# A 3d geoscience information system framework

Der Fakultät für Geowissenschaften, Geotechnik und Bergbau

der Technischen Universität Bergakademie Freiberg

eingereichte

**DISSERTATION**

zur Erlangung des akademischen Grades

doctor rerum naturalium

Dr. rer. nat.,

vorgelegt

von Diplom-Geologe Marcus Apel

geboren am 7. Mai 1972 in Dresden

Freiberg, den 4. Oktober 2004

# A 3d geoscience information system framework

**THÈSE**

pour l´obtention du

Doctorat de l´Institut National Polytechnique de Lorraine

(Spécialité Géosciences)

par

Marcus Apel

Freiberg, 4 octobre 2004

# CONTENTS

# Abstract

Two-dimensional geographical information systems are extensively used in the geosciences to create and analyse maps. However, these systems are unable to represent the Earth's subsurface in three spatial dimensions. The objective of this thesis is to overcome this deficiency, to provide a general framework for a 3d geoscience information system (GIS), and to contribute to the public discussion about the development of an infrastructure for geological observation data, geomodels, and geoservices.

Following the objective, the requirements for a 3d GIS are analysed. According to the requirements, new geologically sensible query functionality for geometrical, topological and geological properties has been developed and the integration of 3d geological modeling and data management system components in a generic framework has been accomplished. The 3d geoscience information system framework presented here is characterized by the following features:

- Storage of geological observation data and geomodels in a XML-database server. According to a new data model, geological observation data can be referenced by a set of geomodels.
- Functionality for querying observation data and 3d geomodels based on their 3d geometrical, topological, material, and geological properties were developed and implemented as plug-in for a 3d geomodeling user application.
- For database queries, the standard XML query language has been extended with 3d spatial operators. The spatial database query operations are computed using a XML application server which has been developed for this specific purpose. This technology allows sophisticated 3d spatial and geological database queries.

Using the developed methods, queries can be answered like: *"Select all sandstone horizons which are intersected by the set of faults F"*. This request contains a topological and a geological material parameter. The combination of queries with other GIS methods, like visual and statistical analysis, allows geoscience investigations in a novel 3d GIS environment.

More generally, a 3d GIS enables geologists to read and understand a 3d digital geomodel analogously as they read a conventional 2d geological map.

# Résumé

Les systèmes d'information géographiques bidimensionnels sont très utilisés dans domaine des Sciences de la Terre pour la création et l'exploitation de cartes. Cependant, les GIS 2D ne permettent pas la représentation tridimensionnelle du sous-sol géologique. Le but de cette thèse est de combler cette lacune et de créer les fondements pour un système d'information géographique 3D ainsi que de prendre part au débat publique sur le développement d'infrastructures pour les données géologiques primaires, modèles géologiques 3D, et géoservices.

Dans cette perspective, les exigences d'un système d'information géographique 3D seront analysées. En réponse à ces exigences, des possibilités d'interrogation de propriétés géométriques, topologiques et géologiques ont été développées et une intégration orientée composant des logiciels de géomodélisation et des systèmes de gestion des banques de données a été créée. Le cadre du système d'information géographique présenté ici est caractérisé par les éléments suivants:

- Enregistrement de données géologiques primaires et des géomodèles dans des banques de données XML- compatibles. Sur la base d'un nouveau modèle de données, les données géologiques peuvent être référencées par des géomodèles.
- Des fonctionnalités d'interrogation géométrique tridimensionnelle, topologique et géologique portant sur les géomodèles 3D et les données primaires ont été développées et implémentées.
- Le langage XML standard d'interrogation "XQuery" pour interroger les bases de données a été complété par l'ajout d'opérateurs spatiaux. Les opérations d'interrogation spatiale 3D sont calculées au moyen d'un serveur d'applications spécialement développé. Ce nouveau type de technologie permet des interrogations spatiales et géologiques complexes sur les banques de données.

Grâce à ces méthodes, des réponses aux interrogations 3D peuvent être apportées, comme par exemple: *"sélectionner tous les horizons de grés qui sont affectés par les failles F"*. La combinaison d'interrogations avec d'autres méthodes SIG, comme par exemple, les analyses visuelles ou statistiques, permettent des travaux géologiques dans un nouveau cadre SIG. Dans un sens plus général, un SIG 3D permet aux géologues, de la même façon que pour des cartes géologiques 2D conventionnelles, de comprendre des modèles géologiques 3D.

# *Zusammenfassung*

Zwei-dimensionale geowissenschaftliche Informationssysteme (2d GIS) werden in den Geowissenschaften intensiv für die Erstellung und Auswertung von Karten genutzt. Jedoch vermögen es 2d GIS nicht, den geologischen Untergrund in drei räumlichen Dimensionen (3d) darzustellen. Das Ziel dieser Dissertation ist es, diesen Mangel aufzuheben und den Rahmen für ein 3d geowissenschaftliches Informationssystem zu schaffen sowie zur öffentlichen Diskussion über die Entwicklung einer Infrastruktur für geologische Aufschlussdaten, 3d Geomodelle, und Geodienste beizutragen.
Der Zielstellung folgend, werden zunächst die Anforderungen an ein 3d GIS analysiert. Entsprechend den Anforderungen wurden geologisch sensitive Abfragemöglichkeiten an geometrischen, topologischen und geologischen Eigenschaften entwickelt und die komponenten-orientierte Integration von 3d Geomodellierungs-Software und Datenbank-Managementsystem geschaffen. Das vorgestellte geowissenschaftliche Informationssystems-Framework ist durch die folgenden Merkmale charakterisiert:

- Speicherung von geologischen Aufschlussdaten und Geomodellen in XML-fähigen Datenbanken. In einem neu entwickelten Datenmodell können geologische Aufschlussdaten von Geomodellen referenziert werden.

- 3d geometrische, topologische und geologische Abfragefunktionalitäten an Geomodellen und Aufschlussdaten wurden entwickelt und als Plug-in für eine 3d Geomodellierungs-Software implementiert.

- Für Datenbank-Abfragen wurde die XML Standard-Abfragesprache XQuery um räumliche Operatoren erweitert. 3d räumliche Abfrageoperationen werden mit Hilfe eines speziell entwickelten Applikations-Servers berechnet. Diese neuartige Technologie erlaubt komplexe 3d räumliche und geologische Datenbankabfragen.

Mit den entwickelten Methoden können 3d Abfragen beantwortet werden, wie zum Beispiel: ”*Selektiere alle Sandstein-Schichten die von der Menge von Störungen F geschnitten werden.*”. Die Kombination von Abfragen mit anderen GIS-Methoden, wie beispielsweise visueller und statistischer Analyse, erlaubt geowissenschaftliches Arbeiten in einer neuartigen 3d GIS-Umgebung. Im allgemeineren Sinn ermöglicht ein 3d GIS Geowissenschaftlern digitale 3d Geomodelle analog zu konventionellen 2d geologischen Karten zu verstehen.

# *Acknowledgements*

This research project has been funded by the Robert-Bosch-Foundation, the Deutsche Forschungsgemeinschaft, and the Association Scientifique pour la Géologie et ses Applications. Thanks to all who contributed financially and made this thesis possible.

During this research project, many scientists and colleagues shared their thoughts with me, and inspired and motivated me. Especially I wish to mention here Prof. Dr. Gerald van den Boogaart, Dr. Guillaume Caumon, Dr. Olivier Grosse, Dr. Eric de Kemp, Dr. Uwe Kroner, Dr. Bruno Levy, Thomas Jerome, and Dr. Sophie Viseur. I am grateful to the whole Gocad Research Group in Nancy who shared their knowledge of geomodeling with me and provided a pleasant atmosphere.
Not only scientists but also staff is essential for research in order to cope with daily tasks and bureaucracy. Thus I wish to thank Madame Cugurno, Frau Klocke and Frau Schmiedgen for their reliable, thoughtful work.

Particularly I owe thanks to Tobias Frank for his very constructive collaboration. He helped significantly to realize my ideas concerning the XML-based geodata management. I hope that our collaboration and friendship will persist, and wish him all the best for his thesis project with the Gocad Research Group.

The manuscript of this thesis owes many improvements to the remarks of its proof-readers. I wish to thank Dr. Jens Richter, Prof. Dr. Jean-Laurent Mallet and Prof. Dr. Helmut Schaeben for their efforts and valuable time.

Prof. Dr. Jean-Laurent Mallet leads the Gocad Research Group, and with his ideas and generous support he opened a treasure of resources for me. Without him, Gocad and thereby one fundamental basis of this work would not exist. I thank him very much for the supervision of this thesis and the great chance to spend one year with his research group.

Finally, I wish to give special thanks to Prof. Dr. Helmut Schaeben. He provided the initial idea for this research project, and as supervisor he has inspired me to accomplish far more than I ever expected. Without his influence, this thesis and my graduate school experience would have unfolded quite differently. I am, and will remain, appreciative of his generosity and support.

# 1. INTRODUCTION

## *1.1 Research field ”3D Geoscience Information Systems”*

Geosciences investigate the Earth which is a spatially three-dimensional (3d) object and evolves through time. Two-dimensional (2d) maps have been the major means of communication between geoscientists ever since the origin of geosciences. The introduction of geographical information systems into the geosciences facilitated the creation and the interpretation of 2d maps by techniques of information technology. In geology, 2d geoscience information systems (abbreviated ”GIS” in this thesis) consisting of an user application environment and a database management system (DBMS) are extensively used to create, manage, query, and analyze georeferenced maps [7]. GIS provide a means to generate abstract models of real world geological situations based on data. Following Bonham-Carter [7], there are five core activities of current 2d GIS that can be applied in geoscience applications:

1. *Data management* is done in a database according to a spatially referenced data model. Data of heterogeneous source, type, and confidence are to be compiled and stored. Geoservices allow to share information between geoscientists.

2. *Data visualization*: GIS facilitate the creation of 2d data views, map models, and the visual inspection of spatial patterns.

3. Combined *spatial and nonspatial querying* is possible because links between spatial features and associated nonspatial feature attributes in a relational database are maintained.

4. *Data analysis* is achieved through combining different map layers and examining them simultaneously to discover their relationships.

5. *Prediction* supports decision making based on multiple factors of spatial information. Knowledge-driven and data driven methods using Bayesian

models have been used to predict mineral potentials [7].

During the last decades a steep increase of acquired digital geoscience data could be observed. This is mostly due to the usage of efficient new methods. Examples are geophysical methods like ground penetrating radar and 3d seismic surveys, remote sensing, geochemical methods like micro-probing, age determination methods like fission track analysis, and digital field mapping techniques [11]. Continuing developments suggest that this trend will in future rather increase than stagnate. The issue to manage, process and interpret these data results in the usage of GIS in the geosciences.

In GIS, geological objects are commonly represented as map objects in two spatial dimensions. While real geological objects are essentially referenced in 3-dimensional space, a 2d map represents a cross-section through a 3d geological space. ”Spatial” extensions of common 2d GIS are at present not applicable for 3d geological applications as they are not capable of representing 3d spatial geological relationships and properties with 3d spatial variation. 2d GIS represent the altitude values $z$ of geoobjects as a continuous function of the geographical coordinates: $z = f(x,y)$. That way they cannot model 3d geological objects which have multiple $z$-values for a single $x$,$y$-value. On the other hand, 3d geomodeling software provides data models and functionality to represent geological situations in 3 spatial dimensions as geomodels.

Geomodeling systems are widely applied in the petroleum and mining exploration industry, geological surveys, and academic science departments. Gocad v.2 ([34], Earth Decision Sciences, Houston/TX) is one of the most evolved geomodeling environments available. Based on the unique discrete smooth interpolation method and a sophisticated topology [34], Gocad allows to build sophisticated structural models honouring heterogeneous input data. It facilitates 3d geomodeling in a unified, geoobject-oriented way and also provides advanced 3d visualization, material property modeling. Thus Gocad can be seen as a core 3d GIS user application.

## 1.2 New aspects of 3d GIS functionality

The increase of digital geodata and the possibility to create regional 3d geomodels results in new, specific needs for geodata management, and extended possibilities for geodata query and analysis. However, both fields are underdeveloped in existing 3d geomodeling environments. This becomes obvious especially when comparing geomodeling software with mature 2d geographical information systems.

*Geodata management.* Current 2d GIS provide specific solutions for data management of 2d raster and vector data in relational databases. Recent developments focus on interoperability and the usage of standards. The OpenGIS Consortium (www.opengis.org) of GIS vendors and institutions is developing standards for geodata exchange, like the GML data model [22]. This can be a starting point for interoperability between 2d GIS databases and 3d geomodeling software.
On the other hand, no appropriate solution for unified storage and query of geological observation data and 3d geomodels exist to date. The available systems including Gocad, Petrel (Schlumberger IS, Houston/TX), and EarthVision (Dynamic Graphics, Alameda/CA) offer solely file-based storage of geomodels. Past attempts to develop database-supported 3d GIS resulted in systems dedicated to a specific database system with non-standard interfaces. It is likely that the ”GeoToolKit” 3d GIS [10], which is based on an object-database and CORBA, is not being used in practice due to these insufficiencies.
While 2d GIS applications are commonly coupled with database servers for storage, available 3d geomodeling software products still use a file-based data storage. This has several drawbacks, for example:

- the access to large datasets is difficult because no query functionality exists,
- no consistent multi-user access exists, and
- the data safety is low compared with database management systems (DBMS).

In order to improve these deficiencies, a change from a file-based data storage to a networked database server is required. From these arguments the necessity to develop a new data management system for 3d GIS data can be deduced.

*Demand for new query facilities.* Geomodeling software allows to create and visualize 3d geomodels of geological situations including their structures and material properties. In addition, the software Gocad comprises extensive capabilities for geodata analysis based on multivariate statistics and geostatistics.
However, the existing functions are not appropriate for sensible geological and spatial queries. Geometrical and attribute queries belong to the core functionality of 2d GIS. One primary concern of this thesis is to develop query methods based on geological, geometrical and topological properties and relationships in

3d space which is impossible in 2d GIS. The investigation of 3d models of large and complex geological situations using queries may lead to new insights and may allow the systematic search of exploration targets.

Due to their complexity and high level of user interaction, statistical data analysis should not be considered as part of the query language. Instead, it may be used separately within a geomodeling application as an expert user component of a GIS.

## 1.3 Outline of this thesis

The objective of this thesis is to overcome the existing shortcomings related to the data management and geologically sensible query functions in 3d geomodeling systems, and to provide a general framework for a 3d geoscience information system. This work shall also be a contribution to the public discussion about the development of an infrastructure for geodata, geomodels, and geoservices. Several research fields of a 3d GIS are examined: data model, query functionality, and data management. Accordingly, this thesis is subdivided as follows:

1. Analysis of the user requirements with respect to the data model, geomodeling functionalities, and data management (chapter 2).

2. Design of a data model based on the user requirements and under consideration of geomodeling software constraints (chapter 3.2).

3. Design of GIS-specific user application functionality; especially for spatial and geological queries (chapter 4). These can be used with an existing 3d geomodeling software in a plug-in technique.

4. Design of a geodata management system incorporating a database server and development of spatial database query functionalities (chapter 5).

Each chapter will have a separate introduction. The developed concepts are supported by a prototypical implementation and by testing the system components using available data. Within this work the creation of a complete 3d GIS system with hundreds of user commands comparable to a mature 2d GIS is impossible. Therefore, the core functionality will be designed and implemented in a generic, standards-oriented and extensible framework.

# 2. REQUIREMENTS FOR A 3D GEOSCIENCE INFORMATION SYSTEM

## *2.1 Concepts and requirements for the data model*

The geological knowledge generally accumulates starting from data available as measurements and observations. Measurements and observations are interpreted and set in context to comprise information. Geological concepts, human thought and intuition often extend this information by ideas. For a geoscience information system it is crucial to represent knowledge in an appropriately approximated and comprehensive way. A data model is formally defined as a set of fundamental conceptual objects and mathematical and logical rules that govern their behaviour. The careful choice of the conceptual objects and their relationships is the key to an intelligible and complete computer representation of geological knowledge. The following paragraphs describe the requirements for a geological 3d GIS data model.

*Related works.* Several data models exist to represent geological field observations and map data in relational data bases, the most advanced and complete of which are the NADM [40] and the NATMAP [12] data models. These models are not designed to represent geological objects including their spatial relationships in three spatial dimensions, but are focused on geological observations and 2d maps. The NADM includes a data model for geological concepts. This can be used as a basis for a data model for geological observations.
Different from those approaches, the OpenGIS consortium (www.opengis.org) created a data model for the representation of geographical data using an geoobject-centric view.

*From observations to models.* A data model for geological data need to account for both observed data and modeled data. Observed data are only available at a finite set of points in 3d space. Observational data captured by geologists in the field are commonly stored in a field book and include qualitative descriptions of the petrographic composition, indicators of stratigraphic facies and age like fossils, fabric and structural descriptions and measurements, drawings, images, and meta-data of sampling locations. Often, samples are being examined in a laboratory and age data and quantitative geochemical and petrophysical data are obtained.

Spatial models are derived by spatial interpolation of data observed at points and can represent geological situations in 3d space. To create a digital geological model using the object-oriented approach, the subsurface space has to be discretized into homogeneous regions based on a chosen parameter. Examples for commonly used parameters are the petrographic composition, the stratigraphic age, or tectonic structures. The choice of the parameters and the classification into subsets is done by a geologist according to the geomodeling project motivation. This may lead to different models which are created by inversion from a single data set, but with a different parameterization.

*Geoobject.* In computer science, the standard method to model the properties and behaviour of conceptual entities is the object-oriented approach. Geoobjects are mappings of spatial entities of the Earth subsurface to abstract computer objects. Commonly, geoobjects are inferred in an iterative and scale dependent way from observational data: evidence is collected and hypotheses are formulated which are either supported or refused by further evidence. In that context, geoobjects are evolving and mutable abstractions which aid in the analytical process of geological interpretation by means of a GIS.

By definition, an object consists of a set of variables and methods which define its properties and behaviour, respectively. A geoobject is characterized by one or more parameters which are constant or vary within an interval at each of its spatial points. These parameters constitute the geoobject definition criteria.

A geoobject is composed of a name, a spatial description including geometry and topology, and a thematic description including geological or numerical properties. Numerical properties are scalar or vector variables which may be defined for every point in space in a discrete way, or globally. Geological properties can be defined for a geoobject and may include alphanumeric descriptions, for example the lithological or structural classification of a geoobject.

A different geospatial data model has been developed by the geographic information system community. The OpenGIS data model [22] uses meta-data to qualify spatial features. No inherent distinction between observed facts and concepts is made. Thereby it intends to represent the real world in a static way, while models are merely treated as meta-data. In geology, this approach is problematic as the geoobjects are commonly not static but change according to the geologists' knowledge based on observations and concepts.

In geology, models can be inferred from observations via a description based on concepts, as depicted in figure 2.1. Here, observation points are associated with property descriptions containing any geological data, knowledge, and assumptions derived from the observation points. Property descriptions are based on conceptual geological models, like stratigraphy or structural evolution. Also based on geological concepts, a set of geoobjects can be defined. Geoobjects represent named geological entities of the Earth crust with spatial and thematic property descriptions. These are in turn associated with the observation points. A geological 3d GIS needs to represent all components of knowledge related to geoobjects and observations.

The value of the parameter which leads to the subdivision of the subsurface into a set of geoobjects is not accurately known at any point in space. For example, the probability p that the lithology is constant within a fixed spatial region is always $p<1$. This is due to the facts that the property value varies in space, and that models are often based on sparse input data and a vague geological concept. Moreover, a problem of GIS systems representing geoobjects by their boundaries is that they cannot account for fuzzy spatial transition zones between geoobjects. A promising approach to quantify uncertainties is a continuous geoobject representation with 3d grids. Here, a membership function property can be assigned to discrete cells. This membership function can, for example, be computed by multi probability field simulation [28].

*Geomodel.* A geomodel is an abstract digital representation of a part of the Earth subsurface. Two main discrete approaches exist to partition the space into a set of mutually exclusive and collectively exhaustive volumes: geoobject-based models and regular grid models. In practice, a complete geomodel comprises both a unique structural model representing the topology and geometry of the subsurface geoobjects (figure 2.2 shows an example), and a property model providing a mechanism to model the material properties of each geoobject.

For discrete spatial property modeling, regular grids are commonly used. This method allows to integrate all available information into one model. Using a membership function, it is possible to model the spatial extension of geoobjects. Major drawbacks are the non-existence of explicit topology and topological concepts

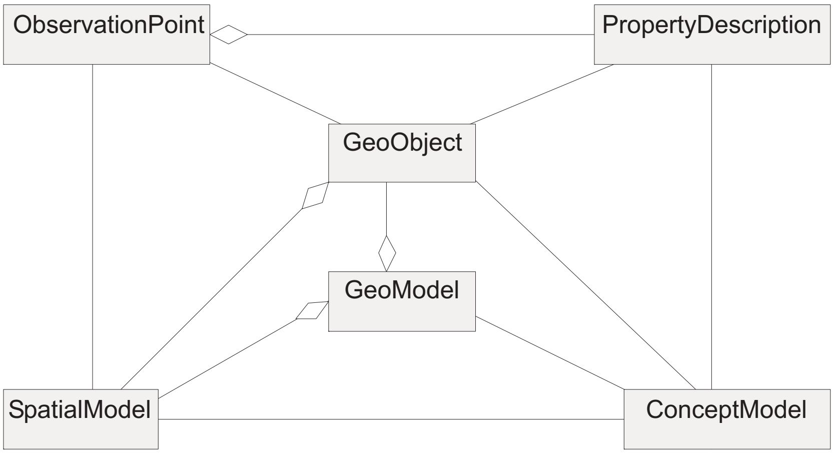


Fig. 2.1: UML diagram of the top-level conceptual data model. Geoobjects are inferred from observations by concepts. Edges denote association relationship, edges with diamonds denote aggregation relationship.

like boundary, which, for example, allows the concise representation of faults. The combination of structural and property geomodels incorporating all available information related to its constituting geoobjects, is also called ”shared Earth model” [20].

In general, it can be stated that representing geoobjects solely by their boundaries is sufficient if the structural geological situation is of primary interest, or if available material property data are too sparse to build a property model. This is often the case for regional geological mapping campaigns, and during the initial state of local geomodeling projects. This work is mainly concerned with the development of GIS-functionalities for geoobject-based models.

