Karlsruher Institut für Technologie

Spatio-Temporal Data Handling for Generic Mobile Geoinformation Systems

Zur Erlangung des akademischen Grades eines

Doktor-Ingenieurs

von der Fakultät für

Bauingenieur-, Geo- und Umweltwissenschaften

des Karlsruher Instituts für Technologie

genehmigte

Dissertation

von

Dipl.-Systemwiss. Paul Vincent Kuper

aus Osnabrück

Tag der mündlichen Prüfung: 11. Juli 2016

Hauptreferent: Prof. Dr. Martin Breunig
Karlsruher Institut für Technologie

Korreferent: Prof. Dr. Mulhim Al-Doori
American University in Dubai

Karlsruhe 2016

Paul Vincent Kuper

Spatio-Temporal Data Handling for Generic Mobile Geoinformation Systems

PhD Thesis, 2016

Referees:

Prof. Dr. Martin Breunig

Prof. Dr. Mulhim Al-Doori

Karlsruhe Institute of Technology

Department of Civil Engineering, Geo- and Environmental Sciences

Geodetic Institute

Englerstr. 7

76137 Karlsruhe

Kurzfassung

Bei der digitalen Modellierung der Erdoberfläche und des Erduntergrunds entstehen große Mengen an raum-zeitlichen Daten. Das Management, die Verbreitung und Nutzung dieser Daten ist daher von essentieller Bedeutung. Diese Arbeit stellt hierfür drei aufeinander aufbauende Bestandteile eines Workflows für den effektiven Umgang mit raum-zeitlichen Daten vor. Dieser beginnt bei der Datenhaltung mit dem Ziel die Daten mobilen Geoinformationssystemen über geeignete Webservices zur Verfügung zu stellen.

Der erste Teil umfasst die effiziente Verwaltung raum-zeitlicher Daten. Hierfür wird ein Verwaltungs-Modell vorgestellt, welches aus den fünf ineinandergreifenden Konzepten Point Tubes, Deltaspeicherung, Topologie Management, Verwaltung thematischer Daten und Netz Komponenten besteht. Die jeweiligen Konzepte werden zunächst auf den Anwendungsfall der Verwaltung raum-zeitlicher Daten auf Basis simplizialer Komplexe übertragen. Anschließend werden diese zu einem umfassenden Verwaltungs-Modell zusammengeführt, welches die Vorteile der Konzepte vereint. Dieses bildet die Grundlage für die Umsetzung innerhalb einer Datenbank Architektur. Das Verwaltungs-Modell ist für verschiedenste Anwendungsfälle geeignet. Dazu gehören massive Veränderungen zwischen aufeinander folgenden Zeitschritten und ein differenzierter Umgang mit thematischen Daten. Unterschiedliche zeitliche Diskretisierungen von Teilbereichen eines 4D Modells werden entsprechend behandelt. Insgesamt wird durch das Verwaltungs-Modell der Speicherplatzbedarf von 4D Modellen reduziert und räumliche sowie raum-zeitliche Operationen wesentlich beschleunigt.

Der zweite Teil umfasst den Austausch raum-zeitlicher Daten und die Zurverfügungstellung raum-zeitlicher Operationen über eine Netzwerk-Schnittstelle mit Hilfe von Webservices. Beispielhaft werden zwei Webservices realisiert, welche auf den Geodatenstandards Web Feature Service und Web Processing Service aufbauen. Hierbei wird gezeigt, wie der Aufbau auf vorhandenen Standards die Entwicklung von Client-Anwendungen und die Integration in bereits existierende Geodateninfrastrukturen erleichtert. Durch die Webservices können raum-zeitliche Daten auf Client-Anwendungen übertragen und raum-zeitliche Operationen auf der Server-Seite durchgeführt werden.

Der dritte Teil des Workflows umfasst einen Ansatz für ein generisches mobiles Geoinformationssystem für professionelle Anwender. Hierfür wird zunächst eine allgemein gültige Definition erarbeitet. Dabei wird erläutert, wie die intensive Nutzung von Geodatenstandards eine geeignete Basis für solch ein System darstellt. Das allgemein gültige Konzept wird anschließend um die Anforderungen raum-zeitlicher Daten erweitert. Dadurch wird eine beispielhafte Anwendung des Verwaltungs-Modells und der Webservices für raum-zeitliche Daten geschaffen.

Abschließend werden alle drei Bestandteile dieses Workflows anhand konkreter Implementierungen evaluiert und als freie open source Software zur Verfügung gestellt. Dabei zeigt sich, dass die Anforderungen an einen effektiven Umgang mit raum-zeitlichen Daten durch den Workflow erfüllt werden können.

Abstract

In the geosciences the modelling of surface- and subsurface-structures produces huge amounts of 3D data. With the introduction of time as a fourth dimension this amount increases even more. Therefore, the handling of such data is most important. Within this thesis, a workflow for an efficient and practical handling of spatio-temporal data is presented.

This workflow consists of three layered parts. The first part is the efficient management of spatio-temporal data. A data management model consisting of the five concepts Point Tubes, Delta Storage, Topology Management, Handling of Thematic Data, and Net Components is presented. These concepts are transferred and adapted to manage spatio-temporal data based on simplicial complexes. The development of a comprehensive model that combines the benefits of these partial concepts, forms the foundation of a realization within a database management system. With this model, it is possible to handle various kinds of spatio-temporal applications. These include massive changes between consecutive time steps and a differentiated handling of thematic and semantic data. Variable temporal discretizations of partial regions of a 4D model are managed accordingly. The data management model reduces the storage requirements of 4D models and accelerates computational operations significantly.

The second part focuses on the development of Web services for the dissemination of spatio-temporal data. For this purpose, two Web services are presented that are based on the OGC standards Web Processing Service and Web Feature Service. These services grant access to spatio-temporal data and operations via a network interface. It is demonstrated that the use of standards facilitates the realization of client applications and the integration into already existing geodata infrastructures. With these Web services spatio-temporal data can be distributed and spatio-temporal operations are executed server-side.

The third part of the workflow is a generic mobile GIS for professional users as a typical application for the spatio-temporal data management model and the related Web services. The intensive use of spatial data standards is explained to be an appropriate basis for such a system. First, a general concept for generic mobile GIS is presented. Afterwards, this concept is extended to include the specific requirements for the handling of spatio-temporal data.

Supplementary, explicit implementations for all three parts of the workflow are provided as free and open source software and are partially integrated into a 3D/4D geodatabase architecture. These are used for a concluding validation and evaluation and confirm the benefits of the workflow within multiple application scenarios.

Table of Contents

List of Figures

List of Tables

List of Algorithms

1 Introduction

1.1 Motivation

The data basis of diverse scientific disciplines consists of data with a spatial reference [Torge, 2003], [Bartelme, 2005], [Li, 2015], [Dao et al., 2015], [Weihed, 2015]. In the 1960s Tomlinson developed the first digital geographic information system (GIS): the Canada Geographic Information System (CGIS) [Tomlinson, 1967]. Since then the field of GIS and geographical information science (GIScience) has developed enormously. In most cases we have to deal with 2D or 3D data.

The amount of spatial data grows rapidly, which is partly due to the continuous improvement of the recording equipment and the distribution in different scientific and industrial disciplines. The research field of GIScience provides essential results and information for spatial issues, such as the identification of a suitable location for a shopping center or an exploration site for the next oil reservoir. The results of location based questions are indispensable even from daily use. For example, every day we can calculate the fastest way to our workplace using current traffic information and geometric analyses.

When it comes to digital spatial data, we have to differentiate between different data types, their dimensions, and the dimension of the space in which the data is present.

For most applications the use of 2D data is sufficient. Such data can for instance be obtained through the digitization of aerial photos or analogue maps.

Nowadays, an increasing amount of data sets come from a wide range of movement and location sensors, e.g. from smartphones or GNSS devices. In most cases we deal with 2D data types of the ISO 19125 (Simple Feature Access) [OGC, 2011], i.e. points, lines and polygons. A typical application for such data is the visualization of thematic data. Topological relationships between spatial objects can be used to establish graphs for various purposes, e.g. routing algorithms [Bast et al., 2014].

Of course, applications on the basis of 2D data have their limitations, e.g. the height and volume of a particular object cannot be handled accordingly.

For this purpose, it is possible to use 3D data such as 3D point clouds, which are for instance gathered by LiDAR (Light Detection and Ranging) or depth cameras, e.g. Microsoft

Kinect[1]. These are often X-, Y-, Z-values referenced to a local Cartesian coordinate system or to a geodetic reference system such as WGS84 (World Geodetic System 1984). Additionally, 3D meshes, structured and irregular (e.g. simplicial complexes) as well as structured and regular (e.g. 3D grids), can be used to describe the surface and/or volume of objects. The origin of such objects can often be found in the modelling of the natural environment [Breunig et al., 1999], [Mallet, 2002], [Xing et al., 2015].

For many applications it is necessary to consider a temporal aspect, i.e. to describe the transformation of a certain region within a period of time. In 2D applications we can model transformations such as the expansion of a city or the alternation of crops on a field. Also, real-time applications such as traffic monitoring can be handled in 2D [Bottero et al., 2013].

An important focus of research, which also is the target area of this thesis, is the modelling of 3D spatio-temporal data that have their origin in the field of earth sciences, particularly *geology*, *geothermal energy* and *geomorphology*.

When 3D data gets extended by a temporal component it is possible to model the transformation of natural structures such as soil layers [Weihed, 2015]. In these fields various modelling tools such as GOCAD[2], GeoModeller[3] or Petrel[4] are used for the development of digital 3D models. In this thesis, we focus on models that consist of *d-simplicial complexes* with a dimension $d \in \{0,1,2,3\}$ [Mallet, 1992], [Mallet, 2002]. Accordingly, point clouds, segment-, triangle- and tetrahedra-nets form the geometric basis to describe a landslide or a geologic layer, respectively. The temporal transformation of such structures is realized by the use of time series that consist of interrelated 3D models. The intermediate period can be supplemented by the use of an appropriate interpolation technique. Due to this procedure we are able to model time-continuous phenomena.

Apart from spatial and temporal dimensions there are additional kinds of data we have to consider, such as semantic or thematic data, e.g. type of use or owner, but also metadata, e.g. recording time or copyright. A more detailed overview of various spatial data models is presented in Chapter 2.

Another important task is to handle and provide such huge amounts of spatial data efficiently to different users and systems. For the handling of geodata, various tools have

[1] Microsoft Kinect is originally a consumer product for the home video game console XBOX. The sensor captures depth and RGB-pictures.
[2] Paradigm GOCAD® – Software suite for the development of geoscientific data.
[3] Intrepid GeoModeller – Software to develop 3D geological models.
[4] Schlumberger Petrel – Exploration software for the development of geoscientific data.

been developed or adapted from other disciplines to the field of GIScience in the last decade. To work with data files is an outdated and no longer contemporary method. In 1976 with GEOQUEL a first approach for a geodatabase has been developed to improve the handling and storing of the ever-growing amount of spatial data [Williams, 1976].

Nowadays we use professional geodatabases, which provide a comfortable data management including a support of various spatial access methods. In addition, such geodatabases often provide simple geometric analysis functions and capabilities for the exchange of information between multiple collaborators and working groups. According to [Worboys and Duckham, 2004] a geodatabase forms the heart of any GIS. Geodatabases that handle time as an additional dimension need to meet a significantly extended set of requirements and are subject of ongoing research [Le, 2014], [Gabriel et al., 2015], [Ohori, 2016], [Breunig et al., 2016].

The support of a temporal component for simplex based data within conventional modelling tools is still limited [Mejia et al., 2015], [Royer et al., 2015]. Therefore, the composition of 4D models is one important task for a 3D/4D geodatabase. By the linking of 3D models to an explicit 4D model, continuously changing phenomena can be managed in a comprehensive way. Thus, the access to individual time steps of the model is facilitated. Additional intermediate steps should be calculated with the help of interpolation routines. A geodatabase developed this way is a good basis for advanced spatio-temporal operations.

When dealing with spatio-temporal phenomena we need to handle huge amounts of data. Therefore, it is essential to improve the internal management by the development of appropriate data structures and data handling concepts, i.e. a *data management model*. This topic is one of the major issues addressed by this thesis.

Another important challenge is the dissemination of data. The access to data should be as standardized as possible for an appropriate exchange of data between client and server. This is a key issue for a productive collaborative cooperation if multiple teams are involved. To this end, concepts for countless different Web services have been developed for various applications, which standardize the communication between the client and a data source, e.g. a database [Fielding, 2000], [Daigneau, 2011]. Unfortunately, no solutions are available that meet the requirements for a standardized access of spatio-temporal data and operations.

A spatial data infrastructure based on standards and with a suitable geodatabase in the backend, however, is useless without appropriate client applications. A conventional client application, for instance, is the commercial 3D modelling software GOCAD, as well as the free and open source visualization package ParaViewGeo[5]. Meanwhile, mobile geographic information systems (GIS) for 2D as well as for 3D data turn into the spotlight of current research [Meek et al., 2013], [Milanes et al., 2014], [Shitkova et al., 2015], [Han et al., 2015], [Zajíčková, 2015], [Kang et al., 2015]. There are different professional solutions available on the market, such as *Cadenza Mobile*[6], *ArcPad*[7] and *GeoTechMobile*[8] which have been developed primarily for the processing of 2D data. Furthermore, current research projects deal with the development of mobile GIS for 3D data. The focus is on the visualization of information in the field. So-called augmented reality (AR) viewer were developed that superimpose the reality on site with computer generated 3D data [Urban et al., 2013]. Due to this promising research field, the present thesis focuses primarily on mobile GIS as a client application for spatio-temporal data.

In summary, this thesis addresses three interrelated challenges:

1) An efficient and practical handling of spatio-temporal data that is based on simplicial complexes.

2) The dissemination of spatio-temporal data and operations based on existing standards.

3) The realization of a generic approach for mobile GIS applications with a focus on spatio-temporal data.

The scope of this work is to develop an effective workflow for the processing of spatio-temporal data which is based on simplicial complexes. The main challenge for this workflow is the efficient management of spatio-temporal data, suitable for an implementation in spatial databases. The second challenge is to provide an appropriate data infrastructure for the exchange of spatio-temporal data. To this end, explicit Web services are to be developed, which support the exchange of spatio-temporal data and the invocation of spatio-temporal operations. The third challenge is to design and develop a generic mobile GIS application, which can be used as a typical application for the management model and the Web services. The individual objectives that lead to such a workflow are subject of the next section.

[5] ParaViewGeo – Open source visualization for geoscience.
[6] Cadenza Mobile – Mobile GIS application for Android and iOS.
[7] ArcPad – Mobile GIS application for Windows Mobile.
[8] GeoTechMobile – Mobile GIS application for Android.

1.2 Objectives and Structure

The primary objective of this thesis is the design and development of a spatio-temporal data management model, which accelerates the data-retrieval and the performance of spatial and spatio-temporal operations. Simultaneously the storage requirements are to be reduced and enhanced capabilities for the control and management of thematic and semantic data have to be provided. Currently, there are multiple prototype database management systems (DBMS) for different spatial data types and applications available [Wolfson et al., 2002], [Breunig et al., 2009], [Gabriel et al., 2015]. This thesis focuses on the management of spatio-temporal data that consist of 0-3D-simplices and form simplicial complexes, cf. Chapter 2.3.2.

Among others, [Polthier and Rumpf, 1994], [Strathoff, 1999], [Breunig, 2001], [Shumilov et al., 2002], [Siebeck, 2003], [Worboys and Duckham, 2004], [Apel, 2004], [Rolfs, 2005] and [Le, 2014] have developed various concepts for the management of spatio-temporal data. This preparatory work represents an important foundation for the development of the data management model that is presented later in Chapter 3. The corresponding requirements of such a model are examined in detail within Chapters 2.4 and 3.

Based on this work a new approach for a data management model is developed, which comprises five sub-concepts as described in Chapter 3.1 to 3.5. For each concept the existing prior work is presented and subsequently adapted to the management of spatio-temporal data based on *d*-simplicial complexes. At the same time, the impact on the different dimensions $d \in \{0,1,2,3\}$ of such complexes is described in detail. The advantages and possible disadvantages of each approach are worked out in particular. Subsequently a comprehensive model on the basis of these five sub-concepts is developed. This model requires less presupposed constraints and provides an expanded functionality while computational operations are accelerated and storage requirements are reduced.

The dissemination of spatio-temporal data is one key challenge for a multi-user collaboration. The specific requirements for such a dissemination that are to be met by an appropriate Web service, are introduced in Chapter 4.1. In Chapter 2.5 a selection of important general Web service fundamentals are presented. Among others, [Pouliot and Bédard, 2007], [Bär, 2007], [Dittrich, 2012], [Wild, 2012], [Breunig et al., 2013b] have developed first approaches for the dissemination of data based on simplicial complexes.

This thesis was written while the INSPIRE (Infrastructure for Spatial Information in the European Community) directive and the German geodata access law (GeoZG) were being implemented. These will facilitate the standardized exchange of spatial data between various European authorities. Therefore, concepts for a standardized access of spatio-temporal data are discussed based on established standards for the dissemination of 2D data. In this context, the use of existing concepts and standards for developing appropriate Web services for spatio-temporal data is analyzed.

An important group of potential clients for such Web services are mobile GIS, which are introduced in Chapter 2.6. Mobile GIS provide opportunities for a productive work in the field, such as different possibilities of visualization and processing of spatial data.

First, the requirements of a generic mobile GIS for professional users are worked out, afterwards a general concept for a generic mobile GIS is presented in Chapter 5. For this purpose, the use of standards for the transfer and processing of spatio-temporal data in the context of mobile GIS is examined. The concept of a generic mobile GIS is based on the strengths of various OGC standards in particular and is therefore closely linked to the development of these standards. The realization of this concept is part of Chapter 5.4.

In the following, the three main objectives of this thesis and their linkage are summarized:

First objective - a data management model for spatio-temporal data

The design and development of a comprehensive data management model for spatio-temporal data forms the core of this thesis. Data from the research field of earth sciences particularly geology, geothermal energy and geomorphology form the data base, whose geometric part is represented by *d*-simplicial complexes. The focus is on the modelling of continuous phenomena.

The intention is to accelerate the data-retrieval and the performance of geometric operations. A reduction of storage requirements and an enhancement of thematic and semantic data handling are part of this objective. The model must be suitable for an implementation in a spatial database.

Second objective - Web services for the dissemination of spatio-temporal data

The second objective focuses on the development of Web services, providing a clear interface between the data management model and client applications such as mobile GIS. To achieve this, existing and standardized solutions for the dissemination of conventional spatial data must be taken into account. These Web services must be capable to provide access to spatio-temporal data and spatio-temporal operations.

Third objective - a concept for a generic mobile GIS for spatio-temporal data

The third objective is the design and development of a generic mobile GIS as one possible client application for the visualization and processing of spatio-temporal data. In a first step the characteristics of a generic and universal mobile GIS need to be worked out in general. Subsequently, the consideration of the specific requirements for spatio-temporal data in this context lead to the development of a generic mobile GIS for spatio-temporal data. Thereby the entire client-server workflow is completed.

For each objective at first the prior work is examined. Based on that, a technical framework is created and presented in detail.

Part of this thesis is not only the development of concepts to the targeted objectives, but also the development of explicit implementations. A practical implementation of the comprehensive data management model based on the geodatabase architecture DB4GeO [Breunig et al., 2009] is presented in Chapter 5.3. DB4GeO was initially developed at the University of Osnabrück and is currently the subject of multiple research projects at the Geodetic Institute of the Karlsruhe Institute of Technology (KIT). An implementation of Web services for the standardized access of spatio-temporal data is described in Chapter 4.3. The practical realization of a generic mobile GIS for spatio-temporal data is part of Chapter 5.4.

[Roddick et al., 2004] noted that a simple visualization of the data itself supports the user in handling a spatio-temporal database. Therefore, a WebGL Viewer is developed, which enables a rapid visualization of data within an Internet browser.

2 Fundamentals and Related Work

This chapter discusses relevant preparations and preparatory work, which are helpful for the further understanding of this thesis. Several different models for the representation of spatial data have been emerged over the years due to different application areas with diverse requirements. Sometimes these models are based upon each other, but they can also differ fundamentally. In a first step, purely spatial data models are introduced and time is subsequently added as a new dimension. A strong focus is on the geometric model of *simplicial complexes*, which forms the basis of the spatio-temporal data that is used within this thesis. In the context of spatio-temporal data modelling, there are different approaches for dealing with time itself. A selection of the most relevant for this thesis is presented. For a general introduction to the mathematical foundations of handling spatial data in Euclidean space the book "GIS: A Computing Perspective" by [Worboys and Duckham, 2004] is highly recommended.

Preparatory work on spatial databases is part of Section 2.4. Hereby the focus is on geo-databases that are specialized on the management of spatio-temporal data.

Furthermore, the technical background for the dissemination of data via a network interface is part of Section 2.5. In particular, *Representational State Transfer* (REST) [Fielding, 2000] and *Simple Object Access Protocol* (SOAP)[9] [W3C SOAP] are discussed. Subsequently, two Web services standards of the Open Geospatial Consortium (OGC) are described: *Web Feature Service* and *Web Processing Service*. These form the foundation for an appropriate infrastructure for spatio-temporal data, which is the subject of Chapter 4.

Prior work in the field of *mobile GIS* and *Location based Services* (LBS) is presented in Chapter 2.6 and form the basis for the realization of a generic mobile GIS for spatio-temporal data, which is part of Chapter 5.

2.1 Spatial Data Models

The existence of multiple models for the mapping of spatial data originate in requirements, which differ depending on the specific application. According to [Peuquet, 1984] it is not possible to develop a universal and simultaneously suitable data model for any

[9] Nowadays often only the acronym SOAP is used because this architecture is neither simple nor limited to objects.

situation. These models possess different representations and ways of processing the data. Another issue is the dimension of the space, in which the data is present.

In a first step, the modelling of spatial data can be divided into two different approaches. The first are *field-based* models realized by *raster data*, e.g. orthophotos that consist of a uniform grid. In many cases factual information is assigned to *cells*. A grid consists of a matrix of cells having a prescribed common extent. Every grid cell owns a certain value, which belongs to the respective cell, cf. Figure 1a. Following this approach, it is possible to describe e.g. the nature of surfaces. Often several of these raster layers are superimposed on one another and brought into relation during analysis functions to generate new information. Such raster data can be handled and stored in image formats such as TIFF or GeoTIFF [Bartelme, 2005].

In the field of 3D geosciences it is also possible to handle raster based data, which are often referred to as *grid* or *voxel*, cf. Figure 1b. In this case these grids are uniformly arranged and structured. Within a 3D-grid data may also be assigned to individual cells. Such data can be e.g. pressure or temperature values. It is also possible to assign more than one value to each cell simultaneously. The distribution is considered to be uniform within each specific cell. The benefit of uniform and structured grid cells is the clear allocation and the possibility to implement fast and simple algorithms. Disadvantages are a potentially high storage requirement if the data is unequally distributed and an inaccuracy emerging through the grid itself [Güting, 1994].

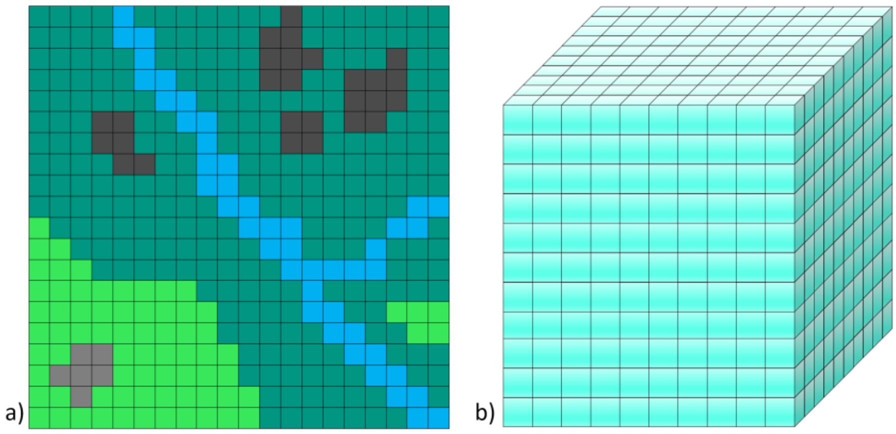


Figure 1 - a) 2D-raster b) 3D-grid (regular)

The second major approach of spatial data models are *object-based* models realized by *vector data*. There exist several different data models, which are presented in the following. A major difference to the raster-based data is the possibility to create spatial data by using *points* that can be linked to *lines* or *polygons*, cf. Figure 2.

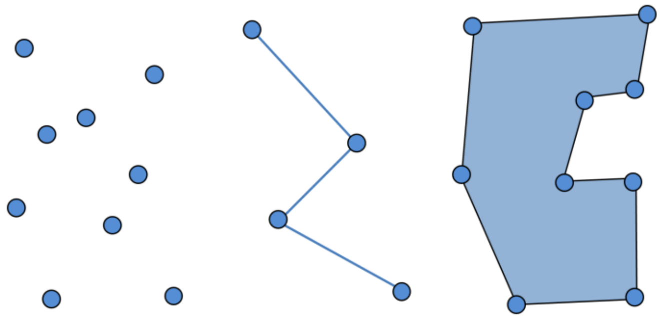


Figure 2 - Various types of vector data: points, linestring, polygon.

In comparison to the raster data this approach improves the flexibility and accuracy of the presentation and handling of the data. This also reflects in a more precise indication of the location of information. Irregular and unstructured grids, e.g. triangulated irregular networks (TIN) are assigned to the field of vector data, see Figure 3. There are several different spatial data models for this kind of node based geodata, which can have various restrictions or constraints for the construction of basic geometries. The so called *Spaghetti Model*, for instance, offers no information about the connection between adjacent geometries. Meanwhile there are topological data models which can handle such information and use them for spatial analysis functions [Peuquet, 1984].

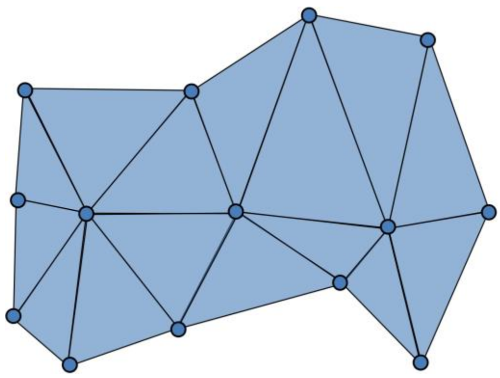


Figure 3 - A triangulated irregular network (TIN)

Although 2D data is commonly used for conventional map applications, the field of 3D vector data becomes increasingly relevant [Breunig and Zlatanova, 2011], [Pouliot et al., 2013], [Gabriel et al., 2015], [Biljecki et al., 2015].

The spatial data itself can contain various information in addition to purely geometric attributes. These include, for instance, *thematic* or *semantic data*[10] (e.g. soil quality) or

[10] Thematic data are primitive attributes such as temperature values. Semantic data are usually more complex and refer to particular entities.

meta-data (e.g. date of the survey). In addition, spatial data, regardless if 2D or 3D, can be expanded to a higher dimension to include a temporal component. This kind of spatio-temporal data is part of Section 2.3.

The usage of a specific data model highly depends on the particular application. 2D Applications often describe spatial relationships between different geometries. An example is the demarcation between streets, houses and meadows. Here different data models are suitable for various problems. Is it purely to the presentation of information? Shall the data be processed any further? Is it important to have topologically distinct geometries? For almost every geometric algorithm or analysis function some constraints need to be matched by the data, e.g. to calculate the area of a polygon geometry, the polygon must be closed and clearly identifiable. A frequently used data model for the description of valid points, polylines and polygons, is the *Simple Feature Access Model* [OGC, 2011]. Within this model such conditions, e.g. the constitution of a polygon, are well defined.

Modelling of natural shapes

[Peuquet, 1984] describes different ways to represent natural surfaces. She compares different data models for mapping spatial data in a 2D space. Thereby, various methods for the tessellation of space are compared with each other. Conventional methods are the use of regular grids of squares, triangles or hexagons. Peuquet comes to the conclusion that irregular structures represent natural data most efficiently. This is partly due to the fact that an irregular structure can be adapted to the distribution of the data itself, while a grid is imposing a precision level based on the resolution of the cells. Following this approach, regions with a higher density of important information can be mapped accordingly. Of the three different grid structures, the triangles are best suited to span an irregular network. This perception can also be transferred to model surfaces in 3D space. Therefore, TINs are useful to reflect the structure of a natural terrain. [Mallet, 1992] confirms this assumption and identified the advantages of the usage of TINs over conventional CAD techniques such as B-splines. Therefore, he initiated the development of the well-known 3D modelling software GOCAD.

Definition of 3D

In addition to the different data models it is also important to define the term "3D data". In the field of GIScience this is done in different ways. The examples in the following Table 1 all use three-dimensional coordinates (x, y, z):

Table 1 - Definition of 3D data

Dimension	Properties
2.5D	Only surfaces, no vertical surfaces and no bulges are allowed. Addition of the height z as an attribute to the geometry. $z = f(x, y)$.
2.7D	2.5D + vertical surfaces are allowed, e.g. walls, houses, slopes, but no bulges. [Förstner, 1995]
2.8D	2.7D + bulges are allowed, e.g. balconies. Only the shell of 3D objects is represented. [Gröger and Plümer, 2005]
3D	2.8D + not only the shell of objects, but the real volume is represented.

The diverse handling of three-dimensional coordinates has its origin in the varieties of particular applications. For instance, when working with a *digital terrain model* (DTM) the height is needed only once for every point within the plane spanned by x and y. In this case the z-value is unique for each (x, y) pair. More z-values would only increase the complexity of the data model.

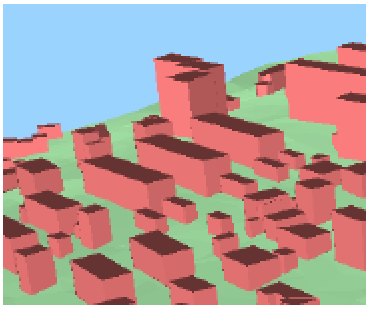
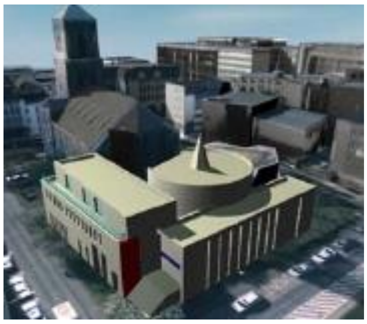
When it comes to the modelling of houses, the opportunity to create surfaces, which are orthogonal to the xy-plane must be provided. Otherwise it is not possible to model walls and other similar structures. This model can be extended by the ability to represent bulges. Accordingly, it must be possible to have multiple z-values for each point in the xy-plane. Such data still represents only surfaces without real volumes.

This will only be achieved by using "real" 3D, where solids are formed by the use of volume spanning primitives; e.g. cubes or tetrahedra. In this thesis the terms three-dimensional or 3D data always refer respectively to the definition of 2.8D or 3D data.

Levels of Detail

Another type to represent different spatial data is the handling of different *levels of detail* (LoD) of a model. In particular, the *CityGML* standard [Kolbe, 2009] defines different LoD for city and building models. These range from a simple Digital Terrain Model (DTM) to detailed house-models including their interiors. The following Table 2 shows the five different LoDs, which are used in CityGML:

Table 2 - Different LoD representations of city models according to CityGML [Kolbe, 2009]

Level of Detail	Example
LoD 0: regional model, landscape (2.5D Digital Terrain Model)	
LoD 1: block model without roof shapes and texture	
LoD 2: detailed model, differentiated roof shapes, with simple textures	
LoD 3: differentiated architectural model with detailed textures	

LoD 4: real object shape with interior model

In this case the different LoDs are related to city and building models that are managed with CityGML. But also other models, which are developed in a different context, use various approaches on handling different kinds of LoD. For instance, in the DFG research project *3DTracks* [Breunig et al., 2011] the handling of different LoDs in the context of CSG-based data[11] are studied using the example of 3D metro track models [Borrmann et al., 2012]. In this case the LoDs also increase with every step, but the procedure itself is completely different. The LoDs of a CityGML model represent a further generalization of the model with each lower LoD (LoD4→LoD3→…→LoD0, cf. Table 2). In contrast, each ascending LoD degree of a CSG model (LoD1→LoD2→…→LoD5, cf. Figure 4) is another refinement of the model itself. Consequently, the model is further developed with each LoD. The following Figure 4 illustrates the different LoDs of a metro track section modelled with CSG.

[11] Constructive Solid Geometry

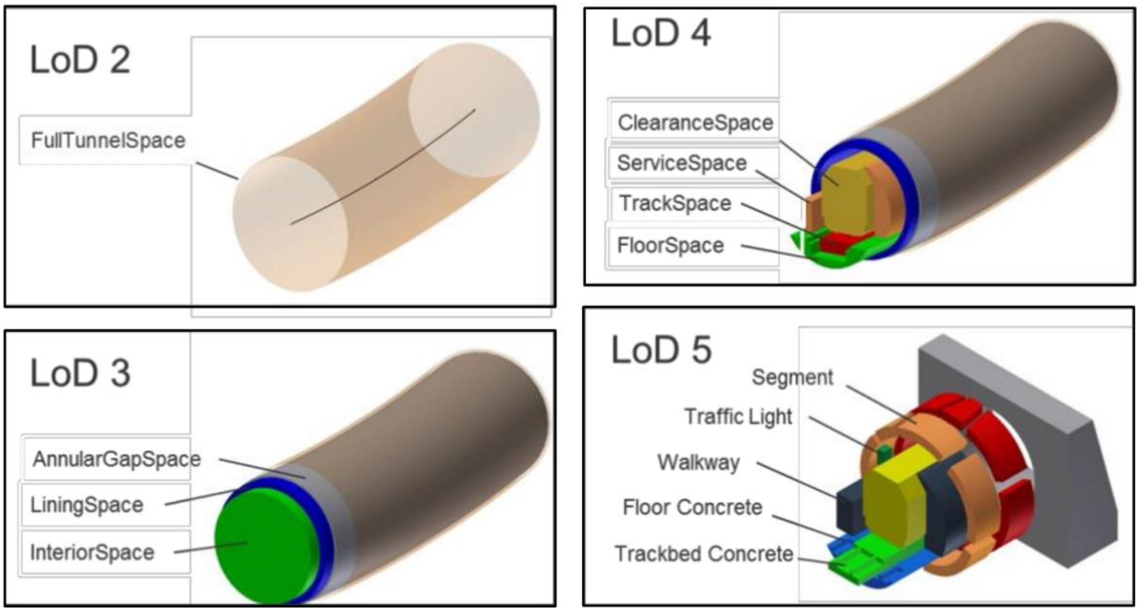


Figure 4 - Representations of different LoDs of a CSG model[12].
Source: [Borrmann et al., 2012]

Other data models also work with generalization levels. Even in conventional map applications different objects are highlighted or displaced depending on the zoom level or chart scale. For models that are developed based on simplicial complexes, it is possible to change the LoD by the use of particular methods, e.g. *progressive meshes* [Hoppe, 1996]. The model of simplicial complexes is introduced in Section 2.2. The idea of progressive meshes will be taken up again in Chapter 3.3.

