geological modeling.

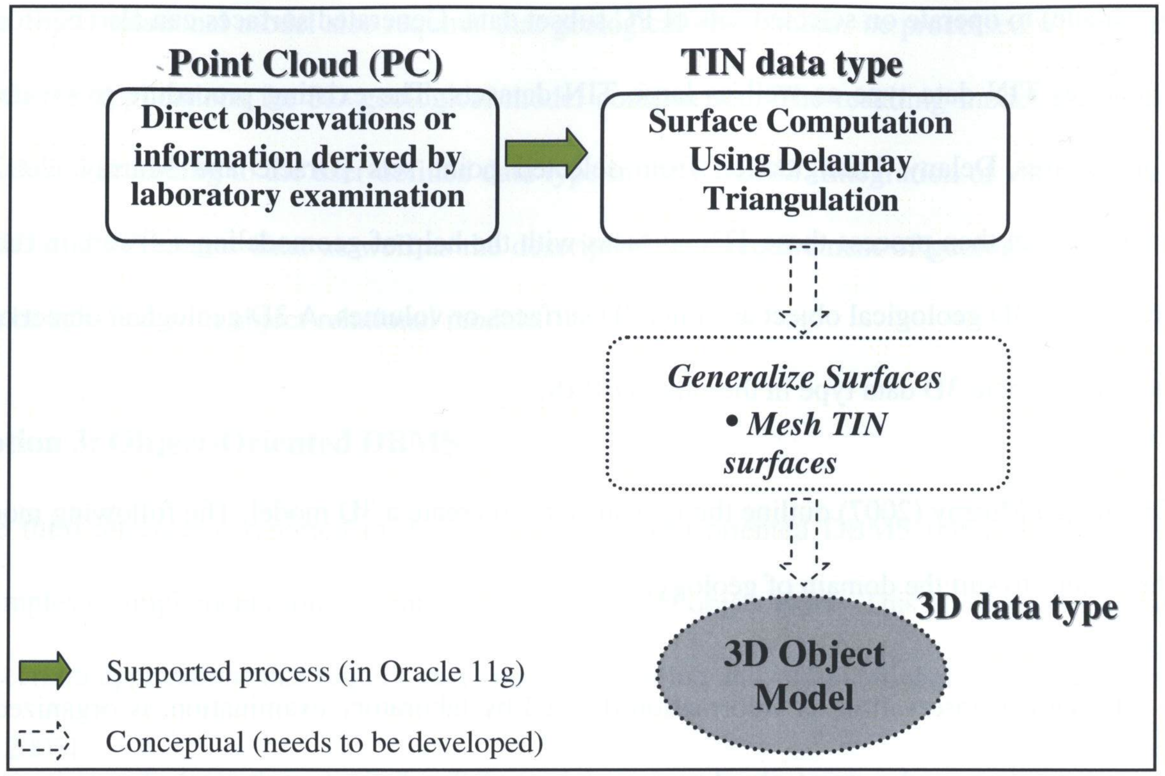

Figure 5.1.a Typical 3D modeling workflow in Oracle (modified after [Kothuri et al., 2007])

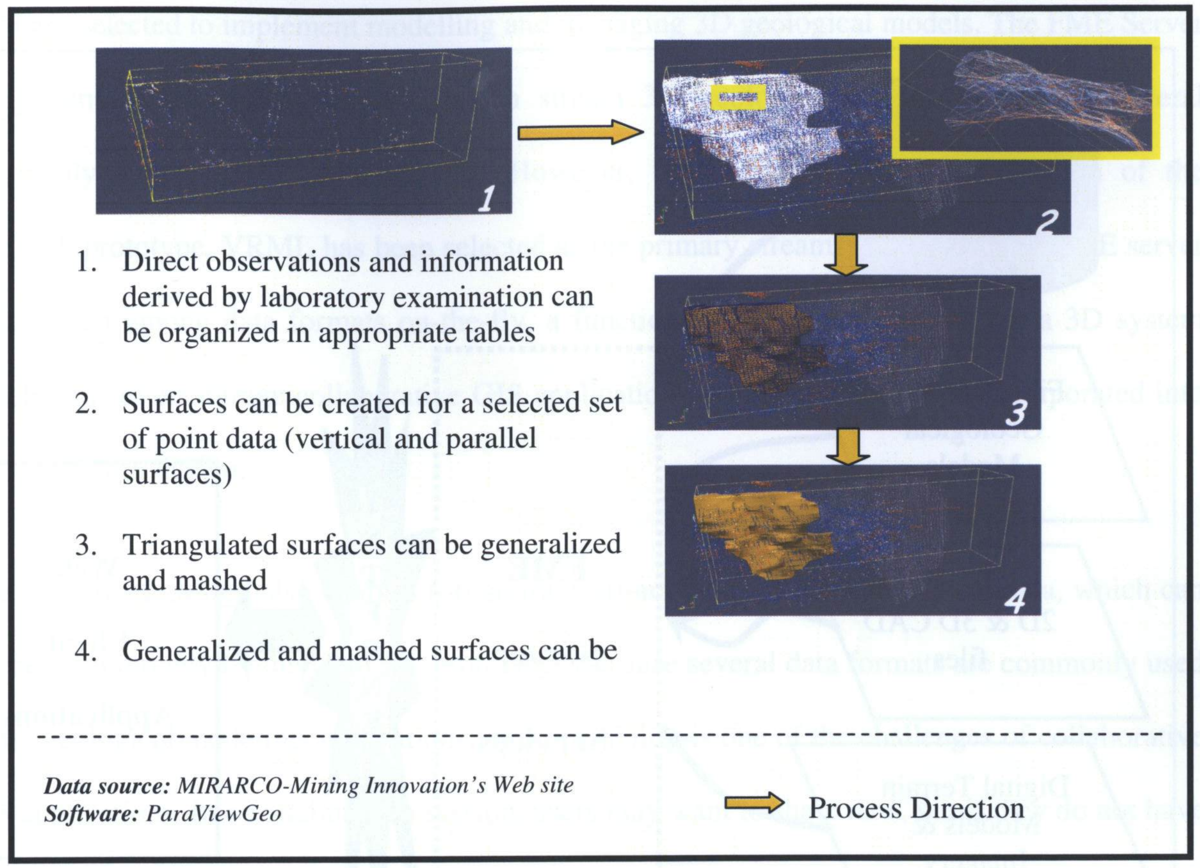

Figure 5.1.b Typical 3D modeling workflow

5.1.5 FME Server for Streaming 3D Geological Models

The capacity to serve data in real time to client applications makes the FME Server a valuable piece of technology, which can play an important role for enabling stakeholder sharing of 3D geological models. This is true even though no agreed-upon common data model exists for ultimate interoperability of 3D data [FME, 2008]. FME Server technology can enable streaming 3D geological models (stored on an Oracle Database) to Web client applications. As well, on-the-fly format translation capacity of the FME Server is a desirable enabler for 3D geological model sharing in a collaborative GIS environment. Figure 5.2 shows how various geological data sources feed into the FME Server to provide geological data to Web client applications.

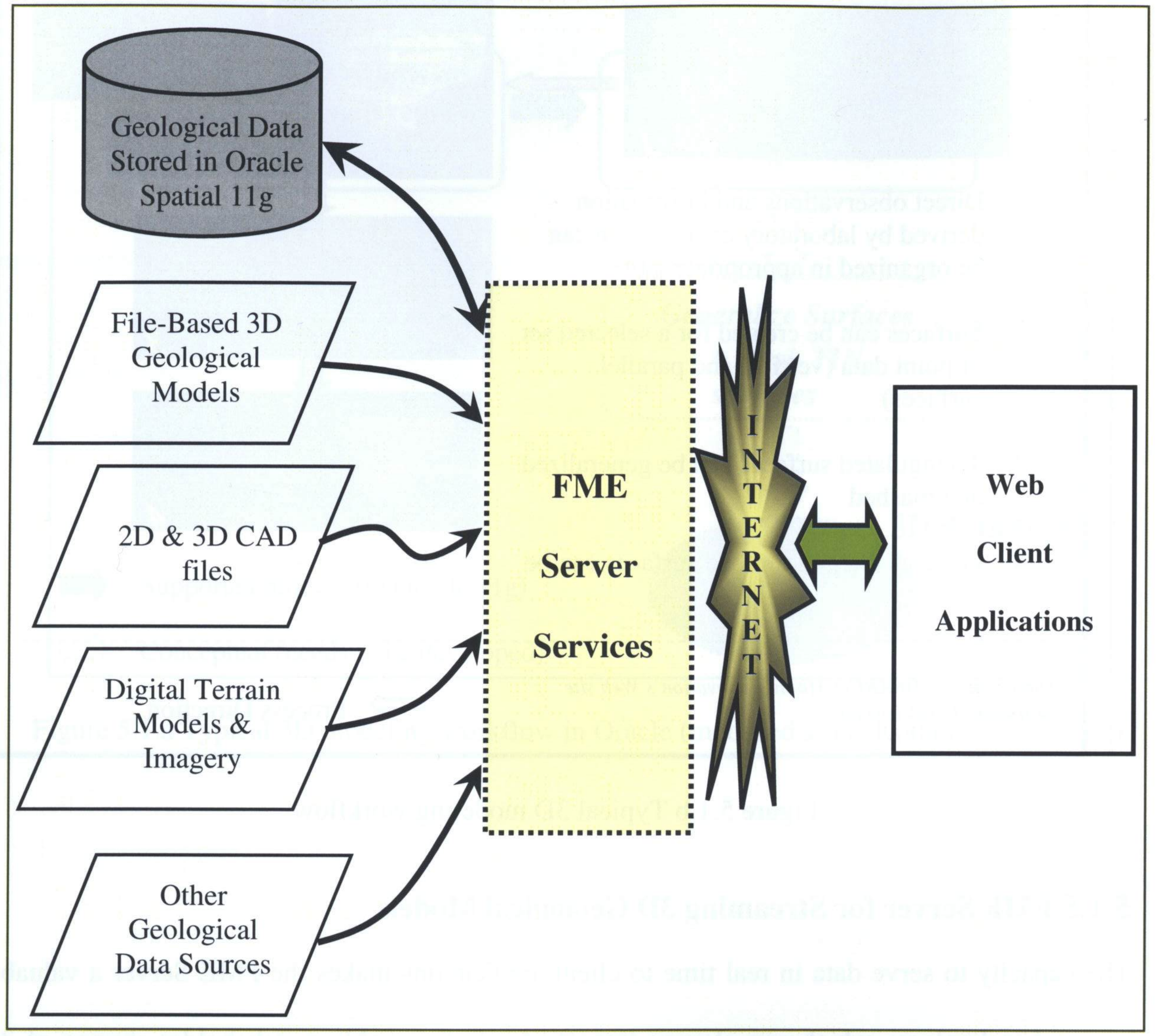

Figure 5.2 Relationship of geological data, FME Server Services and Web client applications

5.1.6 Framework for Geological Data Modeling and Management

A framework, based on geological 3D data modeling requirements and 3D data access and management requirements of the GeoLink prototype, was designed. It demonstrates the possibility of streaming large-scale 3D geological data, which is organized and managed in a DBMS. The Oracle 11g database is the only DBMS providing 3D data types, and it is the

database selected to implement modelling and managing 3D geological models. The FME Server component of this integrated design can stream 3D geological models as one of several commonly supported 3D data formats. However, because of the current limitation of the GeoLink prototype, VRML has been selected as the primary streaming format. The FME server can convert among data formats on-the-fly, a function that is very important for a 3D system which aims to encourage collaborative GIS applications; it has therefore been incorporated into the framework design.