*Topology.* A space can be abstracted as a set of points. Topology adds a structure to these points by defining neighbourhood in a qualitive way: each point in the space knows which points are in its neighbourhood. Geometry and topology relate to each other like absolute and relative location. In chapter 3.1.1 a formal introduction to topology is given. Using a topological model the geological space can be partitioned into subsets representing geoobjects. Geoscientific queries are often topological, and it chapter 3.1.1 of this thesis will show how it is possible to answer them efficiently by focusing on the topology of the space, and not its geometry. For query purposes in a GIS, the macro topology is of particular interest as it describes relationships between the geological objects of a geomodel. A topological model can be helpful to check the consistency of the spatial model

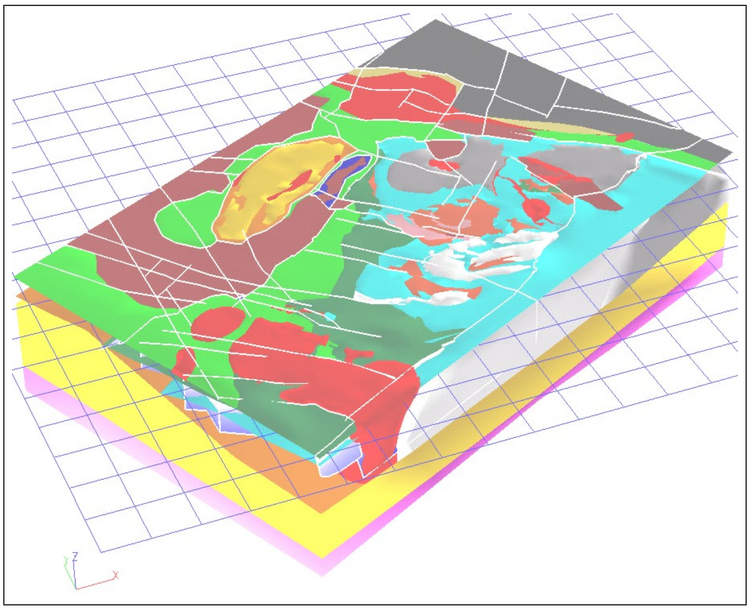


Fig. 2.2: Example of a boundary representation geomodel showing the Erzgebirge region/ Germany. This large-scale crustal model is based on geological and geophysical data and has been created using Gocad software. To examine such regional models, spatial and geological query functionality is required.

with geological concepts, to explore the spatial geoobject relationships, or to select subsets of a set of objects using specific topological criteria.
Topology is implicitly contained in the geometric description of a geomodel, but it can also be stored explicitly. If stored explicitly, fast combinatorial algorithms can be used instead of computational geometry. The separate representation of the geometry and the topology of geoobjects provides the foundation for efficient topological query functions which are based on algorithms with linear complexity.

*Geometry.* The geometry defines the absolute spatial reference of geomodel objects in a coordinate reference system in 3-dimensional euclidean metric space. Since we have to map spatially continuous geological situations to a computer representation, the geometry has to be approximated with finite numbers. For its relative computational simplicity and common usage in the geosciences a local cartesian coordinate reference system is preferred.

*Geological properties.* The semantic description and other non-spatial properties of geomodel objects and their composite elements constitute the geological properties. The set of properties depends on the project purpose. The core geological properties which can be associated with a spatial model include:

1. Material composition. This can be described, for example, by chemical constituents or a classified rock type name. The composition may change as a function of geological time.
2. Time of genetic events, and genesis processes. The time interval of genesis may also be described by a classified stratigraphic name.
3. Structures, strain, strain rate, stress field, temperature. For metamorphic rocks these properties are commonly represented by a function in the stress-temperature-time domain.
4. Physical parameters derived from direct measurements, and geophysical models.

In common GIS, properties of simple data types are stored in a relational attribute database. In a 3d geological GIS, properties of different data type need to be defined for the micro- and macro topological elements such as nodes or surfaces. For geological objects, a wide range of simple data types (for example floating point numbers, strings, binary images) and complex data types (for example a semi-structured sample point description, functions) need to be stored as properties.

## 2.2 Requirements for the geomodeling component

A geoscience information system essentially needs a full-featured geomodeling front-end. For such, the following features are required in particular:

1. a data model for geomodels which allows to represent the topology, geometry and material properties of geoobjects
2. functionality to build sophisticated structural models which honour heterogeneous input data
3. possibility to update a geomodel by re-interpolation, if new input data need to be honoured
4. 3d visualization
5. property modeling functionality including advanced geostatistical methods

These requirements are met by currently available systems. One of the most widely used and evolved 3d geomodeling applications is Gocad (Earth Decision Sciences, Houston/TX). Gocad facilitates 3d geomodeling in a unified, geoobject-oriented way. Its geomodeling capabilities are largely based on the unique discrete smooth interpolation method [34]. A large set of application objects for geomodeling is available: point sets, lines, surfaces, topological boundary models, and irregular and regular grids. Gocad can be seen as a core 3d GIS user application.

## 2.3 Requirements for data exchange and data storage

Storage and exchange of geospatial data between databases and different front-ends like 3d geomodelers, GIS or internet browsers require a format which is capable to represent instances of geomodels and geological observations. By the use of standardized data models and formats a minimum loss of information during data transfer and minimum interface development efforts can be ensured.

*Data exchange using XML.* XML is a mark-up language for documents containing structured information. Structured information contain both content (text, graphics, equations, etc.) and an indication of the semantics of that content. A mark-up language is a mechanism to identify structures in a document. The XML specification defines a standard way to add markup to documents.

During the last years, XML [49] was adopted as the standard mark-up language for data interchange by the ISO, the W3C (www.w3.org), and the OpenGIS consortium (www.opengis.org). Based on XML, the data definition language XML Schema, the comprehensive query language XQuery, and mature tools like parsers and DBMS for the storage of XML documents evolved. An XML Schema is a valid XML document which uses object oriented features in order to define data models for XML-formatted document exchange and database storage. These features include inheritance from existing data types by restriction or extension, complex data types, referencing, and name spaces. The ”OpenGIS Consortium” of GIS companies and user groups created specifications for geographical data exchange and a XML Schema named ”Geographical Mark-up Language” [22].

During discussions with potential users of a 3d GIS it became clear that the numerous requirements could be realized only with a XML based system. Arguments for geodata management based on XML are:

- suitable for both highly structured data like matrices, and semi-structured data like a textual observation description

- possibility to define object-oriented data models
- long-term usability is provided as the documents can be stored in human-readable self-describing text format
- vendor neutral, platform independent, ISO standardized format; XML standards are free available and widely used
- straightforward creation and maintenance of XML schemas. These can easily be adapted or extended for specific, customized applications
- usable by different geoscience front-ends:
  - authoring applications like geomodelers, GIS, field mapping software
  - viewing applications like internet browser using X3D format
  - storage applications like XML-capable DBMS, file system
- available mature libraries provide programming interfaces for fast application development
- comprehensive, extensible query language ”XQuery” [48] is available

A disadvantage is the need for relatively large storage capacities, which are however not a limiting factor. For example, very large XML databases are successfully being used for mission-critical business and logistics projects by numerous companies and agencies [18]. As a future alternative for highly structured data the recently planned binary XML variant by the World Wide Web consortium (www.w3.org) should be considered. This format is especially applicable for data with large storage needs, like high-resolution images and 3d grids.

*Data storage and data serving.* Specific *spatial database* interfaces exist for some 2d GIS like ArcGIS. These are however not appropriate for 3d geoobjects, because the data model is very different and the representation of 3d hierarchical topological models is not supported. Also, semi-structural geological data like observation point descriptions and complex geomodels with hierarchical topology are difficult and time-consuming to map to relational structures. Moreover, the object-oriented character of the data gets lost when geoobjects and observations are distributed over a large set of tables. This can be avoided when XML-supporting databases are used. The 3d GIS proposed in this thesis should have a DBMS-based data management with the following features:

1. client-server architecture, where geomodeling applications act as clients of a database server
2. consistent multi-user access via IP network
3. support for XML and XML Schema
4. XML query language for data access, including geometrical and topological operators
5. high data safety

The coupling of a professional 3d geomodeling software and a DBMS can provide a 3d geoscience information system offering comprehensive spatial and geological query capabilities.

## 2.4 The role of queries in a 3d GIS

*Peculiarities.* Querying geographically referenced geoscience data (geodata) is an essential aspect of a GIS. If one considers non-spatial properties of geodata, several generic data mining tools apply to geodata analysis:

- descriptive statistics, infering multivariate statistics,
- linked map and graph displays, and other visualization tools ,
- association rules detection, sequence discovery,
- predictions using neural networks and memory based reasoning.

These tools are available in several data mining and statistical analysis software packages. However, due to special characteristics of geoinformation, generic non-spatial queries often provide not satisfying results, while queries for spatial properties and relationships are of high interest in geosciences. Particular features of geological information which need to be accounted for by a GIS include:

- Data are referenced in 3d euclidean space. Complex spatial relationships occur.
- Data are referenced in a geological time scale. For the sake of simplicity, geological time can be treated as a scalar property of observation data and geomodel objects. A time-dependent representation of geodata would result in a spatio-temporal GIS.
- Spatial objects can be defined using different classification parameters, for example composition, stratigraphic age, or structure.
- Complex spatial-temporal-property data interactions occur.
- Data uncertainty is often spatially structured; standard statistical methods are thus seldom helpful. Non-stationarity may occur.

A 3d GIS should provide a means to pose both spatial and geological property queries. Such functionality need to be available both within the geomodeling user front-end and at the DBMS. If required, specialized non-spatial data mining and statistical softwares may be integrated with the GIS in an open, component based environment.

The queries important for geological purposes are based on the topology, geometry, and non-spatial properties of geomodel objects. Spatial and non-spatial query operators can be combined with logic expressions to define a comprehensive query language. Applications of such a query language are to select sub-sets, to apply spatial functions, to check the consistency of the geological model, or to obtain property information.

*Geological consistency checks.* If one creates a geomodel, it may not be spatially or geologically sound due to insufficient, inaccurate or imprecise input data, inaccurate geological concepts, or inappropriate geomodeling methods. Also if we add new data to a model, a geomodel may become inconsistent. Such inconsistencies can be detected by checks which make use of a combination of the topological, geometrical, and semantical properties of geoobjects.

For example, if the result of the query: *"Select all fault surfaces of Permian age which intersect Cretaceous horizons.*" is not an empty set, then either i) the age of at least one object is wrong, ii) the geometry of at least one object is wrong, or iii) the geological interpretation of at least one object is wrong (for example, the fault as such is a misinterpretation).

*Selection.* Selection is particularly required to constrain mining exploration targets or environmental damages. Such queries are set-theoretic selections which may contain spatial parameters. The query with the *non-spatial condition ”Select all cells of a set of geoobjects with a geochemical anomaly A.”* gives a set of cells as result. This set of cells comprises a new spatial region. Selection queries may use geometrical, topological, and geological properties of geoobjects. When the source of an ore deposit, or endangered regions nearby a waste dump are of interest, *spatial conditions* need to be added to the query parameters. For example, the query *”Select all fault surfaces which are closer than distance d to the geochemical anomaly A.”* can give clues about the ore source of a deposit, or ground water contamination from a waste dump which is situated above a fault aquifer. Selection queries can also be used to detect geological or spatial relationships between geoobjects.

To illustrate possible spatial queries in a GIS, here are some further examples:

- *”Select the set of fault surfaces with given mean normal direction AND a given geochemical anomaly within a certain distance.”* This query may be used for exploration sensitivity studies for hydrothermal ore deposits.
- *”Select the set of geoobjects with a certain permeability AND which have given faults as their boundaries.”* This type of query can aid in the understanding of fluid movements.
- *”Select the sets of geoobjects which occur in a given stratigraphic succession.”* This can be used to detect stratigraphic patterns.

*Property queries.* Queries for properties do not select any geoobjects, but return property values of these. Properties can be stored in the database and queried there directly, like a query *”Return the stratigraphic age value of geoobject A.”*, or can be computed within an application from other properties. An example for such a property is the euclidean distance between two points, which can be computed from the geometry. Such queries need to be answered by user applications and are not directly required for database requests.

# 3. DATA MODELING

Special features of geological data are their 3d spatial reference. How can spatial data like geoobjects be stored in computers? Geoobjects cannot be represented in a computer directly, as the amount of data required would be infinite. Instead, real world geoobjects have to be approximated and abstracted. The way of representing geoobjects is a very important aspect of a GIS and essentially influences the geomodeling functionality and the query possibilities. The approaches to model geoobjects can be grouped into *geoobject-oriented representations* and *regular grid representations* (also named *field- or raster representations*). When discussing spatial data models, the following parameters have to be optimized:

- accurate and precise representation of real geoobjects including their geological semantics
- computational efficiency and numerical robustness
- representation of geometry and topology

In the following sections the spatial and geological data representations which are being used in current geomodeling systems and geological databases will be investigated for their application in a geoscience information system. Then a data model will be chosen for observational data and geomodels, and formulated using the standardized Unified Modeling Language (UML, [44]), and XML.

## *3.1 Introduction*

### *3.1.1 Representation of space*

*Point set topology.* Geological matter can abstractly be regarded as sets of points on which set-theoretic operators can act and functions and relations may be defined. Topology is the mathematical study of properties of objects which are preserved through deformations, twistings, and stretchings. Point set topology,

also called set-theoretic topology or general topology, is the study of the neighbourhood properties of sets. Sets can thereby be endowed with structure.
The definitions in this section are taken from Goldberg and Bishop [6] and Munkres [35]. Formally, a topology on a point set $X$ is a subset $T \subseteq 2^X$ that satisfies the following conditions:

- the empty set and $X$ are in $T$,
- $T$ is closed under arbitrary unions, and
- $T$ is closed under finite intersections.

A *topological space* is a set $X$ with a topology $T$ on $X$. The sets in a topology on $X$ are called *open sets*, and their complements in $X$ are called *closed sets*. The *interior* $A^\circ$ of a set $A \subseteq X$ is the union of all open sets contained in $A$. The *closure* $\overline{A}$ of a set $A \subseteq X$ is the intersection of all closed sets containing $A$. The *boundary* of a set $A$ is $\delta A = \overline{A} - A^\circ$.
Two spaces are *homeomorphic* if they can be deformed into each other by a continuous, invertible mapping. *Manifolds* are defined to be sets such that the neighbourhoods at all of their points are homeomorphic to a disk.
In geomodeling, surfaces play an important role as they concisely allow to define the spatial extent of geoobjects and structures. In a point set theoretical notion, surfaces are compact and connected spaces with the following property: Each point of a surface has a neighbourhood homeomorphic to either the plane $\mathbb{R}^2$, or the half-plane $\mathbb{H}^2$. Points of the first type are called interior points, and those of the second type are called boundary points. The set of all boundary points constitutes the boundary of the surface. It consists of one or more boundary components, each of which is homeomorphic to a circle. If the surface has no boundary, it is called a closed surface.

*Algebraic topology.* Algebraic topology is the study of algebraic objects attached to topological spaces. Combinatorial topology is a special branch of algebraic topology that uses combinatorial methods. The explicit representation of topology in a computer data structure is possible by combinatorial topology methods which decompose a complex $n$-manifold object into a set of elementary cells.
In 3d geomodeling two main approaches can be distinguished to represent the spatial extension of geoobjects:

- cellular models, and

- boundary representation models.

*Cell complexes.* An (open) n-cell is a topological space homeomorphic to an open ball $\mathbb{E}^n$ of $\mathbb{R}^n$. For $\mathbb{R}^3$ are important cells:

- a 0-cell, called ”vertex”, is an isolated point
- a 1-cell, called ”edge”, is a simply connected curve without ending points
- a 2-cell, called ”face”, is a simply connected open surface without border
- a 3-cell, called ”volume”, is a simply connected solid without (closed) border surface.

A cell complex is a set $K$ of cells in $\mathbb{R}^n$ satisfying two conditions:

- Every face of a cell is a cell in $K$)
- If $P$ and *P’* are cells, then their intersection is a common face of both.

The body $|K|$ of a complex $K$ is the union of all cells. Geomodels based on cell complexes will be discussed in the context of GMaps in section 3.1.2.2.

*Simplicial complexes.* A simplex is an elementary geometric building block in a given dimension: a 0-simplex is a point (node), a 1-simplex is a line segment, and a 2-simplex is a triangle. Abstract simplexes contain no geometric information.
A simplicial complex is as a cell complex whose cells are all simplices. When a subset $P$ of $\mathbb{R}^n$ is the body of a simplicial complex $K$, then $K$ is said to be a triangulation of $P$. A closed surface is a simplicial complex partitioning the plane. Having constructed a simplicial complex, we can divide it into topological and geometric components. The former will be an abstract simplicial complex, a purely combinatorial object, easily stored and manipulated in a computer system. The latter defines the embedding of the vertices of the complex into the geometric space where the complex is realized.

*Geometric space.* A metric space has an associated metric which enables us to measure distances between points in that space. Thereby, topological neighbourhoods are implicitly defined. The geometry defines the absolute spatial reference of geomodel objects in a coordinate reference system in 3-dimensional Euclidean metric space $\mathbb{R}^3$. The geometry and topology of a space are fundamentally related, as they are both properties of the same space. Geometric modifications can alter the implicit topology.

### 3.1.2 Object representation

In geomodeling, the object representation can be considered as a mapping of real world geometrical objects to abstract computer geometrical objects. Object representation tries to model geoobjects as objects which fill an empty geometric space. This can be realized using discrete representations or polynomial functions.

Using object oriented data modeling techniques, geoobjects can be hierarchically composed of discrete topological elements of different dimensionality: nodes, edges, faces, and volumes. Geometrical and other property information can be associated with topological elements. Besides properties, the objects have a set of methods which define their behaviour. The group of objects with the same set of variables and methods is called a class.

While geoobjects intend to model a set of spatial geological matter as complete as possible, computer representations require abstraction from nature. For efficiency reasons, it is common practice to model geoobjects by their boundaries, and model the material properties only in volumes of high interest by the means of 3d grids, like reservoirs and ore bodies.

*Discrete geoobject representation* The method of discrete representation of geoobjects has been investigated by many authors, namely J.L. Mallet [34]. This section gives an overview of the approaches, focusing on the features which are important with respect to a 3d GIS data model.

Discrete object representation is based on regular or irregular tessellations to model the spatial extent of geoobjects using surface partitions or volume partitions. The space between nodes is piecewise interpolated with linear or low grade polynomial splines. Appropriate data structures which allow to define a full topology for geoobjects in $\mathbb{R}^3$ are the boundary representation (BRep) and generalized maps (GMaps). Both approaches model the topology of a geomodel independent from its geometric embedding.

#### 3.1.2.1 Boundary representation (BRep)

The BRep approach models the spatial extent of geoobjects by a discretization of their boundary. Abstractly, a BRep is an acyclic directed graph which corresponds to a combinatorial map. Every node in the graph stands for an element of the BRep. The term element refers to the simplices vertex, edge, triangle, or higher dimensional boundary faces. A triangulated BRep is a type of a BRep, where the boundary faces are comprised by a set of triangulated surfaces. In this

thesis, *macro topology* refers to the BRep, while *micro topology* refers to the internal topology of the simplicial complexes.
A triangulated radial edge Weiler representation [46] as a version of the BRep is implemented in the Gocad v.2 geomodeling software. It has been extensively used, and proved to be applicable as a kernel for a geomodeling software. The Weiler model [46] is based on the algebraic topological connection of maps called radial-edge edge structure which allows to represent non-manifold topology.
Using this technique, the geological space can be partitioned into regions. For example, the space can be partitioned into volume regions using a stratigraphical or lithological classification as criterion. Figure 3.1 illustrates the mechanism by which the radial-edge structure represents non-manifold topological structures. There is a non-manifold condition occurring along the contact between the horizon surface and the fault surface. The radial-edge structure stores the adjacency information of faces about this edge. One can loop radially about the edges and extract the list of adjacent faces.

*Separation of geometry and topology.* The radial edge representation implemented in Gocad maintains a hierarchy of topological elements. Each level of the hierarchy corresponds to various stages of the discretization process as one moves from a geomodel down to a mesh with its associated numerical properties. The abstraction hierarchy is comprised by the following five models:

1. discrete model (geometry model)
2. discrete topological model. It is composed of
    (a) volume decomposition model
    (b) face decomposition model
    (c) edge, border decomposition model
    (d) mesh discretization model

The main purposes for this hierarchy is to provide constraints and inheritance for interactive modeling, and to keep it consistent during the modeling process. In practice, the constraints are commonly given by the input data, and their structural and stratigraphic interpretation.
The discrete model and the topological mesh discretization description represent,

respectively, the conceptually highest and lowest levels in the hierarchy. Following Mallet [34], the *discrete model*

$$M^n(\Omega, N, \phi, C)$$

consists of a triplet composed of the topological graph defined on a set of nodes $G(\Omega, N)$, the functions $\phi$ which define the geometry and numerical properties, and the set of constraints $C$ to be honoured. There are many properties to be associated with the discrete model. Such properties include constraints, and scalar and vector numerical material properties including the geometrical location vector. For example, numerical properties can be stored with topological elements like model regions, or the mesh vertices. By separating the topology and geometry in the data model, one can change the geometry of a model by manipulating the vertex coordinates without changing the topology of the model.

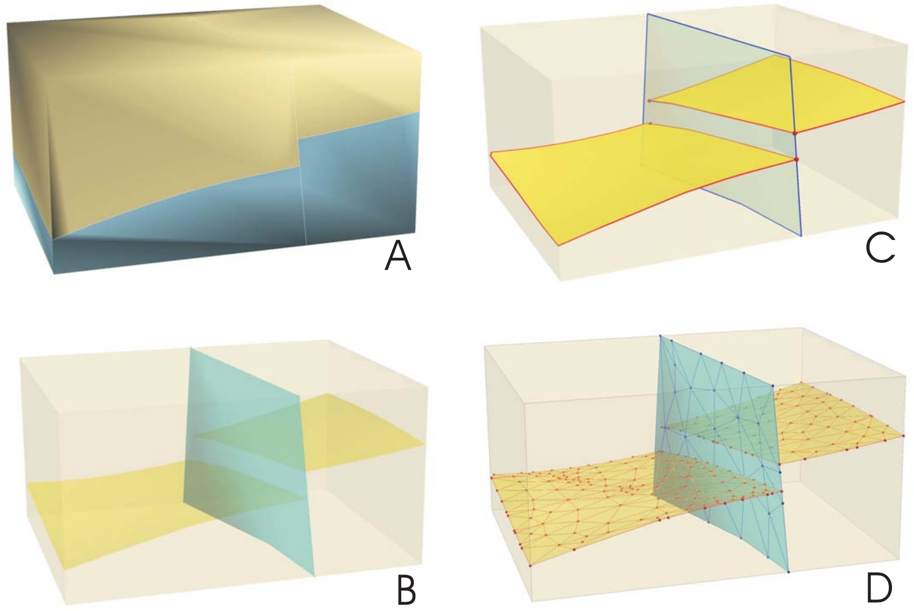


Fig. 3.1: BRep model hierarchy: A) volume decomposition, B) surface decomposition, C) border loop subdivision; a non-manifold condition occurs on the horizon - fault contact line, D) triangulation

*Topological elements.* In the BRep model the topology is represented by the use of topological elements in the adjacency relationship information, rather than by

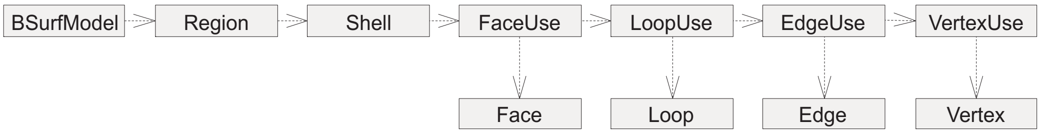


Fig. 3.2: Radial edge topological elements.