Topology

Another important property of spatial and spatio-temporal data are topological relationships, e.g. neighbourhood relationships between geometric primitives. Often those primitives are not explicitly managed within a data model and have to be calculated *on-the-fly* if required. If there are such information available, e.g. about adjacent polygons, topological questions can be resolved. There are also data models that are specialized in topological issues and the management of topological information. A common example from geological modelling is the use of generalized maps (g-maps) [Lienhardt, 1991], [Lévy, 2000]. Due to a data structure of so-called darts, it is possible to use simple algorithms for processing such data. The g-maps are also useful for handling spatio-temporal data and have advantages for different request types in comparison with conventional data models [Butwilowski, 2015].

[12] LoD 1 corresponds to the line, which is represented in LoD 2.

2.2 Simplices and *D*-Simplicial Complexes

The development of 3D models can be distinguished into different approaches. As part of working with *computer-aided design* (CAD) software primitive *constructive solid geometry* (CSG) solids are used to construct 3D objects. These include, e.g. vehicles, engines or other artificial objects. However, in the case of natural forms it is most common to use the model of *simplicial complexes*, in particular for soil layers or any kind of natural surfaces [Mallet, 2002].

A simplicial complex is constructed by the combination of simplices. A *d*-simplex is the minimal object within the respective spatial dimension *d*. A formal definition is given by [Penninga, 2008]:

Definition 1 - Simplices

A d-simplex is the smallest convex set in Euclidian space $\mathbb{R}^m$ with $d \leq m$, containing $d + 1$ points $v_0, \ldots, v_d$ that do not lie in a hyperplane of dimension less than d.

Simplices in $\mathbb{R}^3$ that are suitable for the modelling of natural forms are illustrated in Figure 5.

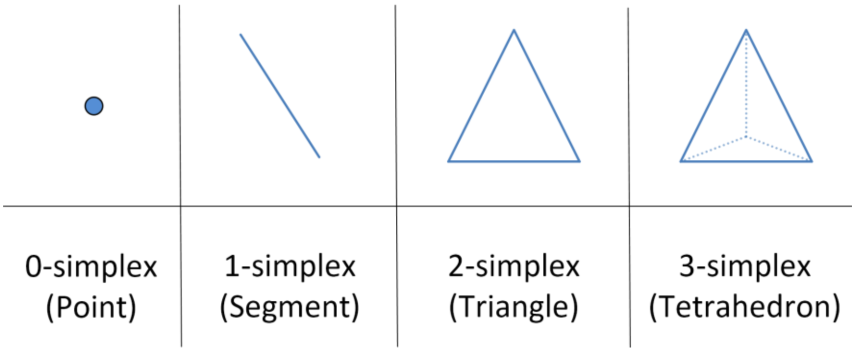


Figure 5 – *D*-simplices of dimension $d \in \{0,1,2,3\}$.

The use of simplicial complexes as a spatial data model was formally introduced by Egenhofer et al. A simplicial complex describes the composition of several simplices to a complex with the following properties [Egenhofer et al., 1990]:

A simplicial complex is a (finite) collection of simplices and their faces. If the intersection between two simplices of this collection is not empty, then the intersection is a simplex which is a face of both simplices. The dimension of a complex C is taken to be the largest dimension of the simplices of C.

Egenhofer et al. describe the model of simplicial complexes as a topological data model, which is suitable for the management of spatial data. The generic model was initially described for the use within a 2D space. Later Balovnev among others used this model also in 3D space [Alms et al., 1998].

Breunig defined faces of simplices and simplicial complexes for arbitrary dimensions as follows [Breunig, 2001]:

Definition 2 – Face of a simplex and simplicial complexes

Let P be a set of (d+1) affine independent points.

Let *conv(P)* be the convex hull of P and $P' \subset P$. Then $S := conv(P)$ is called a *d-dimensional simplex* and conv(P') is a face of S. The set of all faces of a simplex is called *face(S)*.

A set C of simplices is called *simplicial complex*, if

1. For each $s \in C$: $face(s) \subseteq C$
2. $s1 \cap s2 \neq \emptyset \rightarrow s1 \cap s2 \in C$

D-simplices are the simplest way to span a plane in the respective dimension *d*. The advantages of this model include the benefit to develop simple algorithms for geometric operations. Figure 6 shows two examples of simplicial complexes.

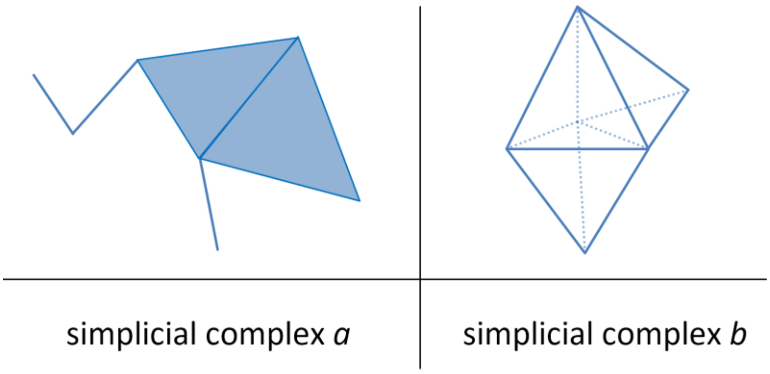


Figure 6 - Two examples of a simplicial complex.

[Worboys, 1992] proposes the term of a *d-simplicial complex* whereas all simplices of a simplicial complex are of the same dimension *d*. This follows the observation that the modelling of natural objects in most cases consist of simplices of just one dimension which form such a *d*-simplicial complex [Bär, 2007]. For instance, a 2-simplicial complex is a simplicial complex that consists exclusively of 2-simplices. In this thesis such *d*-simplicial complexes located in a three-dimensional Euclidean space are referred to as 3D model.

Based on these considerations [Worboys, 1992] developed an object model for an adaption of such data structures into a geodatabase. His object-oriented approach will be revisited in Section 2.4.

According to [Egenhofer et al., 1990] the properties of *d*-simplicial complexes include the following axioms, which are referred to as *completeness axioms*:

- completeness of incidence: the intersection of two d-simplices is either empty or a face of both simplices.
- completeness of inclusion: every *d*-simplex is a face of a (*d*+1)-simplex.

Thus the composition of a simplicial complex is examined in detail. *D*-simplices can share border regions which consist of (*d*-1)-simplices, i.e. corresponding to the case of two adjacent 2-simplices there exist only one shared 1-simplex, cf. Figure 7.

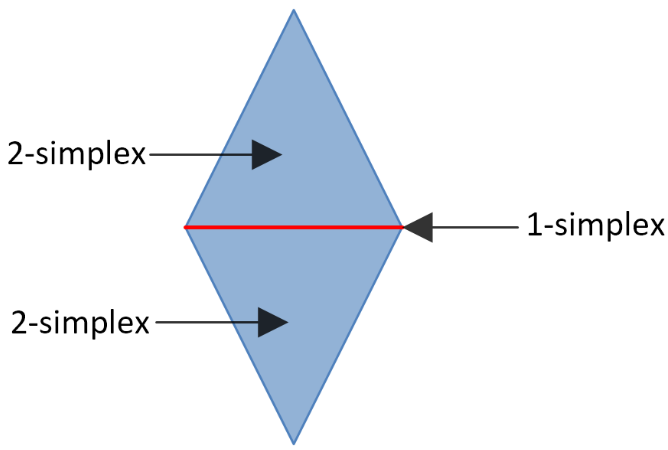


Figure 7 - Two simplices that share a segment and form a simplicial complex.

The model of simplicial complexes forms the foundation for the spatio-temporal data model of this thesis. Therefore, in the following section the concept of *time* is introduced in the context of spatial data. The model of *d*-simplicial complexes for spatio-temporal data is subsequently presented in detail in Section 2.3.1.

2.3 Spatio-Temporal Data

In this section various data models for the modelling of spatio-temporal data are presented and discussed. This begins by outlining the concept of *time* within this context. Subsequently, various concepts are presented, which can be generally applied for the management of spatio-temporal data and are irrespective of the spatial dimension or representation. Thereby, references to the processing of such data in a DBMS are made repeatedly. In Section 2.3.1 spatio-temporal data is described on the basis of simplicial complexes in 3D space.

In most cases spatial information describe a specific temporal condition. Conventional map applications, for instance, often describe the current status (at the time of preparation) of a certain region (e.g. the black forest) in a certain aspect (e.g. hiking map). But when modelling time-continuous phenomena (e.g. the development of Dubai within the last 50 years), *time* needs to be introduced as a new dimension and handled accordingly. Depending on the application there are significant differences in the handling of time.

For many applications, the data is located in 2D space and is based on points, lines and polygons that are expanded to include a temporal component [Güting and Schneider, 2005]. This model can be used to describe changes in conventional 2D map applications. For instance, such a model can be used to model the traffic of cars within a city.

But there are also several applications in 3D space where temporal changes need to be modelled accordingly. For instance, geological changes are modelled in continuous processes. However, when modelling the development of buildings or entire cities, discrete changes are in the focus of interest and have to be specified accordingly [Oosterom and Stoter, 2010].

Among others [Worboys, 1994a], [Güting and Schneider, 2005] studied the handling of time within spatio-temporal data. There are also studies on purely temporal database systems [Snodgrass, 1987], [Jensen et al., 1992]. However, according to [Roddick et al., 2004], the research field of generic, purely temporal data models is almost completely covered.

Referring to spatio-temporal data, the handling of time highly dependents on the data itself. Therefore, based on earlier definitions from the database community, [Worboys, 1994a] proposes to divide time into two separated dimensions: *transaction time* and *valid time*[13]. The *transaction time* describes the date of adding the data into an information system, whereas the *valid time* describes the time of the event, which is represented by the data.

Specific applications may add additional requirements to the management of time. It might be of interest, e.g. for construction projects to differentiate between a *planning time* and a *realization time*. In such a case the time is also managed in two separated

[13] Worboys uses both terms: *database* and *event* time [Worboys, 1994a] as well as *transaction* and *valid* time in [Worboys and Duckham, 2004]. In the following the latter are used.

dimensions. One to reflect the planning of structures and another to reflect the progress of construction. The handling of these issues is part of the DFG research project "3D-Tracks" [Breunig et al., 2011]. [Menninghaus et al., 2016] developed a spatial index which is based on the B⁺-tree to efficiently manage the two temporal dimensions in combination with the spatial data. An example of spatio-temporal data with two different temporal dimensions is shown in Figure 8.

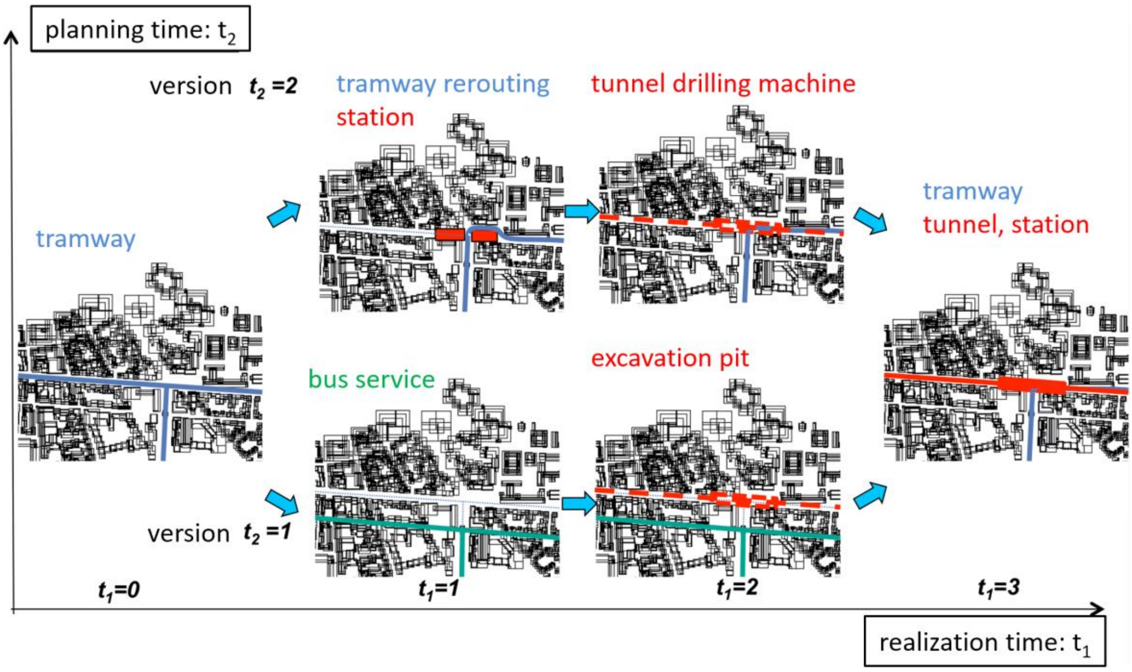


Figure 8 - Representation of the two temporal dimensions "planning time" and "realization time" based on test data of Karlsruhe, Germany.
Source: Andreas Thomsen, KIT (DFG research group "3DTracks")

Another important aspect of handling time is the treatment of *time intervals*. [Allen, 1984] specified various relationships of temporal events which were used by [Breunig, 2001] in the context of spatio-temporal data management. As a result, 6 important relations between two time intervals t_A and t_B were highlighted, cf. Figure 9.

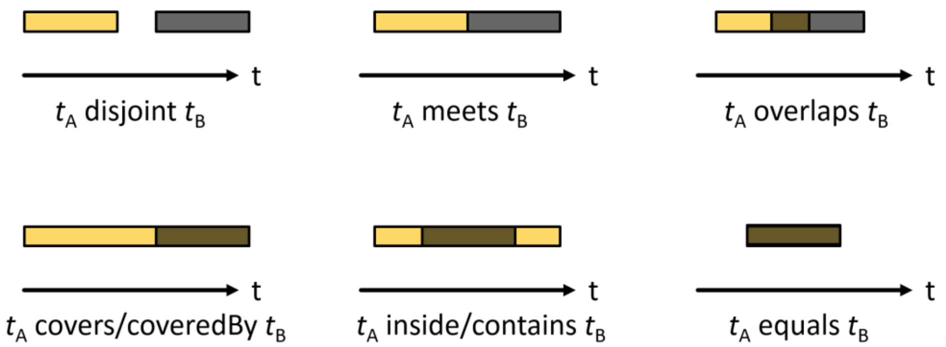


Figure 9 - Temporal topological relationships between two time intervals t_A and t_B.
Source: [Breunig, 2001]

On closer inspection of spatio-temporal data transformation, geometry, topology and attribute values can change in the course of time. Referring to such data, [Abraham and Roddick, 1999] describe 8 different states of temporal change, cf. Figure 10.

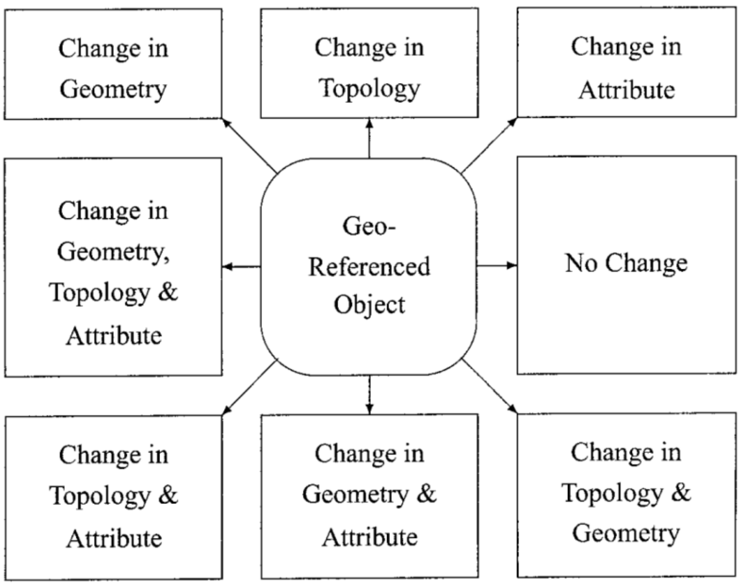


Figure 10 - Different states of temporal change based on a spatial object.
Source: [Abraham and Roddick, 1999]

[Breunig, 2001] comes to the conclusion that for geoscientific applications two different cases should be considered:

1. Time per attribute: various temporal changes of individual attributes of an object, e.g. the geometry of the object.
2. Time per object: temporal changes of the entire object including attributive data.

In the first case the geometry and other attributive data both have their own temporal function. In the second case, there is only one temporal function which is applied to all the different data types of the object. [Breunig, 2001] proposes to separate the changes in geometry and topology from the changes of thematic or semantic data.

Specific geographic information systems, so-called *temporal geographic information systems* (TGIS) were developed for the handling and preparation of spatio-temporal data. Such systems are called *spatio-temporal geoscience information systems* (TGSIS) if the processed data are of geoscientific origin [Le, 2014]. TGSIS focus primarily on objects that are composed of *d*-simplicial complexes and describe continuous phenomena. This topic is subject of the following section.

2.3.1 Simplicial Complexes

As mentioned in Section 2.2 the model of simplicial complexes is frequently used in the context of geoscientific applications for the modelling of natural objects and, in particular, soil subsurfaces and all kinds of irregular surfaces. In most cases these objects are composed of *d*-simplicial complexes, i.e. the elements of the complexes are all from the same dimension *d* [Bär, 2007]. Such models are developed with the help of geoscientific modelling tools such as GOCAD or GeoModeller.

If a continuous process is modelled, the spatial dimension is extended by the dimension of time. This section discusses the specific characteristics of spatio-temporal data that is managed by *d*-simplicial complexes. Thereby it is assumed that the spatial part of the data is located in a three-dimensional Euclidean space.

Different data management concepts have been developed with the target to manage objects based on simplicial complexes in a geodatabase. [Breunig, 2001], [Siebeck, 2003], [Worboys and Duckham, 2004], and [Le, 2014] give a useful introduction to this subject.

The process of modelling the continuous change of geoscientific objects must be addressed in an appropriate data model. Such a spatio-temporal object is from now on referred to as *4D object* or *4D model* and consists of several 3D models representing individual time steps:

$$m\ (t_n) = \text{3D model for time step } t_n$$

$$m_n := m\ (t_n) \times \{t_n\}$$

$$m\ (t_n) + m\ (t_{n+1}) := m_n \cup m_{n+1}$$

$$\text{4D model } m = \bigcup_{n \in \mathbb{N}} m_n = \sum_{n \in \mathbb{N}} m\ (t_n)$$

An example of such a 4D model is shown in Figure 11. This approach for the composition of spatio-temporal data is referred to as *snapshot model* [Langran and Chrisman, 1988], [Renolen, 1997]. In this case time is regarded as continuous and is dedicated to the *valid time* [Le et al., 2013].

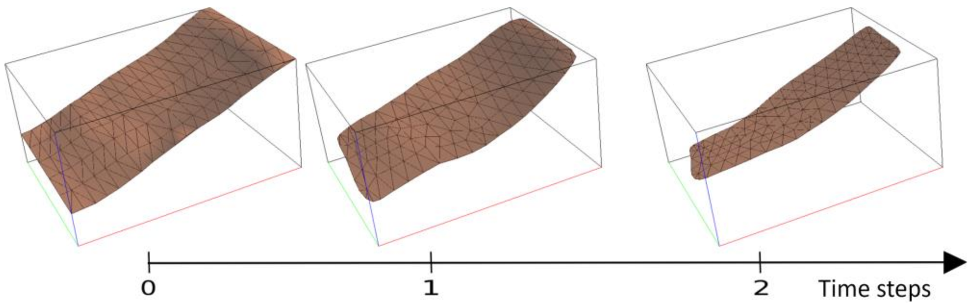


Figure 11 - 4D model composed of three 3D models which represent different time steps.

The following list describes the functionality that need to be covered by a data management model for such 4D models:

- Ability to compose a 4D model based on 3D models that correspond to different snapshots within the time series of the 4D model.
- Ability to extend the time series of an existing 4D model.
- Ability to query a specific state of a 4D model for a specific time $t_x \in [t_{start}, t_{end}]$.

[Worboys, 1994a] introduces the term *ST-simplex* for spatio-temporal simplex and thus describes a simplex that is a bi-temporal object for a transaction time and a valid time interval. He also defines the term *ST-complex*, which describes a collection of ST-simplices. Referring to Worboys such ST-complexes are subject to the following constraints:

1. All spatial projections of constituent ST-simplices must be distinct.
2. The spatial projections of constituent ST-simplices must form a simplicial complex.
3. Any face of a ST-complex must have at least as much temporal referencing as its parent.

Figure 12 illustrates a ST-simplicial complex that consists of 2-simplices.

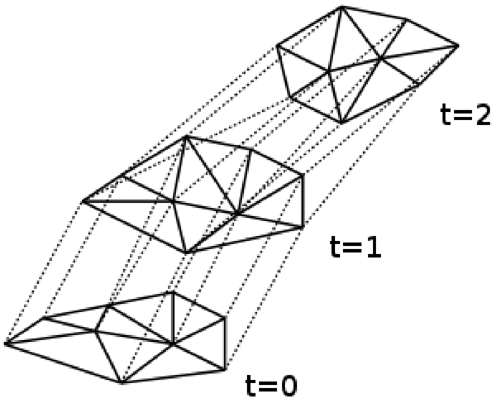


Figure 12 - ST-Simplex for three time steps.
Source: [Rolfs, 2005]

Continuous natural phenomena can be modelled by ST-simplices or ST-simplicial complexes, respectively. [Breunig, 2001] describes the changes of such objects over time and demonstrates three different stages:

- *4D with constant topology*: continuously changing geometry in time.
- *4D with constant geometry*: continuously changing topology in time.
- *4D without restrictions*: continuously changing geometry and topology in time.

Examples of different forms in geometry and topology together with the description of various properties like the equality of ST-objects can be found in [Breunig, 2001]. In addition to a spatial part, spatio-temporal models can consist of *thematic*, *semantic* and *metadata* that need to be handled accordingly.

2.3.2 Data Model Used within this Thesis

This section defines additional terms that are relevant for this thesis. The thesis at hand deals exclusively with spatio-temporal data that is located in a three-dimensional Euclidean space and is based on the model of *d*-simplicial complexes with $d \in \{0,1,2,3\}$, cf. Section 2.2.

In this thesis the following terms are treated synonymously:

0-simplicial complex = point cloud = 0D-net

1-simplicial complex = segment net = 1D-net

2-simplicial complex = triangle net = 2D-net

3-simplicial complex = tetrahedron net = 3D-net

Based on this, continuous phenomena can be modelled by the consolidation of 3D models. Each model corresponds to a single time step of a 4D model and consists of a *d*-simplicial complex, cf. snapshot model described in Section 2.3.1. Such time series are used to model geological phenomena, such as the continuous change of soil subsurfaces. Another example is the modelling of the transformation of surfaces, such as a landfill site.

In this data model time is treated as *linear time*. Therefore, time moves in one direction and has no branches or loops. The net topology, i.e. the mesh of a *d*-simplicial complex, might change between two consecutive time steps. Therefore, also major changes of the shape of 4D objects can be handled appropriately.

During the modelling of continuous changes of natural forms, parts of a certain region sometimes change more quickly than others. A good example for this is a mountain mass of which a partial region is under intense observation because of a particular volatile movement in that region. Figure 13 illustrates an exemplary monitoring of a region with such different volatile partial regions.

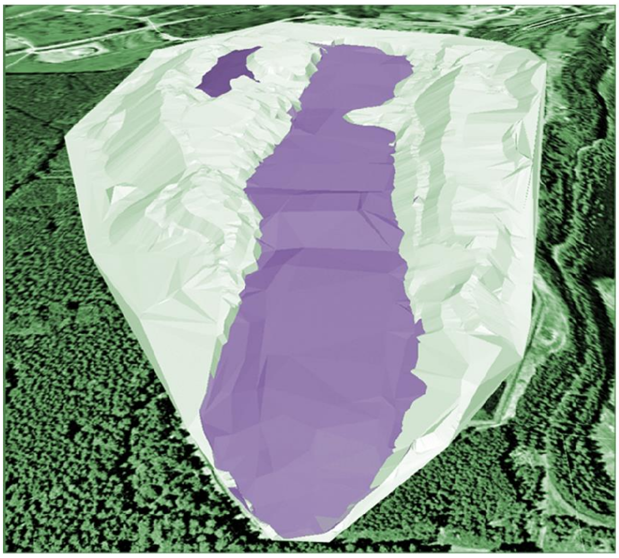


Figure 13 - Region with different volatile partial regions: purple = rapid movement, light green = slow movement.
Source: [Butwilowski, 2015]

Such structures need to be handled accordingly by a data management model. Each subregion represents one contiguous region with a common temporal discretization. The concept of *net components* with different temporal discretizations will be introduced in Chapter 3.2.

This thesis focuses mainly on the geometry of spatio-temporal data, but semantic and thematic data are also significant matters when modelling natural phenomena. During the examination of 4D models the following characteristics are observed:

- A 4D model consists of multiple time steps. Each represents the observed phenomenon at a specific point in time.
- Individual time steps are realized by the use of *d*-simplicial complexes.
- The temporal discretization of the model might be separated into multiple partial regions.
- Semantic or thematic data might be available and must be handled accordingly.
- The net topology might change during two consecutive time steps.

A list of requirements on a data management model for such spatio-temporal data is presented in Chapter 3.

Topological information

In the data management model to be developed later in this thesis, the topological information consist of the composition of single net elements, i.e. simplices and adjacent net elements within a simplicial complex. A single net element of a *d*-simplicial complex has up to d+1 adjacent net elements for $d \in \{1,2,3\}$. Within the data model of this thesis a 0-simplicial complex has no neighbour information. Individual *d*-simplices optionally consist of d+1 links to primitives containing the geometry and semantic or thematic information for its (d-1)-faces. The simplices of every dimension, together with their incidence relationships form the *incidence model* [Ohori et al., 2015]. Such primitives are referred to as *construction elements* within this thesis. An overview of the topological information of the data model is illustrated in Figure 14.

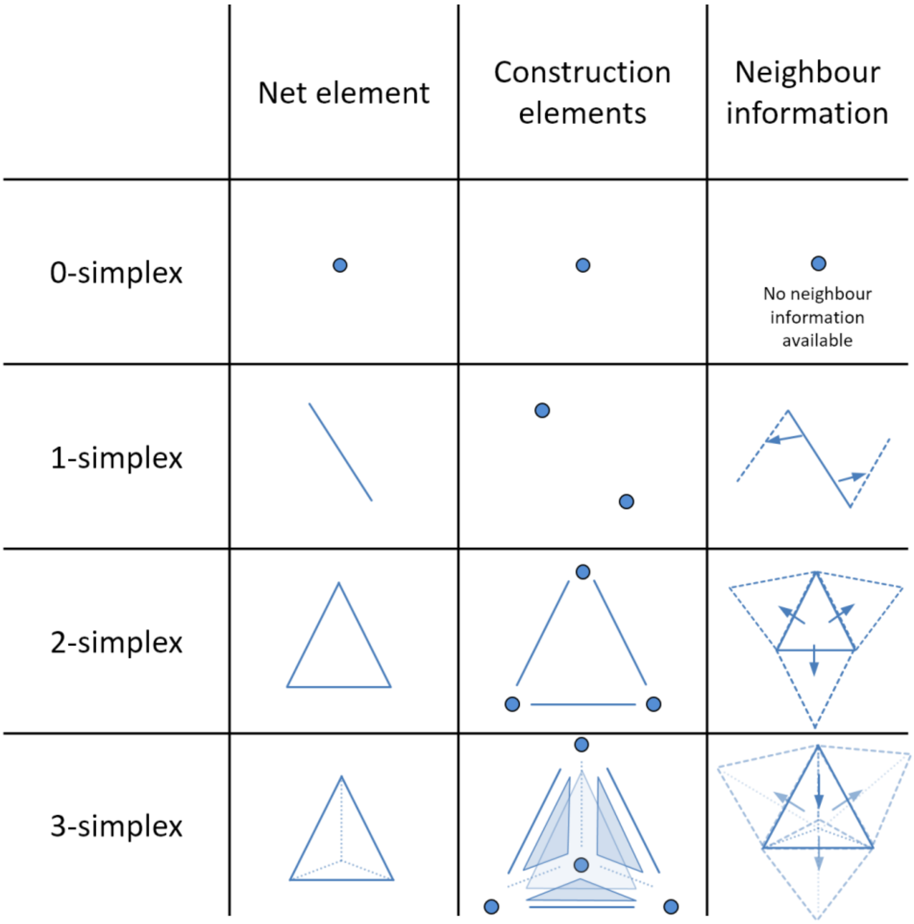


Figure 14 - *D*-simplices with construction elements and adjacent net elements.

2.4 Spatial and Spatio-Temporal Databases

In a database, manageable data consist of entities, i.e. distinguishable objects. These can be related to each other. Due to this structure the *entity-relationship model* is often used for the modelling of relational databases [Chen, 1976]. Entities and their relationships are mapped into a data model, which is subsequently incorporated into a DBMS.

A database has to meet several requirements. First of all, it must be possible to convert existing data into the data model of the database. The data must be clearly distinguishable and be prepared for various queries. The relationship of the individual entities with one another must be mapped accordingly. Afterwards, large amounts of data can be provided to multiple clients simultaneously. Data can be queried, modified, and new data can be added to the database. Depending on the capabilities of the database, routines for further processing of the data can be accessed and invoked.

Requirements for 2D and 3D geodatabases

Similar to conventional databases, the core of a geodatabase is the data model on which the spatial data is mapped. Such a data model is highly related to the particular application [Peuquet, 1984].

[Güting, 1994] describes general properties of a DBMS for spatial data as follows:

1. A spatial database system is a database system.
2. The data model and query language offer spatial data types.
3. It supports spatial data types in its implementation, providing at least spatial indexing and efficient algorithms for a spatial join.

This definition of a spatial DBMS is based on the collaboration with 2D geoinformation systems (GIS). In particular, the third item describes a standard functionality of conventional 2D GIS and does not concern the processing of 3D models that have their origin in geoscientific applications [Mallet, 2002].

In such applications neighbourhood information and the preservation of such topological attributes during the processing of the data is more likely to be in the focus. Also the handling of thematic-, semantic- and meta-data plays an important role when managing 3D or 4D data. Another important aspect is the availability of operations and the data itself via Web services.

Spatial and spatio-temporal databases face the challenge to map such information and meet these requirements. Referring to spatio-temporal databases, they can be divided into two distinct categories [Worboys and Duckham, 2004]: does the data model allow continuous changes, e.g. volcano monitoring, soil movements, etc., or movements such as cars or pedestrians? For the latter, a so-called moving-objects-database (MOD) can be used which is suitable for live applications [Wolfson et al., 2002]. This thesis deals exclusively with the first category.

When managing spatio-temporal data that describe a continuous movement in terms of the first category, there are similar requirements in comparison with spatial databases. But according to [Abraham and Roddick, 1999] new issues and problems arise when adding time as a new component. Thus, the following list summarizes *challenges* for spatio-temporal databases:

- Handling of large amounts of data, especially when dealing with data in 3D space.
- Simultaneous administration of geometry, topology and semantic data.
- Handling the complexity of 2.5D, 2.8D and 3D models + time.
- Dealing with different versions and histories.

- Distinction of planning processes (time = discrete) and geoscientific processes (time = continuous).
- Support of a large variety of client applications.

These challenges lead to the following *requirements* of spatio-temporal databases:

- Efficient storage of huge amounts of data, especially in 3D space.
- Facilities to handle the complex structure of 4D models.
- Ability to answer diverse spatial and spatio-temporal queries.
- Spatial- and/or spatio-temporal-operations on these data types shall be available.
- Facilities to manage temporal transitions of geometry (e.g. location, coordinates) and topology (e.g. discretization) of objects over time.
- Providing the data via a network interface as generic as possible.

It is proposed by [Abraham and Roddick, 1999] to develop specific data structures as part of a new and innovative model for the efficient management of spatio-temporal data.

Generally, the particular data model should be in the focus of development, since no solution can be developed that will simultaneously satisfy all requirements of different data models. This is especially true when the data models differ remarkably, such as CityGML and *d*-simplicial complexes.

2.4.1 Architectures

Referring to the prior work about the management of spatio-temporal data in spatio-temporal databases (STDB), there are various concepts which address parts of the discussed issues. A diversified overview of spatial-temporal databases based on different spatio-temporal data models is available in [Abraham and Roddick, 1999] and [Pelekis et al., 2004].

This thesis focuses on spatio-temporal DBMS for managing *d*-simplicial complexes, which change over time, cf. Section 2.3.1. There are different ways to manage these snapshots while avoiding the storage of redundant information. One popular approach is the use of a *differential mode* to calculate a specific time step with the help of deltas and an initial model [Abraham and Roddick, 1999]. This idea will be picked up again in Chapter 3.5.

Schäben et al. developed an approach for a spatio-temporal DBMS that is based on the object-relational geodatabase PostGIS[14] [Le et al., 2013], [Le et al., 2014], [Le, 2014], [Gabriel et al., 2015]. This approach is capable of handling spatio-temporal data based on the snapshot model and is highly related to the 3D modelling software GOCAD.

The geodatabase architecture DB4GeO is realized as an object-oriented approach [Thomsen et al., 2008], [Breunig et al., 2010]. Both approaches focus mainly on the management of 3D models, but also manage the composition and storage of simple 4D models.