The framework design also enables format translation of external 3D geological data, which can not be used directly by the GeoLink prototype. Because several data formats are commonly used in 3D geological modeling applications, interoperability is one of the challenges of collaborative GIS applications. In a collaboration session, users may want to share data, but if they do not have a tool to convert their data into VRML format, the data cannot be used in the GeoLink collaboration session. The FME Server allows running created services with external data sources. The FME server uploads data, runs the conversion process, and streams back the resulting data which can then be used by the GeoLink Client. The FME Server supports this process, but a mechanism to redirect external data to the FME Service and use the resulting stream may be needed to enable the GeoLink prototype. Figure 5.3 illustrates the framework of geological data modeling and management.

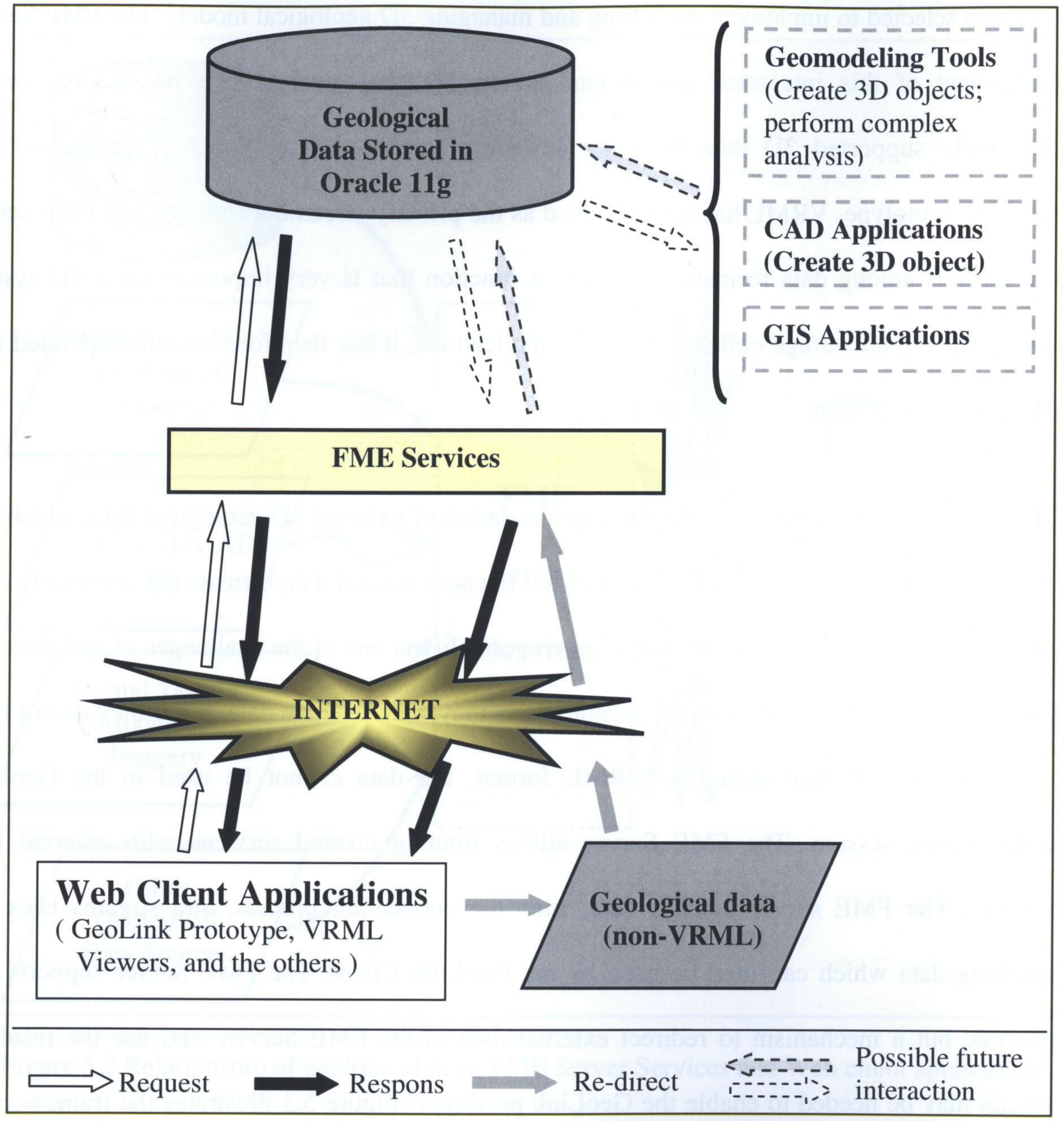

Figure 5.3 Framework of geological data modeling and management

5.2 Prototype Implementation

The prototype system includes an Oracle Server installed and configured on the SIMAL server in the Spatial Information Management and Applications Laboratory at Ryerson University. An FME Server was installed on the same server to enable streaming 3D geological models (stored

on the Oracle database) to the GeoLink collaboration client. FME Workbench was used to manipulate the test data, populate the Oracle database, as well as to create and author FME Server workspaces used for streaming the 3D data. Figure 5.4 shows the implemented prototype system configuration and client interaction.

A test dataset was used to populate the Oracle 3D database layers that were served by the FME Server to 3D web clients, such as free VRML Viewers (Commercial), Xj3D viewer (Open source) and the GeoLink prototype. After implementation of the prototype components, several client applications were used to test performance of the implementation. The following sections discuss the details of the prototype implementation and testing.

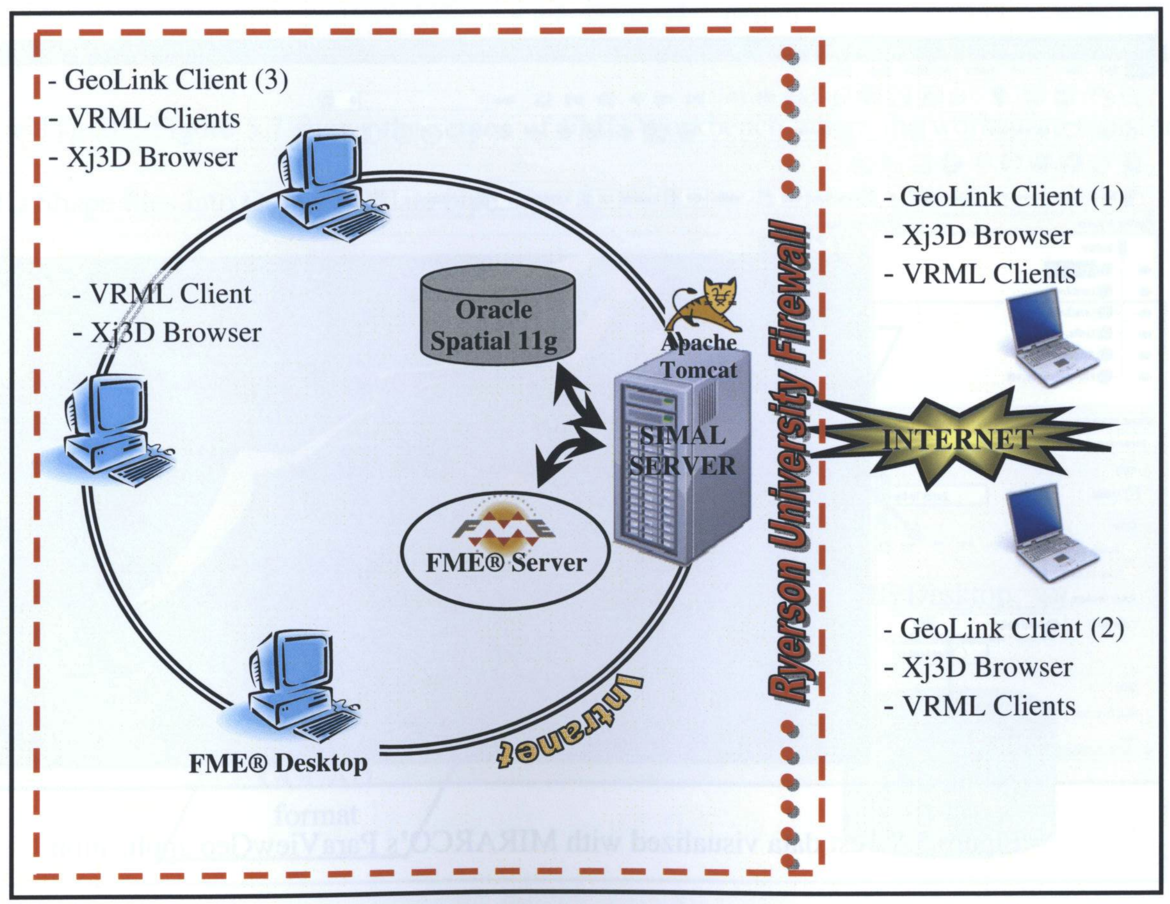

Figure 5.4 Prototype System Configuration

5.2.1 Test Dataset, Data Conversion and Uploading Process

A dataset containing several geological structures, including four lenses, a body of ore or rock, or a deposit thick in the middle and thin at the edges, was obtained from the MIRARCO-Mining Innovation's Web site. The original dataset was in Visualization Toolkit (VTK) format and can be viewed by using open source ParaViewGeo application, which is available at the same website. The company, Kitware, has made Paraview's complete source code downloadable, and MIRARCO has added to the Kitware's base code as a part of a Northern Ontario Heritage Fund project to promote the use of scientific visualization in the mining and exploration industry and to create the ParaViewGeo application. Figure 5.5 shows test data visualized with the MIRARCO's ParaViewGeo application [ParaViewGeo Wiki, 2008].