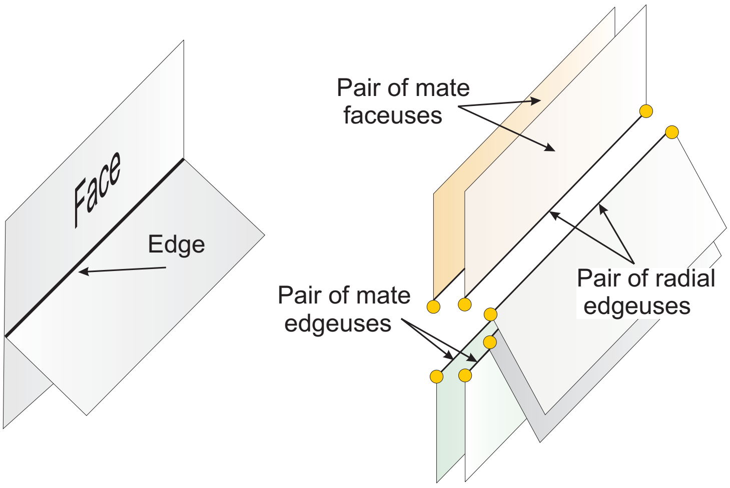


Fig. 3.3: The radial-edge database uses the list of edge-uses, ordered radially about an edge to manage the manifold and non-manifold features.

the topological elements themselves. The basic topological elements are the vertex, edge, loop, face, shell, region, and model (see figure 3.2). For the elements vertex, edge, loop, and face there is a distinction between the existence of the element and instances of the use of the element. This allows multiple topological elements to share the same geometry using pointers. For example, each side of the face is uniquely represented by a faceuse, that means every face is referenced by exactly two faceuses. The introduction of the various use structures greatly simplifies most of the algorithms that modify and query the topological representation. Because of its importance for the development of a 3d GIS based on the Gocad data model, a description of the radial edge topological entities after Weiler [46] is given here :

- A *model* is a single three-dimensional topological modeling space, consisting of one or more distinct (though perhaps adjacent) regions of space. A model is not strictly a topological element as such, but acts as a repository for all topological elements contained in a geometric model. The modeled space is completely partitioned into a collection of regions forming a 3D manifold.

- A *region* is a volume of space. There is always at least one in a model. Only one region in a model may have infinite extent (”universe region”); all others have a finite extent, and when more than one region exists in a model, all regions have a boundary.

- A *shell* is an oriented boundary surface of a region. A single region may have more than one shell, as in the case of a solid object with a void contained within it. A shell may consist of a connected set of faces which form a closed volume or may be an open set of adjacent faces, a wireframe, or a combination of these, or even a single point.

- A *face* is a bounded portion of a shell. It is orientable, though not oriented, as two region boundaries (shells) may use different sides of the same face. Thus only the use of a face by a shell is oriented. Strictly speaking, a face consists of the piece of surface it covers, but does not include its boundaries. A faceuse is one of the two uses (sides) of a face. Faceuses, the use of a face by a shell, are oriented with respect to the face geometry (figure 3.3).

- A *loop* is a connected boundary of a single face. A face may have one or more loops, for example a polygon would require one loop and a face with a hole in it would require two loops. Loops normally consist of an alternating sequence of edges and vertices in an open circuit, but may consist of only a single vertex. Loops are also orientable but not oriented, as they bound a face which may be used by up to two different shells. Thus, it is the use of a loop that is oriented.
  A loopuse is one of the uses of a loop associated with one of the two uses of a face. It is oriented with respect to the associated faceuse.

- An *edge* is a portion of a loop boundary between two vertices. Topologically, an edge is a boundary curve segment which may serve as part of a loop boundary for one or more faces which meet at that edge. Every edge

is bounded by a vertex at each end (possible the same one). An edge is orientable, though not oriented; it is the use of an edge which is oriented.
An edgeuse is an oriented boundary curve segment on a loop-use of a faceuse and represents the use of an edge by that loopuse, or in case of a wireframe edge, by endpoint vertices (”radial nodes”). Orientation is specified with respect to edge geometry. There may be many uses of a single edge in a model, but there will always be an even number of edge-uses (since each use by a face produces two edgeuses, one for each face side.

- A *vertex* is a topologically unique point in space. Single vertices may also serve as boundaries of faces and as complete shell boundaries.
  A vertexuse is a structure representing the adjacency use of a vertex by an edge as an edge point, by a loop in the case of a single vertex loop, or by a shell in the case of a single vertex shell.

- The *tessellation* defines the discrete cellular partition of the boundary surfaces of BRep model into simplices. This comprises the final level in the topological hierarchy. Because of its relatively easy handling in computers the tessellation of surfaces is commonly realized by triangulation (see figure 3.1).

#### *3.1.2.2 Cellular models and generalized maps (GMaps)*

In his study of the combinatorial structure of cellular partitions Lienhardt [32] defined a combinatorial structure called generalized maps (GMap). This mathematical approach defines a class of representations named cellular models since each kind of cell (vertice, edge, polygon, polyhedra, ...) plays an equivalent role within the model. Cellular models may be thought of as a generalization of a BRep. Each element is recursively defined as a discretization of its border into elements of lower dimension.
The following definition is taken from Levy et al.[30]: ”The definition of G-Maps is based on the incidence graph, which describes in a n-dimensional cellular model all the possible paths that can be taken to go from a n-cell to a connected 0-cell, with arcs connecting only k-cells to (k-1)-cells, n>k>0 (figure 3.4). Considering such a graph, it is possible to define relationships between two paths. For instance, in figure 3.4, two paths can be taken to go from face $F_1$ to vertex $V_1$, namely ($F_1$; $E_1$; $V_1$) and ($F_1$; $E_2$; $V_1$); as those two paths are identical but for the edges (1-cells), they are said to be 1-adjacent.

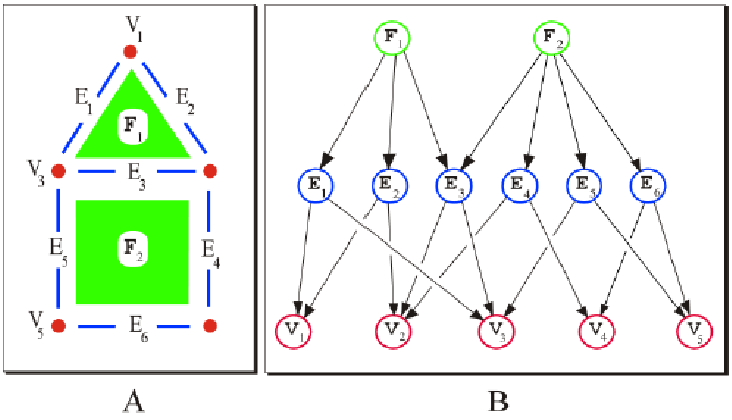


Fig. 3.4: Example 2-GMap (A) with its incidence graph (B)[30]

The properties of these relationships are mathematically well-defined and allow, together with the paths themselves, to describe completely the object. As a consequence, a n-dimensional object can be represented using a unique element type called dart, each dart standing for a path in its incidence graph. The adjacency relations between the paths are directly translated in terms of relations between darts. Thus, a k-adjacency between two paths is represented by a k-link between the corresponding darts 1.” An example 2-G-Map is shown in figure 3.5.

G-Maps allow to generically define polygonal curves, triangulated or polygonal surfaces, tetrahedral or polyhedral volumes, or even arbitrary hypervolumes. Levy [31] proposed the cellular sub-partition of the geomodel space into ”macro-cells” representing volume geoobjects (figure 3.6). In this concept, a higher level GMap called ”frame” defines the explicit relations between objects of the same dimension. The frame of a surface is composed of two oriented faces which are glued by $\alpha_2$ involutions. A cellular volume model can be created by assembling adjacent frames at a common border by $\alpha_2$ involutions. That way, a geomodel can be created by a cellular partition where each 3-cell represents a volume region. Hierarchical GMaps allow to represent space subdivisions by connected oriented surface which is in a way comparable to the radial edge data structure.

The geometry and other properties can be represented separately with each *n*-cell. This can be achieved by associating a discrete model with a GMap. A prototype GMap software library has been implemented (”TopoLab” research project, [30]) and proved to be applicable for geomodeling.

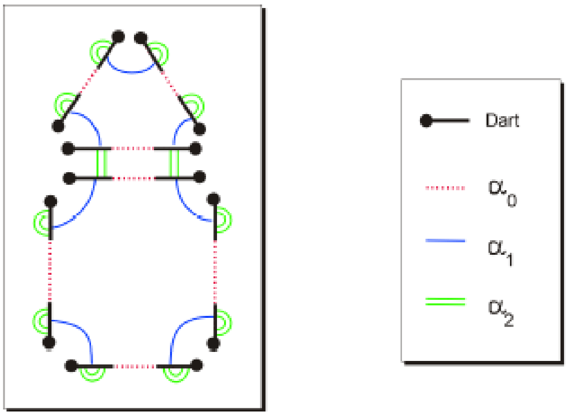


Fig. 3.5: Example 2-GMap with darts symbolized by bullet-headed segments, dotted lines stand for 0-functions, arcs of a circle correspond to 1-involutions, 2-links are represented by double lines [30].

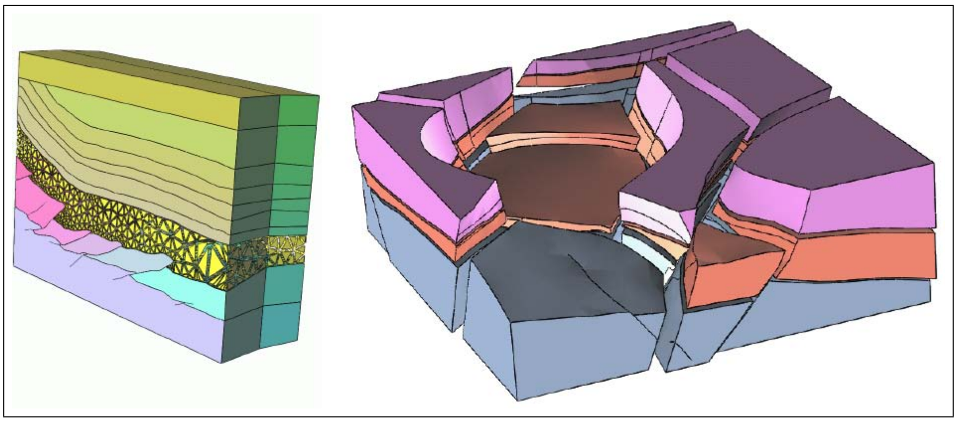


Fig. 3.6: A sub-partitioned GMap surface (left), and a GMap model build from sub-partitioned surfaces (right) [30].

#### *3.1.2.3 Object representations in cellular volumes*

The object representations in cellular volumes can be considered a hybrid of object representation and field representation. Here a geoobject is modeled as a spatial region which is filled with a regular or irregular tessellation composed of $3$-cells. Thus properties can be modeled in a discrete way within the whole geoobject volume. Two different approaches can be distinguished:

1. *3d tessellations with explicit topology.* In Gocad, so called `Solid` objects are composed of connected tetrahedra resembling 3-simplicial complexes. That way a volume region which represents a geoobject and is bounded by triangulated surfaces can be completely filled. The embedding and micro topology of the nodes is explicit defined in the data structure. Also, numerical properties are carried by vertices.
   `Solid` models can be associated with a BRep surface model in order to provide such information. In Gocad, this approach is currently being implemented in so-called `SolidFrame` models [29].
   Using *hierarchical GMap* topology the geomodel objects can be represented as macro cells which are decomposed by irregular 3-cells. That way, a full 3d topological and property model can be created by a cellular volume partition.

2. *Regular 3d grids with implicit topology.* In Gocad, `SGrids` are curvilinear grids with parallelepipedic cells. `SGrids` can be deformed and cut in order to fill the volume region of a geoobject defined by its boundary surfaces. Numerical properties can be carried by the cell nodes or the cell center. As a drawback, for `SGrids` no topological model comparable to BRep models or hierarchical GMaps exists. This makes it impossible to represent complete geological situations. `SGrid` models are being built only at a small scale, like for hydrocarbon reservoirs. Moreover, their creation is relatively time-consuming.

#### *3.1.2.4 Polynomial geoobject representations*

In the past years, Non-Uniform Rational B-Splines (NURBS) have evolved as an essential tool for a semi-analytical representation of geometrical entities encountered in 3d CAD applications. However, splines and polynomial methods fail to represent very irregular complex shapes which are required for geological objects [9], and moreover polynomial functions of higher order are mathematically very difficult to manipulate and have high computational demands. Therefore, such representations may only play a supporting role in geomodel building using sparse data [26].

### 3.1.3 Regular 3d grid representation (Voxet)

A 3d grid defines a continuous space which is decomposed into 3-cells. The cells are adjacent, connected, and addressable by an index. Parallelepipedic or curvilinear grids are commonly used to represent properties in continuous 3d space. Properties can be assigned to the cells or grid nodes, and interpolated at any point in the grid space. For example, a property *geoobject.name* can be defined for a grid, which allows to represent identifiable geological units with a spatial extent. That way geoobjects can be represented as a set of cells filling the interior of that geoobject (figure 3.7). The spatial resolution depends on the cell size of the grid. The geoobject representation using grids is often limited to a particular kind of discretization and connectivity.

Field representations implemented using regular tessellations provide an implicit geometry and topology which can be used for fast set theoretic spatial query computations.

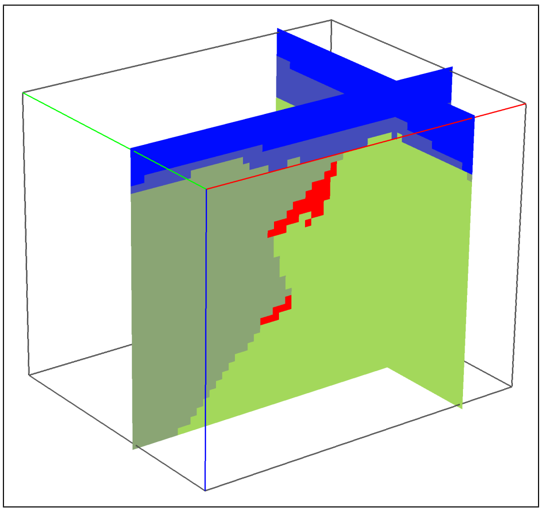


Fig. 3.7: Field representation of 5 geoobject regions filling a Gocad Voxet. 2 sections are shown. (data courtesy Mira Geoscience Ltd.)

## 3.2 Synthesis of a new spatial-geological data model for 3d GIS

### 3.2.1 Comparison of existing spatial geoobject representations

In the following section the different geoobject representations are discussed in respect of their applicability for a 3d geoscience information system.
Discrete object models feature a high flexibility to represent arbitrary shapes at arbitrary resolution, and allow the separation of the geometric embedding and combinatorial macro- and micro topology. General drawbacks are the relatively high roughness and high storage needs. However, in geomodeling a very precise representation is hardly required. In Gocad both the roughness can be minimized and input data be approximated in an appropriate way using the discrete smooth interpolation method [34]. Such geomodels can represent the spatial extent of natural geoobjects. While for geological investigations, 3d mapping and visualization the modeling of volume geoobjects as surface-bounded regions or macro cells is sufficient, quantitative property modeling generally requires a volume discretization into 3-cells.

*Triangulated BRep.* The triangulated radial edge boundary representation has proved to be a stable and efficient data structure used for the Gocad geomodeling software. Geological situations with very sophisticated geometry can be modeled with consistent topology in a fast and user-friendly way. The radial-edge BRep models have the advantage that they encode the full topological structure of an object as well as the geometric information. The representation contains full topology information so that the relationships between vertices, edges, loops, faces and shells are available. The representation of macro- and micro topology in the triangulated BRep model allows for comprehensive query functionality. Drawbacks of the BRep approach are:

1. Eleven data types corresponding to the topological entities and numerous operators are required
2. From a theoretical point of view, it has no concise algebraic foundation as compared to GMaps. 3d geoobjects are represented as the interior volume region of their boundary surfaces and thus are rather ”emulated” instead of represented directly.
3. No discrete 3d volume property modeling is possible. In practice the cellular partition corresponding to the BRep surface model is not appropriate

to comply with the needs of specific applications such as geostatistics. It is necessary to partition these volume regions into smaller 3-cells in order to model spatial property variations inside a geoobject. To achieve this, BRep model regions representing geoobjects can be filled with grids (in Gocad: `SGrids`) or a tetrahedra tessellation (in Gocad: `Solids`). This method is commonly used for property modeling in Gocad.

*GMaps.* GMaps allow to represent geological objects themselves as 3-cells instead of their 2-dimensional boundaries. Major advantages are the genericity and simplicity to represent *n*-dimensional objects using just one data type and operator, and the possibility to represent the micro- and macro topology of geomodels in one consistent data structure with an algebraic foundation. These facts provide the basis for a concise implementation of GMaps as a geomodeling software kernel. A prototype GMap-based implementation has been developed by the Gocad research consortium, which is however not mature to be used as component for a 3d GIS.

With respect to 3d topological query functionality, hierarchical GMap models can provide an elegant way to implement a query language based on the concept of involution. Similarly to the BRep, a comprehensive set of queries based on the micro- and macro topological model can be compiled.

*Volume representation.* In the tessellated volume object representation spatial queries have to be computed using the geometric and explicit topological properties. Such models are useful for fast queries on discrete properties. Regular curvilinear grids or volume tessellations which are constrained by a boundary representation are widely used in practice for discrete property modeling of single geoobjects like hydrocarbon reservoirs or ore bodies.

The disadvantage of the grid approach is that the explicit macro topological BRep model exists independent from the grid objects, which fill BRep model regions representing geoobjects. On the other hand, the use of curvilinear grids makes it possible to model numerical properties of a geoobject in a fast and discrete way.

Tetrahedra-tessellated solids integrated with a BRep surface model, like the `SolidFrame` approach [29], can provide a means to maintain both the spatial extent of geoobjects and their discrete properties in one model.

A promising solution to represent topology, geometry, and properties in a unified way is the development of hierarchical GMap models [31] whose macro cells are decomposed into discrete 3-cells. GMap based volume models are very qualified

for a 3d GIS, as the combination of the topological features of GMaps with discrete volume property modeling leads to a unified geomodel on which a rich set of spatial and property queries can act.
However, in practice such full geomodels are often not accessible because sufficient property data is available only in a very limited region, like a targeted ore body or hydrocarbon reservoir, and not throughout the whole geomodel space. This can lead to a mixed representation, where some well-sampled target regions are decomposed using 3-cells for property modeling, and other regions are just represented by their dividing walls.

*3d grid regular representation.* The grid representation allows fast property computations and topological queries. As the topology between the regular 3-cells is implicitly given, queries for neighbourhoods are straightforward. On the other hand, no macro topological model is defined and queries on the topological neighbourhoods and connectivity of geoobjects have to be computed from the elementary topology of the cells. Another drawback is that the spatial resolution is limited by the cell size, which results in very large storage costs for good approximations of distict features like faults.

### 3.2.2 The feasible solution: a Gocad- and GML-conform spatial geological data model

A geoscience information system framework should be based on a comprehensive data model for geological observational data and geomodels following the requirements given in chapter 2. These include primarily:

1. representation of all 3d geomodels and observational data which are related to one geological situation (see figure 2.1) in one project. Data include particularly the 3d geometry, topology, and textual and numerical geological properties.

2. conformable with the GML [22] specification for geospatial data exchange using XML format

3. compatible with 3d geomodeling software

In order to fulfil these criteria, a new data model has been synthesized by adding the concept of observation points to the BRep-based data model of Gocad. The design of the data model is depicted in figure 3.8. A project essentially contains an identifier, meta-data, and a bounding box for spatial database retrieval. It can reference a set of observations and a set of geomodels. Geomodels reference the observation data they are based on using an identifier. That way it is possible to maintain several models from the same set of data within one project. Geomodels are either boundary surface models, tessellated solid models, or grid models. The next sections describe a new spatial data model which is has been obtained by extending the existing Gocad BRep-based data model. For grids, no new data model has been specified within this work. Instead, the Gocad Voxet regular grid can be used.

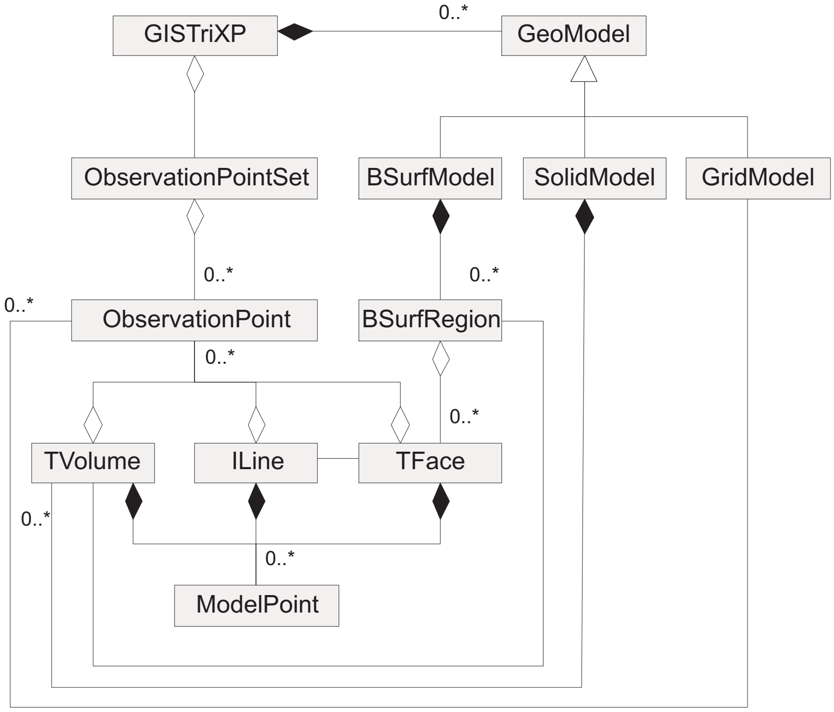


Fig. 3.8: Simplified UML diagram of the spatial macro object model. Note: a "`GISTriXP`" project is an association of both `ObservationPointSet` and `GeoModels`. Geomodels are associated with `ObservationPoints` by aggregation (void diamond symbol), and with `ModelPoints` by composition (filled diamond symbol).