Object-oriented approach

The focus on an object-based spatial data model, compared to field-based, is not necessarily preceded by an object-oriented implementation. However, [Worboys, 1994b] notes that for the management of spatial data the object-oriented approach is superior to the relational or object-relational approach. This is i.a. due to the complex and widely ramified structure of spatio-temporal data [Shekhar et al., 1997].

Among others, [Worboys, 1992] describes the advantages of the object-oriented approach in the context of simplicial complexes. Breunig compares relational with object-oriented databases in the context of mapping spatio-temporal data. As well as Worboys he comes to the conclusion that object-oriented DBMS are superior to relational DBMS, which is particularly related to the following properties [Breunig, 2001]:

- The ability to model spatial and non-spatial data within a unified data model.
- 1:1 mapping between database types and user-defined data types.
- Representation of the objects behaviour.
- Inheritance.
- Polymorphism.

Pelekis supplemented this assessment and points out the following advantages of an object-oriented approach [Pelekis et al., 2004]:

- A single object represents the entire history of an entity.
- Simple queries.
- Efficient temporal data handling.
- Uniform treatment of spatial and temporal data.

[14] PostGIS – an open source geodatabase, based on the object-relational DBMS PostgreSQL.

DB4GeO

DB4GeO [Breunig et al., 2009], [Breunig et al., 2010] has its roots in *GeoStore*, an information system for managing geologically defined geometries [Alms et al., 1998], [Bode et al., 1994], and in *GeoToolKit* [Balovnev et al., 2004] which were both developed at the University of Bonn within the scope of the Collaborative Research Center 350 [SFB350]. The primary focus of the development is on the management of spatial data, which is based on simplicial complexes. From the very beginning there was a strong focus on the communication with geological and geophysical 3D modelling tools [Breunig et al., 2000] such as GOCAD or IGMAS [Götze and Lahmeyer, 1988], [Schmidt and Götze, 1998]. The models that were handled by GeoStore and GeoToolKit mainly represented the structure of the Lower Rhine Basin [Alms et al., 1998].

While the two predecessors GeoStore and GeoToolKit were developed in the programming language C and C++, DB4GeO has been completely developed in Java. A historical overview of the development of DB4GeO is shown in Table 3.

Table 3 - Historical overview of DB4GeO

Time of development	Name of the Software	Programming Language	Innovation
1994	GeoStore	C	First approach of a 3D geodatabase based on a relational DBMS.
1997	GeoToolKit and GeoStore	C++	Upgrade to a self-developed object-oriented solution: GeoToolKit library
1997	GeoToolKit	C++	First spatio-temporal data model
1999	GeoToolKit	C++	Extension by a Delta Storage concept.
2002	DB3D	Java	Re-implementation of the geodatabase in Java.
2005	DB3D	Java	Implementation of a spatio-temporal data model based on ST-simplices.
2007	DB4GeO	Java	Exchange of the database kernel from ObjectStore to db4o. First REST-based Web services.
2010	DB4GeO	Java	Implementation of a spatio-temporal data model for 2-simplicial complexes based on Point Tubes.

The structure of the spatio-temporal database architecture is presented in [Breunig et al., 2010]. It was developed by the use of multiple layers, which depend on each other and fulfil different tasks. An overview of the structure of the 3D/4D geodatabase architecture DB4GeO is shown in Figure 15.

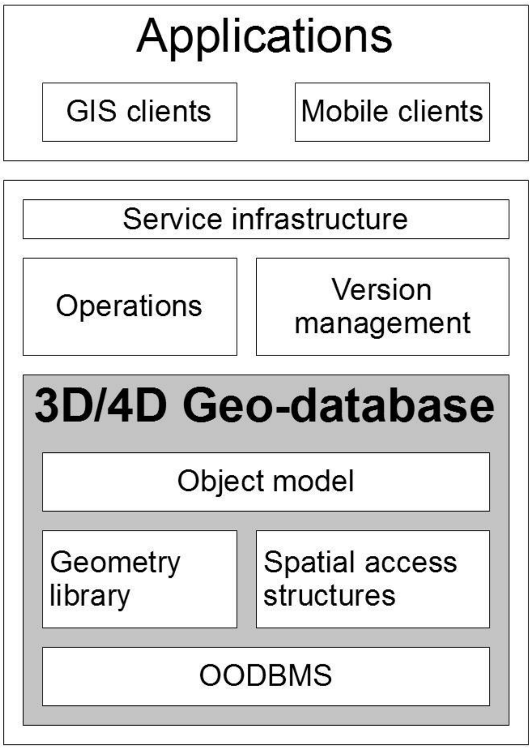


Figure 15 – Structure of the 3D/4D geodatabase architecture DB4GeO.
Source: [Breunig et al., 2009]

The object-oriented DBMS *db4o* [Paterson et al., 2006] handles the persistent data and forms the basis of DB4GeO. On this, a geometry library was built which provides basic 3D geometries. Here the model of *d*-simplicial complexes is supported. There are spatial access structures available, e.g. the R*-Tree [Beckmann et al., 1990] for an accelerated access to the spatial data. Furthermore, DB4GeO provides an object management, spatial operations, as well as a version management. Spatial data can be queried via a service infrastructure [Breunig et al., 2013b].

[Rolfs, 2005] developed a 4D model to add a support for spatio-temporal data. This model provides the ability to manage time series that consist of simplicial complexes. The development was a first attempt to handle 4D objects in DB4GeO. Therefore, the model lacks some important features and has the following drawbacks:

- Once inserted a time series cannot be extended.
- The topology of a 4D object cannot change between consecutive time steps.
- The vertices of an object are stored redundantly at every net element.

- The temporal discretization of a 4D object must be uniform within the entire net.

The 3D kernel of DB4GeO provides a good basis for an implementation of the of spatio-temporal data management model that is part of the outcome of this thesis.

Additional functionality (spatio-temporal operations and access methods)

The management of spatio-temporal data in a spatio-temporal database does not stop at the pure mapping and storing of the geometry itself. So there are other tasks such as the construction of spatial and spatio-temporal indices, providing spatio-temporal operations or visualization, which must be met by a spatio-temporal DBMS [Abraham and Roddick, 1999].

Especially, the development of various spatial access methods (SAM) and spatio-temporal access methods (STAM) represents an important component of a spatio-temporal DBMS, as they accelerate the access to the data significantly. A good entry point to this research field also for multiple time dimensions can be found in [Menninghaus et al., 2016].

2.5 Web Service Fundamentals

The dissemination of data via a network interface is the primary task of a Web service. Furthermore, some Web services offer capabilities to alter existing data and to integrate new data into a system. The invocation of operations is another task that can be handled by a Web service. Geodata should be accessible via the Internet to work, e.g. on distributed systems or with multiple clients on a shared database. From the perspective of client applications, such data infrastructures usually act like a central system. Internally, the data is often stored redundantly and in several separated locations to avoid data loss and gain performance [Ramakrishnan and Gehrke, 2002].

Currently, there are no Web services standards available that focus on spatio-temporal data based on the model of d-simplicial complexes. However, there are some previous achievements in the field of Web services in general and also Web services, which are specialized on spatial data.

In the following basic terms and general concepts for the exchange of data via a network interface are introduced. Subsequently, two Web services standards of the OGC are described: the *Web Feature Service* and the *Web Processing Service*. Both were developed for the exchange of spatial data and for the invocation of analysis functions on spatial data, respectively.

The following terms are used within this thesis when referring to the structure of Web services:

URL

A Hostname (e.g. kit.com) and path (e.g. /departments/index.html) that specifies the location of a resource.

Entity-Body

Actual content of a request, e.g. text.

Idempotence/statelessness

Each call of a specific URL always leads to the same result. Therefore, write operations (such as write, update and delete) should be avoided.

For instance, in the HTTP the operations GET, HEAD, PUT, DELETE and OPTIONS are declared as idempotent. The operation POST, however is not idempotent [Daigneau, 2011].

Resource (or representation)

A *resource* is everything that is clearly identifiable on the Internet, e.g. a text, an image or a 3D model.

Focus information

Information about the focus of the request. In most cases, it is the location of the resource that is retrieved, or to be changed. The term was, among others, introduced and used by [Richardson and Ruby, 2007]. Such information can be located in the entity-body (e.g. SOAP Web services) or in the URL (e.g. REST-based Web services).

XML

Markup language that was specified by the *World Wide Web Consortium* (W3C) for the exchange of data via the Internet. XML is frequently used within various Web services that communicate with a large variety of client applications. Due to the use of XSD schemata files, the structure of an XML file is defined. Such a schema can then be implemented by client applications that are involved, and subsequently the transferred data can be validated automatically. The handling of XML data is simplified by several software libraries[15]. For instance, these provide a parser for the further processing of XML data. The format is *human readable* and can be opened and interpreted by a simple text editor.

Historical origin of Web Services for spatial data

A Web service being composed of several stacked layers to transport various data addressable, securely and well-regulated over a network interface forms the basis for the exchange of geospatial data. This includes a standard for the network access: e.g. *Ethernet*[16], which is located at the lowest level of the protocol stack. An alternative to Ethernet is the *wireless local area network*[17] standard. Both network access standards can form the basis for the *Internet Protocol* (IP), which is used to address data packages over a network. Based on this, there is the *Transmission Control Protocol* (TCP) for the transmission of messages and packet streams. Based on this protocol stack different application protocols are composed, which determine rules for coding the actual messages. One of the most popular application protocols is the *Hypertext Transfer Protocol* (HTTP) which is a document-based protocol designed i.a. for the transfer of web pages on the Internet. The two main features of this protocol are *addressability* and *statelessness*, cf. [Richardson and Ruby, 2007]. According to [Tilkov, 2009] due to its simplicity and distribution "no other application protocol [...] is superior to HTTP regarding to the interoperability features". Many high level Web services are closely related to this protocol. For a more detailed introduction [Richardson and Ruby, 2007] is highly recommended.

Almost all Web services for spatial data use this protocol at least as the basis for the bare transport of the data. An overview of the protocol layers is presented in Figure 16.

[15] Selected examples of XML libraries for Java: JDOM (http://www.jdom.org), Dom4j (http://dom4j.github.io), Xerces Java Parser (http://xerces.apache.org/xerces-j)
[16] Ethernet network standard IEEE 802.3 - http://www.ieee802.org/3/
[17] WLAN network standard IEEE 802.11 - http://www.ieee802.org/11/

Layer	Protocol
Application	HTTP
Transport	TCP
Internet	IP (IPv4, IPv6)
Data link	Ethernet

Figure 16 - Overview of a typical protocol stack.

Based on these fundamental protocols there are multiple approaches that are used to develop Web services for various spatial and non-spatial applications. Popular approaches are *SOAP*, *RPC* and *REST*. SOAP and RPC each form an intermediate layer between the actual Web service and the communication protocols. REST on the other hand uses the basic request methods GET, PUSH, PUT and DELETE provided by HTTP [Alonso et al., 2004].

2.5.1 Architectures

With REST and SOAP two popular general approaches are briefly introduced. With the Web Feature Service (WFS) and the Web Processing Service (WPS) two explicit Web services that are specified for the handling of spatial data are presented.

Representational State Transfer

Representational State Transfer (REST) was originally developed by [Fielding, 2000] and is a software architectural style for the transmission of data between client and server, which is subject to specific rules and concepts. This style pays special attention to the term *stateless*. Each request from a client to a server must include all information needed for a specific request. This approach excludes the common principle of so called *sessions* in which the response to requests of a client depends from information about the client. In return, the setup of a cache[18] is significantly simplified [Fielding, 2000], which can be beneficial for Web services that handle spatio-temporal data.

[18] Cache – A component that efficiently provides frequently requested data.

The implementation of this approach is based on a certain use of the HTTP operations *GET*, *PUT*, *POST* and *DELETE*. The four characteristics of a REST compliant services are *addressability*, *statelessness*, *connectedness* and a *uniform interface*, which are realized by the concepts of *resources*, *URIs*, *representations* and the *references* between the resources among each other. While many Web services are technically classified as *RESTful*, a Web service, which implements the true philosophy of REST is strictly resource-oriented. This is the case when the focus information is located in the URI of the request [Richardson and Ruby, 2007].

SOAP

SOAP is similar to REST an architectural style for the development of a Web service and can be used for the exchange of geospatial data [W3C SOAP]. Compared to a Web service that was built according to the rules of the REST approach, the focus information in a SOAP Web service is not included in the URL, but is located in the *entity-body* of a request.

Most SOAP Web services are considered as RESTful from a purely technical perspective, but for [Richardson and Ruby, 2007] these are considered as "bad" Web services, because the handling differs massively from the "normal" Web behaviour. For instance, the HTTP request methods GET, POST, PUT and DELETE are not used in the form they were originally intended. Using SOAP, the exchange of data as well as the invocation of operations is subject to certain rules, which are usually described as an XML document in the entity-body rather than guided by the request methods. In this case HTTP is just used as a protocol for the pure transportation of the data [Tilkov, 2009], while it is an inherent part of actual REST Web services, as method and focus information is included in the HTTP request and can be processed without processing the entity-body.

With SOAP and REST there are two different architecture styles for a specific implementation of Web services, which are also described by the OGC [OGC Web Service Common] or, in the case of REST, discussed for the application of Geoweb services[19].

[19] OGC RESTful Service Policy Standards Working Group - http://www.opengeospatial.org/projects/groups/restfulswg

OGC Web Feature Service

At the time of the preparation of this thesis the valid specification of the OGC Web Feature Service (WFS) standard is version 2.0. Since this version is already supported by several client applications (e.g. *QGIS*[20], *GeoServer*[21]) as well as realized in an ISO standard (ISO 19142) this version is presented and discussed in the following.

Following the guidelines of the OGC an implementation of a WFS must be able to handle at least three different requests [OGC WFS]:

GetCapabilities

> Query of the capabilities and the existing records of the WFS. Displays general information about the WFS and of available feature types of the specific service. This includes the attribute *ServiceIdentification* which describes the service using predefined keywords and also provides the version of the WFS specification.
>
> Furthermore, information about *ServiceProvider*, *Operations-Metadata*, *FeatureTypeList* and *Filter_Capabilities* are provided. The information is constituted in an XML document.

DescribeFeatureType

> Request of information about a specific feature type. Here the name, the geometry type and any additional attributes incl. their data types are described. In conventional GIS applications a feature type often corresponds to a layer.

GetFeature

> This request returns the actual data. It is possible to include various parameters to restrict the result set. For example, to solely query all features with the attribute material = sandstone or the feature with ID = 42. The results are often returned in a GML format. However, other return formats are also valid.

Requests to a Web Feature Service can be queried by the use of a *Key Value Pair* (KVP) encoded URL. The following URL shows a *GetFeature* request:

```
http://hostname:port/path/wfs?service=wfs&version=2.0.0&
request=GetFeature&typeNames=namespace:featuretype
```

[20] QGIS - An open source geographic information system.
[21] GeoServer - A java based open source geoserver with the capability of handling OGC Web services requests.

This example includes four possible parameters, but only the "SERVICE", "VERSION" and "REQUEST" parameter are required.

Alternatively, the request can be submitted by the use of an XML document via HTTP POST, which is sent to the server. An example of a *GetCapabilities* request message encoded in XML is:

```
<GetCapabilities service="WFS" xmlns="http://www.opengis.net/wfs"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://schemas.opengis.net/wfs/2.0/wfs.xsd"/>
```

The role of SOAP and REST when referring to a Web Feature Service

When sending XML requests via HTTP POST to a server, according to the OGC Web Feature Service specification, optionally the SOAP 1.2 standard can be implemented and used. In such a case, the appropriate XML documents are embedded in the entity-body of a SOAP envelope [OGC Web Service Common]. Therefore, SOAP is only used for the transmission of the XML document. The server response also uses a SOAP envelope which includes the regular XML document. Thus, SOAP is only used solely for the transport of the data.

Furthermore, in 2011 the OGC has launched with the *RESTful Service Policy Standard Working Group* (SWG REST), a working group to develop a RESTful style for OGC Web services in general[19].

OGC Web Processing Service

The *Web Processing Service* provides GIS functionality in the form of various operations and geospatial analysis functions via a Web service. They include simple geometric operations such as buffer or distance functions. These operations are related to spatial data, which must be specified in the context of a request. The actual procedure of such an operation takes place in the backend and is usually processed by a GIS [Neteler et al., 2012]. The result set of the operation may vary depending on the operation itself. An example would be a *Boolean* value (true/false) for an intersect query. If the result of an operation is a geometric object, this is usually returned in a GML format. However, the return value can be of any format.

According to the OGC, an implementation of a Web Processing Service requires at least the following three operations [OGC WPS]:

GetCapabilities

Query of the capabilities of the respective WPS. Displays general information about the WPS and about available processes of the service. Such information is provided within an XML document.

DescribeProcess

Describes a specific process of the WPS. Here, also input and output parameters are specified.

Execute

Request of a specific process. During the request also appropriate input parameter must be specified. The result can be returned in any format, but GML is usually used for the description of geometries.

The specific kind of the processes that must offered by a WPS is not further defined by the OGC. Therefore, the developer of a WPS defines his own processes accordingly. The procedure of the actual analysis function is processed on the server side and is in many cases forwarded to a GIS software.

Requests are handled either via GET request to a specific URL or by transmitting an XML document to the WPS server via a POST request.

2.6 Mobile GIS and Location Based Services

In recent years, the field of *mobile GIS* and *Location Based Services* (LBS) experienced major changes. Among other things, the development of new mobile devices and new mobile operating systems, have strongly supported this trend. In particular, these include the development of modern tablet computers and smartphones, which experienced a strong growth with the introduction of the iPhone in 2007 and the mobile operating system Android in 2008. Due to this development modern smartphones and tablets are equipped with powerful processors and a large variety of sensors. These include sensors such as *gyroscopes*, *accelerometers* and *GNSS receiver* [Chen and Guinness, 2014]. Therefore, it is quite easy to determine the location of such a mobile device. Those conditions and the distribution of these devices in a broad mass market have generated the opportunity for an intensive research in this field, cf. [Shi et al., 2009], [Li and Jiang, 2011], [Meek et al., 2013], [Milanes et al., 2014], [Shitkova et al., 2015].

There are plenty of LBS applications available in different commercial application markets for smartphones and tablets. Most of them respond to database queries such as "Where is the next supermarket, cinema, bar or hotel?". Regularly they provide additional information about how to get there by giving direction instructions and a matching path on a map. There are several examples of research application developments such as *HotelFinder* and *BarFinder* [Rinner et al., 2005]. These LBS are developed to answer location based queries and provide the user location based decision support. Therefore, they use GIS analytic functions, spatial queries and map visualization techniques. [Raubal, 2011] describes how these applications affect the mobile life of its users. Another LBS application that is based on Android was developed by [Kang et al., 2015]. It focuses on a simple development approach for an application about the locations of classrooms.

In addition to popular LBS applications the group of mobile GIS primarily targets professional users. As well as conventional desktop GIS, mobile GIS offer tools for the *collection*, *presentation* and *analysis* of spatial and spatio-temporal data. They have been developed in several variations [Frank et al., 2004], [Menninghaus, 2010], [Deininger, 2012], [Kuper et al., 2014], [Zajíčková, 2015], [Han et al., 2015]. Each solution has its own advantages and there is a rapid development of new solutions. One important aspect for mobile (GIS) applications is a suitable user interface. [Raubal and Panov, 2009] describe a formal conceptual model for an automatic mobile map adaptation which is supposed to reduce the cognitive load for the user of such applications. An abstract model called *Adaptation Model* to reduce the amount of information visible to the user at a particular time was established. Rinner et al. presented a first approach of a suitable user interface for mobile GIS [Rinner et al., 2005], [Rinner, 2008].

[Doyle et al., 2010] examined the advantages of using multi-modal interfaces for mobile GIS. As a result, the mobile GIS application *CoMPASS* (Combining Mobile Personalized Applications with Spatial Services) uses a multi-modal interface to provide an intuitive way of interacting with a mobile GIS. This was achieved by the use of voice recognition and gestures rather than a keyboard and/or a pen. One of the addressed problems was the restricted display resolution and the sensitivity of the user interface for touch events. MAPDD is a modern mobile application that was particularly developed for surveying geodata on public transport [Zajíčková, 2015].

There are also professional solutions for mobile GIS developed by ESRI, Disy, and GeoTech Systems called *ArcGIS for Mobile*, *Cadenza Mobile*, and *GeoTechMobile*, respectively, cf. Figure 17.

Figure 17 – Selection of recent mobile GIS applications.

In most cases current mobile GIS and LBS applications work with 2D data, but there are also prototypical implementations available for 3D data [Dittrich, 2012], [Urban et al., 2013]. Such applications are subject of ongoing research [Runder Tisch GIS e.V., 2014].

3 A Comprehensive Data Management Model for Spatio-Temporal Data

Within this chapter a data management model is developed that optimizes the handling and storage of spatio-temporal data. The structure of the model is subdivided into five separate concepts, which are part of Sections 3.1 to 3.5. Within these sections, the adjustments of the respective concepts for the handling of *d*-simplicial complexes with dimensions $d \in \{0,1,2,3\}$ are described in detail. Subsequently, the consolidation of the individual concepts into a comprehensive data management model is presented in Section 3.6. This includes a technical realization and a detailed description of the internal data structure. The effects of the consolidation and potential limitations are discussed in Section 3.7.

For the modelling of continuous phenomena, huge amounts of spatio-temporal data are generated. A 3D model that represent a single time step of a 4D model usually consists of more than 100.000 net elements [Hu et al., 2010], [Xing et al., 2015]. Since a 4D model consists of multiple of such time steps, there are high demands on the storage capacities, cf. Chapter 2.3. Operations such as translations, interpolations or intersections between models, require lots of computing power and need particular hardware. For these reasons, it is necessary to avoid redundant calculations. The requirement of storage capacity must be kept low in order to relieve the storage medium itself, but also to accelerate retrieval requests and the transfer of data. An appropriate data management model must relate to such issues. The following list summarizes the requirements of such a data management model:

List 1 - Requirements of a model for the management of spatio-temporal data

- Support of various spatio-temporal applications.
- Optimization of calculation processes and avoidance of unnecessary operations.
- Optimization of storage demands.
- Foundation for the implementation of advanced spatio-temporal analysis functions.
- Linking of individual 3D discrete time steps into a continuous 4D model (3D + time).
- Option for an interpolation between time steps.
- Option for a differentiated storage of thematic or semantic information on geometric primitives.
- Handling of transitions of the net topology between time steps.

Five concepts are presented to meet these requirements. The model is based on various prior work, which is partly used in diverse spatio-temporal databases, cf. Chapter 2.4. Three of the five concepts primarily deal with the optimization of *geometric information* of a 4D model. Such information includes:

- The coordinates of vertices, particularly over time.
- The net topology (meshing) of the respective net, particularly over time.
- The management of additional thematic or semantic information that are related to construction elements.

The *Point Tube* concept separates the net topology from the vertices of a *d*-simplicial complex [Kuper, 2010], [Breunig et al., 2013a], [Breunig et al., 2016]. These vertices are managed in so-called Point Tubes, a special time dependent data structure. This concept is part of Section 3.1. Its impact on different net dimensions is described in detail.

When continuous movements of structures are modelled within a 4D model, subregions might behave more volatile than others. To satisfy such a process within a data management model, so called *net components* are introduced in Section 3.2. Due to this concept a division of 4D models in subregions with different temporal resolutions (i.e. discretization of time steps, cf. Chapter 2.3.2) are modelled appropriately.

A special occurrence in the management of 4D models, which are composed of individual time steps, is a change of the net topology, i.e. the meshing of the model. Such a change occurs, for instance, when the model is developed anew for a given time step [Lautenbach and Berlekamp, 2002]. Sometimes the mesh is refined in order to reflect certain aspects of changes. In Section 3.3 a concept is presented that is based on the ideas of [Polthier and Rumpf, 1994] and is capable of handling such net topology changes the by the use of *pre-* and *post-objects*.

Referring to geoscientific applications, the management of thematic and semantic data is as important as the management of geometric and topological data. Therefore, a data management model should avoid any restrictions in that matter. For this reason, in Section 3.4 a concept is introduced that enables the potential to store thematic or semantic data at various locations: at net elements (i.e. 0-3D-simplices) of *n*D-nets as well as at construction elements[22]. Thereby a maximum of flexibility is ensured.

[22] 1-2D-simplices, depending on net dimension.

In Section 3.5 the *Delta Storage* concept is introduced at which successive time steps are examined for similarities to reuse such parts across multiple time steps. This concept is i.a. based on prior work by [Strathoff, 1999] and [Siebeck, 2003].

Some of the sub-concepts have already been applied on *d*-simplicial complexes of specific dimensions. In such a case, the respective concepts are extended and the adjustments and effects for an arbitrary dimension $d \in \{0,1,2,3\}$ are described. An overview of the data management model, which consists of five parts is shown in Figure 18.

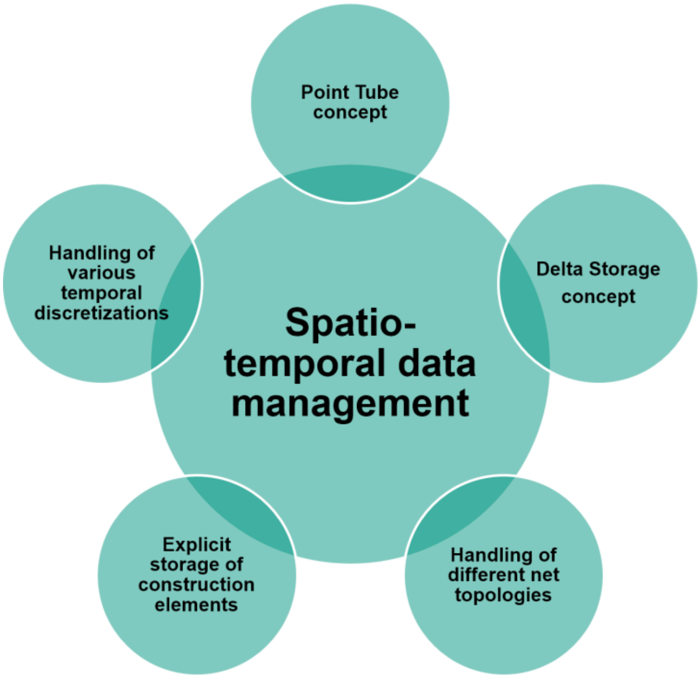


Figure 18 – Data management model for spatio-temporal data composed of five sub-concepts.

The consolidation of the five sub-concepts into a comprehensive model is described in detail within Section 3.6. This includes a description of the interaction of the concepts with one another and also the particular implication and effects of each concept within the consolidation. Subsequently, a comprehensive data management model is developed as a result of this union. It is analyzed if all the identified needs are met accordingly, cf. List 1. It must be ensured that certain constraints and conditions are met or specified, respectively. Such constraints are described and discussed in Section 3.7.1. The result forms the foundation for the implementation of the data management model which is part of Chapter 5.3.

Finally, the impact and benefits as well as potential problems and limitations of the model are presented and discussed in Section 3.7.

3.1 Point Tube Concept

The spatio-temporal data management model of this thesis describes the data handling for continuous processes, e.g. a continuous movement, cf. Chapter 2.3.2. Referring to [Worboys, 1994b], such a process is represented by discrete time steps. In the temporal dimension these time steps form the support points of a 4D model, cf. Chapter 2.3. Accordingly, each time step of a 4D model is a 3D model that consists of an amount of d-simplices which form a d-simplicial complex with $d \in \{0,1,2,3\}$. Partly due to improving recording methods such 3D models consist of vast amounts of data [Hu et al., 2010], [Xing et al., 2015]. When storing or processing such data, an optimization of the performance and the reduction of storage requirements is of fundamental importance.

A concept that addresses this problem is the *Point Tube concept*. The concept is based on the *completeness axioms* described by [Egenhofer et al., 1990], according to which vertices are unique within a simplicial complex. This fundamental idea is also used by the *indexed-face-set procedure* which separates the management of vertices from the corresponding mesh of a 3D model. The procedure is i.a. used by various common data formats, such as the *OBJ* format and the *VRML* format [Carey and Bell, 1997]. This kind of separation was used by [Siebeck, 2003] and [Le, 2014] in the context of managing spatio-temporal data within a geodatabase.

Due to the separation of the vertices from the net topology, the following benefits occur:

1. Under certain conditions, the net topology (mesh) of a model can be reused across multiple time steps.
2. Vertices which are used by multiple elements are stored uniquely.

Based upon this separation the vertices of a net are managed in a special data structure, the so-called *Point Tubes* [Rolfs, 2005], [Kuper, 2010], [Breunig et al., 2013a], [Breunig et al., 2016]. Thus, the movement of these vertices in space is clearly assigned and can be tracked over time. For an appropriate response to a request of a specific time step of the 4D model, the net topology is combined with the Point Tube data structure. An appropriate 3D model can be created for any point in time, once the date is within the time interval of the 4D model. For this propose, the vertices for the result object are calculated by the use of interpolation of all involved Point Tubes.

Hitherto the concept of Point Tube data structures has been studied for 4D models, whose time steps consist of 2D-nets [Breunig et al., 2016]. As part of this thesis the concept is adapted and transferred for the use with point-, segment- and tetrahedron-nets, i.e. d-simplicial complexes with $d \in \{0,1,3\}$. In the following, the details of this transfer and the specific impacts on the particular net dimension are described individually for each net dimension.

Realization for 0D-nets

The realization of the Point Tube concept for 4D models that are based on point clouds (0D-nets), may not reflect benefits at first glance as savings in the storage of the data are not expected. The information of each vertex within a point cloud must be stored and processed individually. There is no net topology that needs to be stored and might be reused. The individual points within a mesh cannot be used multiple times, since there is no existing mesh connectivity. Therefore, a reduction of the storage requirements by the multiple use of vertices is not possible. This also implies to the expectation of possible performance benefits.

Nevertheless, the realization of the Point Tube concept for such 4D models has another added value. Due to the use of the special data structure the following information is implicitly available:

- the unique assignment of individual vertices across multiple time steps.

Due to the structure of a Point Tube each point is traceable unambiguously across multiple time steps. As a result, the interpolation between two time steps of such a net is clearly simplified, cf. Figure 19. This benefit exists regardless of whether the interpolation between the individual time steps is linear, quadratic, cubic or works by another similar procedure.

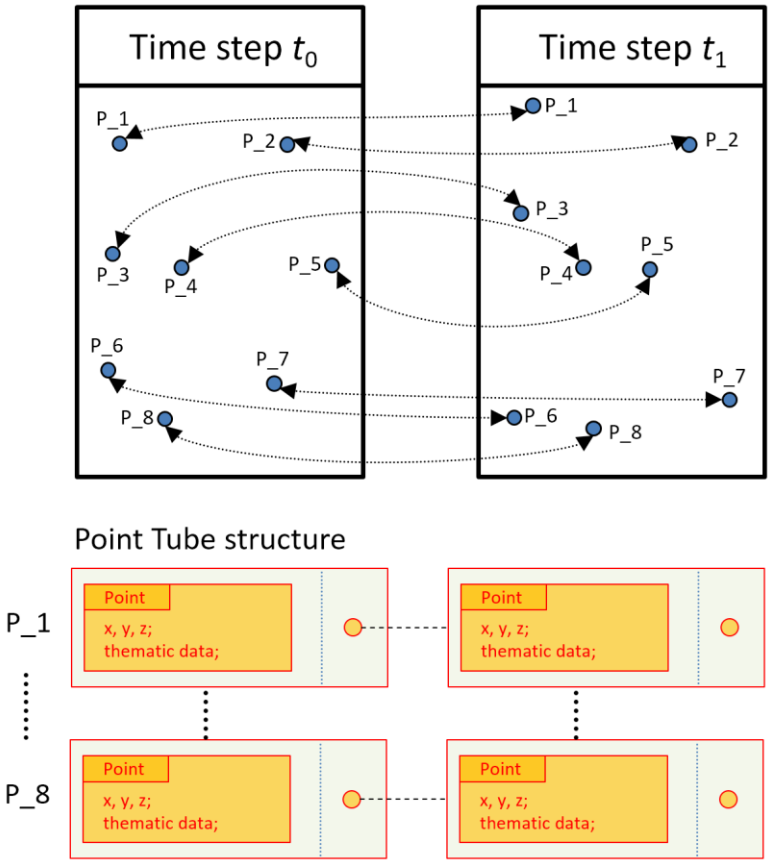


Figure 19 - Two time steps of a Point cloud that are managed in a Point Tube data structure.

A realization of the concept of Point Tubes for point cloud based 4D models is realized as follows:

Storing the initial model m (t_0) for time step t_0:

$$\text{model } m\ (t_0) = \text{unique vertices for time step } t_0.$$

Storing an additional time step m (t_{n+1}):

$$\text{model } m\ (t_{n+1}) = \text{unique vertices for time step } t_{n+1}$$

The stored coordinates of each time step are interrelated in a Point Tube structure and therefore uniquely assigned, cf. Figure 19. In this case, the handling of the net topology top (t_n) is omitted, since such a net topology does not exist for 0D-nets.

Realization for 1D-nets

Within the data model of *d*-simplicial complexes, segment nets consist of connected segments. Individual segments are connected to *i* = {0,1,2} neighbouring segments, cf. Figure

14. There are, e.g. no star-shaped formations. When transferring the Point Tube concept to the handling of segment nets, the storage of redundant information is already avoided with the first time step. Instead of storing the shared vertices of two adjacent segments twice, such a vertex is stored uniquely and is referenced by the particular segments. A model m (t_n) for a specific time step t_n managed by the Point Tube concept is composed as follows:

$$\text{model } m\ (t_n) = \text{net topology } top\ (t_0) + \text{unique vertices for time step } t_n$$

In this case the net topology consists of (maximal) dual associated segments. For queries of a specific state of the 4D model, the vertices, handled in Point Tube data structures, get linked with the net topology. Another saving potential appear in the management of time series. When adding more time steps to an initial model m (t_0), the Point Tube structures are extended with each time step, see Figure 20.