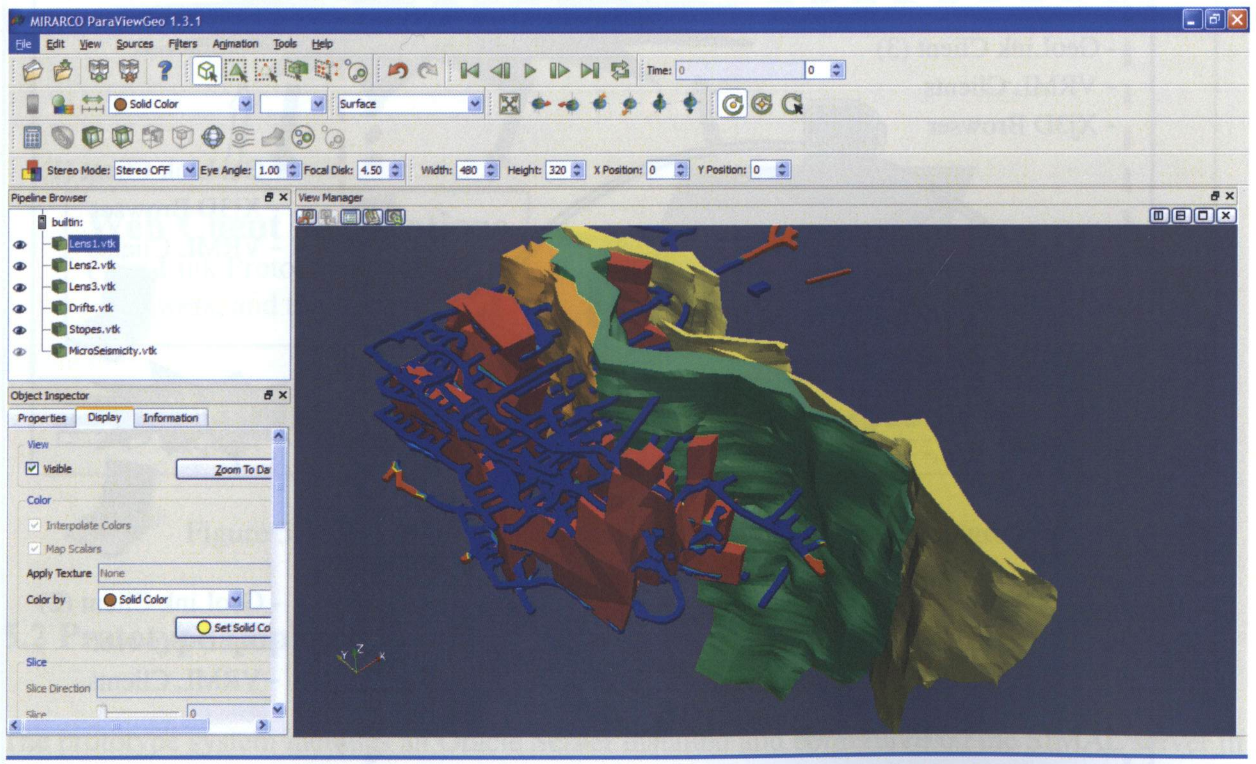

Figure 5.5 Test data visualized with MIRARCO's ParaViewGeo application

VTK, the original geological data layer format, is not supported by FME Workbench and no utility exists that can convert and load VTK format datasets into Oracle 11g. In order to load VTK data into an Oracle database, the VTK files must first be converted into Geological Object Computer Aided Design (GOCAD) ASCII file format (.ts) using ParaViewGeo's GOCAD plugin. Then the GOCAD ASCII file can be converted to Environmental Systems Research Institute (ESRI) 3D Shape files using GOCAD Software. *(GOCAD is the project started in 1989 by Professor Jean-Laurent Mallet at Nancy Université.)*

FME Desktop supports ESRI 3D Shape files, as well as many other commonly used 3D formats. A copy of FME Desktop, provided by Safe Software, was used to translate the 3D Shape files generated by the GOCAD into Oracle Spatial 3D data type in order to populate the Oracle tables. Figure 5.6 shows the flowchart of the data conversion process from the original VTK format to Oracle 11g 3D. Figure 5.7 shows the screen of FME Workbench where the workspace transfers the 3D Shape files into Oracle 3D data type.

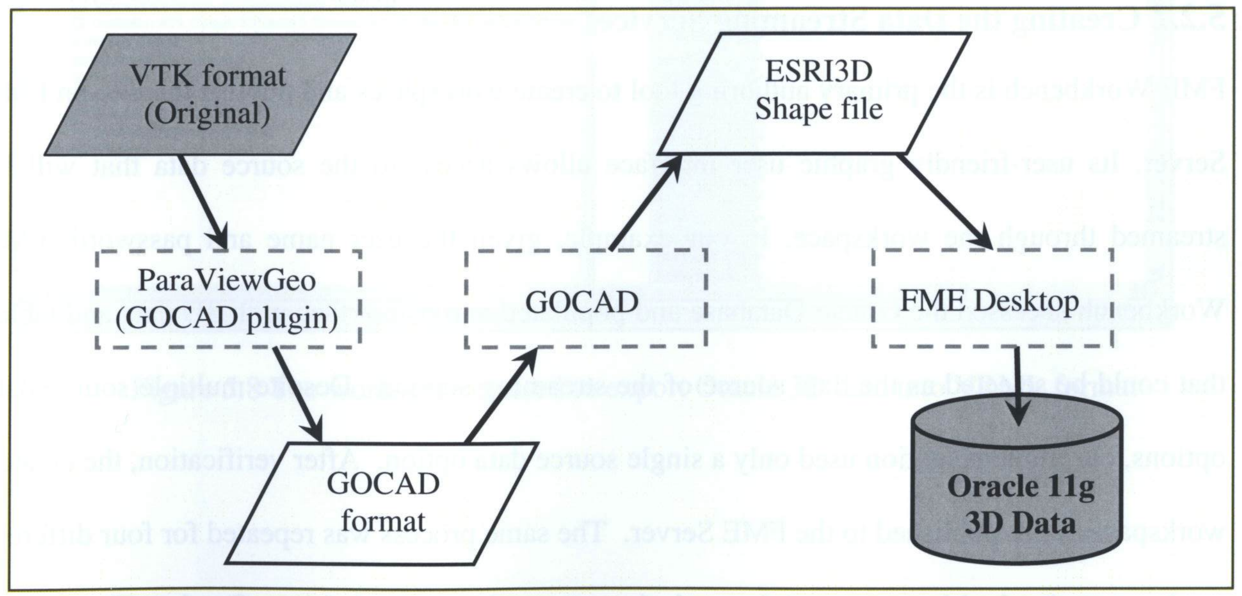

Figure 5.6 Flowchart showing the data conversion process

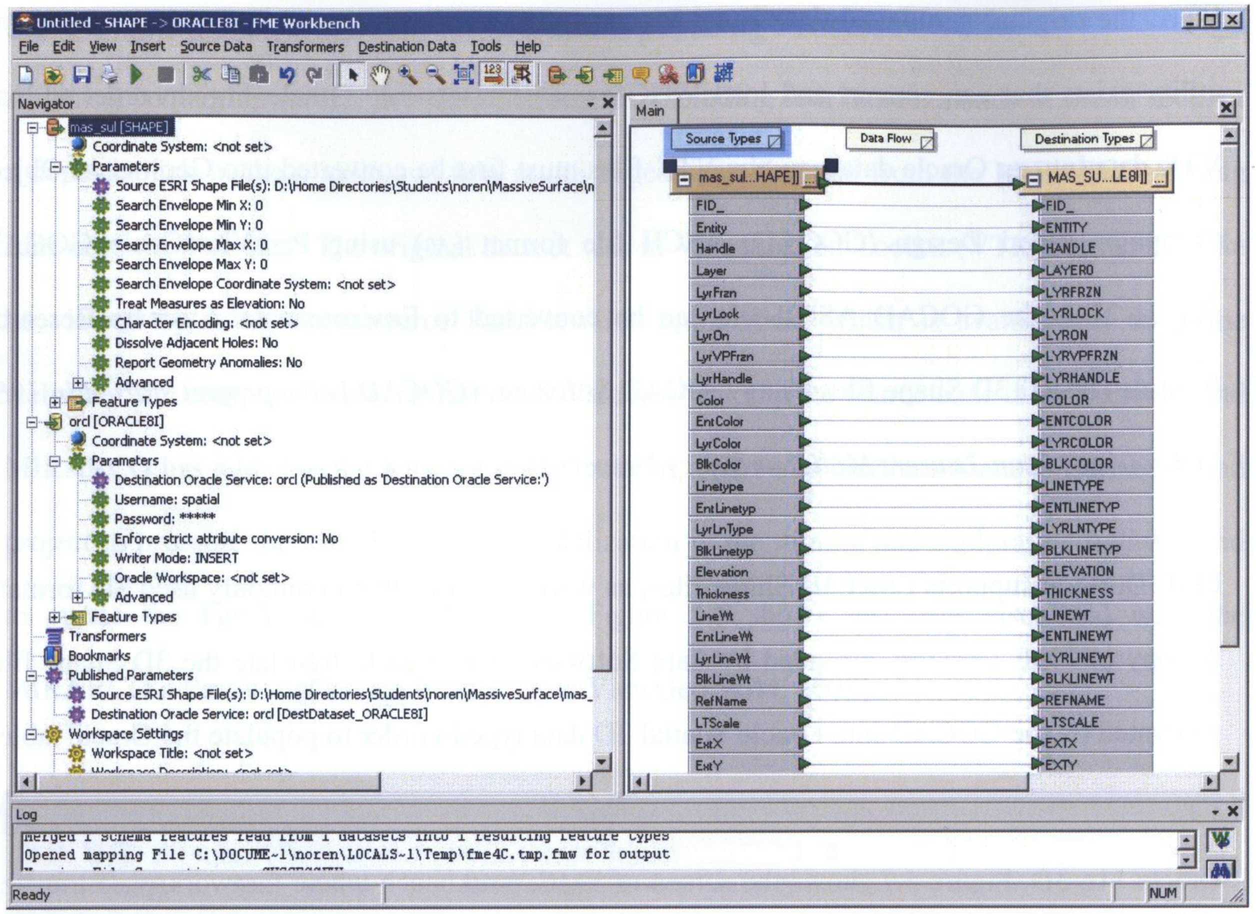

Figure 5.7 FME workspace transfers of ESRI Shape files to Oracle 3D

5.2.2 Creating the Data Streaming Services

FME Workbench is the primary authoring tool to create workspaces and publish them to an FME Server. Its user-friendly graphic user interface allows access to the source data that will be streamed through the workspace. In our example, given the user name and password, FME Workbench accessed the Oracle Database and populated a user interface with 3D data and tables that could be selected as the data source of the streaming service. Despite multiple source data options, our implementation used only a single source data option. After verification, the created workspaces were published to the FME Server. The same process was repeated for four different workspaces that had been created for each dataset previously loaded into Oracle. Then, those

workspaces were published to the FME server using the publishing wizard available in FME Workbench.