### 3.2.2.1 Spatial elements of the structural geomodel

The Gocad data model based on a triangulated radial-edge BRep provides the capabilities to create geomodels according to the requirements posed in section

2. Although the BRep model is in some respects inferior to hierarchical GMap models, the data models are comparable and convertible [14]. Thus a similar set of micro- and macro topological and geometrical queries is possible. Because of Gocad's common usage and sophisticated geomodeling functionality this thesis aims to design and implement a GIS which will integrate with the released Gocad version and be compatible with its data model. The resulting spatial data model used here has the following substantial characteristics:

- topology and discretization. Geomodels are decomposed by a hierarchical explicit topological model (triangulated radial edge ”Weiler” BRep, [46]). Depending on their dimensionality, geoobjects can be represented as surfaces or surface-bounded volume regions. This approach is implemented in the Gocad geomodeling software.

- differentiation between `ModelPoints` and `ObservationPoints`. `ModelPoint` vertices are members of the geoobjects and represent points which are introduced during the geomodeling process in order to achieve models with smoothly interpolated geometry and numerical properties. `ObservationPoints` contain the actual input data.

- object orientation. Geoobjects encapsulate both a spatial and a geologic property description. The topological geoobject data model is compatible with the data model of the Gocad geomodeling software:

    1. `Lines` are used to represent a set of polylines, whose `ModelPoints` and `ObservationPoints` are connected by segments. A `Line` is a set of topologically isolated `ILines`.
    2. `Surfaces` is a set of topologically closed triangulated surfaces called `TFaces`. In the spatial data model, surfaces are used to represent 3d geological boundaries like fault surfaces, stratigraphic boundaries, or other discontinuities.
    3. `BSurfModel` (corresponding to `Model3d` in Gocad) is used to subdivide the geomodel space into topological closed volume regions bounded by a set of oriented `TFaces`. `BSurfModel` is an implementation of the triangulated radial edge BRep model [46]. In addition, `BSurfModel` provides the possibility to group regions to layers or fault blocks. For example, a region can belong to both a fault block and a stratigraphic horizon. The BRep model macro elements can carry properties including geological semantic.

4. `TSolids` represent a set of `TVolumes`. `TVolumes` are tetrahedric 3-cell complexes which can be used to fill the volume regions of `BSurfModel` objects.

The root element is a project `GISTriXP` (The name was chosen to reflect ”GIS”, ”Tri” for 3d, ”X” for XML, and ”P” for project). The children `ObservationPointSet` and `GeoModel` contain observed and modeled data, respectively. `GeoModels` can be subdivided into irregular tessellated surface models (`BSurfModel`), solid models (`SolidModel`), and regular `GridModels` as shown in figure 3.8. The irregular tessellated models may contain also isolated collections of isolated objects: `PointSets`, `PLines`, and `TSurfs`. These collections are not depicted in figure 3.8.
The Gocad objects have been extended by a new object named `MetaGeoObject`, which is an aggregation of faces or regions. This allows to group topological elements which conceptually belong together, like for example a set of conjugate faults, or a set of layers which belong to one stratigraphic group.
The child node `ObservationPointSet` represents the set of input data points. These are referenced by the model objects by an ID and can be shared by multiple geomodels. The `ObservationPoints` can act as constraints for the model building process. An `ObservationPoint` contains all data collected from one geological observation point. `BSurfModels` and `SolidModels` contain pointers to `ObservationPoints`. That way it is always possible to distinguish and query the observed data points which a model is based on.
Besides the `ObservationPoint` references, the `BSurfModels` and `SolidModels` contain a set of `ModelPoints` which represent supporting nodes. These are generated by applications to build smooth triangulations with a good approximation of input data. Opposite to the `ObservationPoint`, they are not referenced but a member of the models.

*Interoperability.* For many geoscience projects in research and resource exploration it is essential that different software components like databases, geomodelers, and GIS can exchange their data without losses. This can be achieved by the use of ISO-standards conform data models.
The data model presented in section 3.2.2.1 is largely equivalent to the Gocad data model, and the object classes representing cell complexes have corresponding elements in the X3D [47] specification and the GML 3.0 ([22]) specification (see table 3.1). This allows the conversion between these formats and maintaining

the geoobject topology and geometry. For example, the triangulated surface class `TFace` in Gocad corresponds to the GML element `Face` and the X3D element `IndexedFaceSet`. A major advantage of the Gocad spatial data model compared to the GML, X3D and GeoToolKit [4] data models is the macro topological model. The extended data model has been created in UML notation [44]. The UML models were transferred to valid XML Schemas for 3d geomodels and geological observation data. Derived by extension from GML and therefore compatible are the following XML elements: abstract `Object` type, simple types including ID and IDREF, geometry types including `Solid`, triangulated `Surface`, `Curve`, `MultiPoint`, `AbstractTopology`, meta-data types, and simple property types. The major differences are the representation of sets of 3d geomodels with 3d topology instead of unique geographic features, and the distinction of contained model points and referenced observation points.

| | Gocad v.2 | GML v.3 | X3D v.1 |
|---|---|---|---|
| | | | |
| 0d - point | Atom | Node | PointSet |
| 1d - line | ILine | Edge | IndexedLineSet |
| 2d - surface | TFace | Face | IndexedFaceSet |
| 3d - volume | TSolid | - | - |
| 3d - macro volume | Model3d | - | - |

*Tab. 3.1:* Table of equivalent classes of topological model objects of the Gocad, X3D, and GML data model.

*Non-spatial elements of the data model.* In order to develop geological query functionality, geoobjects comprise besides spatial data also non-spatial properties which can be defined for the entire geoobject using functions. These primarily include the geoobject definition criteria and describe the semantic of the geoobject represented by BRep-model elements. For example, a geoobject representing a stratigraphic bed can have constant properties `Age`, `Genesis`, `Lithology`; a fault may have constant properties `Type`, `Orientation` of slip and fault plane, and `TimeInterval` of activity. Also spatially non-constant functional properties can be defined for geoobjects. For example, for a model volume region a property stress $S$ can be defined and a function $S \rightarrow$ f(depth) can be assigned to it. Thus, a stress value exists for every point in the model volume region.

Properties can also be defined in a discrete way. In the Gocad geomodeling software, the vertices of tessellated geoobjects contain a vector with numerical properties including the location vector. For each property, a set of constraints can be defined. In the data model provided here, these vertices correspond to `ModelPoints` and `ObservationPoints` in figure 3.8. While `ModelPoints` can carry interpolated numerical scalar and vectorial values, the referenced `ObservationPoints` may possess comprehensive geological information.

#### 3.2.2.2 *ObservationPoint data model*

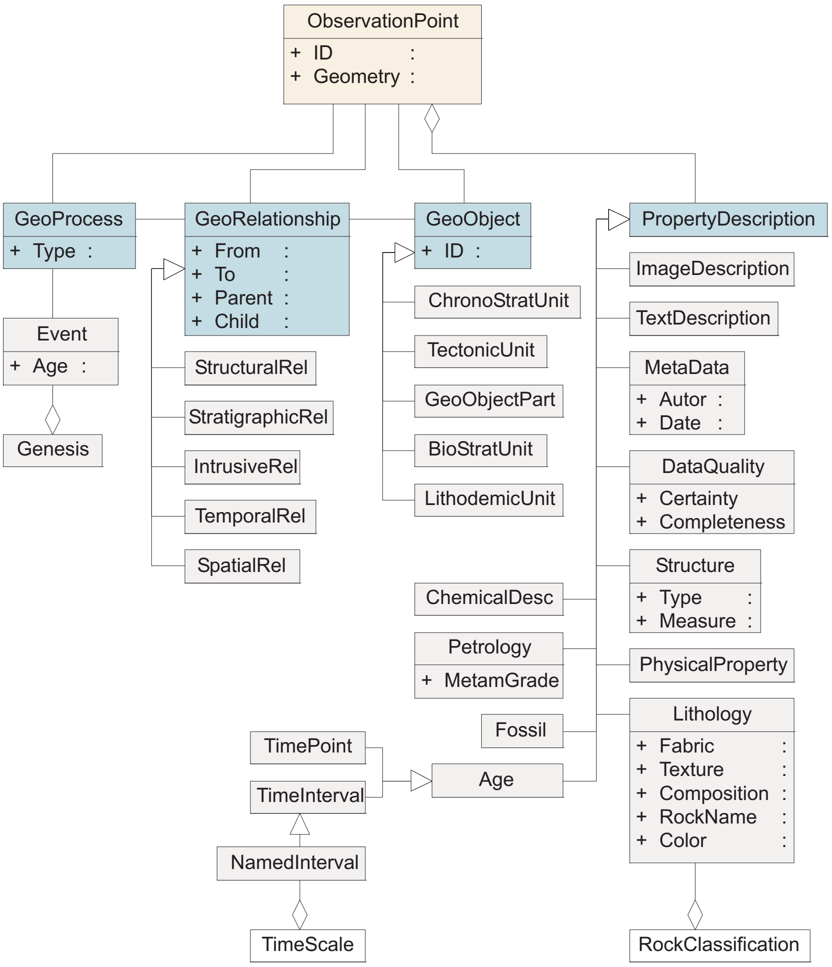


Fig. 3.9: Simplified UML representation of the ObservationPoint data model. The top-level elements are shown in blue. Only top-level associations and attributes are depicted.

The heterogeneous, semi-structured descriptive and quantitative observational data associated with one sampling location are stored as `ObservationPoint` object. Drillhole samples can be modeled as sets of `ObservationPoints` located along a line.
For this work, the NADM 43a [40] conceptual geological data model is adapted as the basis for the observation data model. The NADM 43a is the currently most complete, standardized conceptual data model for geological data. The conceptual data model has been transferred to a logical data model in UML and XML Schema language. Figure 3.9 shows an UML representation of the `ObservationPoint` data model. All properties of complex data types are encapsulated in object classes. An `ObservationPoint` has a geometry attribute and is associated with the following classes:

1. `Properties` cover any type of description and measurements related to the observation point.
   `ImageDescription`, `TextDescription`, and `MetaData` allow to describe general properties of an outcrop like weathering, situation, and metadata including author and sampling date. The `Age` property includes age determination method and error. Two specializations of `Age` exist: `TimePoint` and `TimeInterval`. `TimeInterval` which can be of type `NamedInterval`.
   `Lithology` describes the material properties of a sample and is in association with other properties including `Age`, `PhysicalProperty`, `Structure`, `Petrology`, `MetamorphicGrade`, and `Fossil`.
   `Structures` may be of type contact, fracture, fault, fold, bedding, lineation, foliation, and may have an orientation vector property. `Structures` can be associated with stress and strain physical properties, and may be aggregated in a `CompoundStructure`.

2. `GeoProcesses` include descriptions of diagenesis, alteration, intrusion, volcanism, deformation, crystallization, deposition, erosion, and metasomatism. `Events` are temporally limited occurrences of `GeoProcesses`. The summary of `Events` describes the `Genesis` of an `ObservationPoint` or `GeoObject`.

3. For an `ObservationPoint`, the `GeoObject` class describes its membership of an identifiable volumetric part named ”unit” of the Earth based on properties. Commonly composition, geological age, structures and tectonics, or physical properties are used to define geoobjects.

4. Relationships between geoobjects. A geoobject can have a role property which defines its relationships relative to neighbour geoobjects, for example *”this geoobject was thrust over geoobject B, C”*, *”this geoobject of formation α is stratigraphically included in geoobject B of group β”*. Binary relationships can be generically described by a ”From-To” property or a ”Parent-Child” property. That way also topological relationships can be stored together with their geological semantics.

The `ObservationPoint` UML model has been converted to valid XML Schema documents. The elements can be extended, restricted or substituted to fulfill the requirements of special user groups and tasks. The provided model can almost be seen as a metamodel because several parts can be customized for different purposes.

*Representing geological classification hierarchies.* Geologists commonly group rocks according to their compositional (called ”rock type”) and temporal (called ”geological time scale”) properties into a set of classes. These classes form a hierarchy of varying depth. This classification is commonly hard-coded in database tables for specific needs. However, we need a generic approach to create a hierarchical classification in our data model. This can be achieved by relating each item with its parent using two tables, and use an abstract item as root (see example table 3.2). Thereby no restriction of the hierarchical depth is given, and it can be queried using set-theoretic relational languages like SQL as well as by XQuery.
Instances of the `ObservationPoint` and `GeoObject` properties `NamedAge` and `RockType` can be validated according to `TimeScale` and `RockClassification` XML Schema, respectively. This method provides also the possibility of geoobject generalization according to `Age` and `RockType` properties.

| *UnitID* | *UnitName* | *Rank* |
|---|---|---|
| 0 | root | root |
| 1 | group a | group |
| 2 | formation a | formation |
| 3 | bed a | bed |
| 4 | member a | member |
| 5 | supergroup b | supergroup |
| 6 | group b | group |
| 7 | formation b | formation |

| *DescendantID* | *ParentID* |
|---|---|
| 1 | 0 |
| 2 | 1 |
| 3 | 1 |
| 3 | 2 |
| 4 | 1 |
| 4 | 2 |
| 4 | 3 |
| 5 | 0 |
| 6 | 5 |
| 7 | 5 |
| 7 | 6 |

*Tab. 3.2:* Example conceptual geological unit table and corresponding unit tree table (right).

# 4. SPATIAL AND NON-SPATIAL QUERIES ON GEOOBJECTS

An essential functionality of GIS is the capability to select and investigate geodata by the means of query. While the aspects of relational database queries, 2d spatial and geological queries have been extensively studied in the context of 2d GIS (see, for example, [7], [33]), very few workers consider 3d spatial queries, namely M. Breunig [9]. In addition to examining spatial queries in a topological 3d model, this chapter will emphasize queries which are of particular interest to geology including 3d directional queries, and queries with combined geological and spatial parameters. According to the query parameters, query functions on geoobjects can be grouped into four classes:

- topological queries
- geometrical queries, including buffer queries and direction-based queries
- queries based on geological properties of geoobjects, including semantical queries
- queries based on discrete numerical properties

In the following sections a theory for spatial geological query functionality is developed. In order to prove the concepts, an implementation has been realized as plug-in software library for Gocad and as a query language for XML data management.

## *4.1 Classification and formalization of spatial query functions*

In order to develop generic spatial GIS functions, a query language containing all elements to formulate a spatial query in euclidean space $\mathbb{R}^3$ as mathematical set-theoretic expression is required. The concept of generic database queries [37] can be adopted to examine spatial queries on discrete 3d geomodel objects. The example figure 4.1 represents a set of geological observation points $P$ with property *rocktype*$\{$*granite, limestone*$\}$ and one spatial geoobject region $R \subset \mathbb{R}^3$. The following queries illustrate the different groups of spatial properties that can be distinguished by a query:

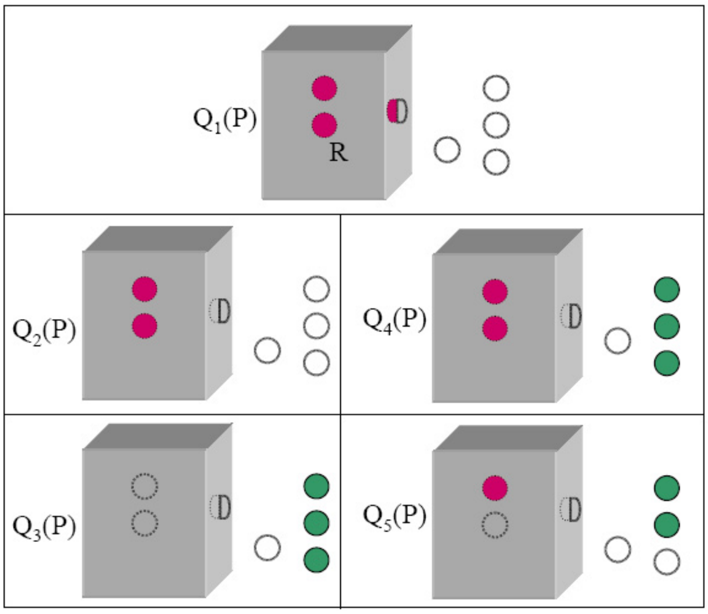


Fig. 4.1: Querying the spatial relationships of observation points. A granite body is symbolized in grey, the surrounding region is limestone. The query results are shown in color: granite observation points are red, limestone points are green.

1. Select the set of points $Q_1(P) \subset P$ with rock type *"granite"*. This query is independent of the spatial location of the points. This type of query is a pure set-theoretic, non-spatial database query, and formal languages like SQL (for relational databases) and XQuery (for XML databases) exist to express them.

2. Select the set of points $Q_2(P) \subset P$ in the topological interior R°. This query is independent of any homeomorphism and it is thus *topology*-invariant. Such queries can only be answered using a standard DBMS if topological relationships are explicitly defined in the data model. This would be the case with queries on spatial regions or faces in a BRep or GMap Model.

3. Select the set of points $Q_3(P) = \{p_0, ..., p_n | n > 1\} \subset P$ that are on a straight line. This group of *similarity*-invariant queries preserves the angles and

cannot be answered by a standard query language acting on a DBMS, but could be handled by the application software or an extended query language.

4. Select set of pairs of points $Q_4(P) \subset P \times P$ that are 11m distant from each other. This group of isometry-invariant queries cannot be answered by a standard query language acting on a DBMS, but could be handled by the application software or an extended query language.

5. Select set ordered pairs of points $Q_5(P) \subset P \times P$, the first is located north of the second, illustrates the *translation*-invariant queries. This group of queries cannot be answered by a standard query language acting on a DBMS, but could be handled by the application software or an extended query language.

For geological purposes the groups of spatial queries considering topology, similiarity, and isometry are of particular interest. Queries can also be grouped by their return type. It is possible to either

- test the relationships between geoobjects and return a predicate or boolean value, or
- operate on geoobjects and return new or altered geoobjects, or
- determine numerical properties of geoobjects.

Table 4.1 shows a classification of spatial queries on geoobjects. Here, topological, metrical and directional queries are distinguished. Because directional data and directional queries are very commonly used in geoscience like structural geology, these are investigated separately from metric queries.

*Formalization of spatial queries.* Algebraically, a point set theoretical query Q on a geomodel can be expressed as

$$Q(p) = \{p|\Lambda(p)\}, where \tag{4.1}$$

- $p$ represents a point variable of a geomodel $M$ embedded in $\mathbb{R}^3$, and $Q(p)$ represents the query result point set with $Q \subset M$. Interpreted for a topological BRep data model, $p$ can be a vertex, or an interpolated point of a boundary surface (e.g. on a triangle or edge), or a point in the topological interior of a volume region.

| | *Topological queries* | *Metrical queries* | *Directional queries* |
|---|---|---|---|
| *Relationships* returning boolean values or geoobjects | disjoint, equal, intersect, covers, covered, inside, neighbourhood (disjoint with common border), connected (by other objects) | in spatial buffer | in direction buffer, relative situation |
| *Operators* returning new or altered geoobjects | union, difference, intersection, complement | translate, reinterpolate, add cell, delete cell,... | rotate |
| *Properties* returning real values | number of intersections, number of neighbours, number of borders, genus, | unary: length, surface, volume, curvature;binary: distance, displacement | binary: angle |

*Tab. 4.1:* Table showing a classification of spatial queries on geoobjects.

- $\Lambda(p)$ represents a combination of terms with quantifiers on $p$ ($\exists$ and $\forall$), functions, boolean operators ($\wedge$, $\vee$ and $\neg$), and predicates.
- a term contains a relation ($<$,$\leq$,=,$\geq$,$>$,$\neq$), or functions.

For example, *"Select the set of points* P *within a buffer of 10m of the fault surface* F*"* can be written as:

$$Q(p) = \{p | \exists f \in F \wedge d(p, f) < 10m\}$$

A geomodel $M$ can define a point set topology on the subsurface space, and a geoobject corresponds to the closure $\overline{R}$ of a point set $R \subset M$. For answering queries which are focused on geoobjects instead of abstract point sets, equation 4.1 can be written as $Q(\overline{R}) = \{\overline{R} | \Lambda(\overline{R})\}$.

*"Select the geoobjects $\overline{R}$ of a geomodel M with a volume v>100m$^3$"* can now be be formulated as:

$$Q(\overline{R}) = \{\overline{R} | v(\overline{R}) > 100m^3\}$$

It is possible to combine multiple spatial and also non-spatial query terms logically using set theoretic operators: *"Select the geoobjects $\overline{R}$ which are at least partly within a buffer of 30m of fault surface S AND have a volume v>100m$^3$"* can be formulated as:

$$Q(\overline{R}) = \{\overline{R} | v(\overline{R}) > 100m^3 \wedge \exists a \in \overline{R} \wedge \exists f \in F \wedge d(a,f) < 30m\} \tag{4.2}$$

## 4.2 Topological queries

### 4.2.1 Theory of topological queries

Combinatorial topological structures such as BRep-models and GMaps are being used to build discrete topological geomodels. The set of geoobjects comprising a geomodel can be abstracted as topological point sets and examined using *point set theoretic queries*. Topological queries are set operations based on topological relations of these sets such as intersection, difference, union, and topological properties such as connectivity and homeomorphism. Three types of topological queries can be distinguished:

- Object queries: "Which geoobjects have a certain relationship with a given object?"
- Relationship queries: "Which topological relation exists between given geoobjects?"
- Topological property queries: "Select all geoobjects with given topological properties."