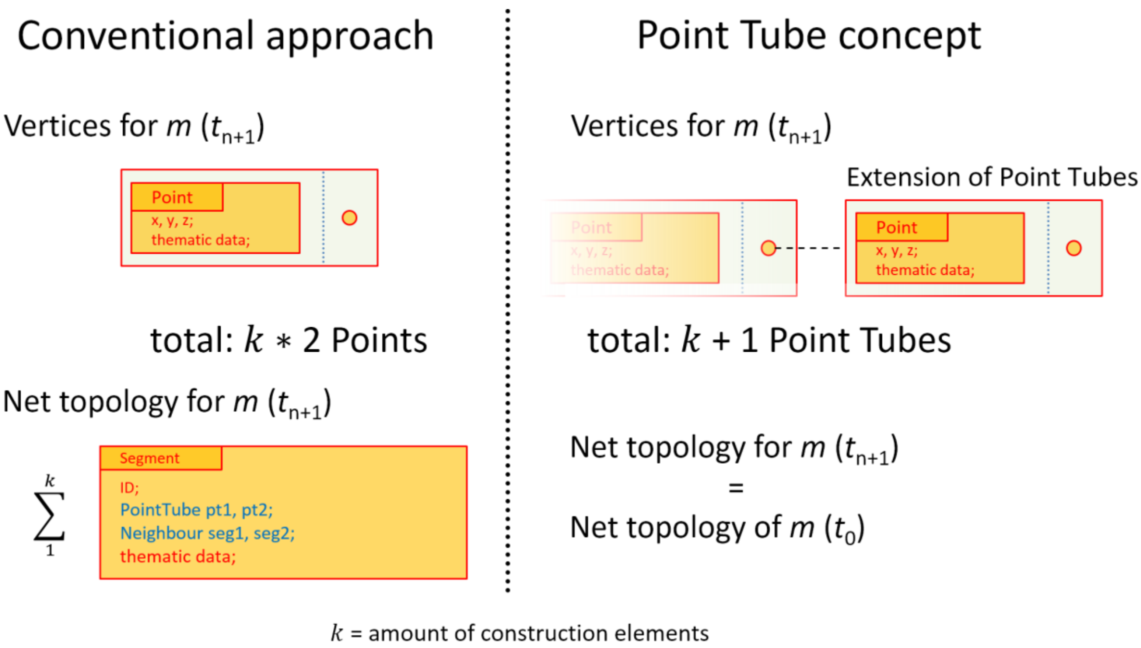


Figure 20 - Handling of a new time step m (t_{n+1}) based on a 1-simplicial complex. Comparison of the Point Tube concept to a conventional approach.

Therefore, the following information is stored for a model m (t_{n+1}) at a given time step t_{n+1}:

$$\text{model } m\ (t_{n+1}) = \text{net topology } top\ (t_0) + \text{unique vertices at time step } t_{n+1}$$

Due to this structure, both the storage costs of the individual time steps as well as the number of interpolation operations are reduced. Vertices used by two segments must be stored and interpolated only once during an interpolation operation. The net topology is applied to the result of the interpolation, and thus the interpolation process is completed. The type of interpolation, e.g. linear, quadratic or cubic is independent of this procedure. Again, every individual vertex of the net is traceable via all time steps of a 4D model.

Realization for 2D-nets

The triangles of a 2D-net consist of references to corresponding Point Tubes. Thus, the vertices are used for multiple triangles. The net topology *top* (t_0) is stored for the first timestep t_0 of the 4D model. When adding additional time steps, the existing Point Tubes get extended. It is not necessary to store information about the net topology, cf. Figure 21.

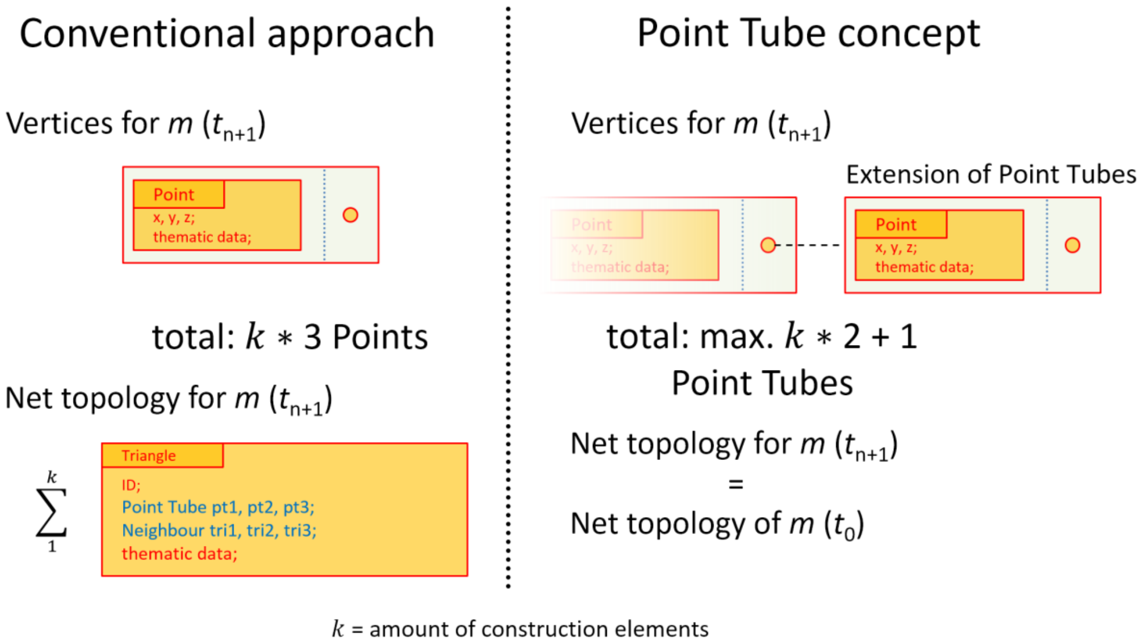


Figure 21 - Handling of a new time step m (t_{n+1}) based on a 2-simplicial complex. Comparison of the Point Tube concept to a conventional approach.

Due to this data structure the storage requirements of a model m (t_n) for a specific time step t_n are reduced to the following information:

$$\text{model } m\ (t_n) = \text{net topology } top\ (t_0) + \text{unique vertices for time step } t_n$$

Each model of a subsequent time step of this time series is stored as follows:

$$\text{model } m\ (t_{n+1}) = \text{net topology } top\ (t_0) + \text{unique vertices for time step } t_{n+1}$$

Since the information about the net topology is already present at time step t_n, it is re-used, which leads to the first part of the storage reduction. The Point Tube data structures provide the vertices, which are valid at the particular time step. The combination results in the model $m\,(t_{n+1})$. Figure 22 shows an example for the time steps t_0 and t_1.

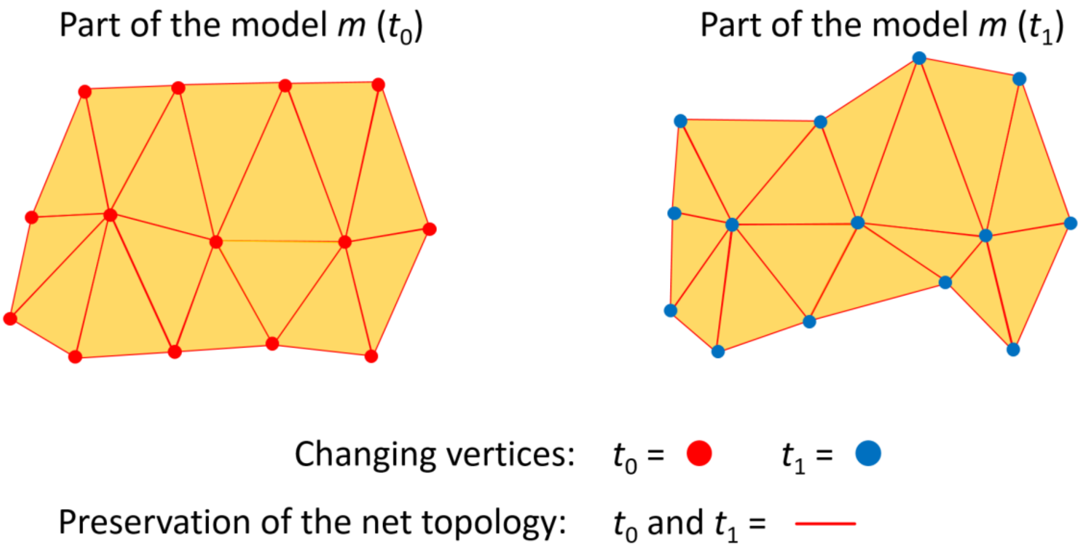


Figure 22 - Obtaining the net topology over time with the concept of Point Tubes.

The second part of the storage reduction is realized due to the multiple use of vertices within the net. Instead of storing three individual points for each triangle, the net topology is used to detect points which are used by multiple net elements simultaneously. Such points are handled in a unique Point Tube and are subsequently used for calculations or retrieval requests, cf. Figure 23.

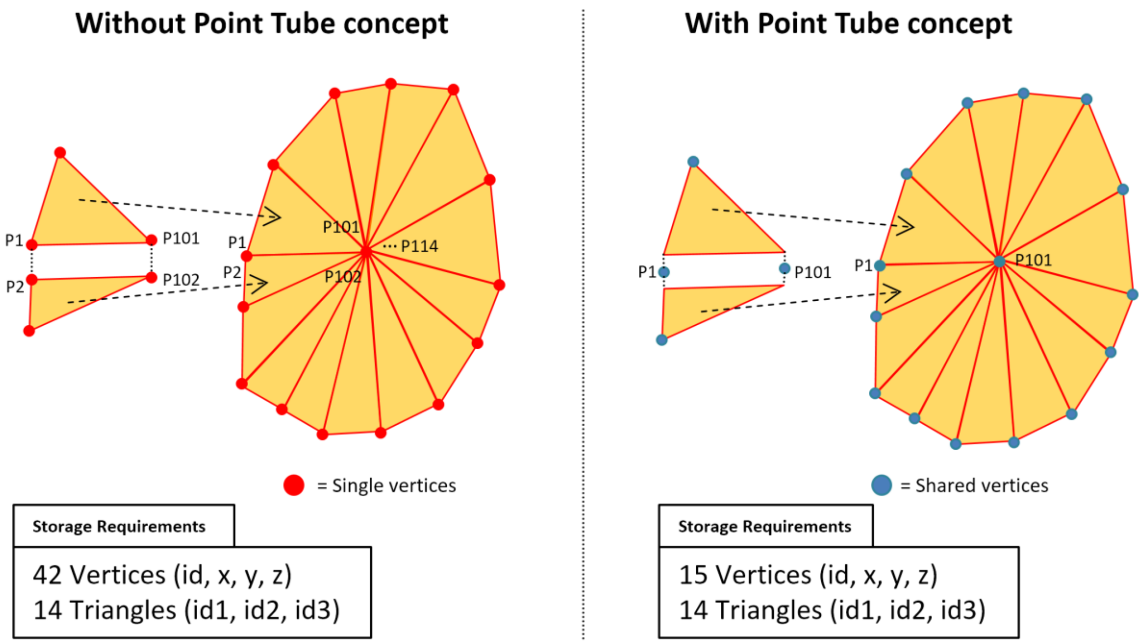


Figure 23 - Use of the Point Tube concept - comparison for a triangle fan with 14 triangles.

In addition to the two effects described above, the performance of several operations and analysis functions is increased: once the vertices of individual triangles are used in a calculation the use of corresponding Point Tubes massively reduces the amount of computations. In the example shown in Figure 23 the points P_{101} to P_{114} of the triangle fan are handled individually if the vertices are stored directly at the triangles. The use of the Point Tube concept leads to a single calculation for the vertex P_{101}. An important calculation is the interpolation between two time steps $m\ (t_n)$ and $m\ (t_{n+1})$. In such a case all existing vertices are interpolated. The use of Point Tube data structures accelerates such an interpolation enormously. The uniquely stored net topology is applied to the result of the interpolation of Point Tubes to complete the composition of the model.

In addition, this concept has a positive effect on the consistency of the net topology. In the alternative scenario of the example in Figure 23 all 14 individual points need to be calculated individually. This can lead to calculation errors. Such errors can produce gaps within a triangle mesh, cf. Figure 24. This is avoided by the use of Point Tube data structures.

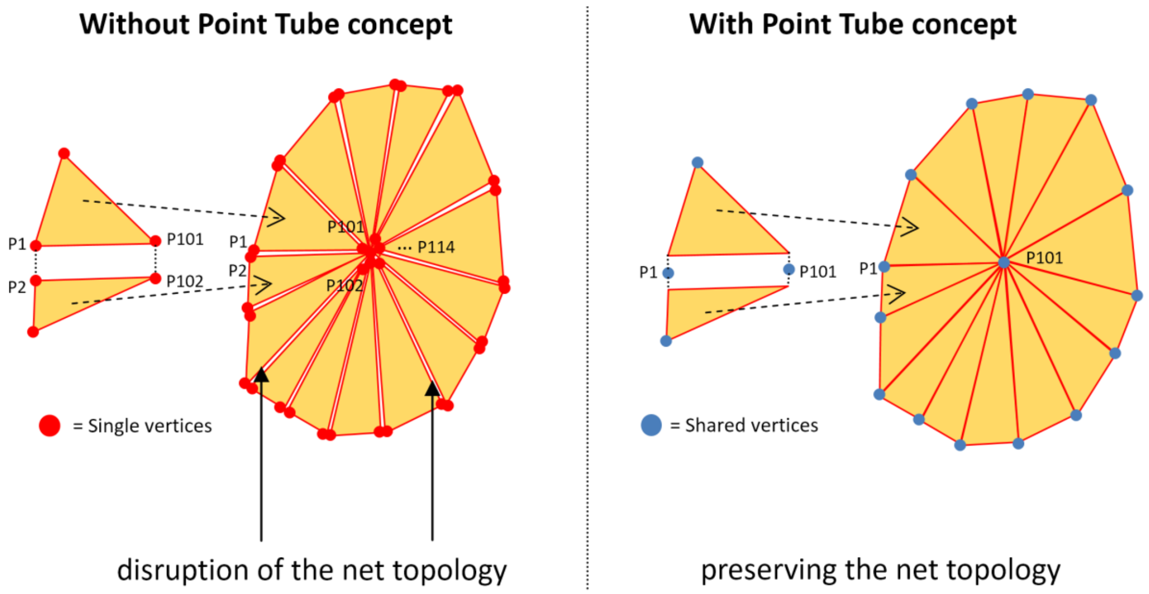


Figure 24 - Different handling of vertices. Left: small errors in displacement lead to a disruption of the net topology; Right: preservation of the net topology due to the Point Tube concept.

Realization for 3D-nets

In a final step, the Point Tube concept must be transferred to 4D models whose individual time steps consist of 3D-nets, i.e. 3-simplicial complexes.

Instead of storing four points at each individual net element (3-simplex), all vertices of the net are stored in Point Tube data structures. Individual net elements are based on references to these Point Tubes. The information about the net topology is managed independently from the vertices and is applied to each time step of the 4D model, as long as the net topology does not change. By joining the vertices and the net topology, a specific state of the 4D model can be developed *on-the-fly* at any time.

In addition, the maintenance of the net topology is enhanced, since computational errors, which might occur during analysis operations, do not tear off the net topology. Figure 25 illustrates the difference between a conventional approach and the Point Tube concept for tetrahedron nets.

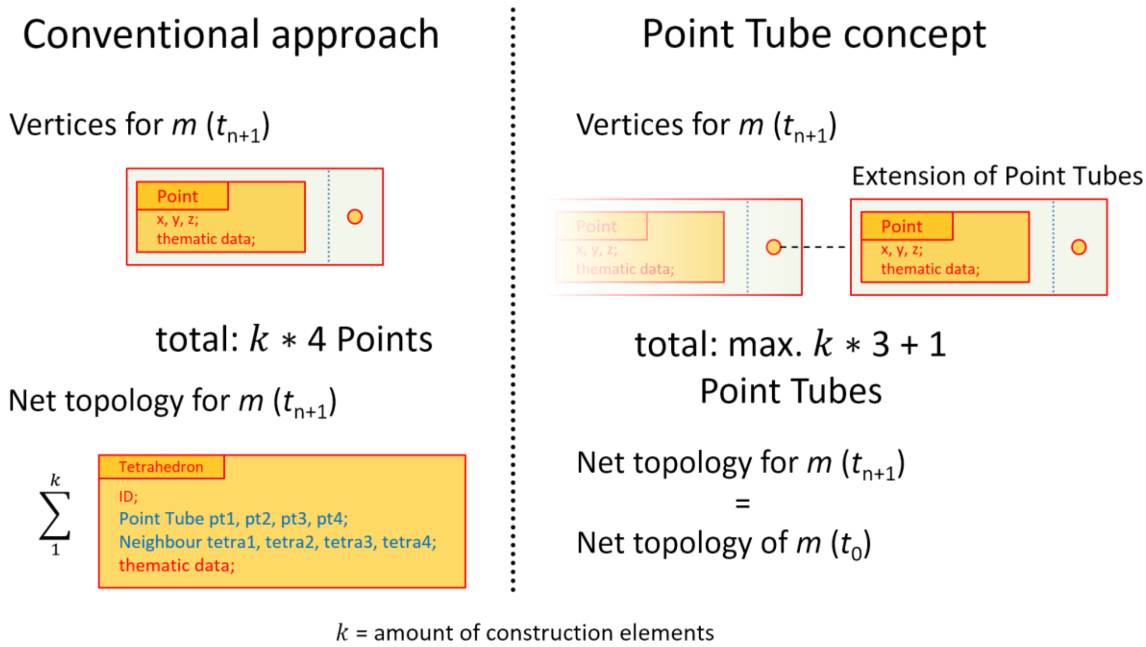


Figure 25 - Handling of a new time step m (t_{n+1}) based on a 3-simplicial complex. Comparison of the Point Tube concept to a conventional approach.

The basic idea to separate vertices from the net topology when managing tetrahedron nets is already beneficial for a single time step, i.e. a 3D model. Thereby the storage of redundant information is avoided. A model m (t_0) at a specific time step t_0 is managed as follows:

$$\text{model } m\ (t_0) = \text{net topology } top\ (t_0) + \text{unique vertices at time step } t_0$$

If additional time steps are added to such a 4D model m the net topology of the model is reused across several time steps. Accordingly, only the new coordinates of the vertices of the new time step are added to the Point Tubes. These are extended with the vertices of the new time step. The procedure is analogous to the handling of 2D-nets. Therefore, a model m (t_{n+1}) of a time step t_{n+1} consists of the following information:

$$\text{model } m\ (t_{n+1}) = \text{net topology } top\ (t_0) + \text{unique vertices at time step } t_{n+1}$$

The information about the net topology top (t_0) is only stored initially or for any change of the net topology and is reused in subsequent time steps. The interpolation between two successive time steps is also simplified, since the vertices used by multiple tetrahedrons are handled uniquely. The net topology is applied to the result of the interpolation of all Point Tubes, and thus the interpolation process is completed. The type of interpolation whether linear, quadratic or cubic is independent of this behaviour. For every net dimension there are certain constraints that must be met for a correct use of the Point Tube concept. The description of such constraints is part of Section 3.7.1.

In summary, the concept of Point Tubes was adapted and transferred without restrictions onto all relevant dimensions of *d*-simplicial complexes. When managing 0D-nets (point clouds) the saving effect due to the reuse of the net topology is eliminated. Additionally, there are no redundant information that can be reduced for 0D-nets. Figure 26 shows an overview of the internal structure of the realization of the Point Tube concept for all net dimensions.

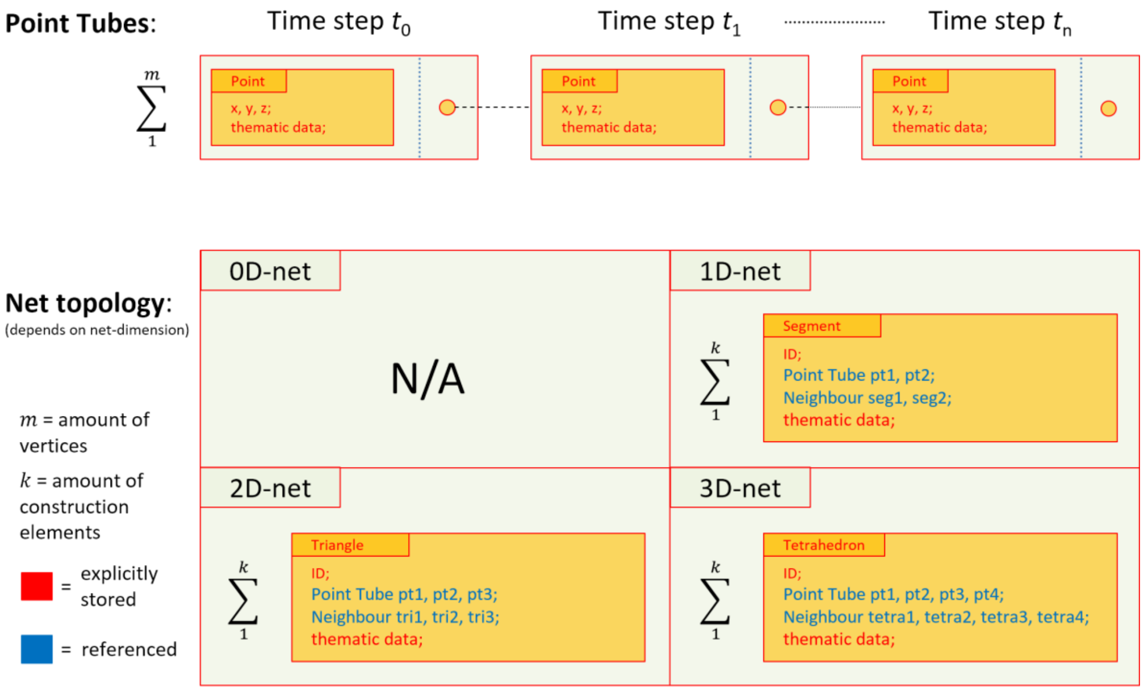


Figure 26 - Overview of the data structure for a realization of the Point Tube concept for 4D models.

3.2 Handling of Different Temporal Discretizations

In this thesis 4D models reflect continuous processes. Such a continuous process is realized by 3D models which refer to individual time steps, cf. Chapter 2.3. Complex 4D models may consist of subregions, which have different temporal resolutions. For instance, when modelling a continuous movement of a glacier, a sub-region is on a slippery slope. The monitoring of such a subregion would be intensified to map the rapidly changing movement within this particular area. Thus, the temporal resolution has increased for this region, cf. Figure 27.

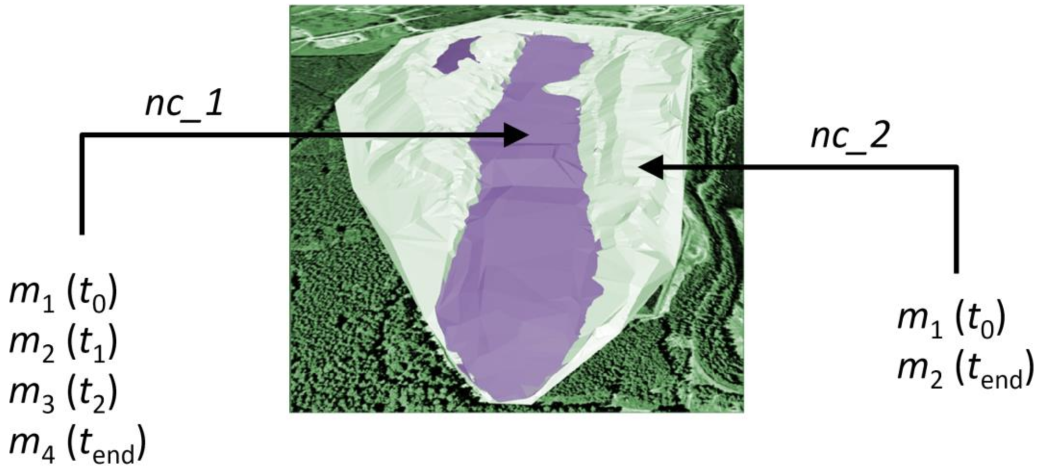


Figure 27 - Different temporal discretizations for subregions of a 4D model.

Within the data management model the ability to handle parts of a 4D model with different temporal resolutions must be provided in order to handle the geometric data of such processes. This is achieved through the introduction of so-called *net components* [Kuper, 2010]. Therefore, contiguous regions with the same temporal resolution are summarized within a net component. The entire 4D model is composed by a combination of all involved net components. From an outside perspective, the 4D model changes continuously as intermediate steps can be interpolated. For the usage of net components certain conditions must be met:

List 2 - Conditions for the usage of net components

- Within a net component, solely the geometry changes, the net topology remains unchanged.
- The net topology at the border regions of two adjacent net components must remain consistent over the entire time interval of the 4D model.
- The time interval $[t_0, t_{\mathrm{end}}]$ of all net components of a 4D model must be uniform.

So far this concept has been studied and implemented for spatio-temporal data based on triangle nets, i.e. 2-simplicial complexes [Kuper, 2010]. Within this thesis the concept is transmitted to other net dimensions that are relevant for the handling of geoscientific spatio-temporal data.

Realization for 0D-nets

Since the concept of different time discretizations for subregions was initially developed in combination with the Point Tube concept, the transfer to handle 0D-nets is trivial. The vertices of each 0D-net component consist of Point Tubes that represent the temporal resolution of the particular component, see Figure 28.

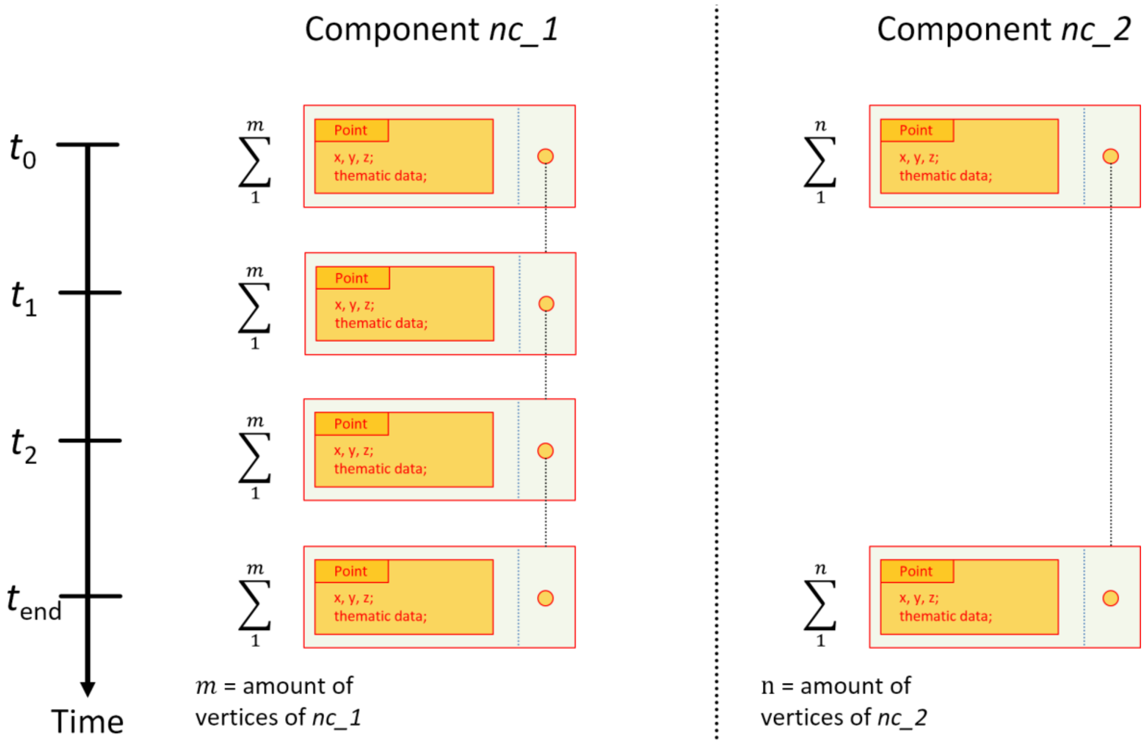


Figure 28 - Internal structure representing the management of a 0D-net, which is composed of two components with different temporal resolutions.

Since there exists no net topology, the Point Tubes from the border region of a 0D-net component can be of the same temporal resolution as the corresponding component these points belong to. There are no overlapping regions and different net components are completely separated from each other.

Realization for 1D-nets

The concept of different temporal discretizations for net components can be applied to the management of 4D models that consist of 1D-nets, i.e. segment nets. If parts of such a model have a different temporal discretization, these parts are aggregated within a new net component. The conditions, associated with the border regions of the net components, must also be met within this case, cf. List 2. Figure 29 shows a 1D-net that consists of two net components $\mathrm{nc\_1}$ and $\mathrm{nc\_2}$ with different temporal resolutions.

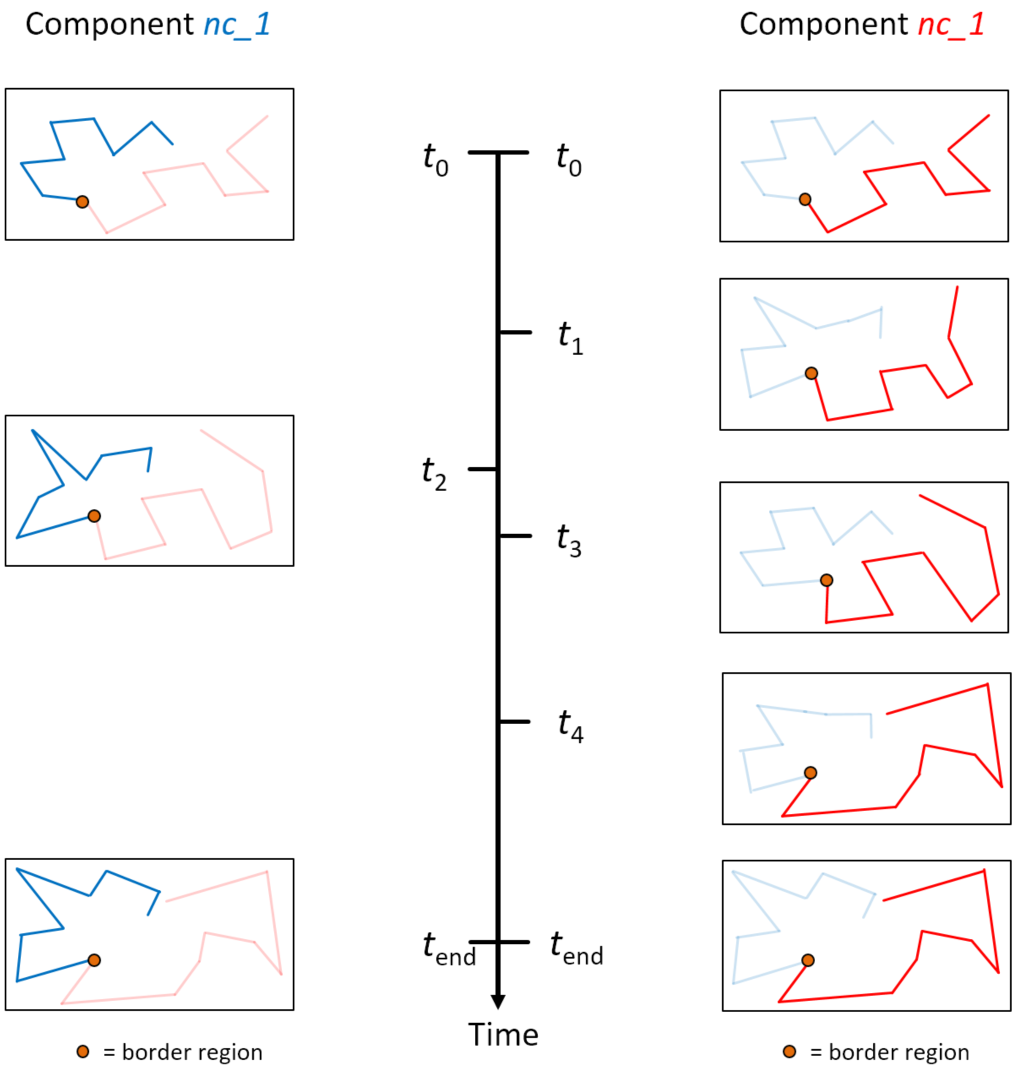


Figure 29 - 1D-net consisting of two net components with different temporal resolutions, i.e. net component *nc_1* with 3 time steps, net component *nc_2* with 5 time steps.

Realization for 2D-nets

Beforehand, the concept of net components with different temporal discretization was already combined with the Point Tube concept. Point Tube data structures belong to a particular net component and share their temporal resolution.

The border regions of such net components must be handled with special attention. Otherwise inconsistencies in the net topology might appear. The border regions must remain consistent even during an interpolation operation between two time steps. For this issue two different approaches are presented in the following. The first approach focuses on the border region and postulates that in two adjacent net components, the time steps of one net component must be a refinement of the temporal discretization of the other component. This is e.g. the case when the net component $\mathrm{nc\_1}$ consists of the time steps m_1 (t_0), m_2 (t_1) and m_3 (t_2) and the net component $\mathrm{nc\_2}$ of the time steps m_1 (t_0) and m_2 (t_2). In addition, the shared vertices of such net components have to be on a common line during the interpolation process, cf. Figure 30.

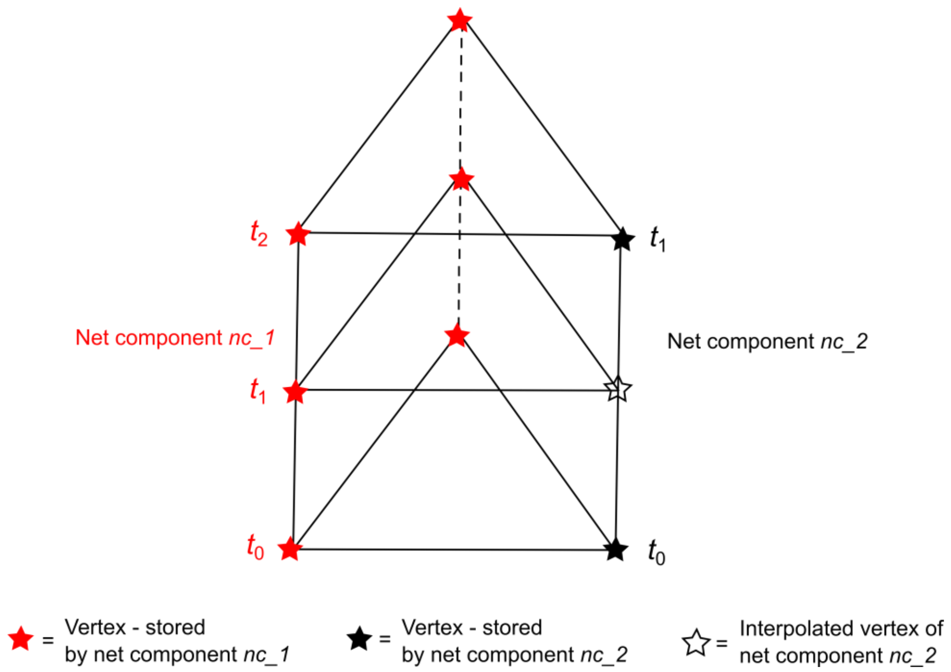


Figure 30 - Spatio-temporal prism representing a shared border region.