Overall, the FME Workbench provides all tools necessary for creating and publishing workspaces to the FME Server. Figure 5.8 shows the Workbench with the workspace that exports the Oracle 3D data into VRML. Figure 5.9 shows the publishing screen of the FME Server.

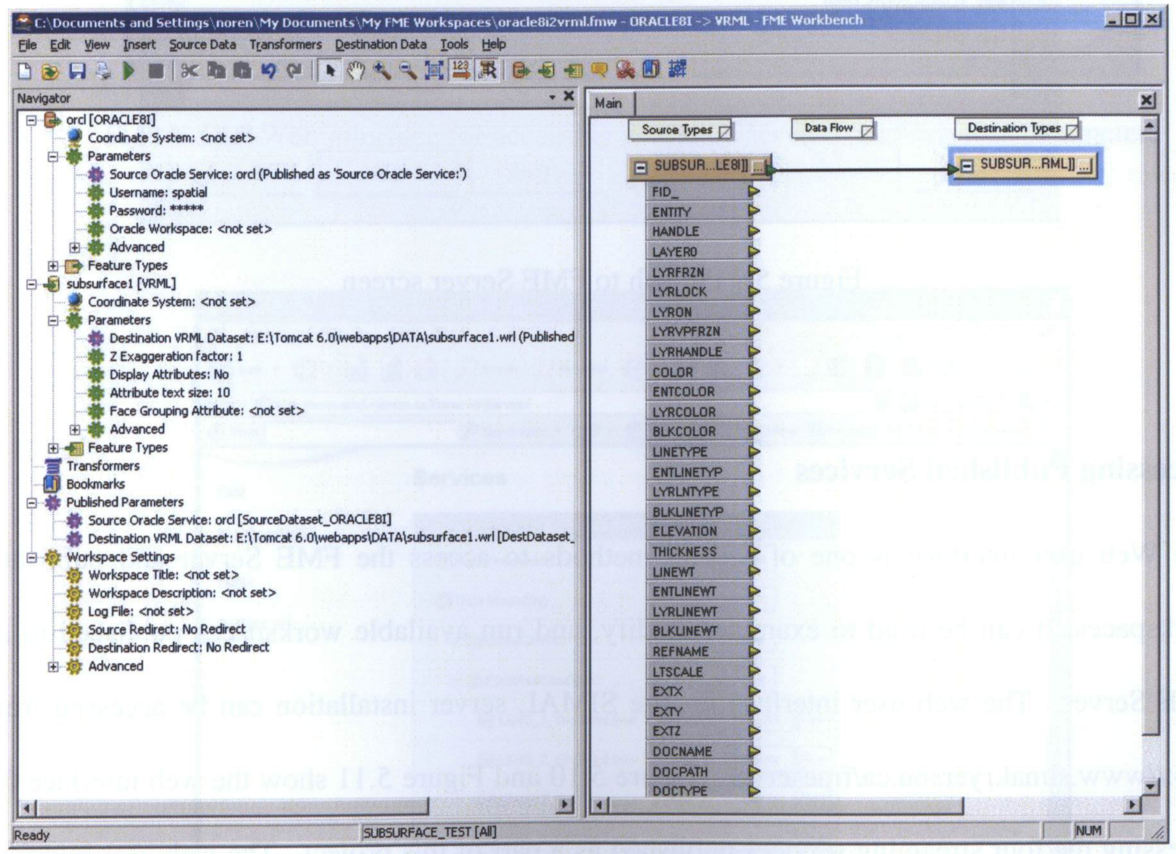

Figure 5.8 The workspace created to export Oracle 3D data into VRML format

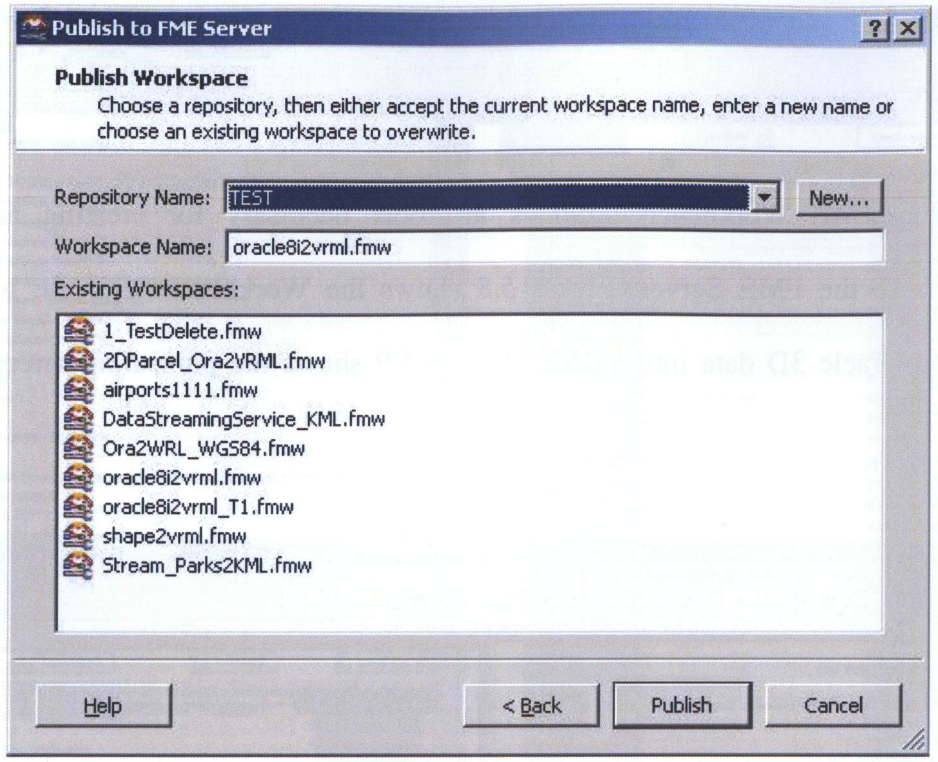

Figure 5.9 Publish to FME Server screen

Accessing Published Services

The Web user interface is one of several methods to access the FME Server and published workspaces. It can be used to examine, modify, and run available workspaces published to an FME Server. The web user interface for the SIMAL server installation can be accessed from http://www.simal.ryerson.ca/fmeserver. Figure 5.10 and Figure 5.11 show the web interface for accessing the four streaming services published as a part of this project. The web user interface also allows examination of published workspace details, such as service request information, as shown in the Figure 5.12.

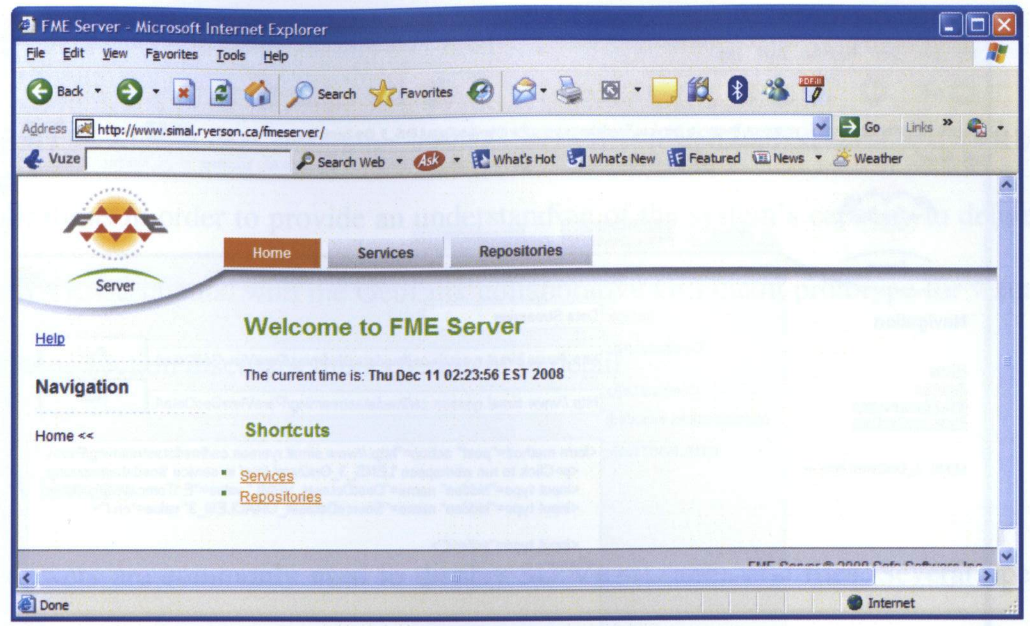

Figure 5.10 Web interface for accessing available services and services parameters