Topological queries on a set of geoobjects can be solved by examining the topological relationships of the set of binary combinations of geoobjects. As shown by Egenhofer and Franzosa [16], point set topological relations based on the notions of interior $R^\circ$ and boundary $\delta R$ can be equated with general set operators. Binary point set topological relationships can be classified by the 3x3 intersection matrix developed by Egenhofer [16, 17]. This formal and complete model is based on the overlapping properties of the interior ($R_1^\circ, R_2^\circ$), the complement ($R_1^-$, $R_2^-$), and the boundary ($\delta R_1, \delta R_2$) of two topological regions $\{R_1, R_2\}$.

$$Rel(R_1,R_2) = \begin{pmatrix} \delta R_1 \cap \delta R_2 & \delta R_1 \cap R_2^\circ & \delta R_1 \cap R_2^- \\ R_1^\circ \cap \delta R_2 & R_1^\circ \cap R_2^\circ & R_1^\circ \cap R_2^- \\ R_1^- \cap \delta R_2 & R_1^- \cap R_2^\circ & R_1^- \cap R_2^- \end{pmatrix} \tag{4.3}$$

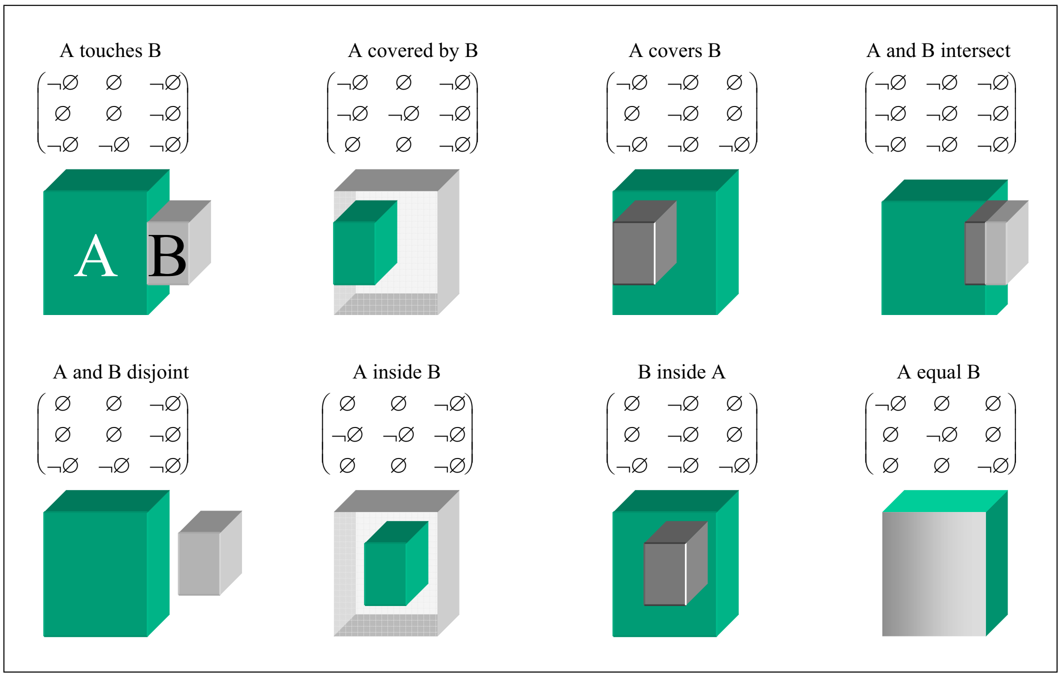


Fig. 4.2: Topological relationships between two 3d topological regions according to the 9-intersection model.

From the matrix $Rel(R_1, R_2)$ 8 realizable topological relationships can be deduced (figure 4.2). For example, *”Select the set of topological regions* R *of a model M which are connected by (i.e. have a common border with) fault F”* can be formulated as point set topological query:

$$Q(R) = \{\overline{R} | \overline{F} \cap \overline{R} \neq \emptyset \wedge \overline{F} \neq \overline{R}\}$$

This model is applicable for binary relationships in $\mathbb{R}^3$. Relationships between three or more geoobjects can be described by examining the combinations of binary relationships. However, the 9-intersection matrix model does not capture the multiplicity of relationships. For example, a query *”How often intersects the straight well curve C the overturned stratigraphic horizon H?”* cannot be answered using this model. Such detailed relationships can be answered by examining the binary relationships of the cells of discretized geoobjects.

The point set topological notions of the interior and boundary, and binary relationships like intersection can be applied to simplicial complexes representing the triangulated faces of a BRep model, or cell complexes. Breunig [9] developed an algebra for topological relationships between 0d-3d simplicial complexes, which can be generalized for cell complexes. Let $R_1$ and $R_2$ be two discrete geoobjects representing two topological regions, it is possible to define the set of possible binary topological relationships if the following sufficient conditions are true:

1. If all cells of $R_1$ are **disjoint** with all cells of $R_2 \Rightarrow R_1 \cap R_2 = \emptyset$
2. If all cells of $R_1$ and $R_2$ are by pairs **equal** $\Rightarrow R_1 \equiv R_2$
3. If a boundary cell of $R_1$ **touches** a boundary cell of $R_2$ from the outside and if all other cells of $R_1$ are disjoint to all other cells of $R_2$ (**neighbourhood**) $\overline{R_1} \cap \overline{R_2} \neq \emptyset$
4. If a cell of $R_1$ **intersects** a cell of $R_2 \Rightarrow R_1 \cap R_2 \neq \emptyset$
5. If a boundary cell of $R_2$ **touches** a boundary cell of $R_1$ from the inside and if all other cells of $R_2$ are inside $R_1$ ($R_1$ **covers** $R_2$) $\Rightarrow R_1 \supset \overline{R_2}$
6. If a boundary cell of $R_2$ **touches** a boundary cell of $R_1$ from the outside and if all other cells of $R_1$ are inside $R_2$ ($R_1$ **covered by** $R_2$) $\Rightarrow \overline{R_1} \subset R_2$
7. If all cells of $R_1$ are **inside** the boundary cells of $R_2 \Rightarrow R_1 \in R_2$
8. If all cells of $R_1$ are **outside** the boundary cells of $R_2 \Rightarrow R_2 \in R_1$

Since this can give different answers for the simplices of an object (e.g. some overlap and some are contained) a superior topological relationships has to be determined: overlap $>$ covers, covered-By, meet $>$ contain, inside [9].

The return value of a topological query can be either a selected set of geoobjects or the type of the topological relationship of a given set of geoobjects. The result set of a relationship query can be deduced from an object query, for example the query *”Is region $R_1$ a neighbour of region $R_2$?”* is true if $R_1$ is contained in the result set of query *”Select the neighbour regions of $R_2$”*.

*Topological properties* are spatial properties which are invariant under topological transformations, like the genus. The genus can informally be regarded as the number of holes in a surface. The related Euler-Poincaré characteristic allows to check the topological validity of a model with manifold topology. It states that

$$V - E + FL - IFL = 2(S - G),$$

where V... number of vertices, E... number of edges, FL... number of face loops, IFL... number of inner face loops, S... number of shells, G... Genus. However, this formula is not valid in non-two manifold topologies encountered in a radial edge BRep. Besides the invariants, the combinatorial topological properties like number of boundary loops or faces of a volume region may be used to characterize geomodels.

### 4.2.2 A concept of topological queries in 3d

#### 4.2.2.1 Queries based on the tessellation

In general, queries on the tessellation of a geomodel, also named ”micro topology”, are not directly required for geological purposes. However, as stated in the last section, topological queries concerning the implicit topological relationships between geoobjects can be answered using the tessellation and the geometry. For example, in order to investigate the relationships between a set of surfaces $S$, the intersection of the surfaces has to be computed. This can be achieved by testing for each combination of the surfaces their sets of triangles for intersection. This gives a boolean result for each pair of surfaces.

In order to avoid unnecessary computations, the intersection is computed in three steps. First the oriented minimum bounding boxes of the surfaces are tested for intersection using the method of separating axes [23, 15]. If the bounding boxes intersect, the search can be refined by computing the octrees of the tessellated surfaces (depicted in figure 4.3) and finally testing the intersection of closely located triangles. The Gocad software provides an implementation of an octree-enhanced intersection test.

For queries whose return value are altered spatial objects, the intersection lines and the re-triangulation of the tessellation have to be computed. Examples for such queries are the join, difference or intersection of a set of triangulated surfaces or volumes. The problem of surface and volume cutting and re-triangulation has been studied by previous workers (for example Euler [19]). This functionality is available in the Gocad software.

#### 4.2.2.2 Queries using explicit macro topology

If a geomodel maintains explicit adjacency relationships between topological entities, then queries on the topology can be performed using the combinatorial database without geometrical computations. The query algorithms have a complexity O(n) which is linear proportional to the number of topological elements involved. Thus, even with very large models rapid query responses are possible. Explicit ad-

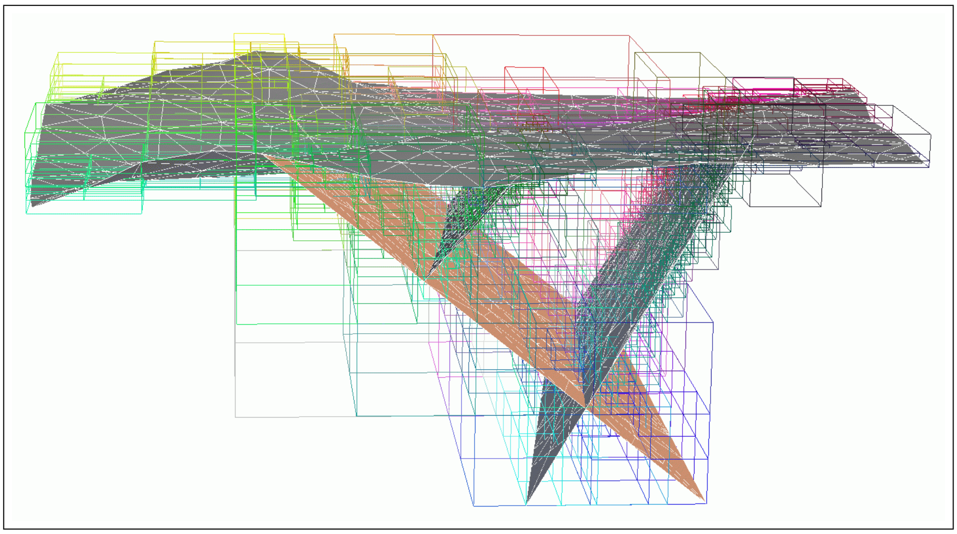


Fig. 4.3: A Gocad BRep model where two fault surfaces are intersecting, two are disjoint, two are touching, and all three are touching the terrain surface. The relationship computation between the triangles is accelerated by the octrees.

jacency relationships can be defined for all topological elements using topological spatial data models like BRep or GMap. According to the query use cases (section 2.4), queries on topological geoobject relationships are of particularly high interest in a GIS. These are aimed at sub-selecting, checking, and analysing the set of geoobjects which constitute the geomodel.

*BRep model.* Queries on topological relationships can efficiently be answered using a topologically manifold triangulated radial edge BRep geomodel. Both the incidence and adjacency relationships of the topological elements can be used to answer queries for connectivity, decomposition, neighbourhood, and orientation. In theory, thirty six element adjacency relationships are possible in a non-two manifold BRep with respect to six basic topological elements (vertex, edge, loop, face, shell and region). For geological query purposes, only the relationships between macro elements with a geological meaning, like region, shell and face, are required. Based on the following three BRep-model topological elements, query functionality has been developed in this thesis and implemented in a Gocad plug-in:

- Loops (also named BFrameElements in Gocad) in a BRep model usually represent the touching line of three or more face-bounded volume geob-

jects called regions. Loops are associated by pointers with topologically meeting faces. This allows to answer queries for the faces meeting at a loop efficiently by iterating through the pointer list.
Topological properties of loops may also be of interest. For example, if a region is bound by one face with $G$ closed loops, it can be deduced that $G$ touching relationships with at most $G$ neighbouring regions occur. Here, $G$ corresponds to the genus of the face.

- The set of faceuses can be queried for their loops, mates, and closedness. Faces without loops constitute closed volumes with one neighbour region. The neighbour regions can be determined from faceuse mates.

- The set of regions can be queried for their boundary faces.

A *connectivity* query like *"Are two faults A, B connected by other faults?"* can easily be answered in a BRep model provided by the Gocad data model, because incidence and adjacency information is stored in double-linked lists which can be searched by iteration. A query function for connectivity which takes two regions or faces as argument and returns a boolean value has been implemented in this work.
In a BRep model, the incidence and adjacency relationships between the topological elements are representable by a hierarchical graph. This could be used for visualization and inspection of the topology of a model.

#### 4.2.2.3 Queries using implicit macro topology

Point set-theoretic spatial queries are applicable for explicit or implicit topological relationships. Implicit topological relationships are defined by the geometrical model but not by the topological model. Thus, they have to be computed from the geometrical model to answer queries. For example, if two surface objects intersect each other, computational geometry algorithms can determine the intersection and thus the topological relationship between these surfaces. Such queries have a higher computational complexity than queries using geomodels with a topological database. In cases where the geometrical model has gaps or other errors which are inconsistent with the explicit topological model, the comparison of the topology computed from the geometry and the explicit topology can provide a means to detect such errors.
General point set topological queries can be translated to combinatorial topology queries acting on discrete geomodel objects as shown in table 4.2.

A method to detect topological relationships from the geometry are intersection tests for a set of topological elements. These can be of a different geometrical dimension. For example, the number of intersection points of a line and a surface can be determined. In general, the result can be either a selected set of intersecting/non-intersecting objects, or a boolean answer if the given surfaces intersect each other, or how many intersections occur. The answers generated by intersection tests provide the basis for further investigations on topological properties of multiple objects, like connectivity.

| *point set topological element* | *Weiler model element* |
|---|---|
| boundary | shell (in Gocad: BFrame) |
| interior | region |
| closure | region + shell |

*Tab. 4.2:* Translation between point set topological elements and Weiler model elements.

*Intersection and union.* The first step of a topological query between geoobjects is to compute their intersection. Starting from the intersection test, it is also possible to compute the difference and union of geoobjects.

Geoobjects to be tested for intersection can be both simplicial complexes like triangulated surfaces without macro topology, and a set of geomodels with internal macro topology. For example, such a request can be *”Compute the intersection of region A, B, where A, B belong to different geomodels.*”. For large sets of geoobjects it is worthwhile to compute the intersection of bounding boxes first. The following algorithm iterates through adjacent topological elements in their incidence order. When the bounding boxes of two faces overlap, the intersection can be computed from the intersection of the triangles of the faces:

```
if bounding boxes of region A, B overlap
  for each face in region A
    if bounding box of face A overlaps bound box of region B
      for each face in region B
        if bounding boxes of face A and face B overlap
          compute intersection of face A, face B
```

For different geomodels, the spatial operators have to respect and maintain the geological semantics of the geoobjects. This will be shown by the following example. The union of different geomodels is of interest if within an area several models exist. Geological surveys often create models with low resolution at regional 3d mapping scale, and versioned detailed models at a local, project oriented scale. Their intention is to unify these models in order to show the regional geological setting of the detailed models. For this purpose, the geoobjects to be unified have to represent the same geological semantic, like parts of an identical fault or horizon. The following method to unify geomodels has been implemented in a Gocad wizard:

1. input: two at least partly intersecting BRep geomodels: regional model *R* and detailed model *D* with bounding boxes
2. cutting the model *R* with model *D*
3. deleting all objects which belong to *R* and are located within the bounding box *D*
4. binary relationships between semantically identical geoobjects of models *R* and *D* are created. The geoobjects are merged by adding the parts derived from model *D* to the parts derived from model *R*. Alternatively, a new union geoobject may be created.
5. post-processing and re-interpolation is required in order to remove gaps between the two models. This can be done by creating surface patches or constraining unwanted internal borders at the former bounding box cutting surface, and re-interpolating the surfaces.

This results in a smooth union of surfaces from two models. However, only identical geoobjects can be merged. If both models are generalized to the same level and contain the same set of geoobjects, this will result in a consistent model. Otherwise, if for example the detailed model is less generalized and thus contains more geological boundaries, these abundant surfaces are not contained in the resulting topological BRep model.

#### *4.2.2.4 Topological queries using volume models*

*Implicit topology - grids.* In 2d GIS, operations on 2d spatial neighbourhoods called filters are commonly used to, for example, smooth data, enhance edges or directional features such as lineaments, characterize textures, or derive shape information such as slope [7]. Spatial filter techniques can also be used in 3d grids. Regular parallelepipedic 3d grids like the Gocad `Voxet` objects have an implicitly defined neighbourhood relationship between the sets of cells. All grid nodes except nodes at the boundary have six neighbour grid nodes. Likewise, a cell shares its boundary with six neighbours. This relationship is being used by many image processing tools. For example, 3d seismic image processing algorithms can detect patterns based on the neighbourhood relationship of cells with similar impedance [27, 3]. Due to the many specific usages and the limited time-frame of this thesis, 3d image processing tools have not been examined.

*Explicit topology - tessellations.* Volume tessellation geomodels like the Gocad `SolidModel` can contain an explicit macro-topology represented by the recently developed `SolidFrame`, and micro-topology comprised by the tetrahedric tessellation. The `SolidFrame` is a radial edge BRep-model created from a 2d-cellular partition of the boundaries of 3d volume regions. That implies that it is easy to navigate through the topological elements and pose topological queries. Possible query functionality is thus similar to the BRep-model examined in section 4.2.2.2. In addition, the tetrahedric volumes filling the `SolidFrame` regions possess an explicit topology between their nodes. However, for geological queries these are not useful as the relationships between tessellation nodes which do not represent geological objects has no geological meaning.

## 4.3 Geometrical queries

Geometrical queries are the most used in 2d GIS, and similar functionality can be made available in a 3d GIS. The geometry of any point of a geoobject can be defined by a location vector triplet [x,y,z] in a Euclidean coordinate system. Based on their geometrical properties, either subsets of geoobjects (buffer queries and relative location queries) or real numbers (property queries) can be obtained as result of queries. Geometrical geoobject queries can be classified as follows:

1. Distance buffer queries, for example: *”Select all geoobjects which are situated within a certain distance of a given geoobject.”*, or:
   *”Select all points of a geoobject which are within a given distance of a given geoobject.”*

2. Property queries. These return a geometrical property like length of a border, highest point of a surface, area of a surface, volume of a closed surface or model volume region, or curvature.

3. Orientation queries can, for example, select the set of faults with a given orientation.

4. Relative location queries, for example: *”Select all geoobjects above, below, between, north, east, south, or west of given geoobjects.”*

### *4.3.1 Distance buffer queries*

Given a geometrical object $O$ and a constant distance $d$ as parameter, the set of geoobjects or the points of geoobjects which are within a given distance buffer $b$ can be determined by computing the distance between the vertices of the objects. This buffer is equivalent to the sum of point buffers of the vertices of $O$. Three different cases of Euclidean distance buffer queries have been implemented in Gocad within this work:

1. *”Select the geoobjects which are completely within the buffer.”* Here, all vertices of the queried geoobject have to fall within the buffer.

2. *”Select the geoobjects which are at least partly within the buffer.”* returns all geoobjects, where at least one of the vertices is within the buffer.

3. *”Select all the points of a geoobject which are within the buffer.”* computes a list of vertices which are within the buffer. This query can either return the list of vertices as points set, or visually highlight the parts of the geoobject which are within the buffer as shown in figure 4.4. This is realized by computing a boolean value telling whether a vertex is within the buffer. This value is used for interpolating the color opacity value $\alpha$ for each triangle.

### *4.3.2 Geometrical property queries*

For the characterization of geological structures often geometrical properties are used. From a 3d geometrical geomodel, the following important properties can be obtained:

- *from one geoobject (unary):* length, surface, volume, curvature of one geoobject. These functions are available in Gocad. To characterize the geometry of a geoobject, like a sediment basin, the ratios of the lengths of axes of the minimum oriented bounding boxes can be used. For example, a query ”Select all stratigraphic geoobjects where the ratios of longest axis

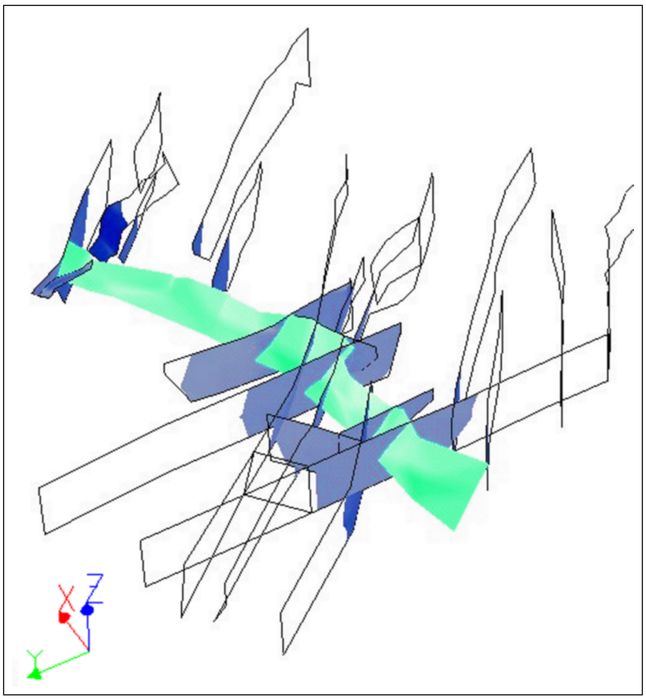


Fig. 4.4: Highlighting the fault parts located within a distance buffer of another fault. The faults are part of the Erzgebirge model [1].

to the other two axes are larger than 5." will select the set of horizons with highly cuspate geometry.

- *from two geoobjects (binary):* minimum and maximum distance between two geoobjects; displacement of the two fault intersection lines of a faulted geoobject. These properties can be computed using standard vector algebra operations.

### *4.3.3 Orientation queries*

Orientation queries form a subgroup of geometrical queries which are similiarity invariant. These can be subdivided in two groups:

- Buffer queries, for example: *"Select all geoobjects whose orientation is within a given directional tolerance."*
- Angle and orientation queries which return an orientation or angle. An example query is: *"Compute the angle between the mean normals of two intersecting surfaces"*

Orientation buffer queries can be useful to select objects with a relatively large geometric extension in one dimension for lines (for example the intersection line of a fault surface with a topographic surface), or in two dimensions for surfaces. In geology, such linear and planar geometries are commonly created by sedimentation as bedding planes, or during rigid deformation as fault planes. Given an orientation $D$ (for example one direction vector $v$, or an azimuth angle $\varphi$ and a dip angle $\delta$) and a tolerance angle $\alpha$, a useful direction query is: *"Select all geoobjects whose mean extension lies within the directional tolerance* $\alpha$ *of* D." This allows to select faults with a certain direction as shown in figure 4.6. To determine the mean extension with spatial dimension $n$ of a given discrete model object $M$, two methods are proposed:

1. For a triangulated surface: Compute the mean of the normals of the set of triangles of a surface. As a mean vector, the sum of triangle plane normals is used (figure 4.5). A measure for the variability of the triangle orientations is helpful, as orientation queries for strongly curved surfaces, like folded faults, may be not appropriate. An option is to use the minimum ratio between the mean normal vector length and the sum vector length as additional query parameter.