Due to these conditions the border regions of adjacent net components remain consistent and valid across the entire time interval. In addition, with this approach the border region of net component $\mathrm{nc\_2}$ with the lower temporal resolution remains independent of $\mathrm{nc\_1}$ between time steps t_0 and t_2.

The second approach proposes the introduction of so-called *hybrid elements* and focuses on the vertices of a border region between two net components. This procedure eliminates the condition that the temporal discretization of the involved net components has to be a refinement of ones another. The vertices of the border region from the component with the finer discretization must no longer be on the trajectory, which is created by the interpolation between two time steps of the coarser component, cf. Figure 31.

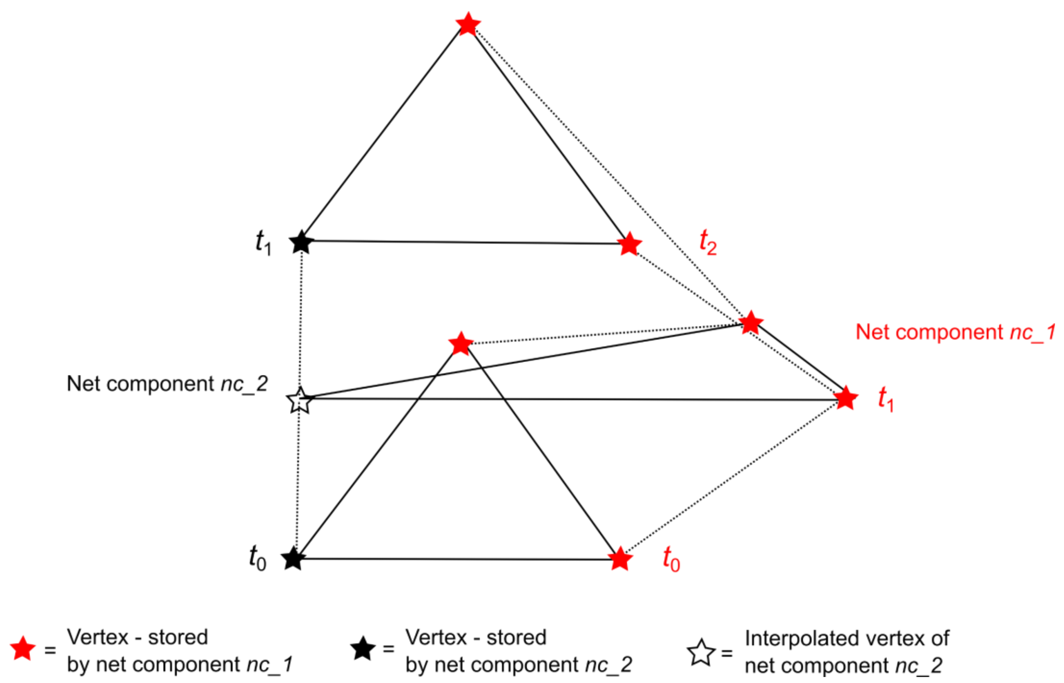


Figure 31 - Spatio-temporal prism representing a shared border region with hybrid elements.

However, it has to be ensured that no data points of the border region enter the boundary of uninvolved net elements. Otherwise this could destroy the net topology, cf. Section 3.7.1.

Due to this concept, it is possible to handle parts of a 4D model with different temporal discretizations. This leads to a major saving of storage space, since rapidly changing parts of a 4D model are handled accordingly while the remaining part of the model is managed in a lower temporal resolution. Figure 32 shows the Point Tube data structure of a 4D model that consists of two net components $\mathrm{nc\_1}$ and $\mathrm{nc\_2}$ with time steps m_1 (t_0), m_2 (t_1), m_3 (t_2), m_4 (t_{end}) and m_1 (t_0), m_2 (t_{end}), respectively.

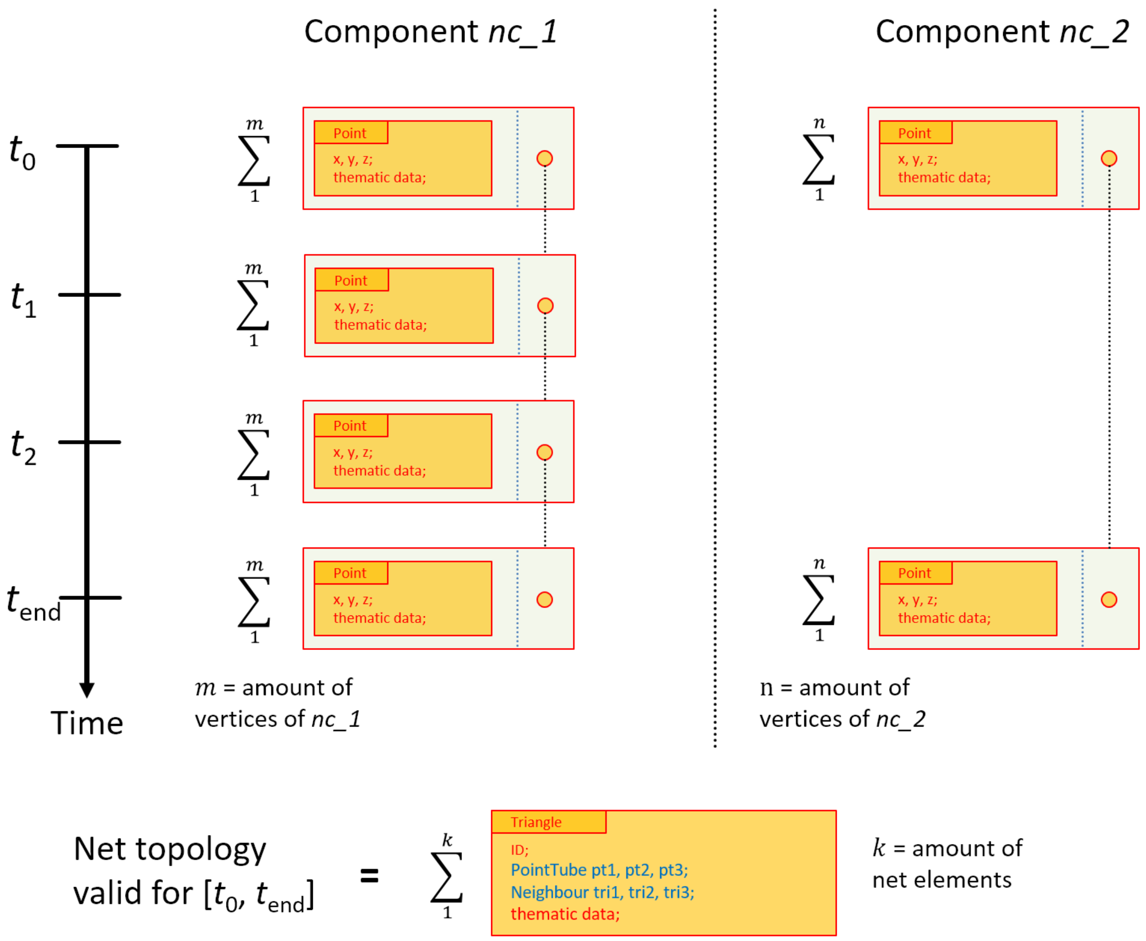


Figure 32 - Point Tube data structure for two net components with different temporal discretizations.

Due to the combination of this approach with the concept of Point Tubes a solution for maintaining the consistency of border regions of adjacent net components is given.

Realization for 3D-nets

The concept can be applied on 4D models that consist of 3D-nets, i.e. tetrahedron nets. The procedure is analogously to the management of 2D-nets. The conditions associated with the border regions must also be met within this case, cf. List 2.

However, the realization of those conditions for the border regions may be much more challenging, when managing 3D-net components with different temporal resolutions. It must be ensured that parts of the border region do not penetrate net elements, i.e. tetrahedra, of other net components, cf. Figure 33. This condition is especially critical when a net component is embedded within another one.

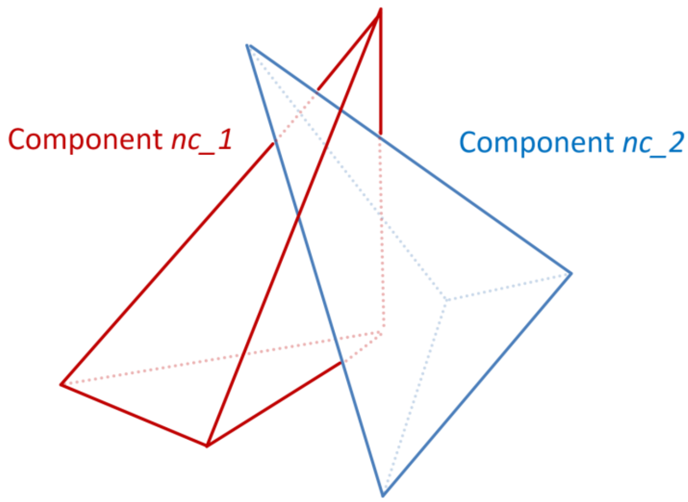


Figure 33 - Violating the consistency of the net topology due to modifications at the border region between two 3D-net components.

The management of such border regions is realized with the help of Point Tubes. The net topology of the border region itself must be kept in a consistent state according to the rules of the Point Tube concept, cf. Section 3.6. When managing adjacent 3D-net components with different temporal resolutions, this problem should be significantly reduced compared to embedded ones.

If an explicit model $m\ (t_x)$ for the specific time t_x is required during a range query or a geometric operation, such a model is formed by the union of all involved net components. If an explicit time step was stored for time $t_x \in [t_{start},\ t_{end}]$ for one or more net components, this specific time step is used for the construction of $m\ (t_x)$. All other net components are created by an interpolation for time t_x. This procedure applies to all net dimensions.

3.3 Handling of Net-Topology Changes

A change of the net topology between two time steps is a critical point for the consistency of a 4D model. However, a change of the net topology at a specific time step is sometimes necessary. For instance, to reflect the transformation of certain regions of the model in more detail. For the modelling of such a transition different concepts have been developed. Two of them are considered and discussed in detail below: the concept of [Polthier and Rumpf, 1994] and the operations by [Hoppe, 1996]. The two concepts are fundamentally different in their approach. The concept of Polthier and Rumpf introduces an additional object for the management of 4D models, the so-called *pre-object*. The concept of

Progressive Meshes (i.e. Hoppe operations) shifts the issue to the interpolation procedure. New dynamic elements with a specific behaviour are introduced that are capable of altering the net topology.

Referring to the field of computer graphics 3D objects are often meshed by triangles for visualization and transmission purposes. In this context, the concept of Progressive Meshes was developed and continuously improved by Hoppe and others [Hoppe, 1996], [Isenburg, 2002], [Kälberer et al., 2005]. The scope of this concept is to represent the transition between two different LoDs of a 3D object continuously. Within a 3D scene 3D objects that are farther away from the viewer than others, are often represented in a minor LoD. As soon as the viewer approaches such an object, the LoD shall increase to visualize more details of the object. In a transition from a lower to a higher LoD, details should emerge as continuously as possible and not be inserted abruptly or just pop in. Such a smooth transition can be achieved by the use of Progressive Meshes operations.

The concept was originally developed for the handling of triangular nets and enables the transformation of an existing triangulated model m_0 to a model m_1 with a reduced amount of triangles. During this procedure the external shape of the model should remain the same. This transformation is reversible and can also be performed in the other direction. The operations of Hoppe and its further developments can basically be divided into two categories:

1. The actual operations *vertex-split* and *edge-collapse.*
2. A function for selecting suitable triangles.

The operation *vertex-split* creates an edge from a vertex. If the operation is performed within a triangle net, the operation results in two new triangles. The inverse operation is called *edge-collapse* which shrinks an edge $\mathrm{e}\ (\mathrm{v_t},\ \mathrm{v_s})$ into a vertex $\mathrm{v_{new}}$. Through this operation, two triangles are removed within a triangle net, cf. Figure 34.

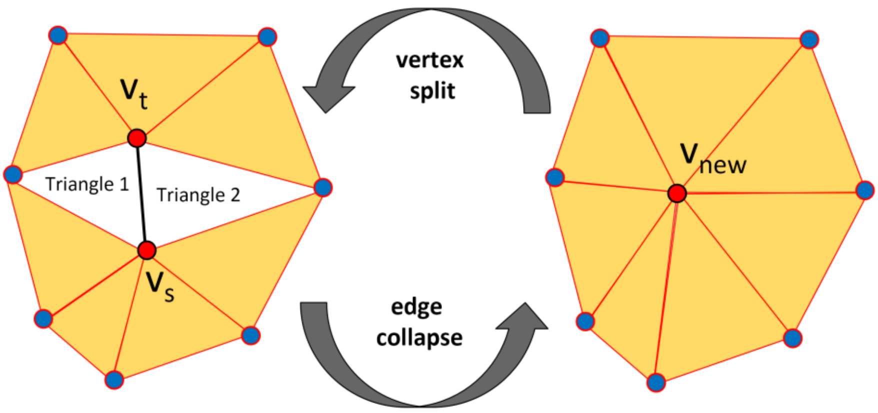


Figure 34 - Edge-collapse and vertex-split operation within a triangle net.

If these operations are applied on the border of a triangle net, it is also possible to remove or add single triangles, see Figure 35.

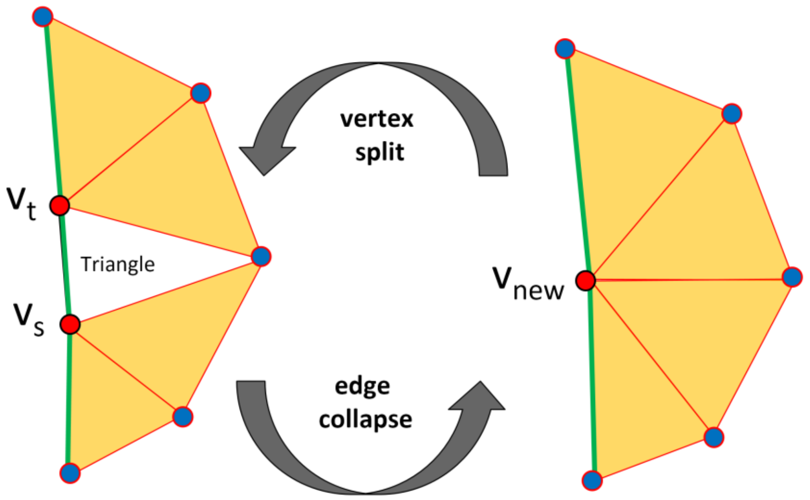


Figure 35 - Edge-collapse and vertex-split operation at the border of a triangle net.

A precondition for this concept is the selection of suitable edges or vertices that shall be manipulated by these operations. These are determined by a function that preserves the external shape of the 3D object as much as possible, if specific triangles are removed from the net. In addition, the choice of v_{new} needs to be handled accordingly. Hoppe therefore suggests the following procedure:

$$v_{new} = (1\text{-}\alpha)\ v_s + \alpha v_t \text{ for } \alpha = \left\{0, \tfrac{1}{2}, 1\right\}$$

An α that leads to most "beautiful" triangles[23] shall be chosen. The concept was primarily designed for two different applications: for a continuous transition between two different LoDs of a 3D object [Hoppe, 1996] and for an acceleration of the transmission and visualization of 3D models via a network interface, e.g. Internet. The principle works by the idea to transmit a reduced model m_0 of a triangle net and subsequently expand this model through vertex-split operations.

Through the use of Progressive Meshes operations it is possible to realize different LoDs. Therefore, objects in a 3D scene that are farther away from the viewer are represented by a model, which was reduced by the use of Progressive Meshes operations. Once the 3D scene changes and the virtual camera approaches the model, the LoD can be in-

[23] Beautiful accounts the proportion of a triangle. Very long and narrow triangles with a small normalized area are to be avoided.

creased by the use of vertex-split operations. These operations can be performed continuously and thereby sudden changes are avoided and an almost continuous transition can be achieved.

Another concept that describes the transition of net topology changes of a 4D model between two time steps has been developed by Polthier and Rumpf. The foundation of this concept is the introduction of a new object, the so-called *pre-object* [Polthier and Rumpf, 1994].

Unlike the concept of Progressive Meshes the concept of Polthier and Rumpf does not provide a continuous transition between two different net topologies. There is also no scope of a progressive transfer or construction of a 3D model. This concept focuses exclusively on the treatment of the transition between two time steps of a 4D model with different net topologies. Polthier and Rumpf based their concept explicitly on a model that changes over time and consists of a series of 3D models that form discrete time steps, cf. snapshot model in Chapter 2.3.1:

$$\text{4D model } m = \text{3D model } m\,(t_0) + \text{3D model } m\,(t_1) + \ldots + \text{3D model } m\,(t_n)$$

In order to reflect the change of the net topology between two of these time steps Polthier and Rumpf propose to store two objects at such an event: a *pre-object* with the net topology $top\ (t_{n-1})$ of the previous time step t_{n-1} and a *post-object* with the current net topology $top\ (t_n)$ at time step t_n. Both objects represent the shape of the geometry of the particular time step t_n. Figure 36 illustrates this approach for the timesteps t_0, t_1 and t_2.

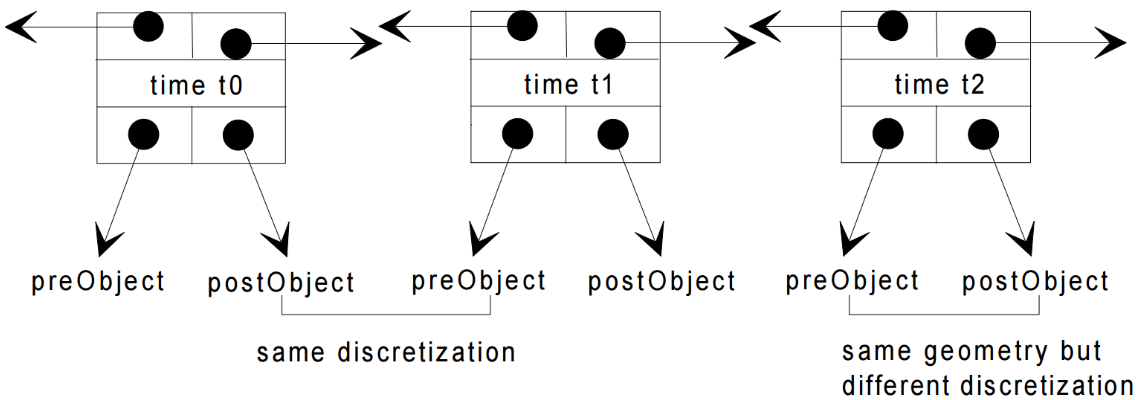


Figure 36 - The concept of using pre- and post-objects proposed by Polthier und Rumpf. Source: [Polthier and Rumpf, 1994]

Accordingly, the management of such a 4D model managed by the concept of Polthier and Rumpf results as follows:

model m = post-object (t_0) with *top* (t_0) + pre-object (t_1) with *top* (t_0) + post-object (t_1) with *top* (t_1) + pre-object (t_2) with *top* (t_1) + … + pre object (t_n) with *top* (t_{n-1}) + post-object (t_n) with *top* (t_n)

Due to this approach it is possible to have a unique assignment and a 1:1 relationship of individual net elements of the model between two time steps. Therefore it is possible to perform a linear interpolation between these time steps, cf. Figure 37. The net topology changes at a specific point in time, while the time and therefore the change in geometry is at rest. The concept has the trade-off that for each time step with a change of the net topology, two objects are stored: one pre- and one post-object. However, the advantage to enable the unique assignment of single elements and therefore a straightforward interpolation between two time steps compensates this disadvantage.

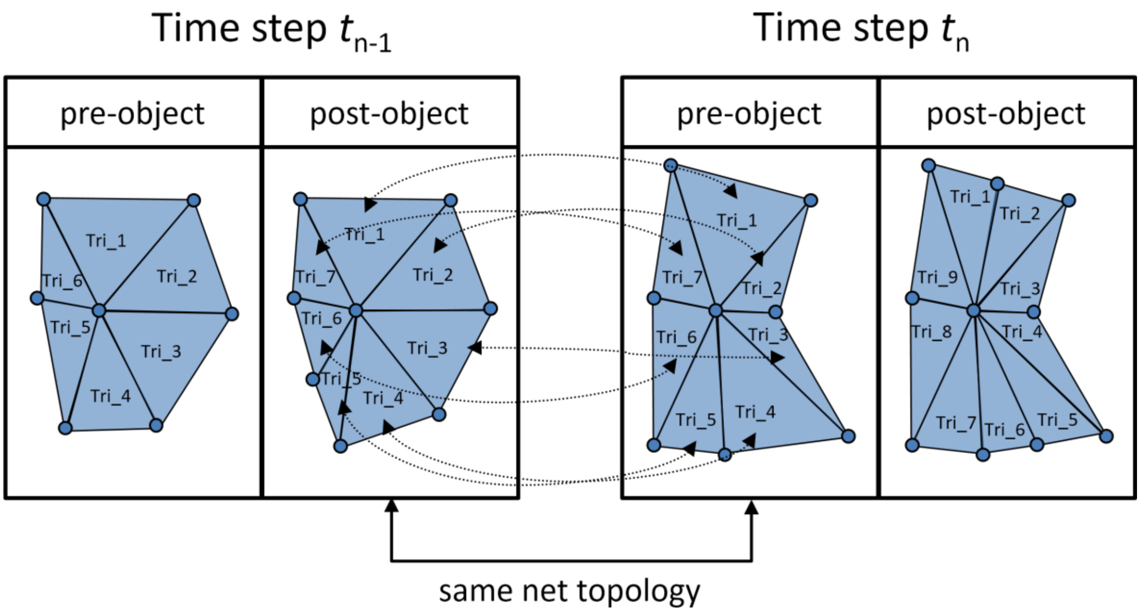


Figure 37 – Two time steps with pre- and post-objects of a 4D model based on 2-simplicial complexes.

The concept to handle net topology transitions of spatio-temporal data by using pre- and post-objects is already used in various spatio-temporal DBMS [Siebeck, 2003], [Le et al., 2013], [Breunig et al., 2013a].

3.3.1 Realization for Additional Net Dimensions

The concept of Progressive Meshes and its further developments are a great asset to the field of computer graphics, since thereby the amount of elements of a 3D model, can be reduced without major changes of the shape of the object. This is an advantage for the transmission of 3D models, e.g. via the Internet, and also for the representation of highly complex models within a 3D scene. The concept can lead to a load relief of the graphics processing unit (GPU). For such applications, the concept of Progressive Meshes is extremely useful. However, this approach cannot be used for the management of net topology changes between two time steps due to two significant problems: within a triangle net the operation vertex-split generates two triangles and the contrary method edge-collapse removes two triangles from the net. An exception exists only for the border regions. The second problem lies in the selection of the net elements that are affected by vertex-split or edge-collapse operations. Here a usual selection process obtains the shape and therefore the geometry of the object. This procedure would be impractical for geoscientific 3D models as relevant information might get lost. The geometry of such a model is not mandatory the most important factor. A different selection process, which, for instance, also takes attribute data into account would highly depend on the particular application. Such a selection process has to be redesigned for every particular 4D model. Furthermore, the Progressive Meshes operations were originally created for a continuous transition. This results in a continuous transition during an interpolation between two time steps. Additional information might be generated during such a transition on which the modeller has no influence. For these reasons, the concept of Progressive Meshes is not suitable for the management of geoscientific data.

In the following the model of Polthier and Rumpf is taken as a basis for a concept that describes a net topology change between two successive time steps. The major challenge of a changing net topology is the assignment of the individual vertices of the respective nets over time. For a simple interpolation, i.e. the description of a movement of the model in space over time, the assignment of elements (0-3D-simplices) and especially the vertices of a net should be unique. This way the movement of the individual elements can be described, cf. Figure 37. The concept of Polthier and Rumpf offers a very elegant solution due to a clear allocation of elements between the time steps for 2D-nets [Polthier and Rumpf, 1994]. Changes of the net topology are avoided between post-object and pre-object and thematic or semantic information can be processed. A transmission and adaptation of this concept, which was originally developed for the handling of triangle nets, onto 0D-, 1D- and 3D-nets is presented in the following.

Realization for 0D-nets

0D-nets, i.e. point clouds that change over time have no natural meshing and no topological information, cf. Chapter 2.2 and 2.3. Nevertheless, a mechanism is needed for the transition of two successive time steps, in which the number of vertices (0D-simplices) have changed, since the individual points of such time steps are not assigned in a 1 to 1 relationship, cf. Figure 38.

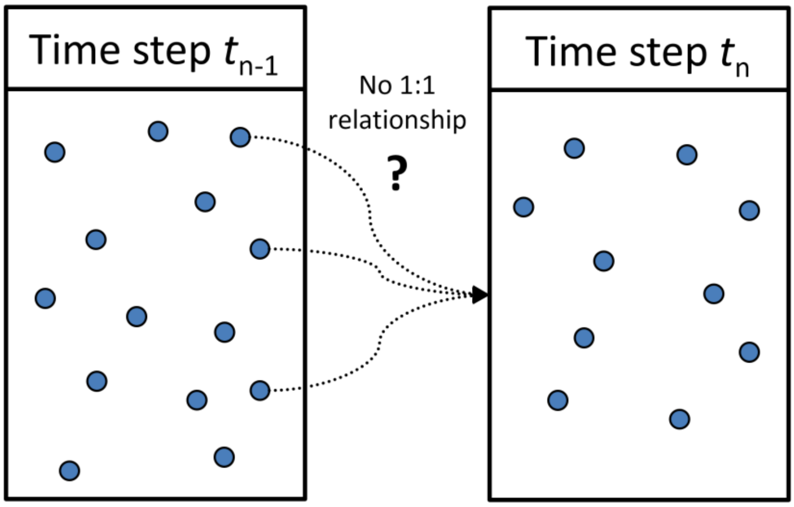


Figure 38 - Two models *m* (t_{n-1}) and *m* (t_n) of two successive time steps t_{n-1} and t_n, with a distinct net topology, i.e. an affine transformation is not possible.

For such situations a unique 1 to 1 assignment is accomplished by the introduction of a pre-object according to the rules of the concept by Polthier and Rumpf. For this purpose, a pre-object for time step t_n is introduced at the end of time interval $[t_{n-1}, t_n]$ of such a transition. Such a pre-object satisfies the following properties:

1. The number of construction elements, i.e. vertices, corresponds to the post-object of the time step t_{n-1}.
2. The geometry (shape of the object) corresponds (as much as possible) to the post-object of time step t_n.

Due to this procedure within the half-open interval $[t_{n-1}, t_n[$ a 1 to 1 allocation is ensured for the net elements and a simple interpolation between these two steps can be realized. Figure 39 illustrates this approach graphically.

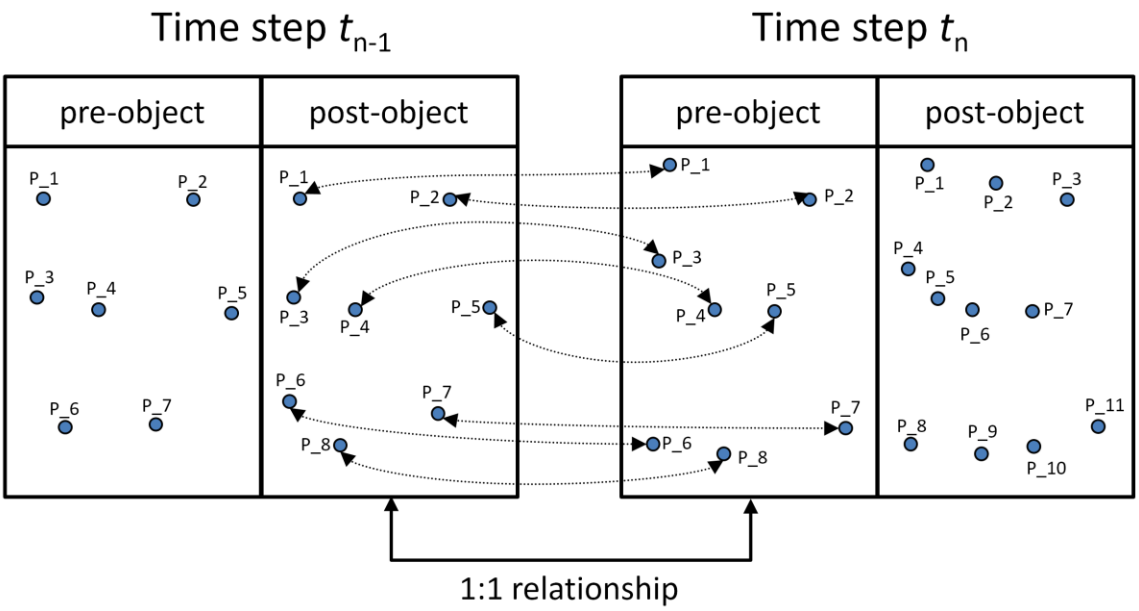


Figure 39 - Due to the introduction of a post-object a 1:1 relationship between the time steps t_{n-1} and t_n is established.

Realization for 1D-nets

Segment nets, i.e. 1-simplicial complexes have a meshing and hence topological information in form of neighbourhood relations, cf. Chapter 2.2 and 2.3. Once the net topology between two models $m\ (t_{n-1})$ and $m\ (t_n)$ of two successive time steps t_{n-1} and t_n changes, there is no unambiguous 1 to 1 relationship between the 3D models of these two time steps, cf. Figure 40.

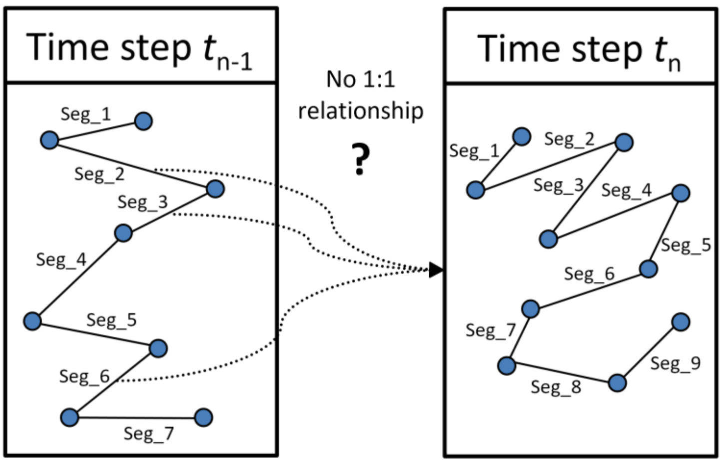


Figure 40 - Two models m (t_{n-1}) and m (t_n) of two successively time steps t_{n-1} and t_n, with a distinct net topology, i.e. an affine transformation is not possible.

In such a situation it is possible to establish a 1 to 1 relationship between the successive time steps by introducing a pre-object at the end of interval $[t_{n-1}, t_n]$. For the pre-object at time step t_n the conditions known from the concept of Polthier and Rumpf are met:

1. The net topology corresponds to the net topology of the post-object at time step t_{n-1}.
2. The geometry (shape of the object) corresponds (as much as possible) to the post-object of the time step t_n.

Due to the introduction of such a pre-object a 1 to 1 relationship of the individual elements is guaranteed during the half-open time interval $[t_{n-1}, t_n[$. Thus, a simple interpolation between these two time steps t_{n-1} and t_n is possible. Figure 41 illustrates this approach graphically.

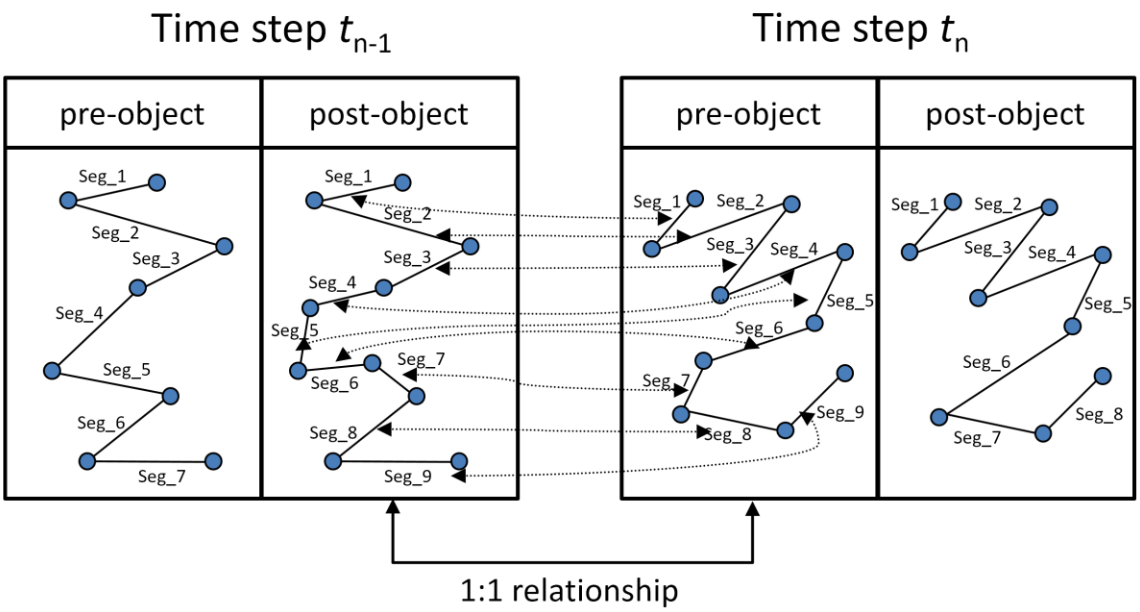


Figure 41 - Due to a post-object a 1:1 relationship between time steps t_{n-1} and t_n is established.