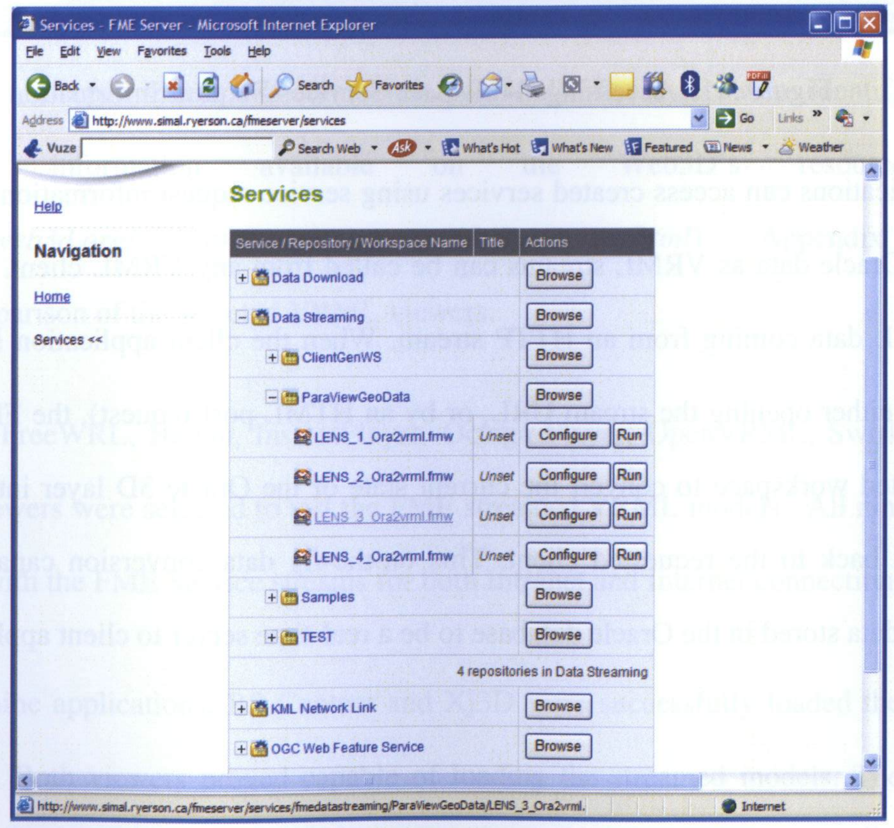

Figure 5.11 Web interface for accessing published Data Streaming services

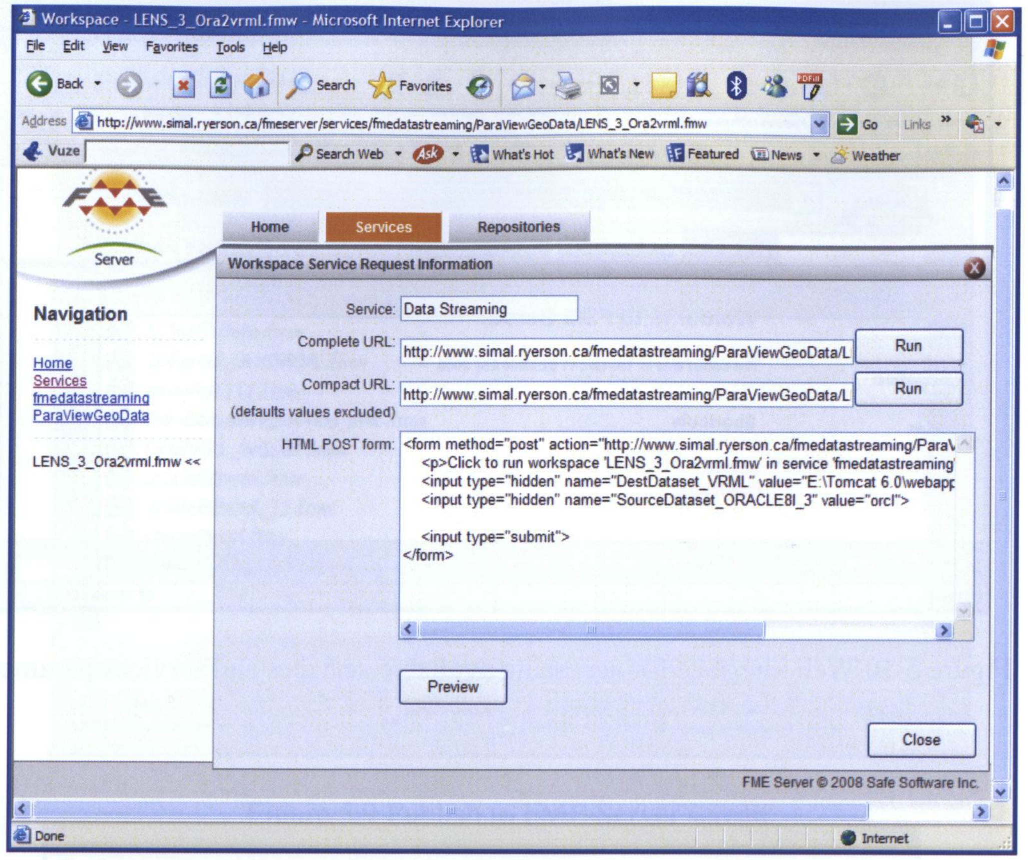

Figure 5.12 Accessing Workspace Services Request Information

Client applications can access created services using service request information. FME Services that serve Oracle data as VRML streams can be called from any VRML client. This client can open VRML data coming from an HTTP stream. When the client application sends an HTTP request (in either opening the stream URL, or by an HTML post request), the FME Server runs the associated workspace to convert the current state of the Oracle 3D layer into a VRML file that is sent back to the requested client. This on-the-fly data conversion capacity allows 3D geological data stored in the Oracle database to be a real-time server to client applications.

5.2.3 Testing Streamed 3D Data

The four 3D geological data streaming services were first tested with several X3D/VRML client web applications in order to provide an understanding of the system's capacity to deliver Oracle 3D data to VRML clients, with the GeoLink collaborative GIS client prototype for visualization. The following section discusses this experiment in detail.

VRML Viewers

VRML viewers are commonly used to display 3D VRML and X3D files. Several open source and commercial VRML tools exist which can be used to view VRML and X3D models. Some of these are stand-alone applications, while a number are open source and commercial plug-in products.

A combination of open source and commercial VRML/X3D viewers were identified for testing based on information available on the Web3D's resources website (*http://www.web3d.org/x3d/content /examples/X3dResources.html*). Appendix A shows a detailed comparison of the selected VRML viewers.

BS Contact, FreeWRL, Heilan, InstantPlayer, Octaga Player, OpenVRML, SwirlX3D, Vivaty, and Xj3D viewers were selected to test the FME streamed VRML models. All nine applications were tested with the FME Service streams for both Intranet and Internet connections.

Among the nine applications, BS Contact and Xj3D were successfully loaded the FME Server data streams. Both viewers proved capable of loading the streamed models. The summary of findings for the tested VRML viewers is shown in Table 5.1.

Table 5.1 Summary of VRML viewer test results

Sofware/Plugin	Can Open FME Stream	Comment
BS Contact	Yes	Can open all the services in 10-30 Seconds
FreeWRL	No	No Windows support
Heilan	No	No VRML support
InstantPlayer	No	Can't load any of the FME Service data
Octaga Player	No	Can't load any of the FME Service data
OpenVRML	No	No Windows support??
SwirlX3D	No	Can't load any of the FME Service data
Vivaty	No	Do not support URL Locations
Xj3D	Yes	Can open all the services in 10-30 Seconds

Figures 5.13 to 5.16 show the BS Contact viewer with LENS 3 and LENS 4 FME streamed geological data layers. Figure 5.17 and Figure 5.18 show LENS 3 and LENS 3 data layers in Xj3D viewer.

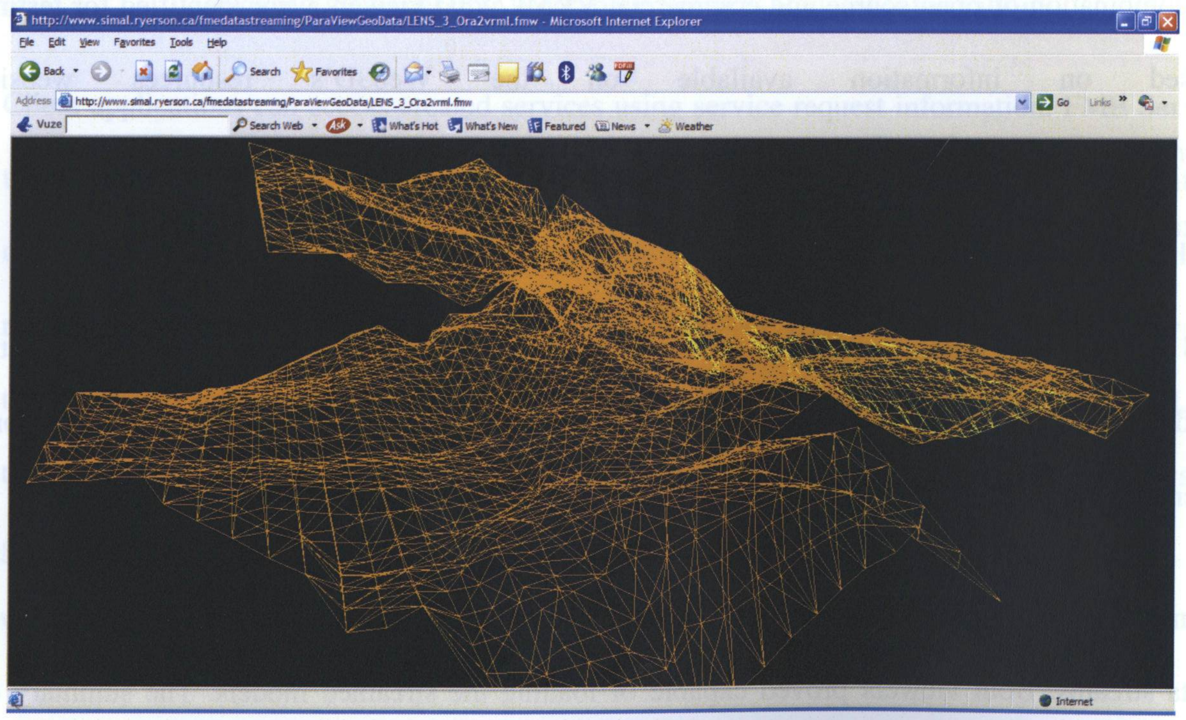

Figure 5.13 BS Contra VRML viewer (FME streamed LENS 3 data layer)