2. For lineaments: Compute the linear regression of the coordinate triples of the vertices of a line. For example, a set of observation points at the intersection line of a fault with the topographic surface can be used for this computation.

Having computed the mean extension, one can test whether a line or surface normal falls within the direction interval. This method can also be used to select observation points with orientation measurements by a query with direction parameter. The proposed methods have been implemented in a Gocad plug-in and successfully tested.

Other orientation queries can be derived from these methods. For example, the intersection angles between two fault planes, or a plane and a line, or two lines can be computed from the normal means.

*Other query methods for orientation data* In addition to orientation buffer queries, a tool has been developed for stereographic projection (Schmidt net) of orientation data, which allows the analysis and visualization of structural geology data. This includes pole, great circle, contour or rose plotting of points with

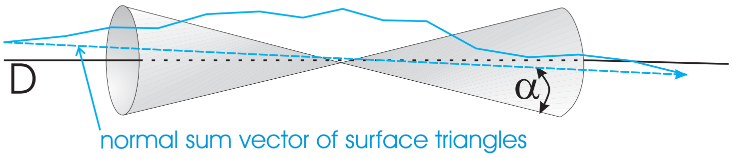


Fig. 4.5: Illustration of the 3d direction query implementation. The function tests if the direction of the sum normal vector of the surface triangles falls within the directional tolerance defined by *D* and α.

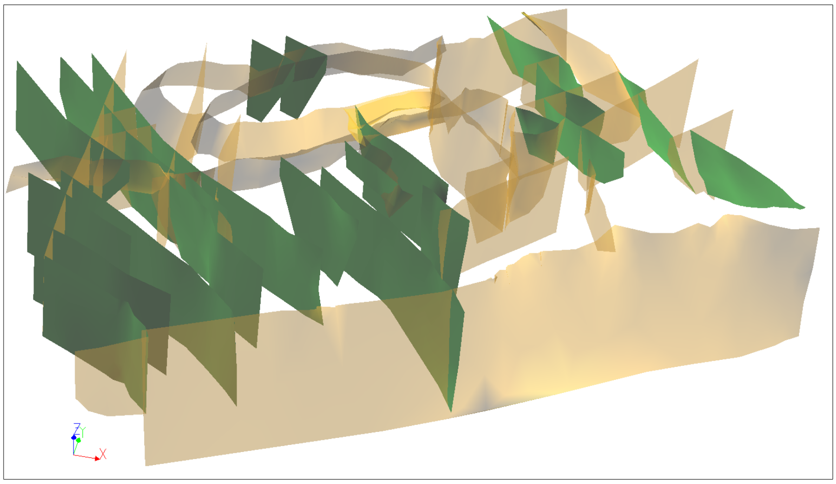


Fig. 4.6: Direction query *"Highlight all faults with a strike orientation 330deg ± 10deg"* applied to a fault network. The faults which are not within the buffer are transparent. The faults are part of the Erzgebirge model [1]

direction property (e.g azimuth and dip, normal vector). Multiple data-sets can be handled, and selected subsets may be highlighted in the plot window and in a Gocad 3D camera. The integration in Gocad allows simultaneous exploration of structural trends and spatial location of the geomodel objects and data points. Another integrated dialog window allows the computation of properties (e.g. plane intersection line). A prototype of a structural analyzer is available as a Gocad plug-in (figure 4.7).

### 4.3.4 Relative location queries

Queries for the relative location result in one of the predicates *above, below, east of, south of, west of, north of* based on the geometry of model objects. Alternatively, it would be possible to give an orientation and angular buffer as query parameter. Two cases can be distinguished:

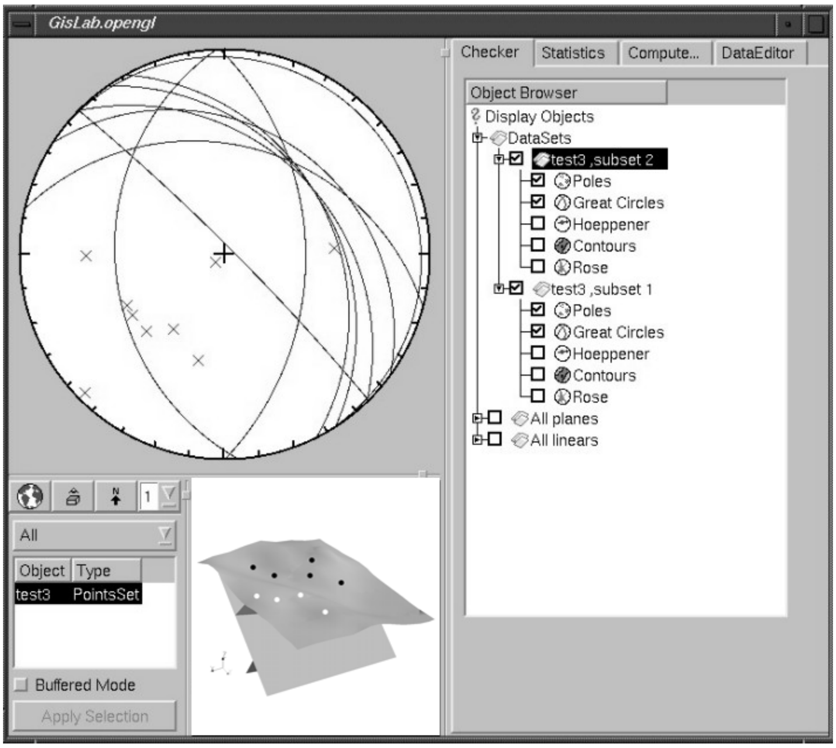


Fig. 4.7: Example of structural analyzer integrated with Gocad.

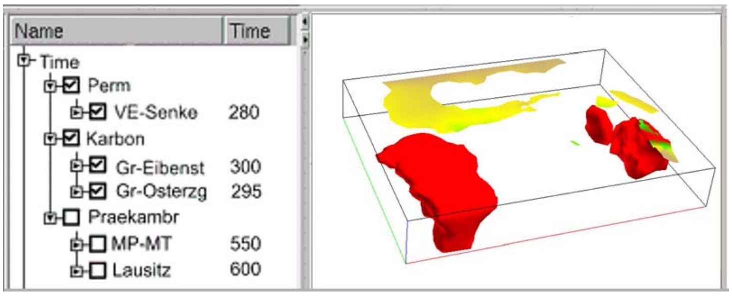
1. *"Select the geoobjects which completely have a relative location P to geoobject A."* That implicates that all vertices of the queried geoobject need to have a relative location *P* with respect to *A*.

2. *"Select the geoobjects which are at least partly have a relative location P with respect to geoobject A."* That implicates that at least one vertex of the queried geoobject needs to have a relative location *P* with respect to *A*.

## 4.4 Queries based on geological and numerical properties

Property queries are based on the non-spatial properties associated with a geoobject. Besides spatial queries, these can provide an important aid to select subsets, reclassify or analyze 3d geomodels. In common 2d GIS like ArcGIS [2], non-spatial properties are stored in a relational database. Here, a SQL-based query language is used for the selection of objects according to their properties, like the geological age or the type of rock, for example.

With the object-oriented topological data model used in this work, properties can



| Name | Time |
|---|---|
| Time | |
| ☑ Perm | |
| ☑ VE-Senke | 280 |
| ☑ Karbon | |
| ☑ Gr-Eibenst | 300 |
| ☑ Gr-Osterzg | 295 |
| ☐ Praekambr | |
| ☐ MP-MT | 550 |
| ☐ Lausitz | 600 |

Fig. 4.8: Simple example of a legend in Gocad. It allows to visually query 3d geoobjects by geological properties.

be declared as variables of topological elements building a geomodel. These can be queried within the Gocad geomodeling software and at the database. Apart from a query language, also graphical methods exist to select geoobjects based on their properties. Such map legends can also be used for a 3d geomodeling application and have been implemented as simple prototype in Gocad, where geoobjects can be selected in a legend list by their age or rock type (figure 4.8).

*Geoobjects.* All non-spatial geological data of a geoobject declared in the data model in section 3.2.2, like stratigraphic age interval, fault type, and meta-data are properties of a whole geoobject. This facilitated the development of Gocad functions to select geoobjects based on these properties. Query terms may be combined using boolean expressions.

*Observation and model points.* The discrete cells of geoobjects can carry properties:

- at the vertices of triangulated surfaces or tetrahedric volume models (in Gocad: `SolidModel`). In the data model provided in section 3.2.2, vertices can be either of `ModelPoint` or `ObservationPoint` type.
- at the *n*-cells of a GMap model. As shown by Levy et al. [30], the GMap approach allows to associate properties to all *n*-dimensional cells of a GMap model. This can provide the basis for property modeling and query methods within a sound combinatorial topological model. This work, however, examines query functionality on BRep models as described in section 3.2.2.

Properties of vertices can be queried by picking them visually, or by query functions. For numerical properties, query functions can be formulated in Gocad with the help of an comprehensive script language. From the query result, a subset of vertices can be defined as a region and be treated separately. However, in Gocad no distinction is made between model points and observation points. Here, extended vertices named ”Atoms” store the geometrical, numerical property, and constraint information. Commonly, input data points are defined as control nodes and thus act as constraints for geometrical and property interpolation. They do not contain geological information.

Geomodeling software users often build geomodels based on input data points. These correspond to `ObservationPoints` presented in section 3.2.2.2. For users it would be advantageous if they could access all available information associated with the input data points directly from the geomodeling software. Two benefits can be delineated:

1. Distinction is made between interpolated `ModelPoints` and `ObservationPoints` as input data. Thereby it is possible to state on which information a geomodel is based on.

2. All information stored with an `ObservationPoint` becomes available during the geomodeling process and during the interpretation.

In order to realize this concept, from the Gocad ”Atom” class a new class ”ObservationPoint” has been derived, which contains a reference to an `ObservationPoint` element in a XML database. Figure 4.9 depicts the class relationships. The absolute database address is defined for the whole geoobject (in Gocad named ”GObj” class). The observation point identifier is contained by each ”ObservationPoint”. Using this lightweight technique, the whole observation point database referenced by a model becomes accessible from within Gocad.
As a database, either XML documents on the file system or a XML database management system can be used. The method of geodata management based on a network-based XML database management system is investigated in detail in chapter 5. Such a system allows to store and serve `ObservationPoint`-documents to geomodeling application clients. Clients can access XML documents using an address. This address is a combination of the database network address and the logical address of the document in the database. Depending on whether the user has sufficient permissions on the database, the observation point may be edited from within the Gocad application and saved in the database.
The following XML document contains an example observation point instance with basic elements:

```
<?xml version="1.0" encoding="UTF-8"?>
<gtx:ObservationPoint>
  <gid>ID12</gid>
  <coord>4589220.5 5639520.2 390.2</coord>
  <GeoObject>
   <ID>OEGneissComplex</ID>
    <LithodemicUnit>Gneiss</LithodemicUnit>
    <ChronoStratUnit>PC</ChronoStratUnit>
  </GeoObject>
  <PropertyDescription/>
  <GeoRelationship/>
  <GeoProcess/>
</ObservationPoint>
```

A visual query method is to select an observation point by picking in the 3d camera, and open the associated XML document with the geological property information. For a nice presentation it is suggested that the XML document becomes processed with a XSL style-sheet resulting in a formatted HTML document viewable by an internet browser.

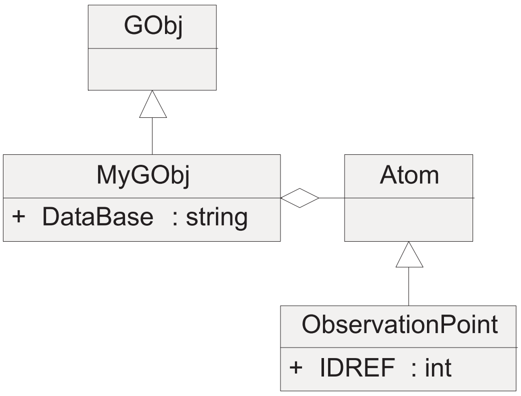


Fig. 4.9: Simplified UML class diagram of the Gocad implementation of class `ObservationPoint`. In Gocad, `GObj` is the base class of modeler objects including surfaces, lines, points sets.

## 4.5 Geological queries using regular grid geomodels

Grid models provide the possibility of fast property queries and computations. Property queries may be supported by geometric and topological parameters. For instance, such queries are applicable in predictive mineral exploration sensitivity studies.

The query *"Select all sets of connected grid cells with a certain electrical resistivity value. The sets should have a certain minimum number of elements."* is a combined topological and property query. In Gocad, non-spatial property queries are possible to formulate using a functional script language. Spatial queries can be computed on the grid cells, and the return value is assigned as a property to the cells. That way spatial queries can be transferred to property queries.

*Geoobjects in grid models.* A different concept for querying geoobjects must be used for grid models than for BRep models. Here, the property values are not assigned to an identifiable geoobject but are defined as a discrete property for a whole geological volume. Such a property may be the assignment of a cell to a certain geoobject. In Gocad it is possible to transfer the membership property from BRep model region to a grid model based on the location of the cells. Using this geoobject membership property, the geometry of geoobjects is defined in the grid model and can be used in property queries. Spatial and non-spatial conditions can be combined using boolean operators to formulate queries. Selections are possible using the property scripting language of Gocad.

Geophysical data can be integrated into a geomodel. It is common practice to derive the geometry of geological structures from 2d or 3d seismic models. Also, it is possible to compute 3d models for the gravity, electric and magnetic potential fields from measurements. Functionality for inverse and forward modeling of geophysical potential fields is available as plug-in for the Gocad geomodeling software [8, 13]. This provides a novel approach for studying the relationship between geological and geophysical models, and thereby validate the models with each other.

## 4.6 Combining a set of geomodels

In 2d GIS, the combination (also named overlay) of several map layers based on the geometrical co-location is a common procedure for the analysis of raster and vector maps. Transferred to a 3d GIS, this means to combine the topological, geometrical, and property information of a set of geomodels. This task seems to be hardly required if a geomodel - per definition - tries to respect all data available related to a set of geoobjects [34]. On the other hand, multiple geomodels from one input data set are commonly created by stochastic simulations, or to test structural geological scenarios using as set of different structural models. However, here each model is built based on the same data set and independently valid according to the modeling criteria. For example, an intersection of two BRep-models representing a structural situation differently is not useful, as the resulting model is likely to be geologically and topologically inconsistent, and the geological semantic of resulting regions is not clearly defined. This arises from the fact that in a geoobject-oriented model the regions are coincident with geoobjects, and the identity of the geoobjects becomes undetermined.

If geomodels with different properties exist or if different topological subdivision should be compared, it is suggested to combine these models using a high-resolution regular grid. That way, grid models and tessellated cellular models can be combined with the advantage that all property information can be examined in one model, and multivariate relationships can be detected. The problem of geoobject identity can be solved by assigning a membership function to the cells. In Gocad, this information can be transferred from a BRep model to the grid model. The disadvantage is that the topological information of a BRep model cannot be transferred to the grid. Due to this fact, a maximum of information can be modeled by maintaining both a grid model and a structural topological geomodel.

## 4.7 Query application - examples

### 4.7.1 Geomodel subset selecting

A common spatial data mining task is to select subsets of the database using conditioned queries. For a geological 3d GIS, topological query conditions are of particular interest because many geological relationships are coincident with topological relationships, and can be discovered by topological queries. For this type of query both the topological and the semantic information of the geoobjects are used to formulate queries on a set of geoobjects:

- ”*Select all model volume regions which are neighbours to region R.*” The result set contains all regions whose topological border at least partially overlaps with the border of region R (figure 4.10).
- ”*Select all model volume regions which are bound only by region R.*” For example, this query can be used to select stratigraphic lenses which are bound by region R.
- ”*Select all fault surfaces which intersect fault surface F.*” The result set contains all faults whose interior partially overlaps with fault F.
- The query ”*Select the neighbouring regions of the selected loops*” can help to discover geological relationships. If, for example, the occurrence of a skarn ore deposit is highly probable at the intersection of an hydrothermal dike and the lower boundary of a limestone horizon, it is possible to pose the inverse query and select the edges where hydrothermal dike boundary faces and lower boundary faces of limestone horizons meet.
- ”*Select all surfaces with free edges.*” In combination with relative geological age data this can be used to detect false fault geometries.
- ”*Select all wells which intersect a certain horizon.*”

### 4.7.2 Spatial and geological consistency checks

*Topological integrity checks.* Point set topological consistency checks can be used to check the integrity of a geomodel. By definition, in a single geomodel each point must belong to the interior of one geoobject. An exception are fuzzy representations, here the cumulative membership function must be equal to 1. The following statements illustrate the usage of this topological separation constraint:

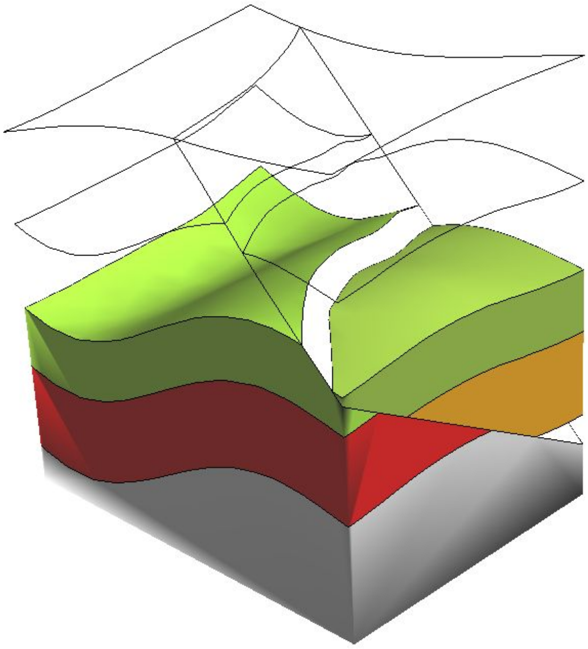


Fig. 4.10: Topological query to a volume model: ”Select all neighbour regions of region $R$ (red color).” Because the topology is explicitly defined, queries for neighbourhood relationships and connectivities can easily be answered.

- If a point belongs to multiple geoobjects then an inconsistency occurs, like crossing of strata. In cellular models including BRep models however, this cannot happen if each spatial region is coincident with one geoobject. In grids, this problem can be avoided by defining geoobjects based on a membership function, which may allow fuzziness.
- There exists only one universe region. Any other point must be contained in a geoobject of the geomodel. Otherwise there are holes in the model. In a BRep model this implies that each region should coincide with a geoobject.

Topological queries also allow to check the validity of a topological model against the geometrical model. In the case of a radial-edge BRep model this implies that two regions may only meet at their shared boundary faces, two faces at their shared loops, and two loops at their shared vertices.

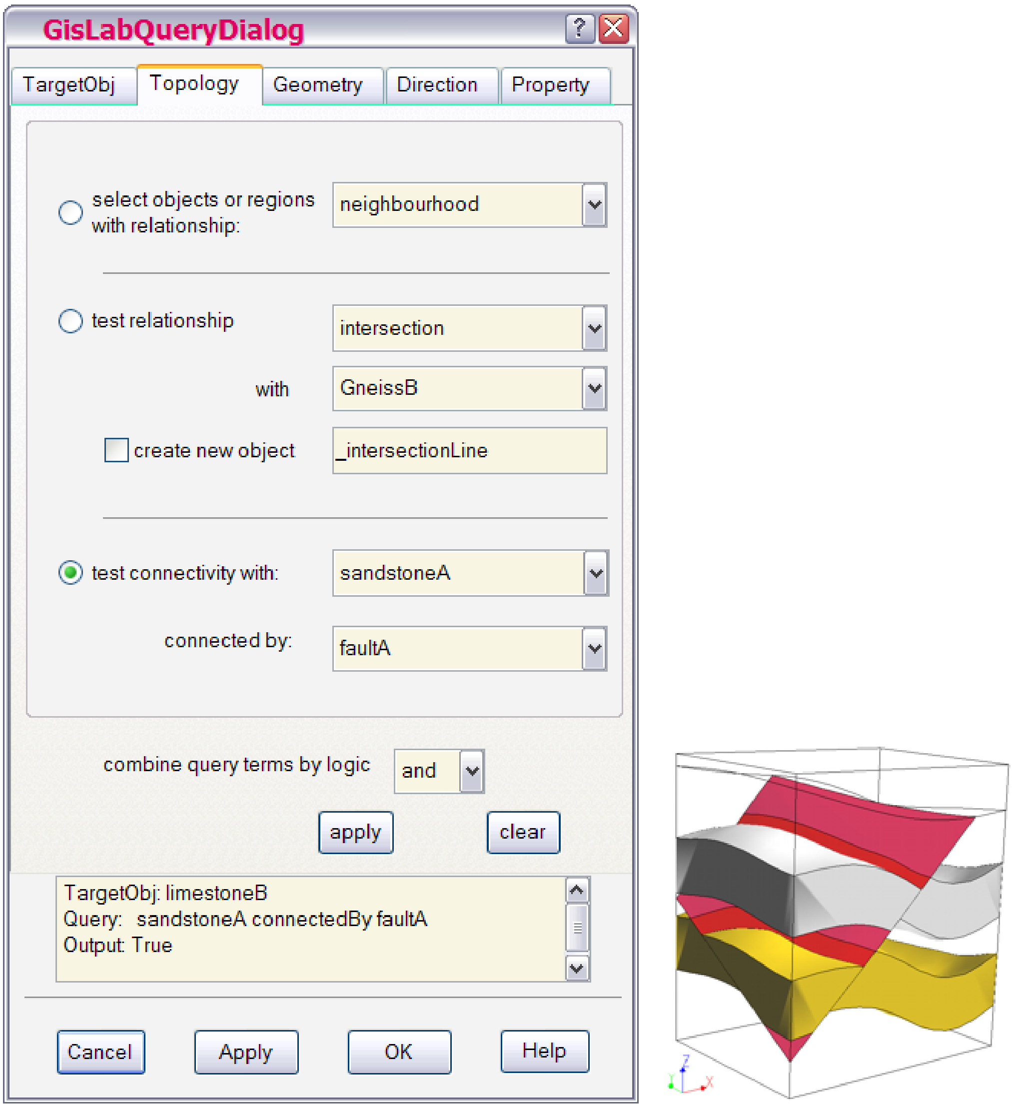
*Spatial-geological consistency checks.* Especially relative geological age and genesis data can be combined with topological data to formulate constraints. Such a constraint is used in Gocad when building BRep model objects from a set of surfaces. Here, geological information is used to correctly name stratigraphic layer regions after their top surface, unless the top boundary is of intrusive origin. The following list contains a list of examples of spatial-geological integrity checks:

1. age and intersection relationships
    - Older faults may not intersect younger stratigraphic horizons.
    - Younger faults may not be juxtaposed by older faults.
2. age, feature type and superposition relationships
    - According to the law of superposition [41], in general older stratigraphic layers may not exist below younger stratigraphic layers (exceptions: normal or thrust fault, overturned fold).