Realization for 3D-nets

Tetrahedron nets, i.e. 3-simplicial complexes, have a meshing and hence topological information in the form of neighbourhood relations, cf. Chapter 2.2 and 2.3. Once the net topology between two models m (t_{n-1}) and m (t_n) of two successive time steps t_{n-1} and t_n changes, there is no unambiguous 1 to 1 relationship between the 3D models that represents these two time steps, cf. Figure 42.

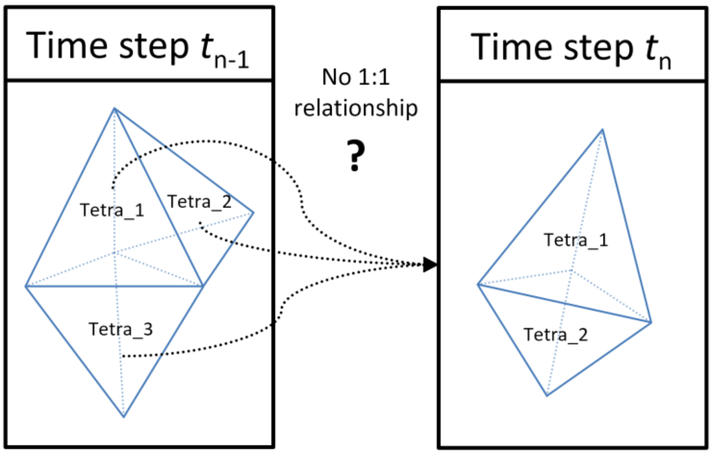


Figure 42 - The two models m (t_{n-1}) and m (t_n) of two successive time steps t_{n-1} and t_n, with a distinct net topology, i.e. an affine transformation is not possible.

In such a situation it is possible to establish a 1 to 1 relationship between the two time steps by introducing a pre-object at the end of the interval [t_{n-1}, t_n]. For the pre-object at time step t_n the conditions known from the handling of 1D-nets are met.

Due to the introduction of such a pre-object a 1 to 1 relationship of the individual elements is guaranteed during the half-open time interval [t_{n-1}, t_n[. Thus, a simple interpolation between these two time steps t_{n-1} and t_n is possible. Figure 43 illustrates this approach graphically.

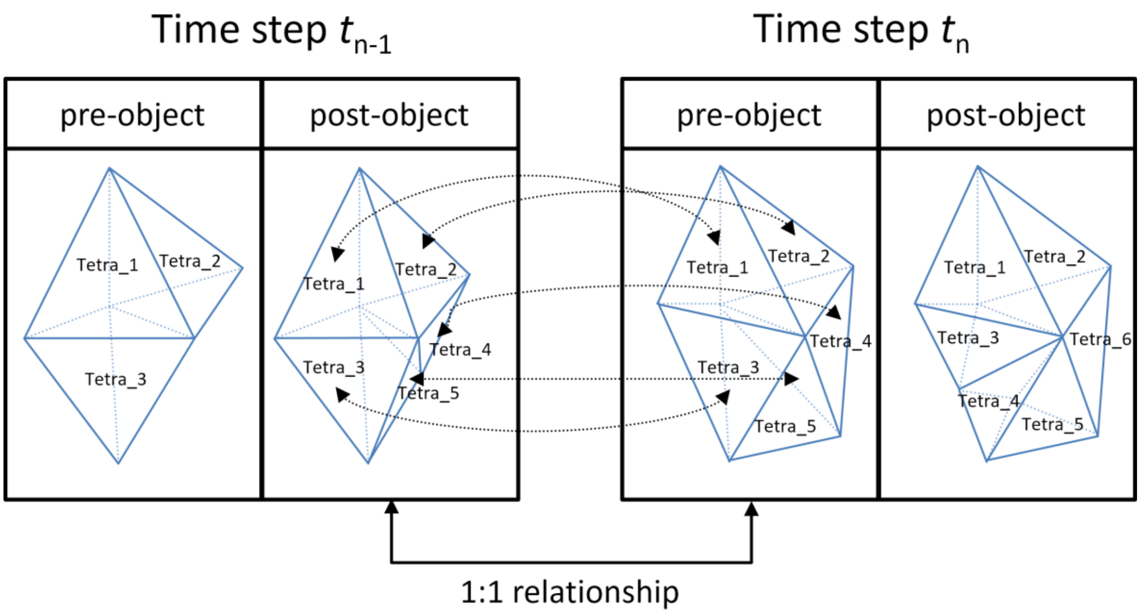


Figure 43 - Use of a post-object to establish a 1:1 relationship between time steps t_{n-1} and t_n.

In summary, the concept of pre- and post-objects by Polthier and Rumpf was transferred with the mentioned adjustments for the handling of 0D-, 1D- and 3D-nets. Apart from the handling of 0D-nets, this procedure follows the conventional rules.

The increased storage requirements, which are caused by the storage of two objects (pre- and post-object) for each time step with a net topology change, might be reduced by the concept of *Delta Storage*, cf. Chapter 3.5 and 3.6.2.

3.3.2 Automatic Construction of Pre-objects

To realize the concept of Polthier and Rumpf the existence of an appropriate pre-object is required. Such an object is required for each time step with a net topology change. Due to this requirement the developer of a 4D model needs to create such a pre-object during the modelling process for all affected time steps.

If such a pre-object does not exist for a time series, it would be a major assistance to create such an object automatically based on existing data. An automated creation of the pre-object is a complex matter and is related to the "vertex corresponding" problem [Le et al., 2013]. In the following, the automated creation of pre-objects is briefly sketched by two different approaches. The common starting point is as follows: within a time series of a 4D model two successive time steps t_{n-1} and t_n are represented by two 3D models $m\ (t_{n-1})$ and $m\ (t_n)$ with a different net topology.

Method 1: Projection of the post-object

By a projection of the model $m\ (t_{n-1})$ onto the model $m\ (t_n)$, a pre-object for the time step t_n can be created. Therefore, at first, the direction of this projection must be determined. Thus the movement of the model has to be considered. For instance, if the entire model moves in the direction of the Z-axis, the model $m\ (t_{n-1})$ is projected in the direction of the Z-axis onto the model $m\ (t_n)$. The border regions of the two models must be handled with caution. The border of the projected model $m\ (t_{n-1})$ has to be reflected on the border of the model $m\ (t_n)$. For this purpose, the extend of the model needs to be shrunk or expanded, cf. Figure 44.

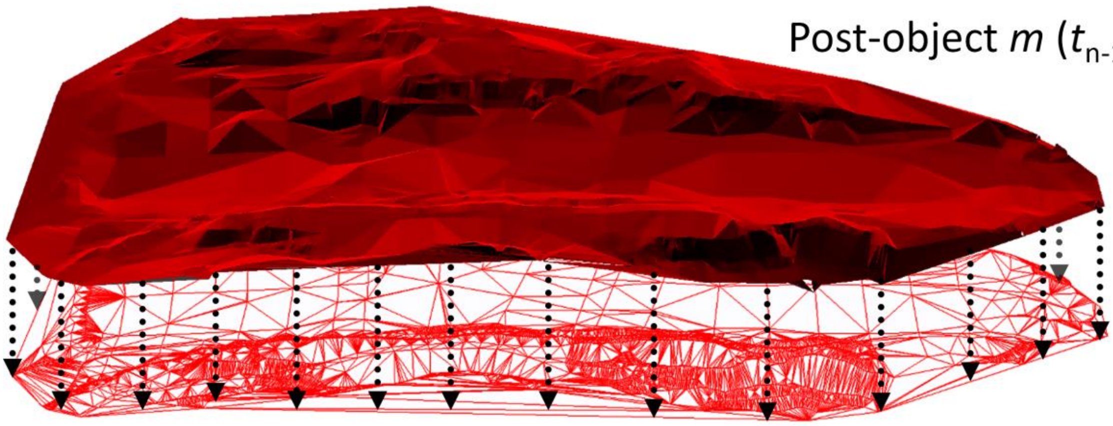


Figure 44 - Automatic generation of a pre-object by the projection of the model m (t_{n-1}) on model m (t_n).

Method 2: Use of Progressive Meshes operations

An alternative way to generate a pre-object is to adapt the associated post-object. In this case the pre-object is developed from a copy of the post-object onto which the Progressive Meshes operations *edge-collapse* or *vertex-split* have been applied on. Due to this procedure it is possible to reduce or increase the amount of elements of the object. In this case these operations must be performed until the number of elements of the pre-object m (t_n) corresponds to the number of elements of the post-object m (t_{n-1}). Afterwards the elements e (t_n) of pre-object m (t_n) must be assigned to the elements e (t_{n-1}) of the post-object m (t_{n-1}). Such a 1 to 1 relationship is required for several spatio-temporal operations such as a suitable interpolation. However, the assignment of the single elements just represents the major challenge. In addition, it must be ensured that the neighbourhood information of affected net elements do not change and thus the net topology is maintained.

3.4 Differentiated Management of Thematic and Semantic Data

For the management of 4D models that consist of simplicial complexes, one question is which kind of geometric data is stored explicitly and what is implicitly available and can be calculated on-demand if required. A *d*-simplicial complex consists of *d*-simplices of the dimension $d \in \{0,1,2,3\}$. These elements form the foundation of such a net. The construction of these elements is of crucial importance for various reasons. In particular, the construction of the simplices is relevant for the management of thematic and semantic data.

For the construction of such elements different approaches can be realized. For instance, the approach that is used in the 3D data model of the geodatabase architecture DB4GeO [Breunig et al., 2010], stores the information of corresponding vertices and adjacent elements on individual net elements. Each net element of a model consists of topological information of the element itself and its vertices: e.g., a tetrahedron consists of four own points ($p1$, $p2$, $p3$, $p4$) and information about the adjacent elements ($tetra1$, $tetra2$, $tetra3$, $tetra4$).

The approach of the Points Tube concept, cf. Chapter 3.1, stores the vertices of a net in Point Tubes data structures that are separated from the net topology. The net topology itself consists of net elements and their adjacent elements. Accordingly, with this approach each individual net element of a model is formed by the composition of the element itself and corresponding references to Point Tubes of the 4D model. A tetrahedron consists of four references to Point Tubes ($pt1$, $pt2$, $pt3$, $pt4$) and the information about adjacent elements ($tetra1$, $tetra2$, $tetra3$, $tetra4$). An overview of these two approaches is shown in Figure 45.

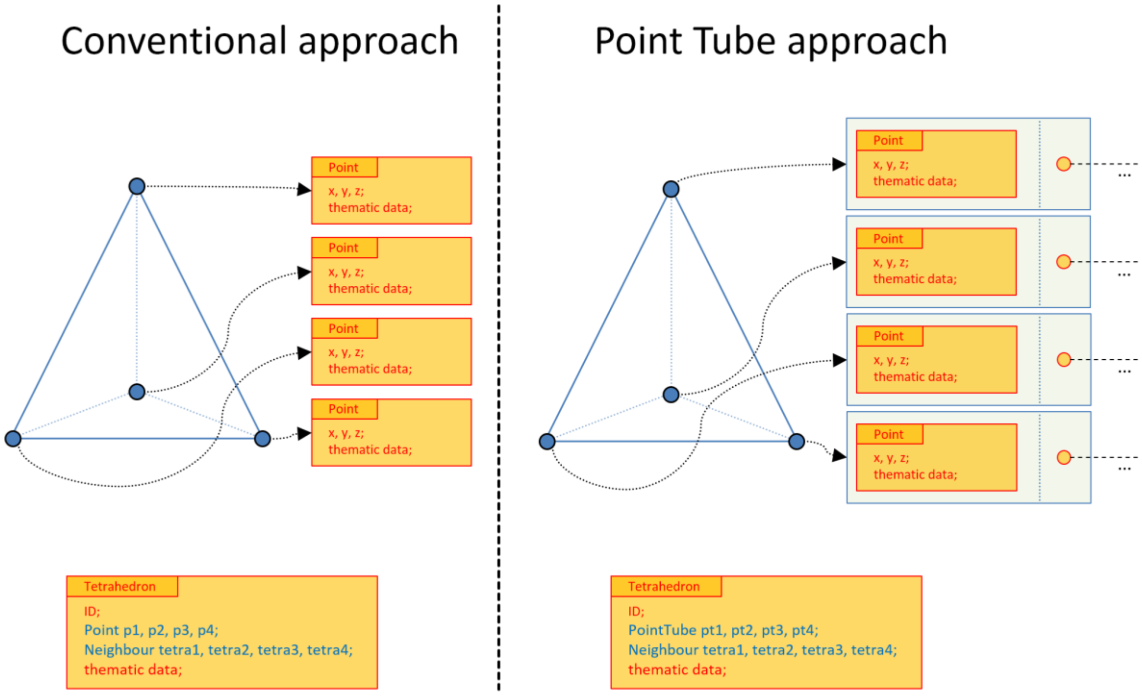


Figure 45 - Conventional and Point Tube approach to store a 3-simplicial complex.

However, both approaches solely manage the vertices and the composition of the current net element. Once details of a particular element are requested or used for geometric operations, these details have to be calculated on-the-fly. For instance, for the request of an outer face of a tetrahedron a temporary 2-simplex is created and used by the points $p1$, $p2$, $p3$ or, following the Point Tube concept, by $pt1$, $pt2$, $pt3$.

Often not only the geometry is of interest and thematic and semantic information of various kinds need to be managed across different application fields. Such data is not solely stored at the elements of the highest dimension (*d*-simplex) or lowest dimension (0-simplex), but also at the elements of any intermediate dimension.

An example is the definition of a boundary surface within a 3D-net, which is semi-permeable, i.e. permeable only for certain molecules. Such a boundary forms its own 2D-net. If these triangles are created only on request, it is not possible to store thematic information at the appropriate places. In this case the boundary surface could only be stored on the involved vertices or tetrahedra. This problem applies also for the management of 2-simplicial complexes with information on 1D-simplices and for 3D-simplices with information on 1D-simplices, respectively.

To ensure high flexibility in the processing of thematic and semantic data, it must be possible to store and manage relevant information at the appropriate places within the 4D model. For this purpose, *d*-simplices must explicitly store *d*+1 references to their explicitly handled (*d*-1)-faces instead of computing these exclusively when needed. A specific model m for the time step t_n of dimension d thus consists of the following components:

$$\text{model } m\ (t_n) = \text{vertices } (t_n) \cup \bigcup_{k=1}^{d} Top\ (k\text{D})^{24}$$

Such a model contains the appropriate *d*-simplices with their incidence relationships. This procedure is also referred to as the *incidence model* [Ohori et al., 2015]. A graphical illustration of the explicit storage for all construction elements of a net element can be found in Figure 46.

[24] *Top* (*k*D) represents the information of the net topology of the respective dimension *k*, e.g. a 3-simplicial complex consists of vertices, segments *Top* (1D), triangles *Top* (2D) and tetrahedra *Top* (3D).

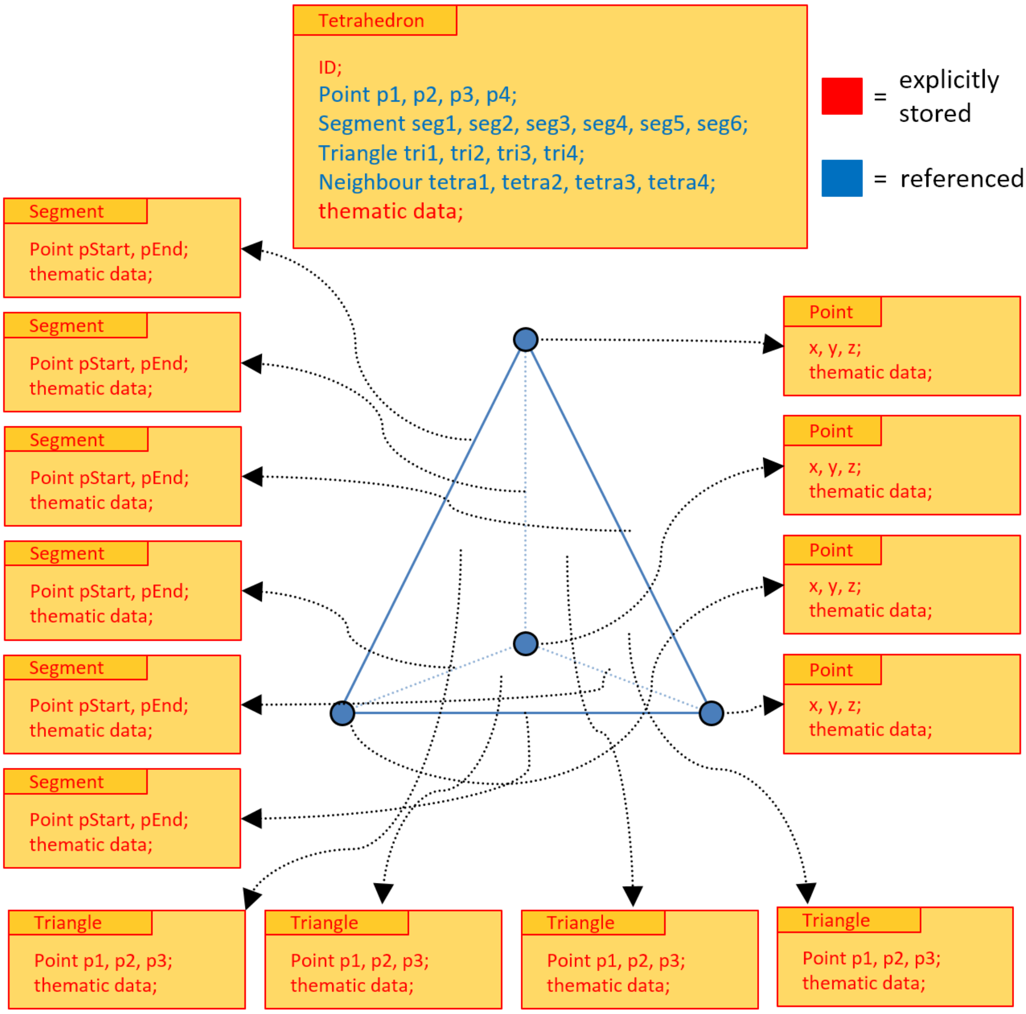


Figure 46 - Explicit storage of construction elements for a single tetrahedron using a conventional approach for the vertices.

At first sight this concept implies a significant increase in storage usage and a decrease of performance. This issue is addressed in Section 3.6.4 and a combination of this approach with the Point Tube concept is presented.

3.5 Delta Storage Concept

The concept of Delta Storage reduces the amount of data that need to be stored when managing 4D models that consist of multiple time steps. The basic idea is to reuse parts of the model that do not change between successive time steps.

In the following a first approach on the processing of time series by using an initial model and separated *deltas* is presented. This approach is based on the idea of incremental updates introduced by [Dadam et al., 1984]. Prior work also exists in the context of handling spatio-temporal data that consist of simplicial complexes. For instance, the concept of Delta Storage was realized in the geodatabase library GeoToolKit [Shumilov and Siebeck, 2001], [Balovnev et al., 2004].

In the following, the details of the transfer and the specific impacts on particular net dimension are described individually for each net dimension.

Realization for 0D-nets

The basic idea to reuse regions, which do not change between one or more time steps, can be used for the management of 4D models, whose individual time steps consists of point clouds, i.e. 0-simplicial complexes.

In a first step, the regions of a 4D model are determined that change between two successive time steps t_n and t_{n+1}. To do this the two models $m\ (t_n)$ and $m\ (t_{n+1})$ are compared. The changing regions form a delta $d\ (t_{n+1})$. In the special case that the complete model changes, the delta $d\ (t_{n+1})$ consists of the entire model $m\ (t_{n+1})$. Therefore, the model $m\ (t_{n+1})$, which corresponds to time step t_{n+1}, is built as follows:

$$\text{model } m\ (t_{n+1}) = \text{model } m\ (t_n) + \text{delta } d\ (t_{n+1})$$

All time steps are managed in the same way and a model $m\ (t_i)$ for a specific time step t_i consists of the following parts:

$$\text{model } m\ (t_i) = \text{initial model } m\ (t_0) + \text{delta } d\ (t_1) + \text{delta } d\ (t_2) + \ldots + \text{delta } d\ (t_i)$$

Since 0D-nets have no net topology, there is no need to manage the consistency of the net topology.

Realization for 1D-nets

The concept of Delta Storage can also be used for the management of 4D models, whose time steps consist of segment nets, i.e. 1-simplicial complexes. The regions of a 4D model, which change between two time steps t_n and t_{n+1}, are stored as a delta $d\ (t_{n+1})$. If the entire model changes, the delta $d\ (t_{n+1})$ consists of the model $m\ (t_{n+1})$. Thus, the model $m\ (t_{n+1})$ of the corresponding the time step t_{n+1} is formed analogously to the realization of this concept on 0D-nets.

Since the concept can be applied to all time steps of the 4D model, a model $m\ (t_i)$ for a specific time step t_i is composed as described above.

4D models, which are built by 1-simplicial complexes, own a net topology that corresponds to the conditions defined in Chapter 2.3. This net topology must be obtained at any time within the time interval $[t_0, t_{end}]$. In particular, geometric operations must not violate the conditions of the net topology. Therefore, the consistency of border regions between the model m (t_n) and the delta d (t_{n+1}) must be guaranteed.

In this case, a border region between a model m (t_n) and a delta d (t_{n+1}) consists of points, cf. Figure 47. If the affected points exist multiple times, minimal deviations might lead to inconsistencies within the model. Such discrepancies may arise by arithmetic errors in geometric operations. A solution to this problem, which can also be used for 2D- and 3D-nets, is presented in Chapter 3.6.

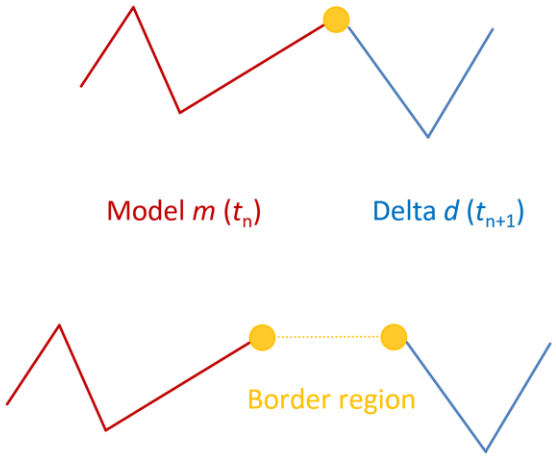


Figure 47 - Border region of a model m (t_{n+1}) which consists of model m (t_n) and a delta d (t_{n+1}) (with exploded-view drawing).

Realization for 2D-nets

The concept is based on the storage of an initial model m (t_0) for the first time step t_0. Subsequently, the particular geometry of a successively added model m (t_{n+1}) is compared with the geometry of the model m (t_n) and the difference between these two models is stored as delta d (t_{n+1}). This procedure is repeated each time the 4D model gets expended by an additional time step. If there are no matching parts in the two compared models m (t_n) and m (t_{n+1}), the delta d (t_{n+1}) is equivalent to the entire model m (t_{n+1}). Therefore, a model m (t_{n+1}) is formed analogously to the realization of this concept on 0D-nets.

If the model of a certain time step t_i of the 4D model is requested, the model m (t_i) is created from the initial model m (t_0) and the assembled deltas d (t_1) to d (t_i). Accordingly, a model m (t_i) of a corresponding time step t_i is composed analogously to the realization of this concept on 0D-nets. Figure 48 describes this procedure illustrative for a model m (t_3) for the time step t_3.

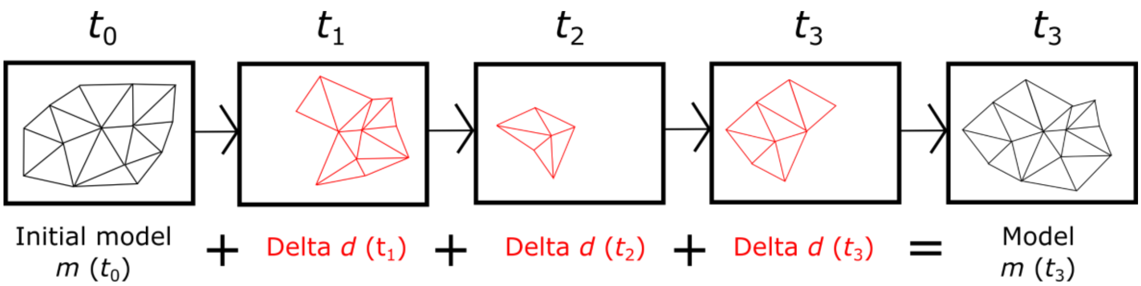


Figure 48 - The use of the Delta Storage concept for a model that is represented by four time steps.

4D models which are built by 2-simplicial complexes own a net topology that corresponds to the conditions defined in Chapter 2.3. This net topology must be obtained during the time interval [t_0, t_{end}]. Geometric operations must not violate the conditions of the net topology. Therefore, the consistency of the border region between the model m (t_n) and a delta d (t_{n+1}) must be guaranteed.

In this case, the border region between a model m (t_n) and a delta d (t_{n+1}) consists of segments, cf. Figure 49. If the affected segments exist multiple times, minimal deviations lead to inconsistencies within the model. Such discrepancies may arise by arithmetic errors in geometric operations. A solution to this problem, which can also be used for 1D- and 3D-nets, is presented in Chapter 3.6.

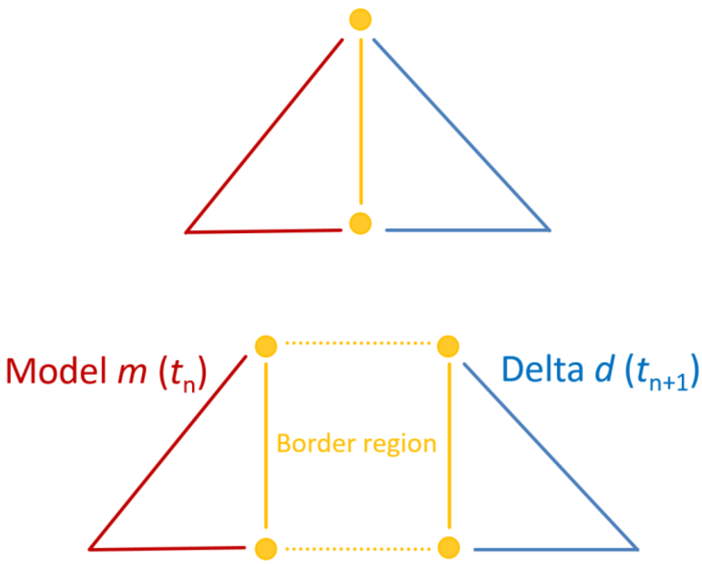


Figure 49 - Border region of a model m (t_{n+1}) which consists of the model m (t_n) and a delta d (t_{n+1}) (with exploded-view drawing).

Realization for 3D-nets

The concept of Delta Storage can also be applied on models, which are based on 3-simplicial complexes, i.e. tetrahedron nets. The procedure is analogously to the realization of this concept on 0D-nets. However, a critical point is the handling of the border region of a delta $d\ (t_{n+1})$ and the model $m\ (t_n)$ of the previous time step t_n, which is built by the initial model $m\ (t_0)$ and the deltas $d\ (t_1)$ to $d\ (t_n)$. Figure 50 describes such a border region of a delta $d\ (t_{n+1})$ and the model $m\ (t_n)$.

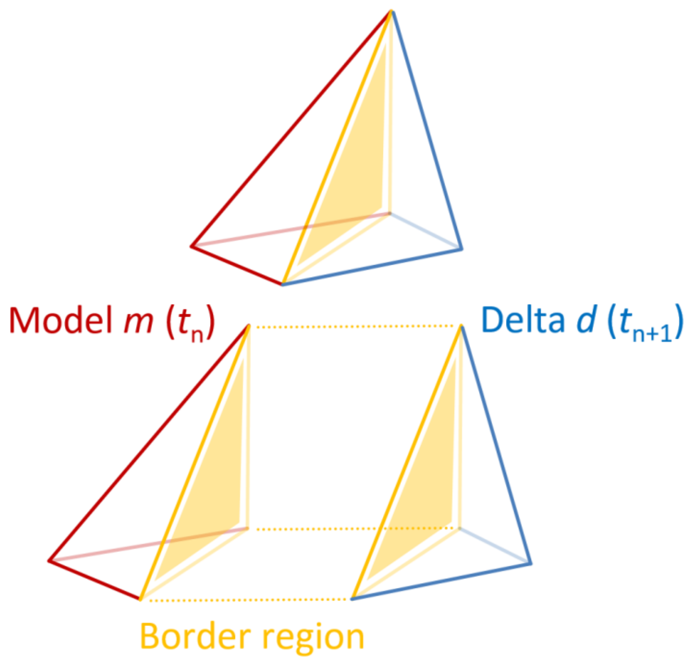


Figure 50 - Border region of a model *m* (t_{n+1}) which consists of the model *m* (t_n) and a delta *d* (t_{n+1}) (with exploded-view drawing).

In this case the border region is composed of a triangle. Thus, the use of tetrahedron models not only increases the dimension of the net elements of such a model, but also the dimension of the elements that form the border region between the model $m\ (t_n)$ and delta $d\ (t_{n+1})$. Equivalent to the realization of the Delta Storage concept on the management of time series that consist of 2D-nets, it is essentially important to keep these border regions in a consistent state.

Otherwise the constraints for the construction of simplicial complexes may be violated, cf. Chapter 2.2. Such a situation may occur through the use of geometric operations which draw parts of a model apart, and thereby invoke a change of the net topology, cf. Figure 51. The consistency of the border regions between the delta $d\ (t_{n+1})$ and the model $m\ (t_n)$ of a tetrahedron based model must be ensured during the entire time interval of the 4D model. One way to address this problem is the use of Point Tube data structures for the management of the vertices of a triangle or a tetrahedron net, respectively. Such a consolidation of the concepts is described in Section 3.6.

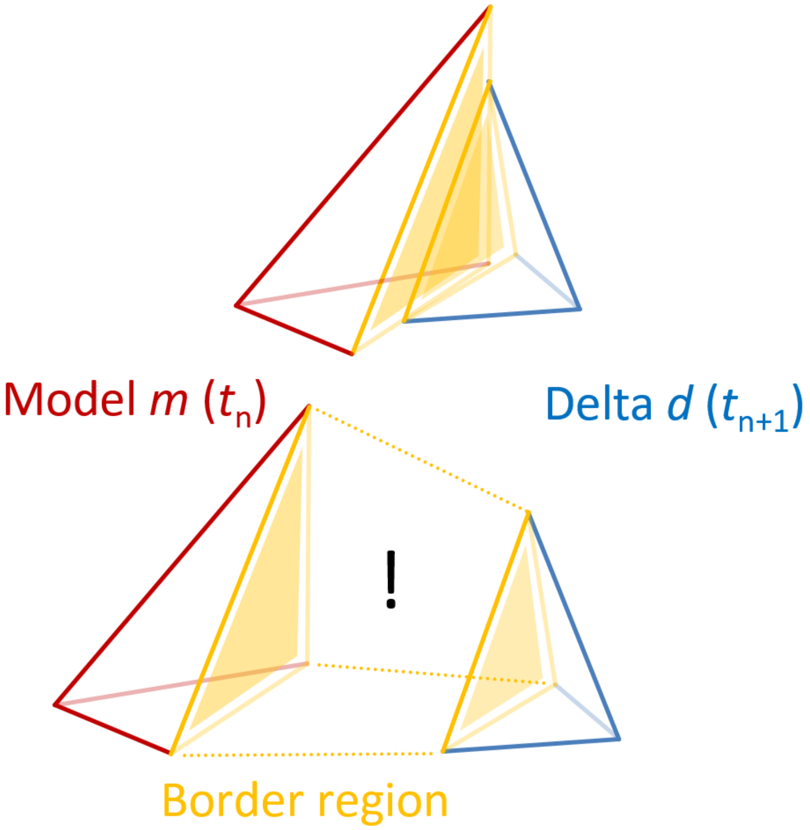


Figure 51 - Inconsistencies between a delta d (t_{n+1}) and the model m (t_n) of a tetrahedron model (with exploded-view drawing).

3.6 Realization of the Comprehensive Model

To cover a wide range of applications and to benefit from the positive effects of the concepts described in the previous sections, the five concepts are consolidated into a comprehensive data management model. However, it must be ensured that these concepts do not adversely affect each other and lead, e.g. to an administrative or storage overhead. Such a situation could occur as soon as a specific concept is unsuitable or not needed for a specific application. On the other hand, it would be beneficial if negative properties of individual concepts are compensated or reduced within the merging process.

The separation of the net topology from the vertices by using Point Tube data structures works as a basic concept. Thereby the individual vertices of a 4D model are traceable over time. Based upon this, the consolidation with the remaining concepts is developed in the next sections. The respective effects on different net dimensions and interactions of the different concepts among each other are pointed out in particular.

First, the merging of the two concepts Point Tube and Delta Storage is described. Subsequently, the integration of the concept for handling net topology changes between two time steps is presented. Afterwards, the consolidation with the concept of net components with different temporal resolutions is described. The extension to include the differentiated handling of thematic and semantic data at construction elements forms the final part of the comprehensive data management model.

3.6.1 Merging of the Concepts Delta Storage and Point Tubes

During the consolidation of the concepts Delta Storage and Point Tubes only 4D models are considered, whose net topology remains constant across all time steps. The combination of these two concepts addresses the question which parts of a model can be referenced and therefore be reused across multiple time steps. It gives an answer to the question "what is the delta $d\ (t_{n+1})$ "?

A delta $d\ (t_{n+1})$ is, as described in Section 3.5, the part of a model $m\ (t_{n+1})$ which changed between two models $m\ (t_n)$ and $m\ (t_{n+1})$ of the corresponding time steps t_n and t_{n+1}. In combination with the concept of Point Tubes, this implies that such a delta forms a set of Point Tubes with an entry for the time step t_{n+1}. The vertices, in combination with the net topology $top\ (t_n)$ of the model $m\ (t_n)$ form the border region between $m\ (t_n)$ and $d\ (t_{n+1})$. These vertices are referenced by both $m\ (t_n)$ and $d\ (t_{n+1})$, i.e. the same Point Tubes are used. Figure 52 shows how the members of the border points x, y and z are used by both the model $m\ (t_n)$ and the delta $d\ (t_{n+1})$.