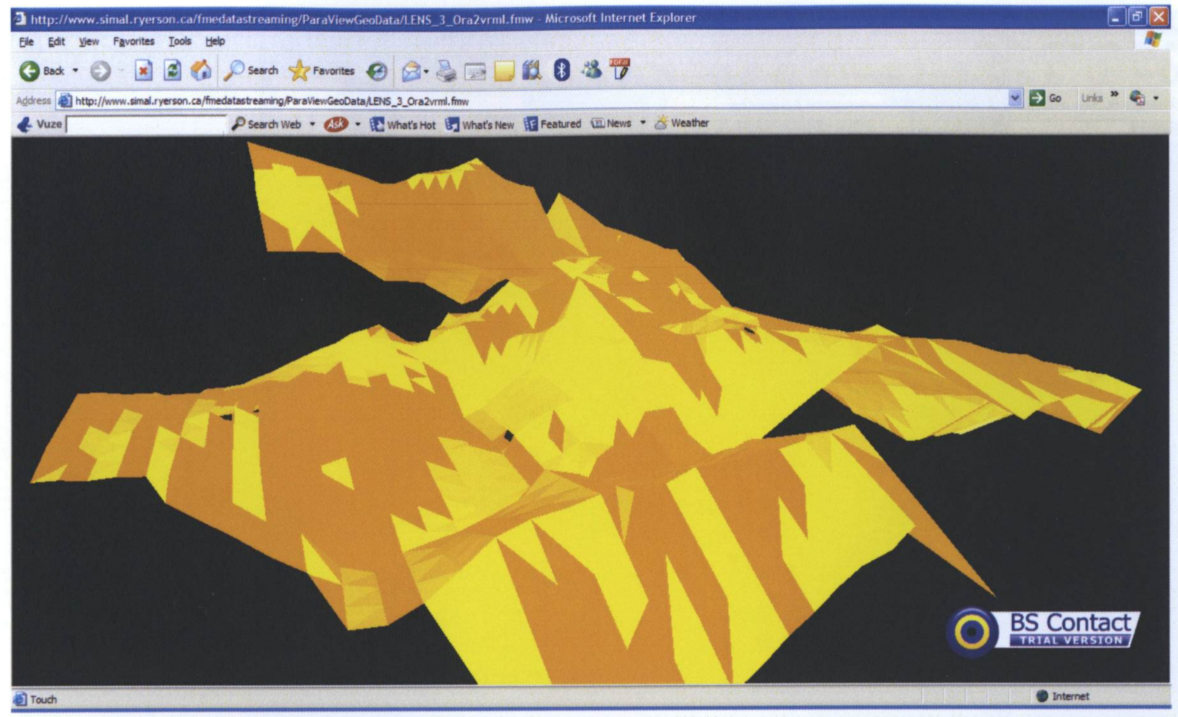

Figure 5.14 BS Contra VRML viewer (FME streamed LENS 3 data layer)

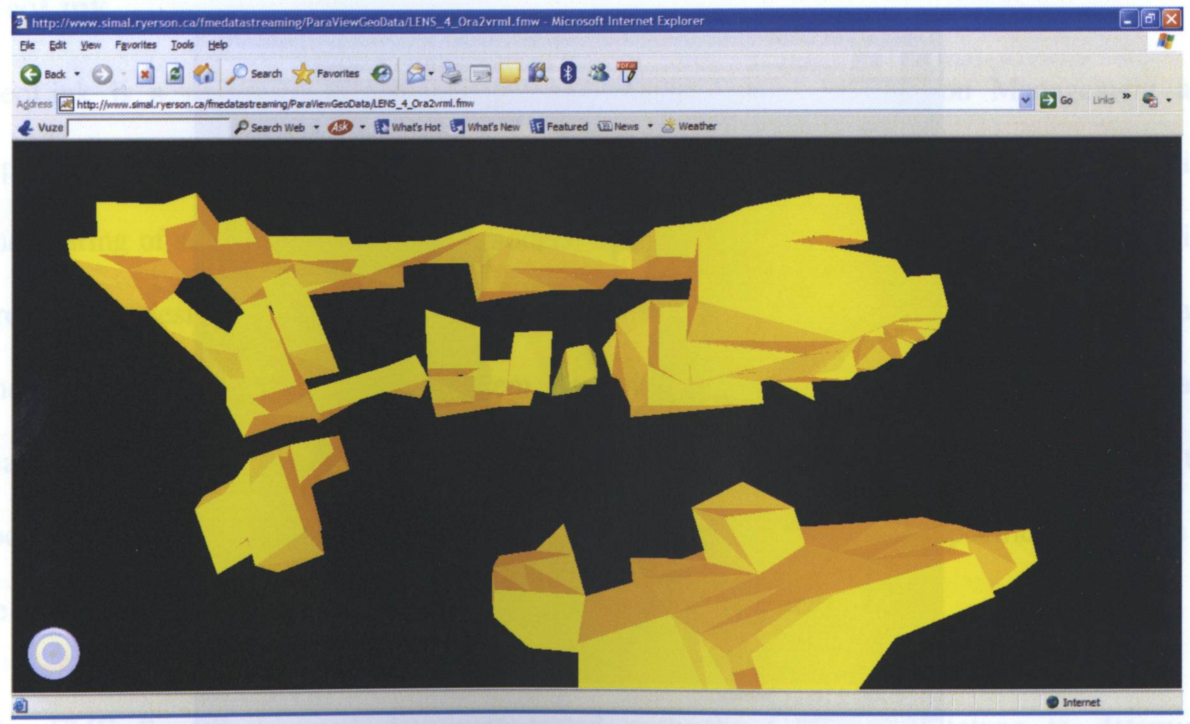

Figure 5.15 BS Contra VRML viewer (FME streamed LENS 4 data layer)

Figure 5.16 BS Contra VRML viewer (FME streamed LENS 4 data layer)

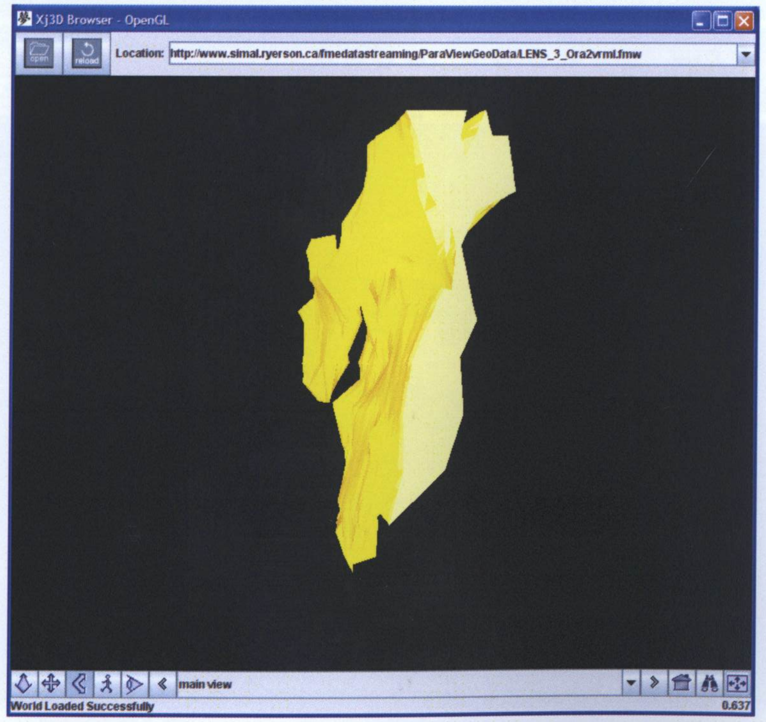

Figure 5.17 Xj3D Browser (FME streamed LENS 3 data layer)

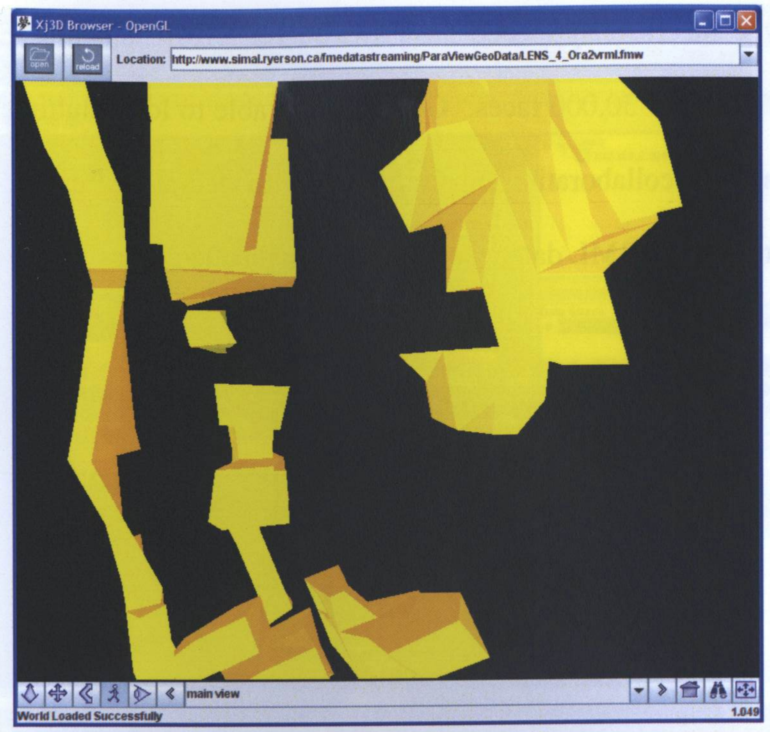

Figure 5.18 Xj3D Browser (FME streamed LENS 4 data layer)

GeoLink

GeoLink is a prototype that was developed at the Spatial Information Management and Applications laboratory (SIMAL) at Ryerson University. This tool allows collaborative modeling and sharing of surface and subsurface models, in a distributed team environment. The GeoLink prototype is designed on principles identified in the collaborative GIS framework study [Chang and Li, 2005]. The prototype system demonstrates how the synchronous sharing and manipulation among geographically dispersed users of 3D subsurface models can be realized; the focus is toward developing capacity that allows various types of domain-specific 3D models to be displayed and shared [Chang et al., 2006].

The FME Server streamed data was tested with the GeoLink prototype to determine compatibility of the FME streamed VRML models with the Collaborative GIS prototype. The

GeoLink prototype successfully loaded the FME Server VRML streams. Each of the streamed models included 10,000 – 30,000 faces. GeoLink was able to load multiple FME streamed data layers and perform the collaboration functions on the loaded data. Figure 5.19 and Figure 5.20 show the FME streamed VRML data visualized within the GeoLink Collaborative GIS