## 4.8 Implementation features

The query functionality presented in the previous sections has been implemented prototypically in a Gocad plug-in. Table 4.3 lists the developed spatial query functions. The plug-in has been programmed platform-independent in C++ and uses the Qt widget library (Trolltech AS, Oslo) and Gocad.

The user dialog shown in figure 4.11 provides an interface for comfortable, interactive formulation of complex queries on Gocad objects. They can be combined with queries for geological information using set theoretic operators. By comparison, the current version of the most widely used 2d GIS, ArcGIS version 8.3, allows spatial and geological attribute queries only successively to be posed for geographical features.



○ select objects or regions with relationship: neighbourhood

○ test relationship intersection

with GneissB

☐ create new object _intersectionLine

● test connectivity with: sandstoneA

connected by: faultA

combine query terms by logic and

```
TargetObj: limestoneB
Query:  sandstoneA connectedBy faultA
Output: True
```

Fig. 4.11: Dialog for query functions in Gocad. First a target geoobject or region is selected, then different queries can be combined and applied. In the example model, a target object *limestoneB* is queried if it is connected with *sandstoneA* by *faultA*. An output window tracks the selected query terms and results.

| Query function | Return Type | Description |
|---|---|---|
| disjoint, equal, neighbourhood, intersect, inside (GeoObject A) | List<GeoObject> | returns a list of GeoObjects with a given topological relationship with the argument GeoObject |
| neighbourhood, inside (Region A) | List<Region> | returns a list of BRep-model regions with a given topological relationship with the argument region |
| connected (Region A, Region B, Boundary C) | boolean | test if two BRep-model regions are connected by a given boundary (fault-)surface |
| intersected (Boundary B) | List<Layer> | returns a list of BRep-model layers which are intersected by a given boundary (fault-)surface |
| buffer (bufferType, distance, GeoObject) | List<GeoObject> | returns the GeoObjects located within the buffer of the argument GeoObject. The first argument allows to choose if the objects may intersect or need to be completely within the buffer. |
| distance (distanceType, GeoObject, GeoObject) | float | returns the distance between two GeoObjects. The first argument allows to choose euclidean minimum or maximum distance |
| relativeSituation(GeoObject, direction) | List<GeoObject> | returns a list of GeoObjects located in the given direction of the argument GeoObject |
| orientationBuffer(direction, angle, curv) | List<GeoObject> | returns a list of GeoObjects with a mean extension in the given direction and buffer angle. ”curv” is a coefficient for filtering rough and curvilinear GeoObjects. |

*Tab. 4.3:* List of spatial query functions developed and implemented as a Gocad plug-in.

# 5. DATA MANAGEMENT FOR 3D GIS - DESIGN AND IMPLEMENTATION

## *5.1 Introduction*

For 3d geomodeling projects a large amount of geodata needs to be stored and served to multiple users during a long period of time. Commonly, geological databases are maintained for decades. Therefore, for 3d GIS projects it is essential to have an efficient and reliable data management.

After a review of existing methods for spatial geodata management, a novel method for 3d geodata storage and query is presented in this chapter. As pointed out in section 2.3, from the current perspective XML-based data management can provide an efficient means for long-term availability of geodata. Two specific requirements must be honoured by the database server:

1. to store observation point data and geomodels as valid XML documents according to the data model defined by XML Schemas, and

2. to provide geological and spatial query functionality from user applications like Gocad against the database.

### *5.1.1 Summary of existing approaches to spatial geodata management*

*2d spatial data in relational and object-relational databases.* Several object-relational database systems offer support for primitive 2d spatial data types. Examples of such systems are Oracle and PostgreSQL [38]. These databases can store geometrical primitives in relational tables. This technique is also used in common geoinformation systems like ArcGIS. Here, the ArcSDE spatial database engine acts as an interface between the application software components and relational database systems. 2d GIS objects have a relatively flat data structure compared to topological 3d geoobjects. For 2d geometrical objects the mapping to relational structures can be efficient.

Güting [24] developed an georelational algebra based on 2d-spatial geometric object types. This approach can enable geometrical and topological query support by the implementation language of the database system [24]. Recent spatial databases, like Oracle 9i Spatial, provide support for 2d spatial queries.

*3d spatial data in object oriented databases.* The concept of a georelational algebra [24] was extended by Breunig [9] to support 3d geometrical objects. This algebra acts on so-called extended simplicial complexes (abbreviated: ”e-com”). It does not consider high-level topological data types like BRep models. However, the storage and query of such models is important for geological modeling. The extension of the e-com algebra by the support of high-level topological models could provide a sufficient algebra for 3d spatial query processing.
During the ”GeoStore” project [10] the storage of 3d geoobjects in an object-oriented database has been examined. Although it allows the persistence of complete geoobjects, this approach is not used in the geomodeling community. This might be due to the fact that users are forced to use an uncommon, proprietary database system.

*File based geomodel storage.* Currently, geomodeling software packages including Gocad solely offer file based storage of geomodels. For example, in Gocad two ways of file based data storage are implemented:

1. import and export of selected objects as ASCII-files
2. import and export of all objects of a Gocad session as binary project file

The two Gocad formats allow to store the geometrical, topological, property and graphical appearance information of all application objects including point sets, polylines, triangulated surfaces, Weiler-models, wells, and grids. For Gocad and comparable geomodeling systems, no efficient geomodel storage approach using a database management system with standard interfaces is available.

### 5.1.2 Spatial data in document centric object oriented data bases - XML

Geodata management based on XML is a solution meeting the requirements stated in chapter 2.3. Functionality for the storage of both data-centric geomodels and document-centric geological descriptions can be implemented using a native XML database server. A native XML database is a database that has a XML document as its fundamental unit of logical storage and defines a logical data model for a XML document. It stores and retrieves documents according to that model.

The logical model can be defined using the XML Schema language. The XML Schema documents can be used to validate the integrity of data. Currently, the ”Tamino” native XML database by Software AG is the only mature product which is in use for large-scale industry projects. Besides Tamino, several XML database development projects exist, for example ”Xindice” by the Apache group (www.apache.org). Key advantages compared to relational database systems are:

- Native XML databases are most suitable for the storage of document-centric XML data. This relates back to their respective data structures: the tree/node hierarchy versus open, tabular data entities. When working with documents, as opposed to pieces of data, queries typically will result in larger amounts of data being requested. XML-aware content indices can provide a faster data retrieval mechanism when a large amount of document data is requested. The parsing of large XML data constructs is faster than the equivalent processing required when retrieving and then assembling this information from a relational data source [18].
- Support of query technologies designed specifically for the XML representation format. This opens the door to sophisticated query statements that would not be possible with many of the XML-enabled relational database platforms [18].
- Native XML databases can be positioned alongside relational repositories as cache and pre-validator [18].

*XML and relational databases.* Native XML databases are designed to accommodate and manage a XML document structure independently from its content. Being able to differentiate the actual data from markup, including processing instructions and entity references, is beyond the ability of typical relational database platforms [18]. For data-centric XML documents, however, relational databases that have been extended with XML support are still the way to go [18]. It should be noted that data-centric XML support is provided by all major database systems, including Oracle 9i, IBM DB2, and Microsoft SQL Server. These systems can use XML as exchange format and map XML documents internally to relational structures for storage. However, Oracle and Microsoft are currently developing a new generation of databases which support XML as a native data type.

*Queries.* For querying a XML database, the comprehensive query language XQuery can be used. XQuery is a functional language where each query is an expression. XQuery expressions fall into six broad types:

1. path expressions: XQuery supports path expressions that are a superset of those currently being proposed for the next version of XPath, a graph-based XML search language. The example query:

   ```
   /gtx:GISTriXP[@ID="project1"]
   /gtx:GeoModel[@ID="model1"]
   //gtx:observationPointRef
   ```

   selects all `observationPointRef` elements which are contained in ”model1” of ”project1”.

2. Element constructors: In some instances, it is necessary for a query to create or generate new elements. Such elements can be embedded directly into a query in an expression called an element constructor.

3. A FLWR expression is a query construct composed of FOR, LET, WHERE, and a RETURN clauses. A FOR clause is an iteration construct that binds a variable to a sequence of values returned by a query (typically a path expression). A LET clause similarly binds variables to values. A WHERE clause contains one or more predicates that are used on the nodes returned by preceding LET or FOR clauses. The RETURN clause generates the output of the FLWR expression.

4. A conditional expression evaluates a test expression and then returns one of two result expressions. If the value of the test expression is true, the value of the first result expression is returned; otherwise, the value of the second result expression is returned.

5. Quantified expressions: XQuery has constructs that are equivalent to quantifiers used in logic. The SOME clause is an existential quantifier used for testing to see if a series of values contains at least one node that satisfies a predicate. The EVERY clause is a universal quantifier used to test to see if all nodes in a series of values satisfy a predicate.

6. Expressions involving user defined functions: Besides providing a core library of functions similar to those in XPath, XQuery also allows user defined functions to be used to extend the core function library.

The most comprehensive XQuery language (www.w3.org/TR/xquery) implementation currently exists for the Tamino database. XQuery has become an ISO and W3C standard in 2003. Using the XML Schema spatial data model defined in this thesis, spatial queries which are based on explicitly defined topology and geometry can be answered with the Tamino XQuery implementation.

- Queries based on the bounding box geometry of the geoobjects include the following: inclusion, exclusion, and intersection. The bounding box is stored as pre-computed element of geoobjects. For example, the set of observation point references *obs* of a `BSurfModel` which are located within a cuboid bounding box can be queried using a FLWR-expression:

```
FOR $obs IN
  /gtx:GISTriXP[@ID="project1"]/gtx:GeoModel[@ID="model1"]
  //gtx:observationPointRef
WHERE $obs/gml:coord/X>23000 AND $obs/gml:coord/X<63000
  AND $obs/gml:coord/Y>37000 AND $obs/gml:coord/Y<223000
  AND $obs/gml:coord/Z>-1000 AND $obs/gml:coord/Z<1000
RETURN $obs;
```

- Queries based on explicitly defined topological relationships of a BRep-geomodel can be computed. The set of neighbour regions *reg* of a `BSurfModel` region *R* can be obtained by the quantified FLWR-expression:

```
FOR $reg IN
  /gtx:GISTriXP[@ID="project1"]/gtx:GeoModel[@ID="model1"]
  //gtx:modelRegion
WHERE SOME $bface IN $reg//gtx:bface SATISFIES
  $reg//$bface = R//$bface
RETURN $reg;
```

However, for requests requiring more demanding computational geometry algorithms a standard database query language like XQuery is not sufficient. Commonly required algorithms, like coordinate transformations and graph-based topological queries, have a linear computational complexity of O(n). For geometrical algorithms in 3d space, as for minimum distance computation between two triangulated surfaces, a computational complexity higher than O(n*log(n)) with a n>>1000 can occur. Such algorithms are not part of general database query languages. The ”X” in the ”XQuery” stands for eXtensible, which suggest the possibility to add new functionality to the language which could process spatial computationally demanding queries.

## 5.2 A new concept of a generic XML/component-based 3d GIS

The combination of all data management, geomodeling and query components constitutes a 3d geoscience information system (figure 5.1). To prove the concept of a XML-based 3d GIS, a prototype system based on the following core components and concepts has been developed and will be discussed in the next sections:

- The data model defined using XML Schema language covers geomodels and observation points as described in section 3.2.2. This data model determines the common interface for the GIS software components. In order to facilitate spatial database queries an extension for the XQuery language has to be defined and implemented.

- Persistent data storage is realized using a XML database server. The native XML database system Tamino provides the currently most comprehensive XML query language implementation based on the XQuery specification [48], and is available free of charge to developers and non-commercial users. Therefore, Tamino has been chosen for this implementation. However, other XQuery-supporting XML database servers can also be used.

- For spatial query processing, a middle-ware component between the database and the user application is required. This can be an application server which computes demanding 3d spatial XQuery requests.

- For communication between the GIS components XML documents are exchanged using the HTTP protocol and the internet protocol (IP).

- XML database input and retrieval is possible from a set of XML-enabled clients, for example the geomodeling application Gocad.
  For geological observation data input, a method which generates HTML forms automatically from XML Schemes has been developed. These forms are able to save the input data as valid XML documents. That way the observation point data can be stored consistently in the XML database.

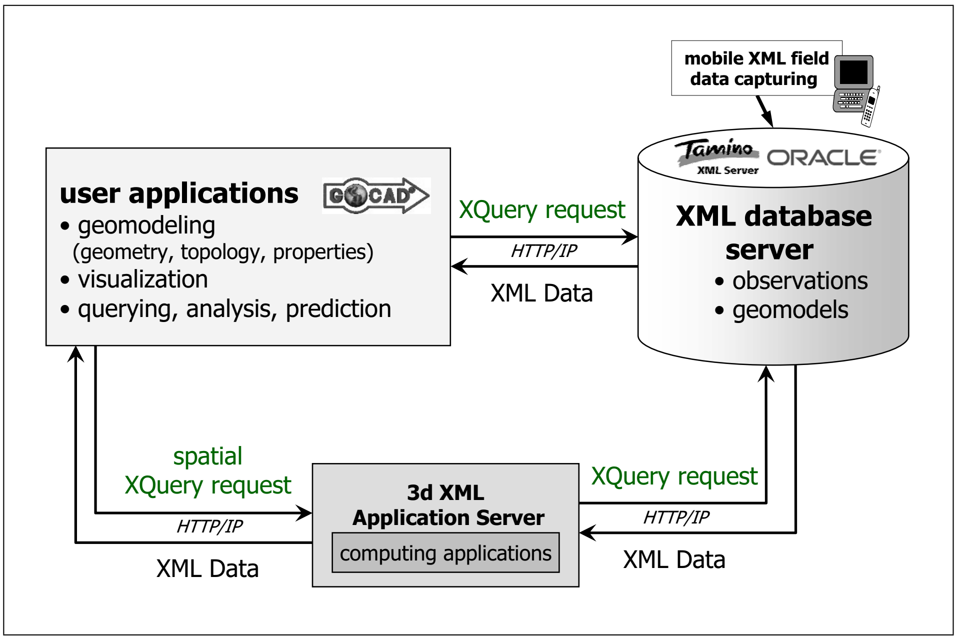


Fig. 5.1: Design of the 3d GIS framework. Queries can be posed from user applications like Gocad. Standard XQuery requests can be sent directly to the database, while spatial XQuery requests are processed by an application server. The application server obtains the data of a project from the database and sends the computed result set to the user application.

### *5.2.1 XQuery extension*

A database containing sets of `BSurfModels`, `SolidModels`, or separate surfaces and lines, often has no topology defined between these geoobjects. In that case, point set-theoretic spatial queries can be formulated and answered using topological relationships computed from the geometry. The binary relationships disjoint, equal, neighbourhood, intersect, outside, inside defined in section 4.2.1 need to be formulated in XQuery language and computed. Also, the altering topological queries union, intersection, and difference, geometrical queries involving distance, volume, surface, buffer, relative situation, orientation, and geometrical transformations may be performed using a spatially extended XQuery language. Table 5.1 shows the spatial XQuery functions which have been defined in an extension XML schema document.

The spatial query statements are included in XQuery requests. The return value of XQuery requests is always a valid XML document which can be served to an application software like Gocad. The following example returns the list of geoobjects of a project database which are intersected by geoobject *"faultSurface1"*:

```
FOR $gobj IN /gtx:GISTriXP[@ID="project1"]//gtx:GeoObject
WHERE  intersect($gobj, gtx:GeoObject[@ID="faultSurface1"]) = true
RETURN $gobj;
```

The result is returned as a XML document containing the complete geoobjects.

| *Query function syntax* | *Return type* | *Description* |
|---|---|---|
| disjoint, equal, neighbourhood, intersect, inside (GeoObject) | List<GeoObject> | returns a list of GeoObjects with a given topological relationship with the argument GeoObject |
| intersection, union, difference (GeoObject, GeoObject) | GeoObject | returns a new GeoObject which is the result of an operation on the argument GeoObjects |
| buffer (bufferType, distance, GeoObject) | List<GeoObject> | returns the GeoObjects situated within the buffer of the argument GeoObject. The first argument allows to choose if the objects may intersect or need to be completely within the buffer. |
| distance (distanceType, GeoObject, GeoObject) | double | returns the distance between two GeoObjects. The first argument allows to choose euclidean, manhattan, or maximum distance |
| relativeSituation (GeoObject, direction) | List <GeoObject> | returns a list of GeoObjects situated in the given direction of the argument GeoObject |
| inside, outside, buffer (GeoObject) | List <ObservationPoint> | returns a list of ObservationPoints with the given relationship to the argument GeoObject |

*Tab. 5.1:* Table showing topological and geometrical extensions to the XQuery language.

### 5.2.2 Design of the new data management system

*Component-oriented query computation* A new approach for computations of high complexity on spatial geodata in XML format is required. This cannot be realized using a standard database query language. Instead, queries should be answered with the support of an application server because of the following arguments:

- Spatial query requests with high computational demands have to be computed. The required algorithms have a complexity $\geq$O(n) and cannot be formulated efficiently by the means of a query language.

- Data base management systems offer API's (Application Programming Interfaces) to improve the functionality of the query language and the functionality of the database server. For example, using the Tamino XTension API in order to enhance the XQuery functionality would be a possible solution. This allows to develop computational algorithms in a performant COM-enabling programming language. However, the development of a GIS adapted to a proprietary database programming interface would make the system dependent on one proprietary database management system.

- Following the data independence paradigm, in information systems the application logic is commonly separated from the database server. This can be achieved by the means of an application server. The application server communicates with user-application clients, and manages the data retrieval from the database and the computation of requests. Such an architecture can be used to provide a flexible three-tier GIS environment (figure 5.1).

- Several application servers are currently in use, for example IBM WebSphere [39] and Oracle AS [45]. They are designed for e-business and can cope with $>$10 requests/second, which is far more than required for serving a 3d GIS community with less than 1000 users.
  On the other hand, their interfaces support applications written in interpreted languages and scripts, like Java, .Net, and PHP. These languages are not optimized for efficient processing. However, efficient processing is required for efficient spatial numerical and algebraic computations of high complexity $>$O(n) and a typical data volume of $>>$1MB. A high-performant application server with a C++ API is suggested.

- A component-based information system with standardized exchange protocols and formats can be used in a distributed computing environment. For example, spatial computing applications can run on a remote computer and act as clients for an application server [21]. This method frees processing resources for the application server and the database server. Such a distributed system allows an efficient geodata management if high computational demands exist.
- For high performance data access the application server should offer a database driver API for custom database driver development. This will allow to use the application server with different XML database systems.

*Implementation features.* The design of the data management and query processing suggested in this thesis has the following characteristics (see figure 5.1):

- An application server, named ”*XAppS*”, is located between user application clients like Gocad and a XML database. It offers programming interfaces for both computing applications named ”*XApp*” and database drivers.
- XML database support. Currently, only the XML Server Tamino by Software AG is supported. However, the prototypical application server implementation [21] offers a ATL/COM-based driver programming interface to integrate further database management systems. The interface abstracts the functionality for:
  1. Initiating and terminating a database connection session.
  2. Requesting, storing and updating XML data.
  3. Transactions including commit and rollback.
- *XApp* are fast software components which compute query results. The integration of distributed applications is possible but not supported by the prototypical version [21].
- A Gocad plug-in has been developed as a client-side interface which allows
  1. to formulate and send extended XQuery requests to the application server (standard XQuery requests can be posed directly to the database server),

   2. to receive responses from the application server or from the database server,
   3. and to generate native Gocad application objects from the XML representation.

- For data transfer with user application clients and databases, the application server uses XML documents via HTTP. That way, the database and computing applications may run on other machines than the application server.

- Not every request of the Gocad client to the database is served by the application server. Only requests which are formulated using extended spatial XQuery operators (see table 5.1) are computed by a *XApp* called by the application server.

- due to its generic design the *XAppS* is not limited to 3d spatial computations but can be used to solve any computationally demanding requests, if appropriate computing applications are provided.

*Architecture of the application server.* The design of the internal architecture and the prototype implementation of the *XAppS* has been realized by T. Frank [21]. The *XAppS* consists of software components which interact using ATL/COM interfaces [21]. This allows to use a generic software design including templates. All components of the *XAppS* were programmed in C++. Figure 5.2 shows the main components of the new application server (after [21]):

- Request Server: The *XAppS* is receiving requests from clients and returns the results to them.

- Parser: The Xerces C++ parser is used to convert XML elements into C++ objects.

- Manager: The *XAppS* configuration is stored in a XML file. The Manager uses this file to extract the server settings used at server startup.

- Server Engine: The Server Engine integrates the single components to the *XAppS* application. It includes the logic for both the standalone and the system service version of the *XAppS*. Both versions are realized within one executable.