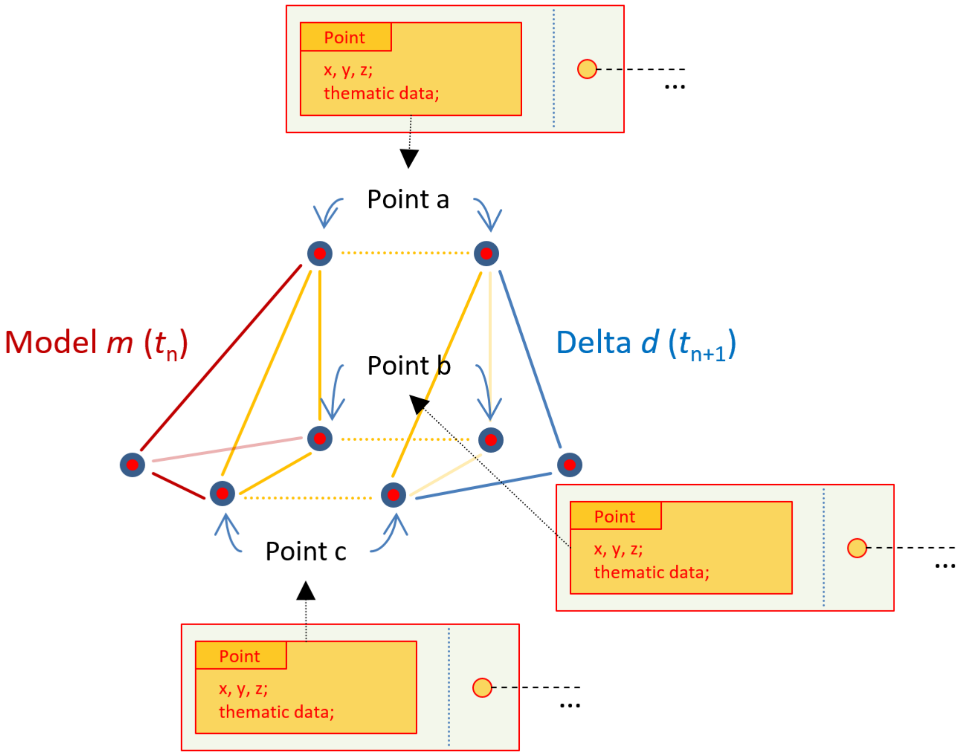


Figure 52 - Representation of a 3-simplicial-complex using the Delta Storage concept in combination with the Point Tube concept.

When creating a new 4D model, the initial model m (t_0) is managed internally by the use of Point Tubes data structures, cf. Section 3.1. Thereby the Point Tube structures are initialized and the net topology is managed separately. Subsequently, the differences between the models m (t_n) and m (t_{n+1}) are determined each time when another 3D model is added as a subsequent time step. These changes form a delta d (t_{n+1}). As only the vertices of the delta d (t_{n+1}) have changed, the corresponding Point Tubes are extended, while the Point Tubes of the unchanged regions reference to the last explicitly saved vertex, cf. Figure 53. The net topology is not altered between these time steps. For this reason, it is not stored multiple time and the net topology top (t_0) is reused. Dealing with 1D-, 2D- and 3D-nets top (t_0) consists of the collection of net elements. For 0D-nets top (t_0) consists of the collection of corresponding vertices. The following figure illustrates the internal structure combining the concepts Point Tubes and Delta Storage for three time steps t_0, t_1, and t_2. Three Point Tubes are described explicitly as examples of the internal structure. The total amount of Points Tubes corresponds to the amount of existing vertices in the net.

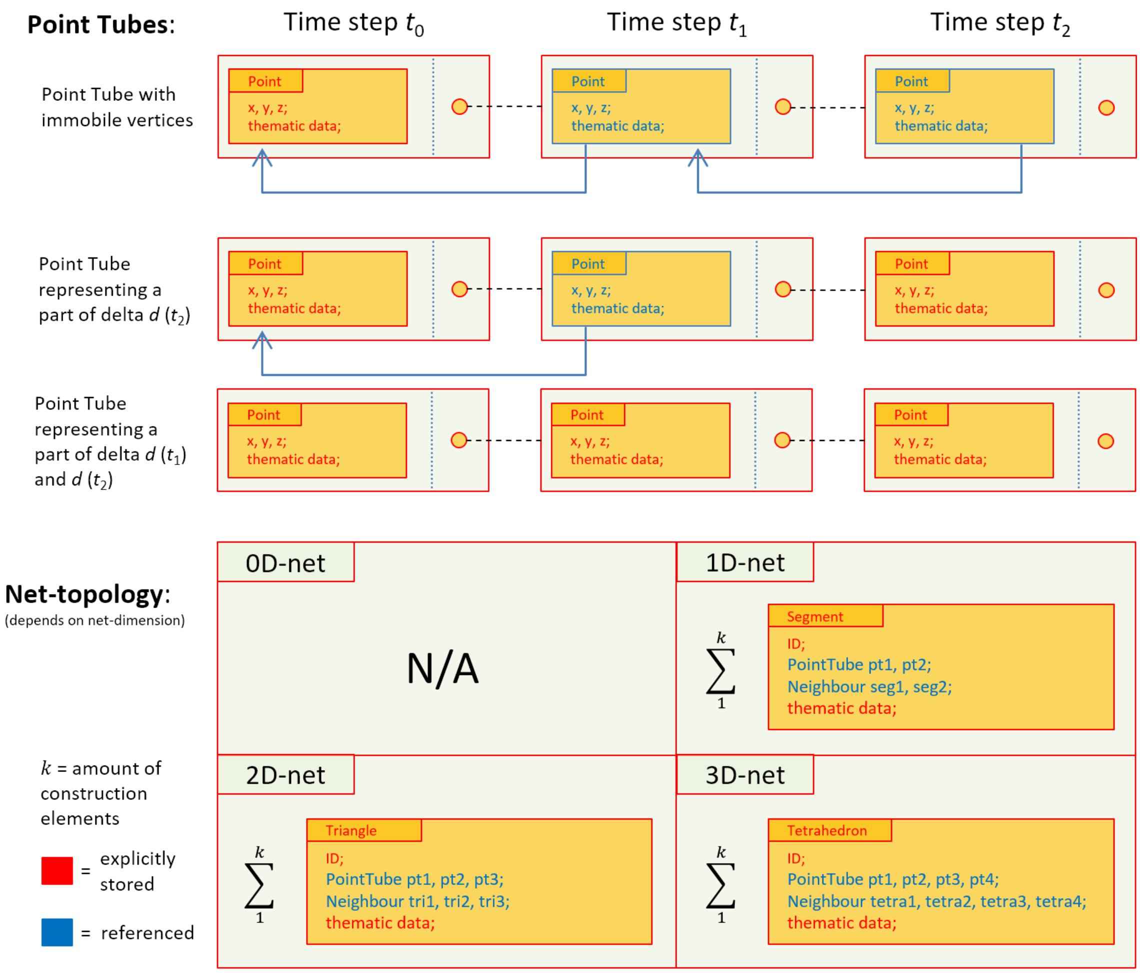


Figure 53 - Overview of the internal structure for the combination of the concepts Point Tubes and Delta Storage.

Effects of the consolidation

Due to the combination of the Delta Storage concept with the Point Tube concept inconsistencies at border regions are avoided, cf. Chapter 3.5. By using the same Point Tubes for vertices that belong to a border region and are used by both the model m (t_n) and delta d (t_{n+1}), the net topology is robust against arithmetic errors. As a result, the conditions for simplicial complexes are met at any time, cf. Chapter 2.2. Figure 52 illustrates this for a model that consists of 3-simplicial complexes.

This effect is also valid for the management of 1D- and 2D-nets. In these cases inconsistencies within the borders regions between a model m (t_n) and a delta d (t_{n+1}) are avoided, as the vertices of the border regions are used by both, cf. Figure 54.

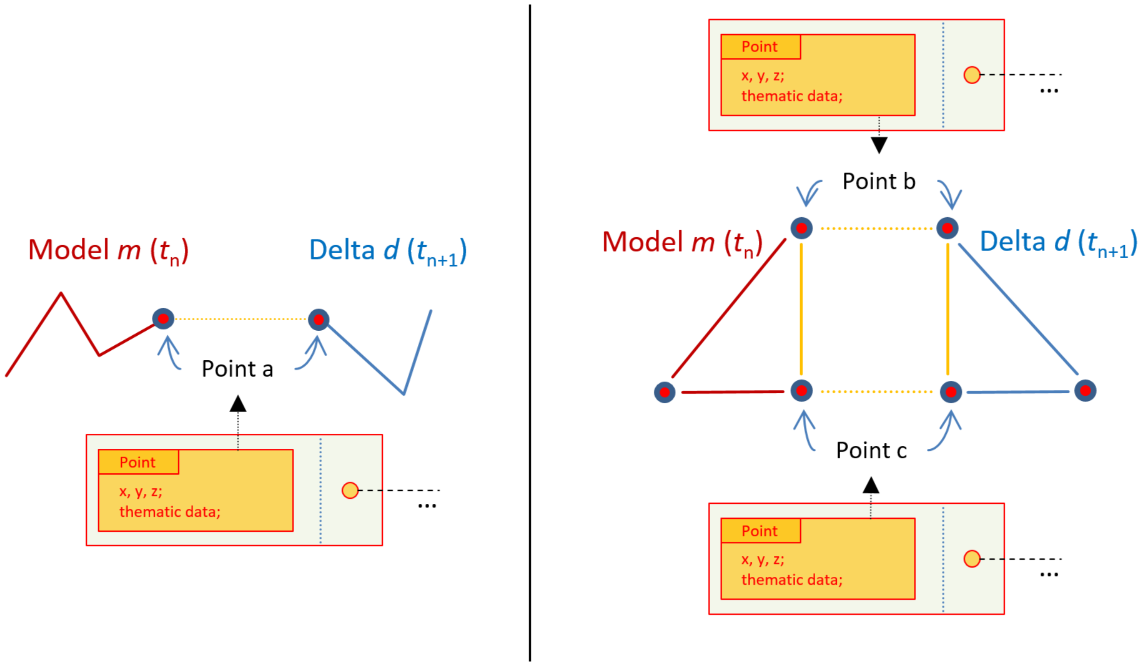


Figure 54 – Combination of the concepts Point Tube and Delta Storage: representation of the border region of a 1D-net (left) and a 2D-net (right).

This is especially important for the interpolation between two time steps. A model m (t_x)[25] created by interpolation may consist of more than one delta, cf. Chapter 3.5. Thereby the problem of inconsistencies may occur at multiple border regions. Combined with the Point Tube concept these border regions have been unambiguously coupled to each other and can no longer be "torn apart".

In addition, the positive effects of the two concepts are both active without any disadvantages. Due to the Point Tube concept storage space is saved and the performance of spatial and spatio-temporal operations is improved. In combination with the Delta Storage concept the required storage space is reduced even more. However, this depends on the size of the respective delta. If there are large similarities between the models of two time steps, the effect increases. Additional computations that are necessary for the realization of the two concepts are only required during the insertion of a 4D model into the data management model.

[25] t_x is not an explicit time step, but part of a valid time interval of m.

3.6.2 Consolidation with the Concept for Handling Net Topology Changes

The Delta Storage concept, in combination with the Point Tubes concept, can be extended by the concept for handling topology changes. For this purpose, the time steps with a change of the net topology, are subsequently studied in detail.

For each time step with a change of the net topology two models are required: a model $pre\ (t_{\text{change}})$, which represents the pre-object and a model $post\ (t_{\text{change}})$, which represents the post-object at time step t_{change}, cf. Chapter 3.3.

These objects are still required when combining this concept with the concepts of Delta Storage and Point Tubes and applies to all net dimensions. The combination has an impact on the internal representation, management and construction of the two models $pre\ (t_{\text{change}})$ and $post\ (t_{\text{change}})$.

In this case the parts of the model are reused, which do not change between $pre\ (t_{\text{change}})$ and $post\ (t_{\text{change}})$, analogous to two successive time steps without a change of the net topology. The changing regions between the two models $pre\ (t_{\text{change}})$ and $post\ (t_{\text{change}})$ form a delta $dpost\ (t_{\text{change}})$, which includes these changes. The model $post\ (t_{\text{change}})$ of a time step t_{change} is therefore formed as follows:

$$\text{model } post\ (t_{\text{change}}) = \text{model } pre\ (t_{\text{change}}) + \text{delta } dpost\ (t_{\text{change}})$$

The model $pre\ (t_{\text{change}})$ and the delta $dpost\ (t_{\text{change}})$ are hereby developed analogously to the rules of the Delta Storage concept, cf. Section 3.5. If the two models $pre\ (t_{\text{change}})$ and $post\ (t_{\text{change}})$ differ entirely from each other, the delta $dpost\ (t_{\text{change}})$ is equivalent to the model $post\ (t_{\text{change}})$.

Due to the usage of the Point Tube concept the reusable parts consist of the vertices that were used by both the model $post\ (t_{\text{change-1}})$ and $pre\ (t_{\text{change}})$. This also applies on the models $pre\ (t_{\text{change}})$ and $post\ (t_{\text{change}})$.

According to the rules of the Point Tube concept, the net topology is reused between all time steps without a net topology change. For a time step that involves a change of the net topology, it might be possible to reuse parts of the mesh.

The procedure is applicable for 1D-, 2D-, and 3D-nets. For 0D-nets (point clouds) the processing and reusing procedure of the net topology is skipped.

The data structure to realize the consolidation of these concepts is illustrated in Figure 55.

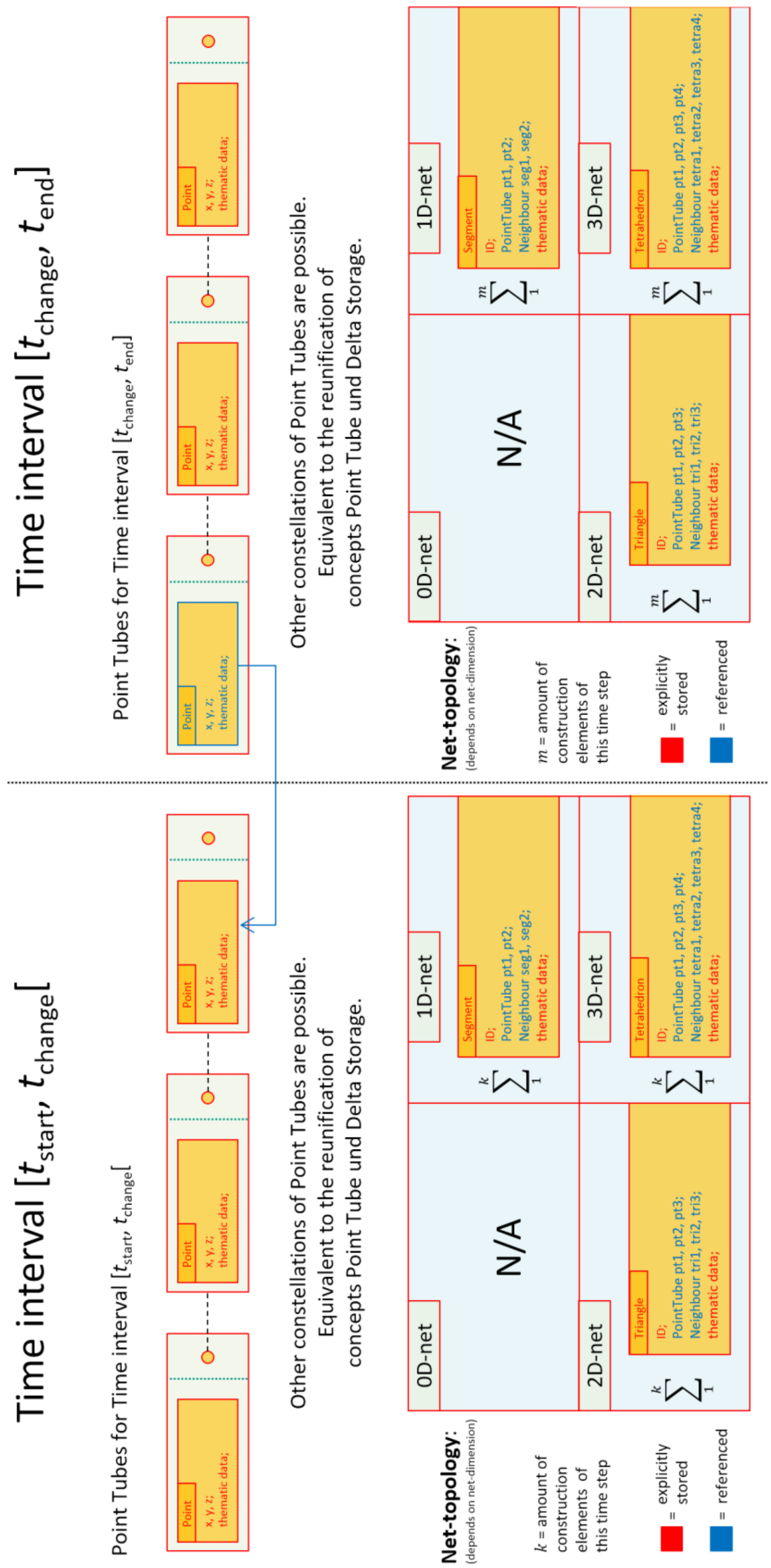


Figure 55 - Data structure to realize of the consolidation of the concepts Point Tube, Delta Storage and handling of net topology changes.

Effect of the consolidation

The consolidation of the three concepts complements the data management model to manage 4D objects, whose net topology changes within its time series. Therefore, the spectrum of applications is expanded significantly.

For all four net dimensions $d \in \{0,1,2,3\}$ the advantages of the respective concepts are retained within the consolidation. The amount of data that need to be stored, is reduced for the net dimensions 1 to 3 by using Point Tube data structures. For all dimensions, we obtain a clear assignment and a manageable allocation of vertices of the net elements. The separation of geometry and topology leads to a simplification and acceleration of geometric operations on the nets.

The identification of stationary parts of a model between two time steps, is expanded on the identification of differences between pre- and post-objects. Therefore, the storage of redundant information is further reduced.

The consistency of the net topology is guaranteed during the entire time interval of the model and is independent of the net dimension. Thus the three concepts complement each other without provoking any disadvantages. The application of the combined concepts based on sample data is shown in Figure 56.

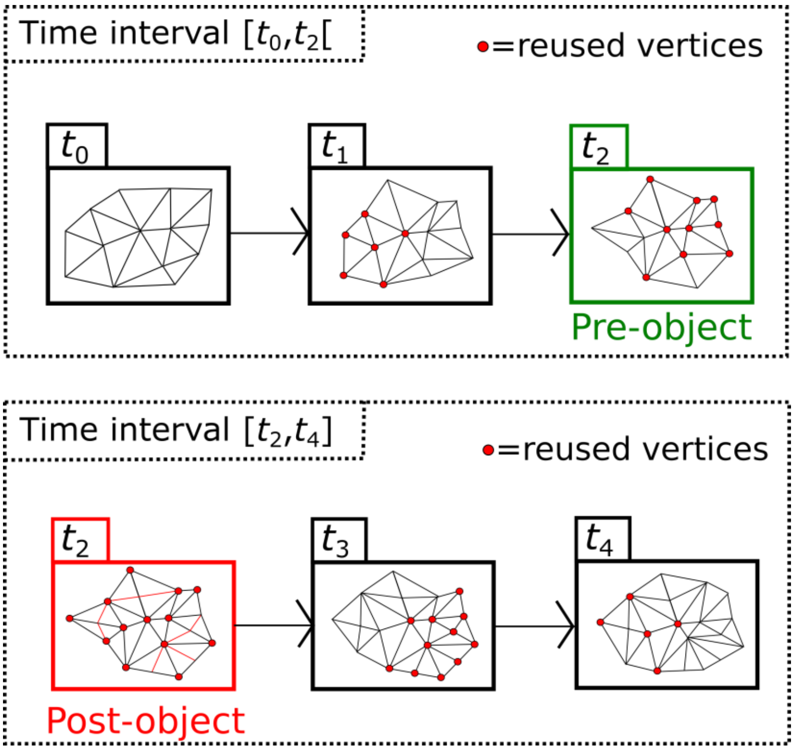


Figure 56 – Combined use of the concepts Point Tube, Delta Storage, and Topology Management.

3.6.3 Consolidation with the Concept for Handling Different Temporal Discretizations

Net components with different temporal discretizations support the management of 4D models with different volatile subregions. One important issue is the consistency of the time intervals of all involved net components. During the consolidation the time intervals are split into time intervals in which the net topology remains unchanged. These are the half-open intervals $[t_{\text{post}}, t_{\text{pre}}[$ or $[t_{\text{change-1}}, t_{\text{change}}[$, respectively. Such a time series gets completed with the interval $[t_{\text{changeN}}, t_{\text{end}}]$. All net components of a 4D model must be valid within the entire corresponding time interval.

Since this concept is directly related to the concept of Point Tubes, the consolidation of these two concepts has already been described in Chapter 3.2. The individual Point Tubes contain the vertices of the respective net component in the respective temporal discretization. For the border regions between such net components a sharing of net elements only occurs in higher dimensional structures, starting from 1-simplicial complexes. In this case the particular vertices are immutable assigned to a fixed net component.

Due to the consolidation with the concept of Delta Storage the vertices of individual net components are treated as if the components were independent *d*-simplicial complexes. The Point Tube structures are thus constructed as described in Section 3.6.1. The assignment to the corresponding net topology must be ensured. This is achieved by the introduction and usage of a unique *ID* for the Point Tubes and net components. Due to the usage of the Delta Storage concept unaltered vertices can be identified and referenced in the respective Point Tubes.

The merging process with the concept for the management of net topology changes is divided into the following sections:

- The management of a 4D model is divided into *n* stages, each stage begins with a change of the net topology and corresponds to a time interval.
- For time intervals $[t_{\text{change}}, t_{\text{change+1}}[$ with a constant net topology, the net topology is saved as $top\ (t_{\text{change}})$ and is used for the entire interval. The net topology is always applied to the union of all net components.
- If the net topology changes, the new net topology is stored as $top\ (t_{\text{change+1}})$. It is valid starting from the date of the introduction and is used till a further change.
- As a requirement for a time step with a net topology change, all involved net components need to have an explicit time step, see Figure 57.
- A net component is exclusively valid within a certain time interval $[t_{\text{change}}, t_{\text{change+1}}[$ and may differ in its temporal discretization from other net components within the same time interval.

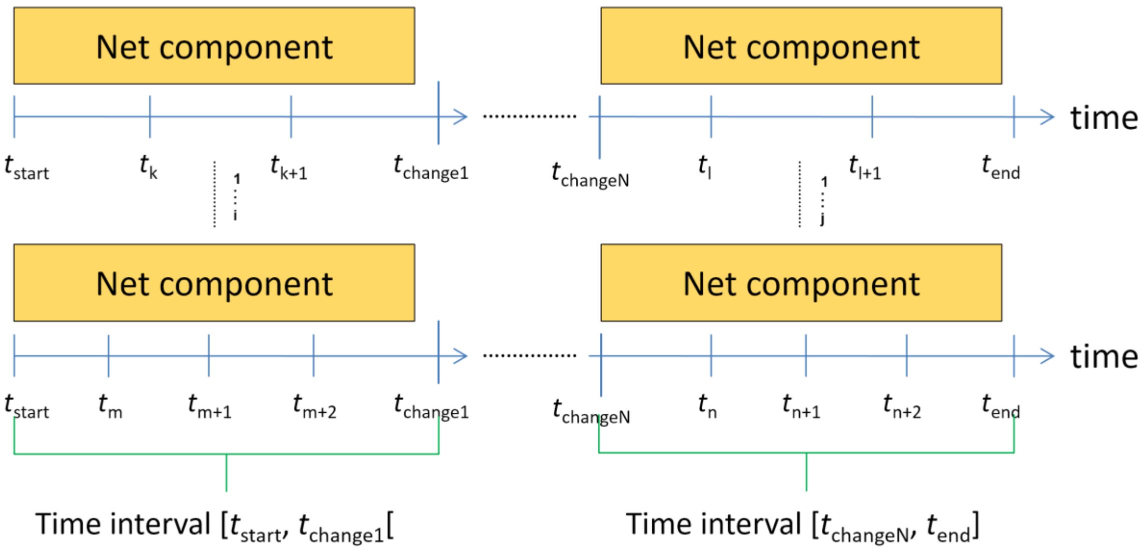


Figure 57 - Various net components with different intervals, which converge together for pre- and post-objects.

As discussed in Section 3.6.2 it is necessary to build new Point Tube structures for each post-object. However, when mapping the 4D model onto the internal data structure, the vertices that occur in both the pre- and the post-object are identified and referenced. The information of the corresponding vertices get transferred to the new Point Tube structure of the post-object.

If the particular 4D model is a 0D-net, there is no net topology that must be treated. However, a pre-object is still required to represent the spatial structure (i.e. shape) of the object at time step $t_{change+1}$ with the amount of vertices of time step t_{change}.

When working with 1-3D-nets (i.e. d-simplicial complexes with $d \in \{1,2,3\}$) the procedure is analogous to the one described in Section 3.6.2. The net topology of the entire net is stored for each time step with a net topology change (including the initial model) and is used for the entire time interval $[t_{change}, t_{change+1}]$.

Effect of the consolidation

If the net topology of a 4D model is not constant during the entire time interval the consolidation of the concepts implies specific conditions: each time step with such a change requires a pre- and a post-object. For this purpose, all involved net components need an explicit time step in the respective time interval.

Similarities between pre- and post-objects may be reused through the use of the Delta Storage concept, whereby storage space is saved. Further, the consolidation benefits from the specific advantages (reduction of data volume and the acceleration of certain operations) by the use of Point Tube data structures. For the periods without a net topology change the concept of Delta Storage is applied on the individual net components. Thereby storage space is saved analogously to the administration of regular *n*D-nets, cf. Section 3.6.1.

3.6.4 Consolidation with the Concept for a Differentiated Management of Thematic and Semantic Data

The concept for a differentiated handling of thematic and semantic data by an explicit storage and management of construction elements, can be realized in combination with the Point Tube concept. The vertices of all dimensions of the mesh Top (1D) up to Top (3D) of an *d*-simplicial complex are handled separately. Therefore, a specific model, e.g. a 3-simplicial complex for a time step t_{n} consists of the following information:

$$\text{model m } (t_{\mathrm{n}}) = \text{vertices } (t_{\mathrm{n}}) \cup \bigcup_{k=1}^{d} Top\ (k\mathrm{D})$$

The storage of thematic and semantic information takes place at the elements of each dimension of the mesh, cf. Section 3.4. The information are built by references to the elements of the lower dimension, e.g. Top (1D) is built from references of individual vertices. Accordingly, the various meshes of a specific model, e.g. Top (1D), Top (2D) and Top (3D) can contain appropriate semantic or thematic information.

Due to the Point Tube concept the vertices are stored separately from the net topology. This basic concept of the separation of vertices and the net topology combined with the management of vertices in Point Tube data structures has advantages in terms of performance and storage requirements as discussed in detail in Chapter 3.1. However, following the Point Tube concept, the mesh is only stored in the respective dimension of the net component. Therefore, e.g. a 2D-net component is a set of linked triangles that represent the net elements and form a 2-simplicial complex. These net elements reference on three vertices that are managed by Point Tubes data structures. The mesh of the construction elements can only be calculated *on-the-fly* and is not explicitly stored for 2D- and 3D-nets.

With the expansion of the comprehensive model by the concept for the explicit storage and management of construction elements an *n*D-net component therefore consists of the respective Point Tubes and the meshes Top (1D), ..., Top (n-1D), Top (nD).

At first sight this concept significantly increases the storage usage. But due to the combination of this explicit storage with the concepts Point Tubes and Delta Storage the additional storage overhead is considerably reduced since the individual meshes Top (1D) to Top (nD) only consist of references of the vertices and thematic or semantic information. Additionally, the information are not stored redundantly. Faces (segments in triangle nets or segments and triangles in tetrahedron nets), which are used by multiple elements in the higher dimension, are managed uniquely. Multiple usage of such objects lead to references to the same face, cf. Figure 58.

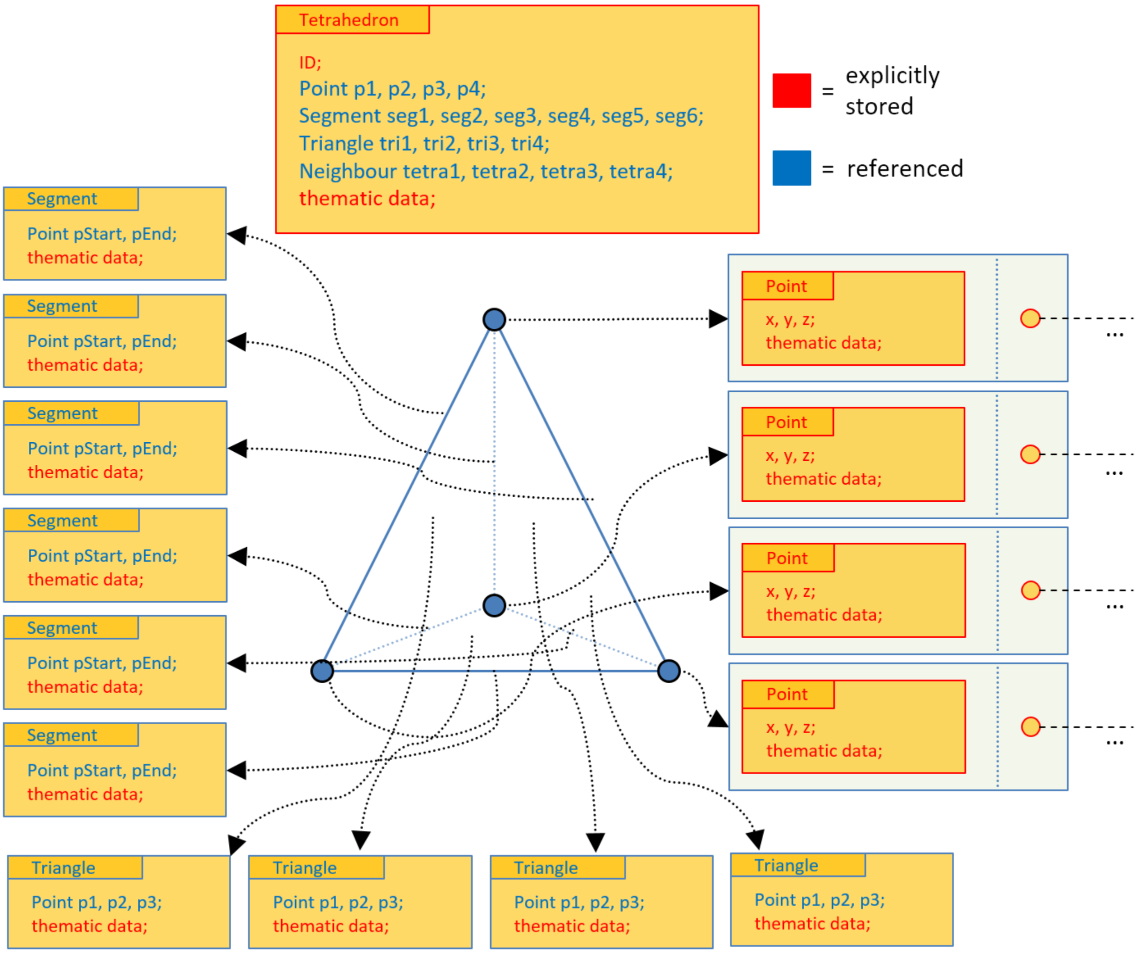


Figure 58 - Internal structure of a tetrahedron (3-simplex).

The concept of topology transitions, i.e. change of the net topology between two time steps, is realized as described in Section 3.6.2.

The concept to store construction elements explicitly does not interfere with the linking of 3D models that form a 4D model. Therefore, it is sufficient to examine a single time step. An exception to this are regions that are reused due to the Delta Storage concept. In such a case potential changes of thematic or semantic information has to be considered. In such a case it is only possible to reuse and reference the information about the mesh itself, but not the thematic or semantic information, respectively.

Effects of the consolidation

Due to the construction of a non redundant incidence model to handle the construction elements, the user is able to manage thematic or semantic information directly at the appropriate places. This extends the set of applications that can be handled by the comprehensive data management model. In combination with the concepts Point Tubes and Delta Storage the storage and computational requirements are significantly reduced for 4D models that are constructed by d-simplicial complexes with $d \in \{1,2,3\}$. However, additional computing capacity is needed to develop the structure of such a 4D model and to assign the constructional elements to the net elements without redundancy.

It is not to be expected that all applications need the explicit storage of thematic or semantic data at construction elements. In such cases it is sufficient to handle the vertices and the net topology of the respective net dimension following the principle of the Point Tube concept. Due to the additional storage requirements for this concept for the handling of $Top\ (1\mathrm{D})$ to $Top\ (n\text{-}1\mathrm{D})$, this concept should be optional.

The other concepts are independent of the explicit handling of construction elements. They can be calculated from the available data, if the concept of their explicit storage is not active at any time if required.

3.6.5 Summary of the Consolidation

The consolidation of the five different concepts was accomplished without any additional limitations or drawbacks. An exception is the concept of the explicit storage of construction elements. This concept should be only used if needed and therefore be optional. It should be enabled as soon as thematic or semantic data need to be stored and handled directly on the construction elements of a 4D model.

Figure 59 illustrates the use of the consolidated concepts. In this example the 4D model starts with time step t_{start} and consists of two net components in the first time interval $[t_{\mathrm{start}}, t_{\mathrm{change}}[$ and one net component in the second time interval $[t_{\mathrm{change}}, t_{\mathrm{end}}]$. The first component consists of five and the second of three time steps. There is a change of the net topology at time step t_{change}. The usage of the explicit storage of construction elements is exemplarily shown for the first time step t_1 in Figure 60.