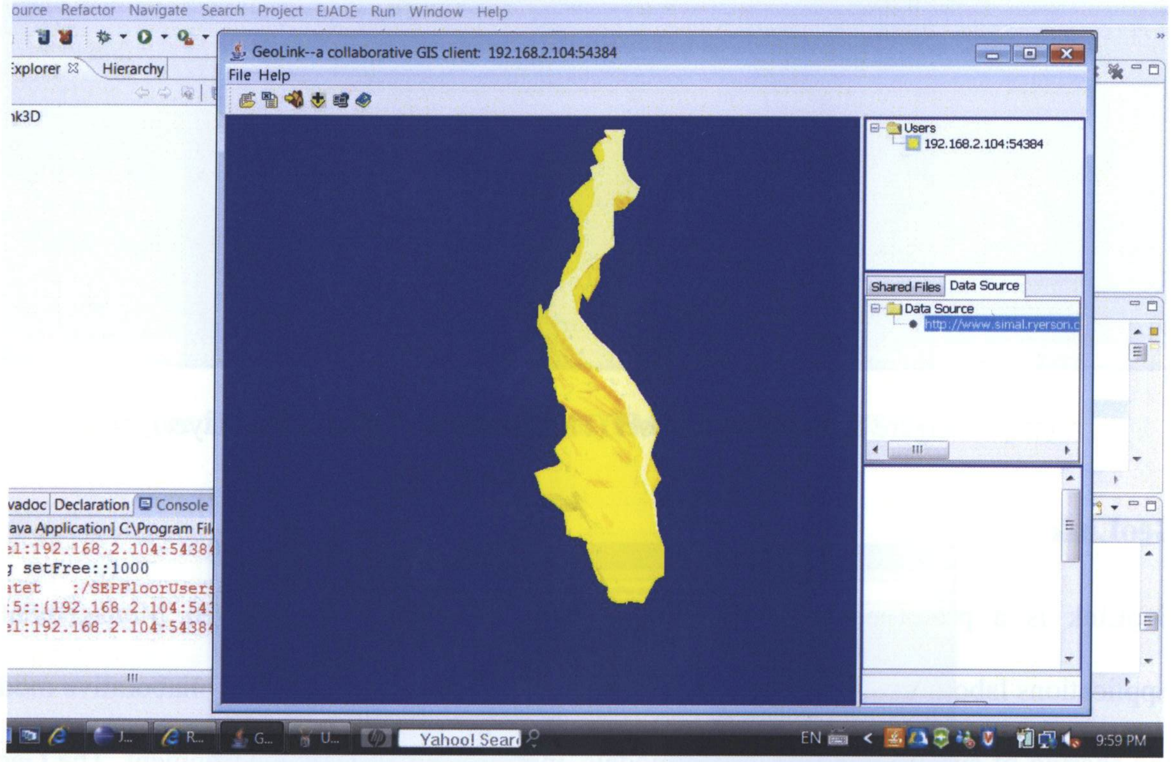

Figure 5.19 GeoLink Browser (FME streamed LENS 3 data layer)

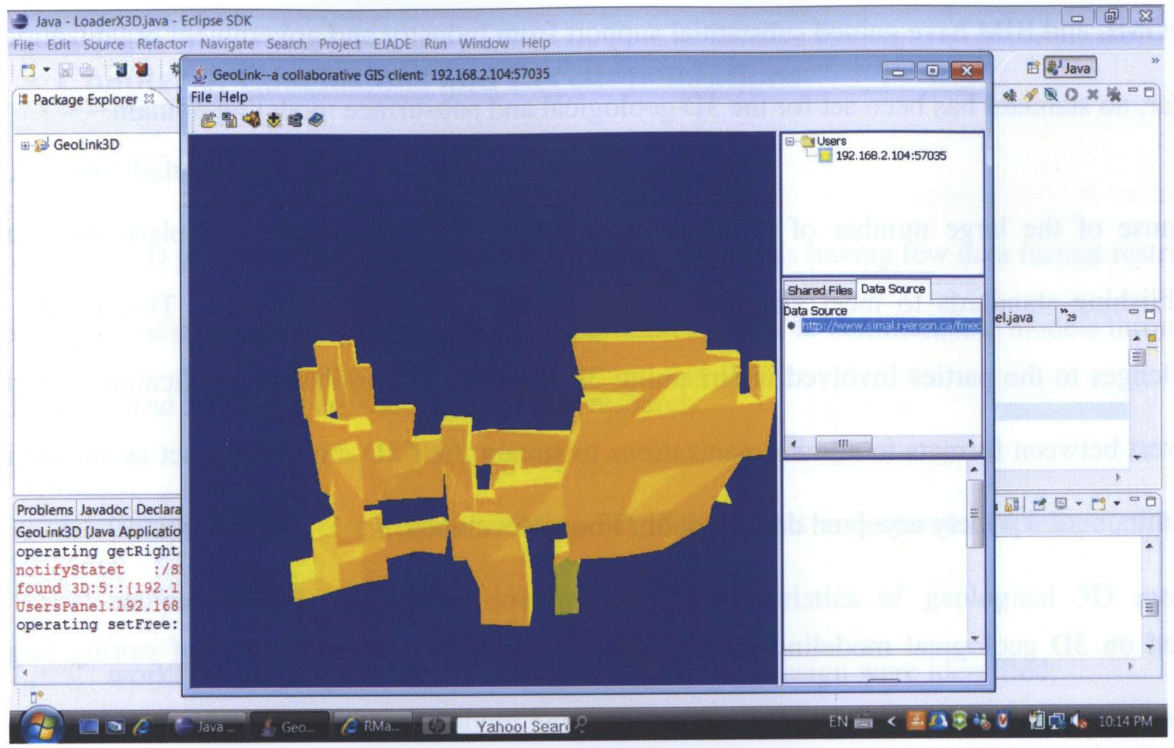

Figure 5.20 GeoLink Browser (FME streamed LENS 4 data layer)

5.2.4 Summary

Graphic representation of complex real world objects used in a geospatial models can assist a more rapid understanding of reality, something only possible when there is a high level of abstraction of the objects. A well-designed geospatial model can be used to perform tasks that are less convenient, too expensive or practically impossible to perform in the real world.

Spatial data, generated by different private and public organizations, is stored in a variety of data formats based on the organisation's platform preference. Because geological data accumulates over a long period of time and comes from many different sources, data compatibility is important for sustainable 3D geological modeling. Several initiatives to address interoperability issues in 3D modelling have been developed over the past decade. In the 3D modeling domain, CityGML and BIM have gained substantial support from industry and government organizations. So far, no standard has been set for the 3D geological and subsurface modeling domain.

Because of the large number of organizations involved, and the number of platforms used, establishing standards to meet all requirements of 3D modelling is difficult. This results in challenges to the parties involved in streaming 3D models. A middleware application that can convert between formats to allow organizations to stream their 3D models can act as an interim solution, until a widely accepted data model has been developed.

Based on 3D geological modeling requirements, a prototype system capable of storing large-scale 3D geological data and streaming the underlying model directly to Internet clients has been designed and implemented for GeoLink. It can stream different data formats, which can be organized as HTML responses. Several VRML clients, including the GeoLink Collaborative GIS prototype, can use the streamed data. GeoLink has been developed as a part of the topo3D project.

Chapter 6. Conclusions and Recommendations

Accomplishments

A robust 3D geological modeling framework along with tools having few data format restrictions is important steps toward real-time 3D collaboration systems to communicate models through the Internet. The following are the results of this research:

1. Comprehensive research on existing 3D data models and system implementations was completed and the results documented. Characteristics of geological 3D modeling workflow and requirements for a robust 3D system design were identified;

2. A technology-driven integrated framework capable of managing the 3D geological modeling workflow and supporting real-time 3D data access capacity was designed. Geological 3D data modeling requirements, and 3D data access and management requirements of the GeoLink prototype were modeled into an integrated framework design. This design enables format translation of external 3D geological data which is not directly usable by the GeoLink prototype;

3. A prototype system using Oracle Spatial 11g and FME Server applications was implemented. Some characteristics of the integrated framework design, such as storing large datasets and streaming geological models from Oracle 3D to a GeoLink Colloboratibe GIS client over the internet, were successfully tested.

Limitations of the Study

Because the prototype had been developed for a proof of concept, the results described in this thesis are subject to several possible constraints and limitations:

1. Because of geological modeling process complexity and dependency on manual processing, the identified process steps, and the order in which they occur, are not well defined. Therefore, the identified geological 3D modeling requirements may be too general;

2. Spatial 3D analyzing capacities was discussed, but these functions were not tested or modeled in the prototype. This is due partially to the client application's limited capacity to utilize 3D spatial queries and partially to the coding requirement for implementing these features at the DBMS or FME server;

3. The prototype system is capable of translating between most of the well-known spatial data formats. However, only Oracle 3D data was streamed as VRML format to feed the GeoLink, collaborative GIS prototype. Therefore, there may be constraints in data translating and streaming of some of the other supported formats.

Concluding Remarks

The Oracle 11g can offer a robust, efficient data-modeling environment for 3D geological modeling processes, providing that 3D data types can be used for storing all types of geological data and knowledge. However, 3D spatial functions and operators are not fully supported by Oracle, and no existing geomodeling tool, CAD or GIS application supports Oracle's 3D data

types. Until Oracle and geomodeling tools mature adequately to accommodate the functions and operations required for a sustainable form of 3D geological models, this research area will remain relevant.

An FME Server can be integrated into any existing solution to enable streaming 3D data, regardless of the underlying data format. The data access requirement set by the GeoLink Collaborative GIS client can be satisfied without data security concerns and without overwhelming the collaboration clients with data conversion and data transformation issues.

Future Work

This thesis provides a framework to manage the geological 3D modeling process and to communicate 3D models to client applications over the Internet. However, further research can provide a better understanding of areas not addressed here, including:

Editing the model on the client side and re-saving the modified version on the Oracle database. This operation may possibly be implemented, but may require some programming on the FME Server. The client application may also need further development to provide editing tools.

1. Converting Oracle's 3D data type to VRML worked for this project, but conversion to other formats and addressing data quality issues were not within the scope of this project.

2. All 3D models, but specifically 3D geological models, are big in size. Because of Oracle's capacity to store very large 3D models, a Level of Detail (LOD) implementation, on either Oracle or FME Server, may be the subject of further study.

3. Most 3D modeling formats, including VRML, use texture mapping to render picture-like models. This topic was not included within the scope of this project.

References

Abdul-Rahman, A. and Pilouk, M. (2008). "Spatial Data Modeling for 3D GIS", Springer-Verlag Berlin Heidelberg.

Altmaier, A. and Kolbe, T.H. (2003). "Application and solutions for Interoperable 3d Geo-Visualization", Available: http://www.citygml.org/docs/Altmaier_und_Kolbe_PhoWo2003_englisch.pdf

Apel, M. (2006). "From 3D geomodelling systems towards 3D geoscience information systems: Data model, query functionality,and data management", Computers & Geosciences 32 (2006) 222–229.

Blessent, D. (2006). " GeoTopo3D project", 8th GEOIDE Annual Scientific Conference. Banff, Canada, May 31 - June 2. Accessed on June 11, 2007, Available: http://geotopo3d.scg.ulaval.ca/documents/2006_Blessent_GeoTopo3D_presentation.pdf

Chang, E.Z. and Li, S. (2005). "VRML-Based 3D Collaborative GIS: A Design Perspective", Web and Wireless Geographical Information Systems, Springer.

Chang, E.Z., Li, S. and Kirkwood, D. (2006). "3D Synchronous Collaborative Modeling", 8th GEOIDE Annual Scientific Conference. Banff, Canada.