- Application Dispatcher: The Application Dispatcher loads the appropriate *XApp* and dispatches the client request to this application.
- *XApp* API: application programming interface for the communication with *XApp* computing applications

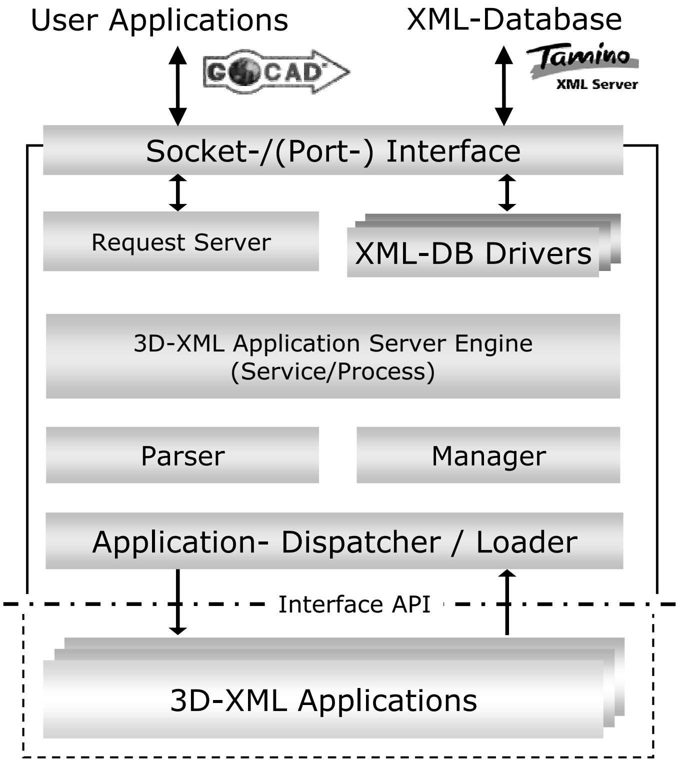
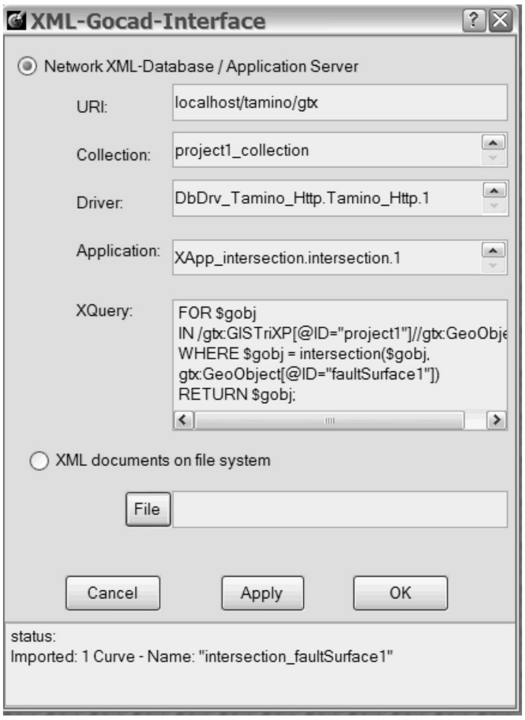


Fig. 5.2: Design of the XML application server (from [21])

*How are XQuery requests processed?* Spatial XQuery request can be sent to the application server by an application software like Gocad. Gocad has a client-side interface which features a query user dialog, and functionality to wrap a XQuery in a XML request document. Besides the XQuery, the request includes the logical database address and the name of the processing *XApp*. The name of the required *XApp* is obtained by parsing the XQuery request and searching for spatial query functions.

This request is interpreted by the application server, and required geomodel data

are thereupon obtained from the database. The application server calls then a specified application, which computes the result set. The result will subsequently be sent to the requesting application. Figure 5.3 shows an example user dialog, which allows to retrieve Gocad-objects in XML format from a database server. Alternatively, if no spatial query operator is contained in the request, the XQuery may also be sent directly to the XML database server. Also, XML documents containing Gocad objects may be stored and retrieved using the file system.
The Gocad client-interface has been implemented in C++ and is platform-independent.



● Network XML-Database / Application Server

URI: localhost/tamino/gtx

Collection: project1_collection

Driver: DbDrv_Tamino_Http.Tamino_Http.1

Application: XApp_intersection.intersection.1

XQuery:

```
FOR $gobj
IN /gtx:GISTriXP[@ID="project1"]//gtx:GeoObje
WHERE $gobj = intersection($gobj,
gtx:GeoObject[@ID="faultSurface1"])
RETURN $gobj;
```

○ XML documents on file system

File

status:
Imported: 1 Curve - Name: "intersection_faultSurface1"

Fig. 5.3: Query dialog in Gocad: a database and driver is specified, and a XQuery is formulated. The parameters are wrapped in a XML request document and sent to the *XAppS* via HTTP. The result is returned as XML document to Gocad, and subsequently converted into Gocad objects. In this example, the line of a surface intersection is returned.

### 5.2.3 XApp - applications to compute the result of XQuery requests

*XApp* applications are called by the application server using a request document in XML format containing data and - if required - function parameters for query

processing.
Many queries based on implicit topology can be answered using geometrical intersection tests. Two example spatial query operators from table 5.1 were chosen as example *XApp* application: intersect(GeoObject, GeoObject) and intersection(GeoObject, GeoObject), which return a boolean result or a new geoobject, respectively. The implementation has been realized in C++ in cooperation with T. Frank [21].

*Algorithms.* To compute the results of 3d geometrical and topological XQuery requests, several algorithms are available. Computing algorithms for queries on 3d simplicial complexes have been extensively studied by Breunig [9]. Algorithms for queries on macro-topological geomodels and directional queries have been discussed in sections 4.2.2 and 4.3.3. The *XApp* applications can call API functions implemented in user application software. In particular, algorithms implemented in Gocad have been used for the development of test applications.

*Example: XApp.intersection.* The example query *"Select the intersection lines of all surfaces of a project which with* faultSurface1." can be formalized in XQuery language

```
FOR $gobj IN /gtx:GISTriXP[@ID="project1"]//gtx:GeoObject
WHERE $gobj = intersection($gobj, gtx:GeoObject[@ID="faultSurface1"])
RETURN $gobj;
```

and answered using the application *XApp.intersection*. This application tests the geoobjects for intersection and returns the intersection polylines. The application dispatcher (figure 5.2) launches the *XApp.intersection*. It contains a function *ServeRequest()* which covers the core application logic (figure 5.4). First, this function retrieves the required data from the database. The *Intersect()* algorithm then tests whether the geoobjects intersect. For this example, an external Gocad function for intersection test is called. This Gocad function is based on a recursive octree computation and octree traversal for common cells.

## 5.3 Discussion

The data management is a core component of the 3d GIS framework presented in this work. The approach provided here fulfills the requirements stated in section 2.3. Major steps and contributions are:

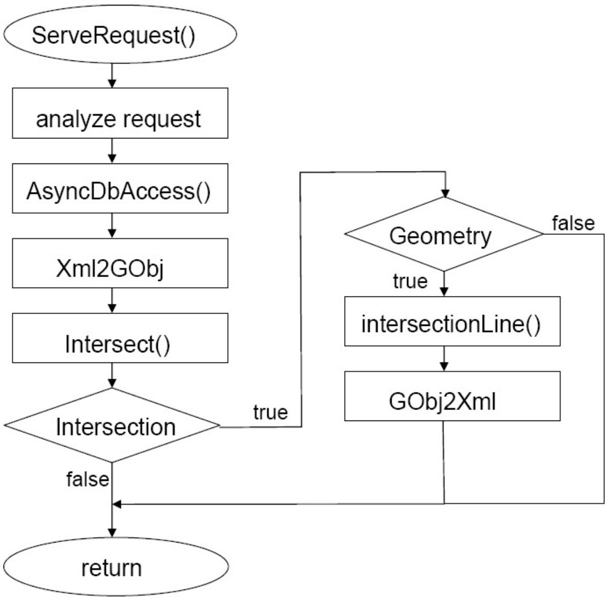


Fig. 5.4: UML diagram of the example computing application XApp.Intersection (from [21]).

1. Design and implementation of an example XML database according to the 3d spatial XML schema data model described in section 3.2.2.
2. Extension of the XQuery language with a set of spatial query operators for 3d geoobjects. This was necessary because computationally demanding, complex spatial queries cannot be solved by the means of a standard query language like SQL or XQuery.
3. Development of a concept to answer non-spatial and simple spatial requests directly using XQuery, and computationally demanding spatial queries using a middle-ware between the user application and the DBMS. In this example implementation, such spatial queries are posted from a Gocad application, and computed by fast C++ applications called by an application server.

Besides the capability to handle spatial XML data, a main advantage of the developed system is its genericity. This can be characterized by the following facts:

- It allows to integrate further databases capable of storing XML-documents. This follows from the application server driver API [21].

- It allows not mere database selection queries, but the use of *XApp* as tools for other computationally demanding data manipulations is possible, like fast data transformations, or down-sampling of tessellations for web-serving of geomodels.

- Due to the generic programming interfaces it is possible to integrate further databases in the information system, and to enable the usage within an heterogeneous environment of GIS tools. Any geospatial application front-end can be used together with the application server if appropriate computing applications, data converters and client interfaces exist, for example:
  - 2d GIS. Recent releases of large 2d GIS vendors support GML as data exchange format, which eases the integration of such systems.
  - web front-ends for observation data input, including mobile devices. For example, a HTML-based form prototype has been created which allows to automatically save the form data as valid XML documents according to the observation point schema. This facilitates the efficient and consistent observation data storage by field geologists.
  - web front-ends for display of geomodels or raw data in XML format, including X3D for 3d visualization

The application server has been designed to compute set based requests on geospatial data stored in XML databases. The query result set is efficiently computed directly from the geoobjects in the XML database. The storage of an oriented bounding box (OBB) tree [23] with 3d geoobjects along with the usage of the method of separating axes [23] for intersection tests may enhance the XML query processing speed significantly. A conceptually similar approach is used by common relational spatial database systems for 2D GIS-applications like the Oracle 9 Spatial database. They rely on precomputed spatial indexes such as R-Trees to compute fast search results.

The performance of spatial XML query depends primarily on the database server and can be optimized by the usage of indexes and efficient query formulation. Using Tamino version 3.2, for indexed highly structured data, like triangulated surfaces, the processing speed is 75-85% compared to common relational database servers [43]. For document-centric, semi-structured data like observation point descriptions the processing speed of native XML-databases is higher than their relational counterparts [18].

# 6. CONCLUSIONS

*Summary of achievements.* In this work, new technologies for a 3d geoscience information system were developed. These result in a XML-based 3d GIS framework, which overcomes existing deficiencies in the current geomodeling software environment. Similar to common 2d GIS, the 3d GIS framework is not a single-component application. Instead it is a system integrating several software components based on a novel data model incorporating geomodels and geological observation data. The main achievements are twofold:

1. For a GIS, querying is an essential part of data analysis. In this work, 3d spatial and geological query functions have been analysed and designed systematically and subsequently implemented as a Gocad Plug-in. For example, functions for the selection of geoobjects of a 3d geomodel by their topological, geometrical, and geological properties have been made available.

2. In order to improve the data management for geomodeling applications, a new concept which involves a XML database management system and which offers spatial query possibilities is suggested. A data model has been created in XML Schema language for geomodels and for geological observation points. This allows to store observations and geomodels as valid XML documents in a XML database. Data can be retrieved by Gocad or other applications via the internet by non-spatial or 3d spatial XQuery requests. For that purpose XQuery has been extended to include spatial operators. Spatial queries which are computationally demanding can be processed by a fast application server, which has been developed by T.Frank [21].

This 3d GIS framework is designed to support the data management and data analysis of large-scale 3d geological projects which usually incorporate large sets of spatial geological observations and 3d geomodels. The data management is designed to store and retrieve spatial sampling data, knowledge, concepts, and

models in an open, extensible, object-oriented, standard-conform way as XML documents.
The Gocad plug-in has been implemented in ANSI C++ [42], and relies solely on the Gocad and Qt software libraries. Gocad is distributed by Earth Decision Sciences, Houston (http://www.gocad.com), and Qt is distributed by Trolltech SA, Oslo (http://www.trolltech.com). Both components are available for common operating systems including Microsoft Windows and Linux. The only platform-dependent component of the GIS framework is the Microsoft COM-based application server, while user applications and databases can run on any internet-enabled operating system.

## 6.1 Critical reflections

As the title word ”framework” suggests, this is not a complete 3d GIS environment comparable to well-established, full-featured 2d GIS. This relates mainly to the following points:

1. The XML Schema data model is limited to geological observation points and BRep geomodels. Additional data types, especially wells, need to be handled for many projects. For wells, no common industry standard exists to date. Thus custom solutions are needed.
   No XML storage is provided for grid models. It is suggested to add this in future using the planned binary XML standard. Alternatively, the storage of grid data as binary large object (BLOB) in Gocad format is possible using XML.

2. The spatial query functionality developed in this work is not intended to be complete. Due to the many and very complex data types and the huge amount of theoretically possible operations involved this is unrealistic and is often also not useful to implement. Instead, geological, geometrical, and topological queries have been theoretically investigated, and a set of spatial and geological property queries which are considered useful for geological investigations has been implemented.

3. In contrast to 2d geographical data processing, geological data analysis and modeling of complex geological scenarios is often a difficult task due to the many parameters involved, different geological concepts, and sparse data. In a sense, a geomodel can be considered as a highly probable assumption

of a geological situation based on observation data and geological concepts. Using geostatistical methods, Gocad also allows to model the spatial and property uncertainties associated with a geomodel. This model is, however, a static one, and does not contain information about its geological genesis in terms of its spatial and property changes through time. Many important geological issues however are related to the genetic processes and relationships and the structural and compositional evolution of a geological situation. Such queries need to be handled by a spatio-temporal geoscience information system which captures the evolution of geoobjects and their relationships in geological history.

4. The creation of large databases of geological observations and of complex, regional geomodels composed of a large number of geoobjects requires still enormous efforts. Similarly, large 2d GIS projects including database creation commonly need several years to be completed. For real-world use of the functionality developed here resulting in new geoscience knowledge, a large database is required. For small models it is often appropriate to use simply visual analysis and come to the same result as by a sophisticated query. For this work such a large database was not available. Instead, the functionalities implemented were tested separately using relatively small data sets.

The core user application, the Gocad geomodeling software, is a mature application but nevertheless continuously under development in terms of its functionality, and in the long-term probably also in terms of its data model. Therefore, the GIS framework needs to keep track with Gocad in order to be usable. The other components, namely the XML database server, are also actively developed. Nevertheless the chances for a long usability of the GIS framework are high for to the following reasons:

1. The data model and interfaces in this work are based on XML. The superset of XML named SGML is an ISO standard since 1985, and XML itself, XQuery, and XML Schema are as well ISO standards. Due to their common usage in the world wide web they are actively developed. This implies that this is not short-lived language but a mature, well-supported, long-term solution for storing data as self-describing documents.

2. The triangulated radial-edge BRep topological data model used in Gocad and for XML storage is as well a stable, well studied data model. Other

approaches, like hierarchical GMaps [14] and the recently developed GML standard [22] for geographical data exchange, are convertible to this data model.

Due to the usage of XML, the observation point and BRep-model data models can easily be adapted to custom needs. This is facilitated by the object-oriented nature of the XML Schema language which is used for the data model definition: elements can be derived by extension or restriction, and new elements can be created.

The application server, as spatial database query processing component, has been developed in a generic way. That means that it is possible to use it with other databases and user front-ends which are able to process XML. It can be extended in functionality by developing new computing applications in a plug-in fashion.

## 6.2 Outlook

The following developments are suggested to extend the implementation of the system, or to make it more user-friendly:

- Completion of the implementation of the set of *XApp* for 3d spatial XML database queries using the application server.

- Implementation of features for user-friendly formulation of complex, combined queries. This could be achieved by an interactive graph, where the nodes represent basic query operators. Similar approaches have been used for fourth generation programming languages, like the OpenDX data explorer [36].

- Integrating artificial intelligence functionality. Probabilistic neural networks have successfully been used for mineral potential mapping in 2d GIS [25]. Applying this methodology in to 3d geomodels may result, for example, in intelligent prediction of 3d regions with high mineral potential.

- An interesting aspect is the usage of the application server within a spatio-temporal GIS. Because of the lack of common database management systems to efficiently represent temporal information related to geoobjects, like geometrical, topological, or non-spatial property changes, the application server could be the way to add custom logic to the GIS.

# BIBLIOGRAPHY

[1] Apel, M. 1999: 3D-Modellierung des östlichen Saxothuringikums. Unpubl. diploma thesis, TU Bergakademie Freiberg.

[2] ArcGIS user manual 2003, E.S.R.I., Redlands/CA.

[3] Bakker, P.: 2002: Image structure analysis for seismic interpretation. Ph.D. thesis, Delft University of Technology, ASCI Dissertation Series 78, Delft.

[4] Balovnev, O., Breunig, M., Cremers, A.B. 1997: From GeoStore to GeoToolKit: The second step. Proc. of the 5th International Symposium on Spatial Databases, Berlin, Germany, Lecture Notes in Computer Science, Vol. 1262, Springer, p. 223-237.

[5] Bartl, S. 2003: XML Schemas für geologische Daten und dreidimensionale Gocad-Modelldaten. Unpubl. diploma thesis, Technische Universität Freiberg, 46 p.

[6] Bishop, R. and Goldberg, S. 1980: Tensor Analysis on Manifolds. Dover, New York.

[7] Bonham-Carter, G.F. 2000: Geographic Information Systems for Geoscientists: Modeling with GIS. Pergamon Press, Oxford. 398p.

[8] Bosch, M., McGaughey, J. 2001: Joint inversion of gravity and magnetic data under lithological constraints. The Leading Edge 20, SEG, p. 877-881.

[9] Breunig, M. 1996: Integration of Spatial Information for Geo-Information Systems. LNES No. 61, Springer Verlag. 171 p.

[10] Breunig, M. 2001: On the way to component-based 3D/4D Geoinformation Systems. LNES No. 94, Springer Verlag. 179 p.

[11] Brodaric, B. 1997: Field Data Capture and Manipulation Using GSC Fieldlog v3. US Geological Survey Open-File Report 97-269

[12] Broome, J., Brodaric, B., Viljoen, D., Baril, D. 1993: The NATMAP Digital Geoscience Data-Management System. Computers and Geosciences, vol. 19, no. 10, p. 1501-1516.

[13] Brottka, K. 2003: 3D Modellierung und Inversion in der Gravimetrie. Unpubl. diploma thesis, TU Bergakademie Freiberg.

[14] Conreaux, S., Levy, B., Mallet, J.-L. 1998: A cellular topological model based on generalized maps: The gocad approach. Gocad Meeting 1998 Proceedings, Nancy, 17p.

[15] Eberly, D. 2001: Intersection of Convex Objects: The Method of Separating Axes. Documentation and C++ source code, http://www.magic-software.com. See attachment (CD-ROM).

[16] Egenhofer, M., Franzosa R. 1991: Point-set topological spatial relations. International Journal of Geographical Information Systems, vol. 5, p. 161-174.

[17] Egenhofer, M. 1993: A model for detailed binary topological relationships. Geomatica vol. 47, Canadian Institute of Geomatics, p. 261-273.

[18] Erl, T.: Service-Oriented Architecture: A Field Guide to Integrating XML and Web Services. Prentice-Hall.

[19] Euler, N. 1999: Modelisation volumique de reservoirs petroliers: constructions et mise a jour. PhD thesis, Institut National Polytechnique de Lorraine, Nancy.

[20] Fanchi J.R. 2002: Shared Earth Modeling: Methodologies for Integrated Reservoir Simulations. Gulf Professional Publishing.

[21] Frank, T. 2003: Design and Implementation of a 3d XML application server. Unpubl. M.Sc. Thesis, University Freiberg, 97 p.

[22] OpenGIS Geography Markup Language (GML) Implementation Specification v.3 2003, http://www.opengis.org. See attachment (CD-ROM).

[23] Gottschalk, S., Lin, M.C., Manocha D. 1996: OBB tree: A hierarchical structure for rapid interference detection. Proceedings of ACM Siggraph'96. 24 p.

[24] Güting, R.H.: Geo-Relational Algebra: A Model and Query Language for Geometric Database Systems. In: Schmidt J.W. et al. (Eds.): Advances in Database Technology. Proc. of the Int. Conf. on Extending Database Technology, Venice, 1988, 506-527.

[25] Harris, D.P., Pan, G. 2000: Information Synthesis for Mineral Exploration, Oxford University Press.

[26] Kemp, E. de 2000: 3-D Integration and Visualization of Structural Field Data: Tools For Regional Subsurface Mapping. PhD Thesis, Université du Québec.

[27] Keskes, N. 1994: Image analysis techniques applied to 2D and 3D seismic data. Proc. SPIE Vol. 2301, p. 111-116

[28] Labat, L., Shtuka, A. , Mallet, J.-L., Coureaud, B. 2002: Facies Simulation Using Membership Functions and Multi P-Field. ECMOR VIII. Meeting Proc., Freiberg, 8 p.

[29] Lepage, F. 2002: Triangle and tetrahedral meshes for geological models. Terra Nostra Vol. 4., IAMG Conference 2002 Berlin, p. 105-110.

[30] Levy, B., G. Caumon, G., Conreaux, S. 2000: TopoLab: A generic implementation of G-Maps.

[31] Levy, B. 1999: Topologie Algorithmique: Combinatoire et Plongement. PhD thesis, INPL, Nancy.

[32] Lienhardt, P. 1994: N-Dimensional Generalized Combinatorial Maps and Cellular Quasi-Manifolds. Journal on Computational Geometry and Applications, volume 4, p. 275-324

[33] Malczewski, J. 1999: GIS and Multicriteria Decision Analysis. J. Wiley Sons, Hoboken, NJ.

[34] Mallet J.-L. 2002: Geomodeling. Oxford University Press Monograph.

[35] Munkres, J. R. 1975: Topology: A First Course. Prentice-Hall, Englewood Cliffs, NJ.

[36] Thompson, D., Braun, J., Ford R. 2003: OpenDX Paths to Visualization. Visualization and Imagery Solutions, Missoula, MT.

[37] Paredaens, J., Van den Bussche, J., Van Gucht, D. 1994: Towards a theory of spatial database queries. Extended abstract, Proc. 13th Symp. on Princ. of Database Systems, Minneapolis, p. 279-288.

[38] Rigaux, P., Voisard, A., Scholl, M.O. 2002: Spatial Databases: With Application to GIS. Morgan Kaufmann.

[39] Sadtler, C. et al. 2003: IBM WebSphere Application Server - Express V5.0 Handbook. IBM Redbooks.

[40] Soller, D.R., Berg, T.M. 1999: The National Geologic Map Database - A Progress Report; USGS Open File Report 99-386.

[41] Steno, N. 1669: De solido intra solidum naturaliter contento dissertationis prodromus. Florence.

[42] Stroustrup, B. 1987: The C++ Programming Language. Addison-Wesley , Reading, MA.

[43] Tamino XML Server 3.1 White Paper 2003. Software AG, http://www.softwareag.com/tamino. See attachment (CD-ROM).

[44] Unified Modeling Language (UML) Specification v.1.3 2003, http://www.omg.org. See attachment (CD-ROM).

[45] Vandivier, S., Cox, K. 2002: Oracle9i Application Server Portal Handbook. Osborne ORACLE Press Series, Berkeley.

[46] Weiler, K. 1988: The radial edge structure: a topological representation for non-manifold geometric boundary modeling. Geometric Modeling for CAD Applications, Elsevier Science Publish., p. 3-36.

[47] eXtensible 3D Specification 2003, http://www.web3d.org. See attachment (CD-ROM).

[48] XQuery 1.0: An XML Query Language, W3C Working Draft, http://www.w3.org/TR/xquery. See attachment (CD-ROM).

[49] Extensible Markup Language (XML) 1.0. W3C Recommendation 6/2000, http://www.w3.org. See attachment (CD-ROM).

[50] XML Schema Specification, http://www.w3.org. See attachment (CD-ROM).

# LIST OF FIGURES

# LIST OF TABLES