Chena, T.K., Abdul-Rahmana, A. and Zlatanova, S. (2008). "3D Spatial Operations for Geo-DBMS: Geometry vs. Topology", The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences. Vol. XXXVII. Part B2. Beijing.

Döllner, J. and Buchholz, H. (2005). "Continuous level-of-detail modeling of buildings in 3D city models", 13th annual ACM international workshop on Geographic information systems, ACM Press, New York, NY, USA, pp. 173-181.

Gröger, G., Reuter, M. and Plumer, L. (2004). "Representation of A 3-D City Model in Spatial Object-Relational Database ", FIG Commission IV, WG IV/1, Available: http://www.isprs.org/istanbul2004/comm4/papers/308.pdf

Gröger, G., Kolbe, T. H. and Czerwinski, A. (2007). "CityGML specification document version 0.4.0" Reference # OGC® project document: OGC 07-062, Available: http://portal.opengeospatial.org/files/?artifact_id=22120.

Haist, J. and Korte, P. (2006). "Adaptive Streaming Of 3D-GIS Geometries and Textures for Interactive Visualisation of 3D City Models", Fraunhofer Institute for Computer Graphics, Fraunhoferstr. 5, 64283 Darmstadt, Germany.

Heuel, S. and Kolbe, T.H. (2001). "Building Reconstruction: Dilemma of Generic versus Specific Models", KI - Zeitschrift für Künstliche Intelligenz.

Kaufmann, O. and Martin, T. (2008). "3D geological modelling from boreholes, cross-sections and geological maps, application over former natural gas storages in coal mines", Computers & Geosciences 34-278–290.

Kolbe, T.H. (2007). "CityGML – 3D Geospatial and Semantic Modelling of Urban Structures", GITA/OGC Emerging Technology Summit 4, Washington D.C.

Kothari, R., Godfrind, A. and Beinat, E. (2007). "Pro Oracle spatial for Oracle Database 11g", Springer-Verlag New York inc.

Latuada, R. (2006). "Three-Dimensional Representations and Data Structures in GIS and AEC, Large-scale 3D data Integration: Challenges and Opportunities, Taylor & Francis Group, LLC.

Marcus, A. (2004). "A 3D geosciences information system freamwork", PhD Dissertation, Doctorate de l'Institut National Polytechnique de Lorraine.

Nebiker, S. (2003). "Support For Visualisation And Animation in A Scalable 3d GIS Environment –Motivation, Concepts And Implementation", International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences, Vol. XXXIV-5/W10, Commission V, WG V/6

Oosterom, P.v., Stoter, J., Quak, W. and Zlatonova, S. (2002). "The Balance Between Geometry and Topology", Symposium on Geospatial Theory, Processing and Applications, Ottawa.

Oracle and Murray, C. (2007) Oracle, Oracle Spatial Developer's Guide 11g Release 1(11.1).

Perrin, M., Zhu, B., Rainaudb, J.F. and Schneiderb, S. (2005). "Knowledge-driven applications for geological modeling", ELSEVIR, Journal of Petroleum Science and Engineering 47 - 89–104

Pouliot, J., Lachance, B., Brisebois, A., Rabaud, O. and Kirkwood, D. (2003). "3D geological modeling: Are GIS or CAD appropriate?", Proceedings of ISPRS Workshop, WG II/5, II/6, IV/1 and IV/2 Joint Workshop on Spatial, Temporal and Multi-Dimensional Data Modelling and Analysis, October, 2-3, Quebec,Canada.

Pu, S. and Zlatanova, S. (2006). "Integration of GIS and CAD at DBMS Level", The third Dimension", UDMS, Available: http://www.gdmc.nl/publications/2006/ UDMS%2009-07.pdf

Rahman, A.A., Zlatanova, S. and Pilouk. M. (2001). "The 3D Software Development: global efforts from researchers and vendors", Geoinformation Science Journal, Vol. 1, No. 2.

Safe Software (2008a). "FME® Server Administrator's Guide", Accessed on December 2008, Available: http://docs.safe.com/fmeserver/pdf/FMEServerAdminGuide.pdf

Safe Software (2008b). "FME Installation and Licensing Manual ", Accessed on December 2008, Available: http://docs.safe.com/fme/2008/pdf/FMEInstallationLicensing.pdf

Safe Software (2008c). "FME Server Architecture", Available:http://www.safe.com/technology/documents/whitepapers/FME-Server-Architecture-Whitepaper.pdf

Shi, W., Yangi, B. and Li, Q. (2003). "An object-oriented data model for complex objects in three-dimensional geographical information systems", Int. J. Geographic information science, Vol. 17, No. 5, pp. 411-430

Stoter, J. and Zlatanova, S. (2003). "Visualization and editing of 3D Objects Organized in a DBMS", Proceedings of the EuroSDR Com V. Workshop on Visualisation and Rendering, 22-24, Enschede, The Netherlands, 16p.

Quadt, U. and Kolbe, T.H. (2005). "Web 3D Service", Open Geospatial Construction Inc, discussion paper, Available: http://www.opengeospatial.org

Wiki (2008) "Open Source - The Free Software Alternative", Available: http://paraviewgeo.mirarco.org/index.php/About

Worboys, M. and Duckham, M., (2004). "GIS: A Computing Perspective, Second Edition", CRC Press.

Wu, Q., Xu, H. and Zou, X. (2005). "An effective method for 3D geological modeling with multi-source data integration", Computer and Geosciences 31-35-43

Zlatanova, S., (2000). "3D GIS for Urban Development", PhD Thesis, Graz University of Technology.

Zlatanova, S., Rahman, A.A. and Pilouk, M. (2002). "3D GIS: Current Status and Perspectives", Symposium on Geospatial Theory, Processing and Applications, Ottawa.

Zlatanova et al., (2004). "Topological models and frameworks for 3D spatial objects", ELSEVIER, Computer and Geosciences, (30) pp. 414-428

Zlatanova, S., Holweg, D. and Coors, V. (2004). "Geometrical and Topological Models for Real-Time GIS", Disaster and Risk Management- Real time GIS @UDMS.

Zlatanova, S. (2005). "3D Geometries in Spatial DBMS", Innovations in 3D Geo Information systems, Springer, Lecture notes in Geoinformation and Cartography.

Zlatanova, S. and Stoter, J. (2006). "The role of DBMS in the new generation GIS architecture", Frontiers of GIS architecture, Springer, pp. 155-197.

Appendix A - Player support for X3D components by www.web3d.org

Table key

Yes all nodes, all fields supported for all levels of this component (though some bugs may be present)
Partial some nodes and fields supported
level # which component level number (1-4) is supported (found at end of each component specification)
no no support provided
? unknown, need status report

Players, versions, and X3D Conformance Certification	BS Contact	FreeWRL	Heilan	InstantReality	Octaga Player	OpenVRML	SwirlX3D	Vivaty	Xj3D
	v7.1	**v1.21.2**	v0.14	beta 5	v2.3.0.2	v0.17.9	v2.1.7	v1.0 build 900	1.0
	Interchange Profile	**Interchange Profile**	**none**	**none**	**none**	**none**	**none**	**Interchange Profile**	**Interchange Profile**
File Encodings									
- XML (.x3d)	yes	yes	yes	yes	yes	?	yes	yes	yes
- ClassicVRML (.x3dv)	yes	yes	no	yes	yes	yes	yes	yes	yes
- Compressed Binary Encoding (.x3db)	no	no	no	partial	no	no	no	no	yes
X3D component list									
CAD geometry	yes	no	no	yes	yes	partial	yes	no	yes
Core	yes	yes	partial (not Proto)	yes	yes	yes	yes	yes	yes
Cube map environmental texturing	yes	partial	no	yes	yes	no	no	partial	no
Distributed interactive simulation (DIS)	no	no	no	no	no	partial	no	no	yes

Environmental effects	yes	yes	level 2	yes	yes	partial	yes	yes	yes
Environmental sensor	yes	level 2	no	partial	yes	partial	yes	yes	yes
Event utilities	yes	yes	yes	yes	yes	partial	yes	yes	yes
Followers	no	no	no	yes	yes	no	no	no	no
Geometry2D	partial	yes	no	yes	yes	partial	yes	no	partial
Geometry3D	yes	yes	level 3	yes	yes	partial	yes	yes	yes
Geospatial	partial	yes	no	no	partial	partial	no	no	yes
Grouping	yes	yes	level 2	yes	yes	partial	yes	yes	yes
Humanoid animation (H-Anim)	yes	partial	no	partial	partial	partial	partial	yes	yes
Interpolation	yes	level 3	level 2	yes	yes	partial	yes	yes	yes
Key device sensor	yes	yes	level 2	partial	yes	partial	yes	yes	yes
Layering	yes	no	no	no	partial	no	yes	partial	no
Layout	yes	no	no	no	partial	no	yes	partial	no
Lighting	yes	yes	level 2	yes	yes	partial	yes	yes	yes
Navigation	yes	level 2	level 1	yes	yes	partial	yes	yes	yes
Networking	level 2	level 3	no	yes	yes	partial	yes	yes	yes
NURBS	yes	no	no	yes	partial	partial	yes	partial	no
Particle systems	partial	no	no	no	yes	no	no	partial	yes
Picking sensor	no	no	no	no	no	no	no	yes	yes
Pointing device sensor	yes	yes	no	yes	yes	partial	yes	yes	yes
Programmable shaders	yes	yes	no	yes	yes	no	partial	no	no
Rendering	level 3	level 4	level 4	yes	yes	partial	yes	yes	yes
Rigid body physics	partial	no	no	?	yes	no	no	no	partial
Scripting - ECMAScript	yes	yes	no	yes	yes	partial	no	yes	yes
Scripting - Java	partial (external)	partial (external)	no	yes	yes	partial	no	no	yes
Shape	yes	yes	level 2	yes	yes	partial	yes	yes	yes
- FillProperties node	yes	yes (requires runtime shader support)	no	no	partial	partial	no	no	partial

- LineProperties node	yes	yes	no	partial	yes	partial	no	no	yes
Sound	yes	yes	yes	yes	yes	partial	yes	yes	partial
Text	yes	yes	no	yes	yes	partial	yes	yes	partial
Texturing	yes	yes	level 1	yes	yes	partial	partial	yes	yes
Texturing3D	yes	no	no	yes	yes	no	no	no	partial
Time	yes	yes	level 2	yes	yes	partial	yes	yes	yes
Players	BS Contact	FreeWRL	Heilan	InstantReality	Octaga	OpenVRML	SwirlX3D	Vivaty	Xj3D