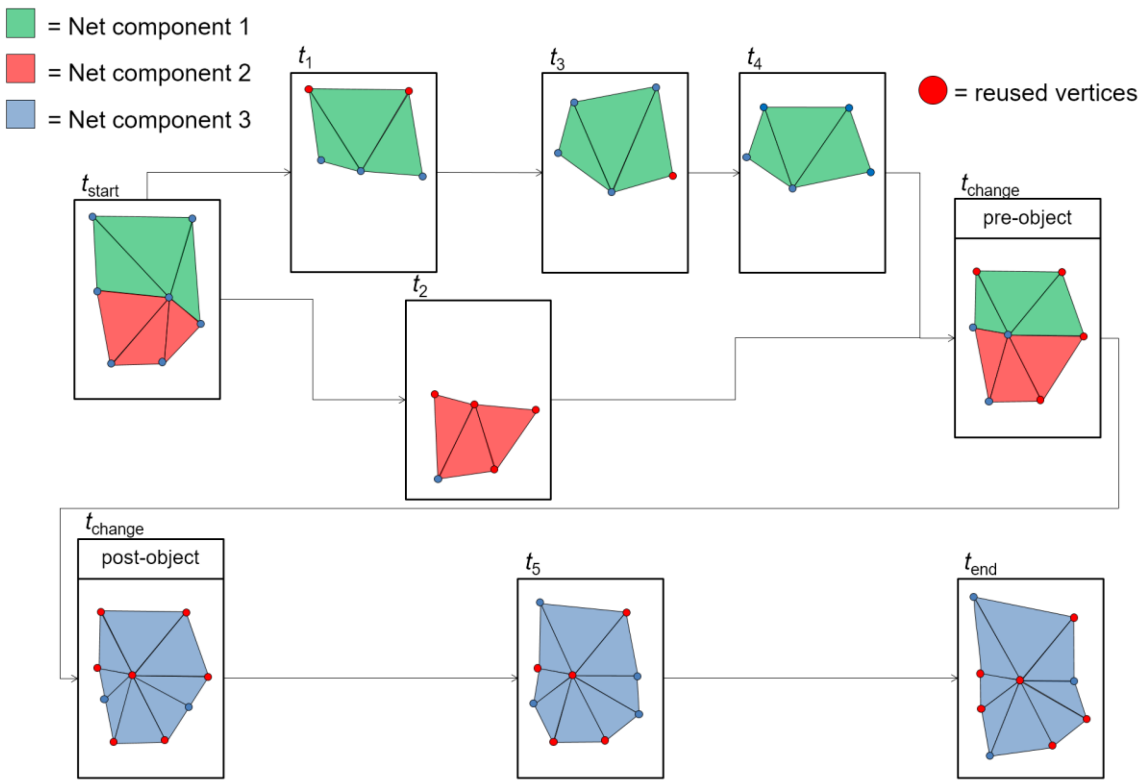


Figure 59 - Use of the five consolidated concepts.

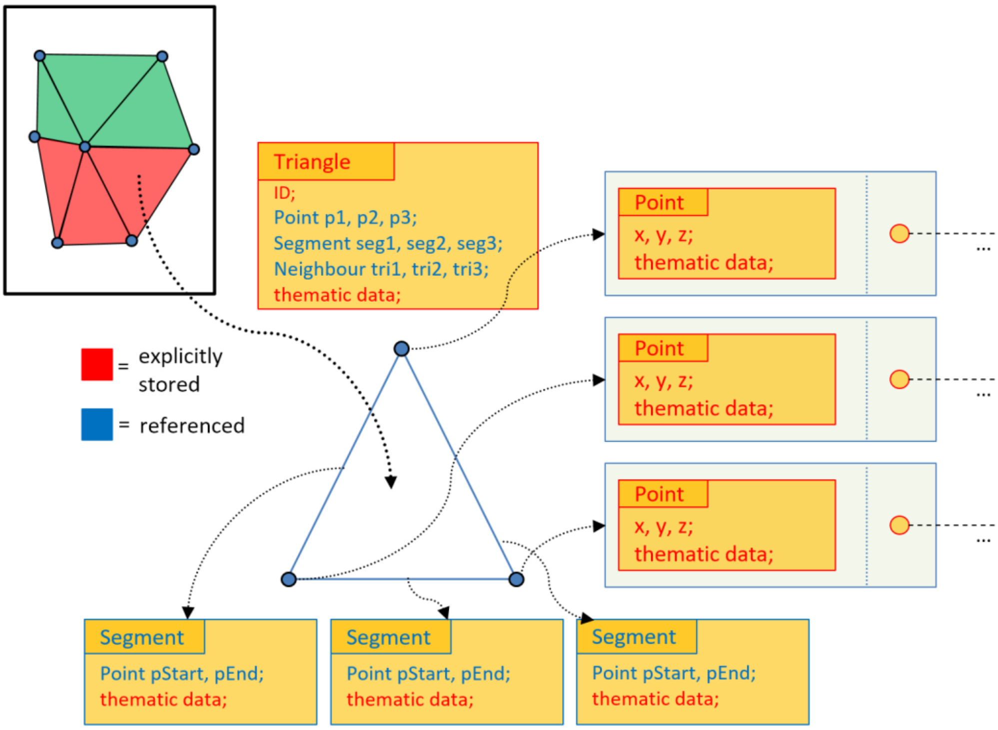


Figure 60 - Detailed view of a net element used in the 4D model of Figure 59.

Further details on a specific implementation of each concept within the implementation of the comprehensive data management model realized with an object-oriented programming language is part of Chapter 5.3.

3.7 Discussion

In the previous sections, a model for the management of spatio-temporal data that consists of five partial concepts has been presented. These concepts were examined and described individually. Afterwards they were adapted to meet the specific requirements that are relevant for this thesis. Subsequently, the combination of the individual concepts into a comprehensive data management model was realized and the impact of the consolidation was described in detail.

In this section the advantages and disadvantages of the result are discussed critically. For this purpose, a distinction is made between the effects that are dedicated to the interaction among the concepts and the effects of each individual concept within the comprehensive model.

If the comprehensive model is compared with the individual sub-concepts, it is clear that the sub-concepts do not adversely affect one another and can be combined in the described manner. Possible negative effects (trade-offs) of the individual concepts are not reinforced by a consolidation, but in some cases are even significantly reduced, especially in combination with the concepts Delta Storage and Point Tubes.

However, the use of the sub-concepts adds some additional computation and/or storage requirements. In consequence these are also required for the realization within the comprehensive model. It was possible to partially reduce the impact of these effects by the consolidation of the concepts.

The additional requirements of computation or storage capacities are divided into the following two categories:

1. Additional requirements on computing capacity

   For the concepts Point Tubes, the handling of different temporal discretizations and Delta Storage additional calculations are expected: for the composition of Point Tube data structures, the division into different net components and the examination of two time steps on unchanged subregions. However, for a 4D model these additional computational operations have to be performed only once and solely at the insertion of such a model into the data structure.

Once the 4D model is inserted into the geodatabase, computational resources are saved by the use of these concepts.

2. Additional requirements on storage capacity

The concept for handling net topology changes demands additional storage capacities: additional information of a pre-object have to be stored for every time step with a net topology change. This additional storage space is needed for the entire administration period of a 4D model. This also applies on the transfer of the data, once the complete 4D model needs to be transferred. However, this effect occurs only if a change of the net topology between two time steps is present. Otherwise, the realization of this concept does not need any additional storage space.

The concept for the explicit handling of construction elements requires additional storage space, as more information need to be handled and stored. This allocation of storage space remains active within the entire administration period of a 4D model.

However, under certain conditions the additional storage requirements are reduced significantly due to the concepts Delta Storage and Point Tubes, cf. Section 3.6.4.

The computational costs for the realization of the five concepts is only needed during the initial storage (insertion) of a 4D model and for the extension of an existing time series. Subsequently, the appropriate data structures are available for spatio-temporal queries and spatio-temporal operations.

It is expected that the process of inserting data into a geodatabase usually is a unique event during the administration of a 4D model. Afterwards the time series of such a model can be extended, but primarily data retrieval- and processing requests are expected. Additionally, during an insertion of new spatial or spatio-temporal objects, index structures such as the R*-Tree or the Octree are often initialized [Beckmann et al., 1990]. The costs that are required for the realization of the comprehensive data management model are expected to be less than the costs for the construction of such indices. Therefore, the additional temporary computational effort for the data management model is regarded as justifiable.

Additional storage space that is required by the concepts for the handling of net topology changes and the explicit handling of construction elements is needed within the entire lifetime of a 4D model. But the concept for the handling of net topology changes is used only if such a change occurs. Otherwise the use of the concept does not require additional storage space.

For the concept of the explicit storage of construction elements the additional storage space is needed for each time step with a change of the net topology, i.e. also for the time step t_{start}. This applies to all 2D- and 3D-nets, cf. Chapter 3.4. As mentioned before, the concept of the explicit storage of construction elements is not required for all applications and should be available optionally.

When working with 2D spatial databases it is common that, before inserting any data, the user of the database defines a *space* that fits the specific application, i.e. determining attributes and data types. For such a space policies and constraints can be established and be applied on the data that is handled within this space. This approach can be adapted to handle spatio-temporal data. Such a space definition i.a. specifies whether the construction elements of the net elements are managed explicitly or developed on-the-fly when needed. Due to this approach the allocation of additional storage space is avoided if an explicit handling of construction elements is not necessary.

The most important positive effects of the comprehensive data management model compared to a conventional approach are:

1. Reduction of storage space

   By separating the net topology from the vertices of a 4D model it is possible to save storage space, since the net topology is reused for multiple time steps. Furthermore, due to the realization of the indexed-face-set procedure the multiple storage of vertices at net elements is avoided. Construction elements are also handled uniquely if used by multiple net elements. Thus the *completeness axioms* are respected, cf. Chapter 2.2, and storage space is saved when managing 4D models based on *d*-simplicial complexes with $d \in \{1,2,3\}$.

   By the use of the Delta Storage concept the storage requirements are reduced for 4D models that do not change in partial regions within multiple time steps. For such regions only references to existing information need to be managed. The storage of redundant data is avoided.

   Due to the introduction of net components with various temporal discretizations different volatile subregions of a 4D model are treated accordingly. Therefore, it is possible to avoid the storage of the entire model in a high temporal frequency, which leads to an elimination of entire time steps for suitable subregions.

2. Acceleration of spatial and spatio-temporal operations

   Due to the introduction of Point Tube data structures many calculations, such as the translation of a 4D object, have to be performed exclusively for the vertices. These are separated from the net topology and can be accessed without the use of net elements. Vertices that are used by multiple net elements (within 1-3D-nets) are treated uniquely, which accelerates the processing of operations. In addition, the use of the Point Tube concept obtains the consistency of the net topology as the mesh is resistant against possibly arithmetic errors.

3. Extended field of applications

Due to the concept to support net topology changes between two time steps it is possible to handle 4D models that consist of regions which significantly change over time. Such changes are appropriately handled by the comprehensive data management model.

The concept for handling different temporal discretizations of partial regions extends the range of possible applications. Due to this concept it is possible to manage 4D models that consist of multiple regions with different volatile movements. This is the case, e.g. once a region changes faster than other regions and the monitoring is focused on this particular area. Due to this concept, it is possible to embed such regions in an appropriate spatio-temporal context. The entire 4D model is managed collectively in a single entity.

The explicit storage and handling of construction elements provides additional options to manage semantic and thematic information within a 4D model. Although, these construction elements can be generated on-the-fly if needed, due to the integration of this concept it is possible to store such information at the appropriate geometries permanently.

The reduction of the storage requirements by the use of the Point Tube concept is only relevant for 4D models that are based on *d*-simplicial complexes with $d \in \{1,2,3\}$, as 0-simplicial complexes do not have net elements that share vertices. Nevertheless, by the unambiguous assignment of vertices across the time steps of a 4D model, some important operations are supported, such as the interpolation between two time steps. This benefit applies on all net dimensions. Such operations benefit not only from the unambiguous structure of the individual vertices across different time steps, but also from the cumulative access to all vertices of a 4D model.

The use of the Delta Storage concept leads to a reduction of the required storage space, if regions are present that do not change between consecutive time steps. The examination of such regions consume computing resources. Therefore, this concept should only be activated with a promising data base. If such unaltered regions are present, the 4D model should be mapped onto the internal data structure by the use of the Delta Storage concept. In the following such a 4D model can be used without any restrictions.

The identified improvements for the acceleration of operations and calculations by the use of the Point Tube concept are expected to increase in higher dimensional nets, due to an improved treatment of vertices.

Managing 4D models whose net topology changes over time, extends the range of applications that can be handled by the data management model. As this concept does not require any additional resources, if the net topology does not change, it can be permanently activated without any negative impact.

The concept of net components with different temporal discretizations does also not require any additional resources when it is not needed. In such a case, the entire net consists of one single component. Therefore, only positive effects are expected and the concept should be activated permanently.

The explicit storage of construction elements extends the range of applications in particular when handling thematic and semantic data. In this case additional computational effort for the development of such construction elements, as well as additional storage requirements for the management are expected. Therefore, this concept should be activated exclusively if an explicit handling of construction elements is required. Negative effects are partly compensated by the combination with the concepts Delta Storage and Point Tubes. Table 4 summarizes the positive effects of the comprehensive data management model according to the individual sub-concepts:

Table 4 - Benefits of the particular concepts in storage and computational costs

Concept	Storage costs	Computational costs
Pont Tube concept	0 to O(n)[26]	0 to O(n)[26]
Delta Storage concept	0 to O(n)[27]	0
Handling of various net topologies	N/A	N/A
Use of net components	0 to O(n)[28]	0
Explicit handling of construction elements	N/A	O(n)

[26] Depends on net dimension
[27] Depends on net composition
[28] Depends on model composition

In summary, it can be stated that the previously discussed requirements are met by the comprehensive data management model due to the consolidation of the five sub-concepts, cf. List 1.

A particular implementation of the model based on the geodatabase DB4GeO is part of Chapter 5.3. A typical workflow based on the implementation with test data is presented in Chapter 5.3.5.

3.7.1 Constraints and Limitations

The presented data management model is suitable for 4D models that meet the specifications outlined in Chapter 2.3.2. Additionally, there are constrains that need to be met by an appropriate 4D model for a correct usage of the comprehensive model. These partly depend on the dimension of the particular net. The handling of 0-simplicial complexes is uncritical, but for d-simplicial complexes with $d \in \{1,2,3\}$ it must be assured that the structure of the particular simplicial complex is valid at any time. One key issue is the compliance of neighbourhood information of the net elements. A change of such information would result in a violation of the simplicial complex structure within a time interval. This should only be possible during an explicit change of the net topology and is than handled appropriately, cf. Section 3.3.

Another important constraint that refers to the compliance of the simplicial complex structure is a correct behaviour of border regions. These are particular vulnerable if a 4D model consists of multiple net components or immobile parts that are handled by the concept of Delta Storage, cf. Sections 3.2 and 3.5. In such a case the net topology must be in a consistent state within the entire time interval. It is the responsibility of the developer of a 4D model to consider these issues and to obtain the consistency of the net topology during the entire time interval. An implementation of the comprehensive data management model might report a violation but cannot solve it automatically. In practice such constrains are already respected during the construction of a single time step, i.e. a 3D model. This is partly due to the constraints which are imposed by 3D modelling tools such as GOCAD.

Another issue concerns thematic and semantic data. Currently the comprehensive model has advanced capabilities for handling such data at net elements and construction elements. Although the issue of interpolation was addressed for the geometric part of the model, there is no solution for an appropriate interpolation of thematic or semantic data. Currently thematic and semantic data are considered to be constant between two time steps.

4 Infrastructure for the Dissemination of Spatio-Temporal Data

The access to a central and common data storage is a key issue for a productive and collaborative workflow. To address this issue, Web services are used for the realization of data infrastructures and provide a regulated access to such data storages, cf. Chapter 2.5. Thus, multiple clients and users can simultaneously access the latest information and work on a common data basis.

In this chapter a generic approach for the dissemination of spatio-temporal data is presented. The requirements on an effective data infrastructure for spatio-temporal data are examined in Section 4.1. Subsequently, the individual components of such an infrastructure are investigated and illustrated. Explicit concepts for the realization of Web services for spatio-temporal data are presented and implemented in Section 4.2 and 4.3, respectively. These form the foundation for a standardized data exchange between various client applications, such as generic mobile GIS, cf. Chapter 5.

4.1 Requirements on a Web Service for Spatio-Temporal Data

To provide spatial and spatio-temporal data through a network interface to multiple user or client applications, it is essential to build an effective data infrastructure with appropriate Web services. Such a Web service for spatial data is also referred to as *Geo-Webservice*. The main task of a Geo-Webservice is the bi-directional transfer of spatial and spatio-temporal data. For this purpose, it is necessary to reply to requests from client applications and to transmit the corresponding data in a format that can be handled by such a client. It was proven that the use of standardized access methods is of crucial importance in order to cover a broad spectrum of client applications [Nogueras-Iso, 2005]. In terms of spatial data handling the development of such standards has mainly been influenced and promoted by the OGC, cf. Chapter 2.5.1. This includes the realization of the INSPIRE directive within the EU member states [INSPIRE, 2007].

The rough structure of a Geo-Webservice can be described as follows: on the server side, where the actual Web service is running, a data source is available, e.g. a geodatabase. Such a data source either contains the appropriate data or is able to store new information. It is also possible that a Geo-Webservice provides both options. The data specified by a request can either be transmitted to the client or integrated into the data source that is linked to the Web service. Usually such a service is realized as a client-server communication via the Internet and primarily serves as an interface for the exchange of data. For the implementation of Web services design patterns such as REST or SOAP can be used, which were introduced in Chapter 2.5.

In addition to the transfer of spatial and spatio-temporal data, a spatio-temporal Web service should offer the option to invoke various operations. The following list summarizes the requirements on a Web service for spatio-temporal data:

- Data-retrieval operations.
- Option to add or modify data.
- Various spatial queries, e.g. bounding box (BBox) queries.
- Various spatio-temporal queries, e.g. on the spatio-temporal representation of objects.
- Spatial or spatio-temporal operations on available data such as an interpolation or the average speed of objects.

The data retrieval for spatio-temporal data differs from the query of purely spatial data. Referring to spatial data, the return value can be determined by one or more of the following attributes:

- A unique identifier, e.g. object-ID.
- A range query using a BBox.
- Semantic or thematic attributes, e.g. type = highway.

Referring to spatio-temporal data, the result of a request may also be determined by these attributes, but the attribute *time* is added as an additional parameter. Accordingly, when handling spatio-temporal queries the following temporal parameters are taken into account:

- Time step, an explicit time step of the amount of time steps that are part of a 4D object.
- Time interval, a subset of the time interval that is valid for the correspondent 4D object.
- Time, a specific date within the valid time interval of a 4D object.

Referring to Web services it is common to clearly separate the process of transferring the data from the invocation of operations. When processing purely spatial data such operations are, e.g. *intersect* or *contains*. Web services for spatio-temporal data should provide access to spatio-temporal operations. As mentioned before, the requests of such operations need to include additional temporal parameters such as time intervals, time steps or particular dates that need to be processed accordingly. Typical examples of spatio-temporal operations on two spatio-temporal objects *A* and *B* implemented within this thesis are:

- *A covers B*: checks if object *A* covers object *B*, i.e. if $A \supseteq B$ within a time interval $[t_{\text{start}}, t_{\text{end}}]$.
- *A contains B*: checks if object *B* is always completely contained in object *A*, i.e. if $A \supset B$ and border(*A*) ∩ border(*B*) = ∅ within a time interval $[t_{\text{start}}, t_{\text{end}}]$.
- *A intersects B*: checks if the two objects *A* and *B* intersect, i.e. $A \cap B \neq \emptyset$ within a time interval $[t_{\text{start}}, t_{\text{end}}]$.
- *Interpolation (A)*: the object *A* gets interpolated for a specific date $t_x \in [t_{\text{start}}, t_{\text{end}}]$.
- *AverageSpeed (A)*: calculates the average speed of object *A*.

In addition to the indicated functionality of a Web service for spatio-temporal data the interoperability of such a service is of crucial importance. A Web service should be usable by a large amount of different client applications and the implementation of new client applications should be kept simple.

4.2 Realization of Web Services for Spatio-Temporal Data

Web services form the foundation of a data infrastructure and are frequently used in different applications. For this purpose, various Web technologies were developed such as *Simple Object Access Protocol* (SOAP), *Web Services Description Language* (WSDL), *Universal Description, Discovery, and Integration* (UDDI) [Alonso et al., 2004] and *Representational State Transfer* (REST) [Fielding, 2000], which were partly introduced in Chapter 2.5.1. The wide variety of Web services and Web technologies is partly due to different application scenarios and also a historical evolution. In this section the structure of Web services for spatio-temporal data is realized based on Web technologies that are most suitable to meet the discussed requirements.

For the separation of requests of spatial data and the call of spatial operations with the *Web Feature Service* and the *Web Processing Service* two independent standards for the development of Web services have been established by the OGC, cf. Chapter 2.5.1. These standards define operations as well as important parameter values that must be provided by such a Web service. The concept of an infrastructure for the dissemination of spatio-temporal data is based on these two standards to ensure the highest possible level of interoperability and to benefit from already established software components.

For the transfer of spatial data that is developed according to the Simple Feature Access standard, the OGC standard Web Feature Service is already popular[29]. However, the data model of this thesis is based on *d*-simplicial complexes and differs fundamentally, cf. Chapter 2.3.2. Therefore, two Web services for spatio-temporal data based on *d*-simplicial complexes are presented in the following. These use conventional WFS and WPS standards as a basic principle.

4.2.1 Web Feature Service

Spatio-temporal data should be available via a Web service for various client applications. To simplify an implementation of the *communication interface* on the client side and to exchange information with a broad base of already existing clients, the WFS standard of the OGC forms the basis of this Web service. The main focus is on the dissemination of spatio-temporal data by an appropriate realization of the operations, which are required according to the standard, cf. Chapter 2.5.1.

In Section 4.1 the requirements on a Web service that is specialized in spatio-temporal data have been described. These form the basis of the subsequent development of a Web service for the dissemination of spatio-temporal data. The result set of a query is not only determined by spatial, semantic or thematic attributes, but also by *temporal attributes*. Such a behaviour can be controlled with the help of WFS operations. In the backend requests are processed by a geodatabase that implements the comprehensive data management model presented in Chapter 3[30]. Therefore, the operations of an appropriate spatio-temporal WFS are structured as follows:

GetCapabilities

> Presentation of general information about the WFS including a list of all available *feature types*. In this case a feature type corresponds to a 4D object, managed within a 3D/4D geodatabase.

DescribeFeatureType

> Describes a specific feature type, i.e. a specific 4D object. This includes the *name* of the 4D object, the *geometry type* and a description of *additional attributes* incl. their data types. An important attribute is the *time interval* that describes the valid

[29] Supported by PostGIS, Q-GIS, GeoServer among others.
[30] An exemplary implementation based on DB4GeO is part of Chapter 5.3.

time interval of the 4D object. Another important attribute is *interpolation* that describes the type of interpolation used, e.g. linear interpolation.

GetFeature

A GetFeature request depends on the parameters that are provided through the request. If a complete feature type is requested, the result of this request is the entire 4D object, including all available time steps. When providing a *timestamp*, an explicit representation of the 4D object for this specific *date* can be requested. Such a representation (i.e. a 3D object) is generated in the backend by the geodatabase (e.g. by linear interpolation) and transferred to the client afterwards. In addition, the return object can be specified by the use of *semantic* or *thematic attributes*, e.g. type = sandstone.

These three operations form the basis of a WFS for spatio-temporal data. In this case the actual data is handled by a 3D/4D geodatabase in the backend.

Each operation includes various parameters that are specified within a request. Some of these parameters are optional, whereas others are mandatory. An overview of the relevant parameters is presented in Table 5.

Table 5 - Parameter for a spatio-temporal WFS query

Parameter	Definition	Potential values
SERVICE	Particular spatio-temporal service	WFS
REQUEST	WFS request operation	GetCapabilities, DescribeFeatureType, GetFeature
VERSION	WFS version	1.1.0
TYPENAME	Specification of the spatio-temporal data	MyNew4DSpace:Piesberg
TIME	Specification of the date	1989-01-01T00:00:00[31]

It is possible to invoke requests to the WFS via a HTTP GET call on a corresponding URL. Therefore, a *GetFeature* query is assembled as follows:

```
http://localhost/myWFSService/wfs?REQUEST=GetFea-
ture&TYPENAME=MyNew4DSpace:Piesberg&VERSION=1.1.0
```

[31] Time format: ccyy-mm-ddThh:mm:ss - https://www.w3.org/TR/NOTE-datetime

This specific request has no information about a particular date. At this point the WFS supplies the entire 4D object. If a specific timestamp of a 4D object is requested, the information must be provided within the request. Furthermore, the submitted date must be included within the time interval of the corresponding 4D object. Such a time interval can be requested, e.g., via a *DescribeFeature* query. A *GetFeature* request on a 4D object for a specific *date* is composed as follows and responds with a 3D object:

```
http://localhost/myWFSService/wfs?REQUEST=GetFeature&TY
PENAME=MyNew4DSpace:Piesberg&VERSION=1.1.0&TIME=1989-
01-01T00:00:00
```

As an alternative to GET requests on a particular URL it is possible to use POST requests with an XML document, cf. Chapter 2.5.1. Here the corresponding parameters, cf. Table 5, are integrated into the document as XML tags.

In both cases the requests are forwarded to a geodatabase. An overview of the structure of a WFS for spatio-temporal data is shown in Figure 61.

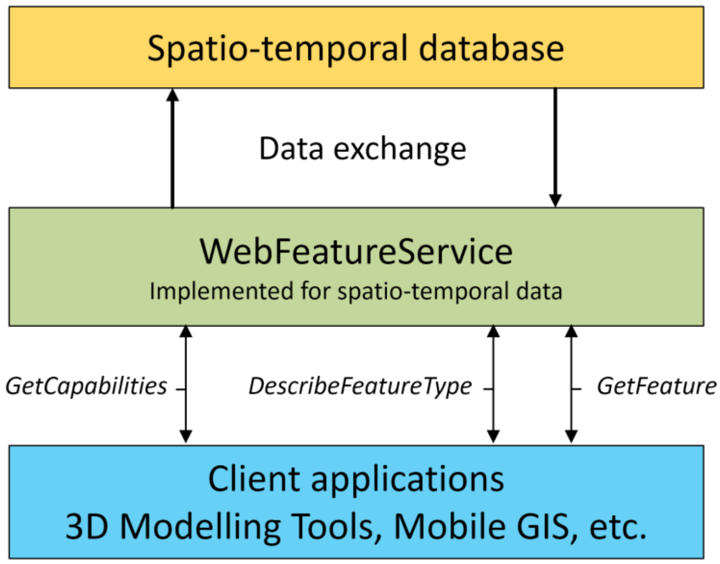


Figure 61 - Structure of a WFS for spatio-temporal data.

An explicit implementation based of the open source REST-framework library *Restlet* is part of Section 4.3.1.

4.2.2 Web Processing Service

Spatio-temporal operations shall be provided to various client applications via a Web service. A standardized approach provides such operations to a wide variety of clients and facilitates a simple implementation of new client applications. For this reason, the WPS standard of the OGC forms the basis for such a Web service. This Web service focuses exclusively on the provision of operations and analysis functions. For a realization of the WPS standard at least three operations are required by the standard and need to be implemented, cf. Chapter 2.5.1.

Following the principles of the WPS standard, the spatio-temporal operations and analysis functions are implemented and provided as *processes*. Each process operates independently and with an individual set of input parameters. In the backend such requests are processed by a GIS or a 3D/4D geodatabase. The minimal set of required operations for a WPS for spatio-temporal data are structured as follows:

GetCapabilities

> Provides general information about the WPS and delivers a list of available spatio-temporal operations that can be handled by the service. Such a list can be used by client applications to represent an overview of the capabilities of the specific WPS.

DescribeProcess

> Provides detailed information about a specific spatio-temporal operation. This includes a brief description of the operation itself and a list of all required and optional input parameters including their data types. In particular, among others these input parameters include the specification of a *timestamp* or a *time interval* and the involved data that is required for a specific operation.

Execute

> This operation starts a specific *process*. For this purpose, the corresponding and sometimes optional input parameters are submitted. The return value can be specified without restrictions and can be determined by an optional input parameter. The actual procedure takes place in the backend, e.g. in a 3D/4D geodatabase. Advanced spatio-temporal operations are, e.g. *averagespeed (A), A intersects B* or *A contains B*.

The three operations *GetCapabilities*, *DescribeProcess* and *Execute* form the foundation of a WPS for spatio-temporal data. The actual processes are performed within a 3D/4D geodatabase. Requests are invoked via a HTTP GET call to a corresponding URL. Therefore, the corresponding parameters must be committed in a key-value pair (KVP) representation. Table 6 provides an overview of suitable parameters.

Table 6 - Parameter for a spatio-temporal WPS query

Parameter	Definition	Potential values
SERVICE	Particular spatio-temporal service	WPS
REQUEST	WPS request type	GetCapabilities, DescribeProcess, Execute
VERSION	WPS version	1.0.0
IDENTIFIER	Particular spatio-temporal operation	averagespeed, intersect, contains
DATAINPUTS	Additional data input	TIME=1989-01-01T00:00:00; Spacename=MyNew4DSpace; Objectname=Piesberg

A typical *execute* request for the invocation of an intersect process is composed as follows (red = required parameters):

```
http://localhost/myWPSService/wps?REQUEST=Execute
  &IDENTIFIER=Intersect&VERSION=1.0.0
  &DATAINPUTS=[ObjectID1=23;ObjectID2=42;
  TimeStart=1989-01-01T00:00:00;TimeEnd=1991-01-01T00:00:00]
```

This query returns the value `true`, if there is an intersection between the two objects *23* and *42* within the time interval [1989-01-01T00:00:00, 1991-01-01T00:00:00] and `false` otherwise. The required input parameters and the correspondent data types for this request can be inquired via a *DescribeProcess* query. Different return values can be specified for other processes.

An alternative to the HTTP GET requests on a particular URL is a HTTP POST request with an XML document. In each case such requests are forwarded to a 3D/4D geodatabase or a GIS in the backend and is processed accordingly. An overview of the structure of such a WPS for spatio-temporal data is shown in Figure 62.

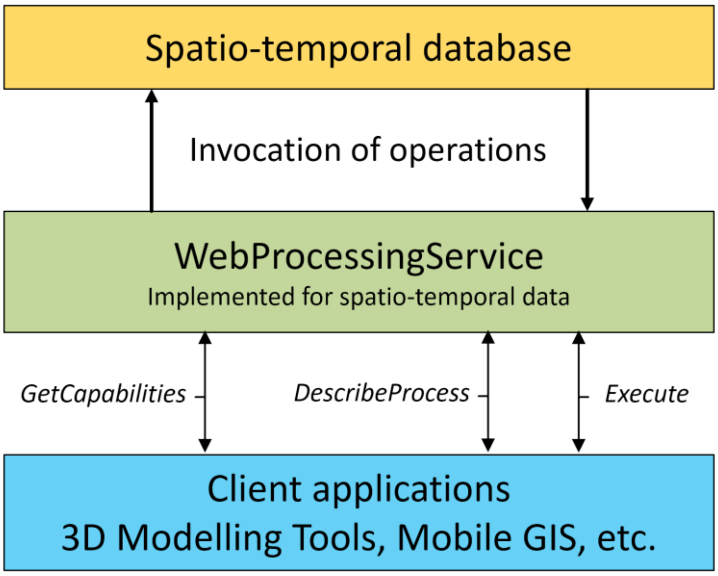


Figure 62 - Structure of a WPS for spatio-temporal operations and analysis functions.

An explicit implementation based of the open source REST-framework library *Restlet* is part of the following section.

4.3 Implementation and Integration into a 3D/4D Geodatabase

Due to an implementation of Web services for spatio-temporal data, a link to various clients such as generic mobile GIS can be established. During the design of such Web services the processing of requests for spatio-temporal data and spatio-temporal analysis capabilities were clearly separated and divided into two independent Web services. However, both Web services (WFS and WPS) are based on the same foundation as the workflow and the request handling basically follow the same principles.

First, it must be determined how to address the Web service. The specifications of the WFS and WPS instruct that queries are directed to an appropriate URL in two different ways: using a KVP-technique or with the help of XML documents, cf. Chapter 2.5. Basically, an OGC Web service can be rather associated to the category of SOAP services since requests are particularly specified by the use of XML documents. However, the implementation of a proper SOAP interface is optional and a pure REST interface is currently still under development, cf. Chapter 2.5.1. For this reason, the two common standard methods (KVP and XML) are implemented within this thesis. Therefore, the following modules are required:

- A *Web Application Framework* (WAF) for the processing of URL requests and the transport of data via the HTTP.
- An XML parser for the processing of XML requests.
- Connection to a 3D/4D geodatabase for the processing of spatial and spatio-temporal operations in the backend.

For the specific implementation of the Web services the Restlet[32] framework is used as a WAF. This framework provides an appropriate solution for the processing of the HTTP request types GET and POST and can be implemented directly on top of the API of a 3D/4D geodatabase. Due to a direct access of the API it is possible to maximize the functionality of the Web services. There are two types of XML parser to choose from, which differ fundamentally from each other: *Document Object Model* (DOM) and *Simple API for XML* (SAX). In this case a SAX parser is used due to a better memory handling. The 3D/4D geodatabase in the backend is realized by the geodatabase architecture DB4GeO, cf. Chapter 2.4.1 and 5.3.

4.3.1 Web Feature Service

The implementation of the WFS for spatio-temporal data is based on a WFS that was implemented for the exchange of 3D data based on *d*-simplicial complexes [Breunig et al., 2013b]. Therefore, an existing REST-based data access and processing framework for 3D data is used [Breunig et al., 2013a].

The main classes for the development of the WFS are the classes `OGCServer` and `OGCServices`. The class `OGCServer` starts the actual server on a system. Therefore, a port, e.g. 8182 must be specified. Afterwards the server is accessible via a specific URL, e.g. `http://myServer.gik.kit.edu:8182`. Requests are accepted at this URL. Therefore, so called *resources* are assigned to specific paths of the URL. Specific requests are assigned to corresponding resources and are processed further in appropriate classes. Following this approach, the WFS resource class `WFSObjectResource` was developed for the handling of specific KVP-requests and is registered as follows:

```
router.attach("/{projectname}/wfs?{query}",
WFSObjectResource.class);
```

[32] Restlet framework – an open source framework for the development of RESTful APIs: http://www.restlet.org

The class `WFSObjectResource` is a standalone resource class that is based on the class `BaseResource`. Here, the actual processing of the WFS request takes place. Initially, the request type (*GetCapabilities*, *DescribeFeatureType* or *GetFeature*) is determined and the parameters of the request are prepared accordingly. Depending on the request type a result is prepared, e.g. for a DescribeFeatureType request an XML document that contains the information about the 4D object is sent to the client. An overview of the class `WFSObjectResource` is illustrated in Figure 63.

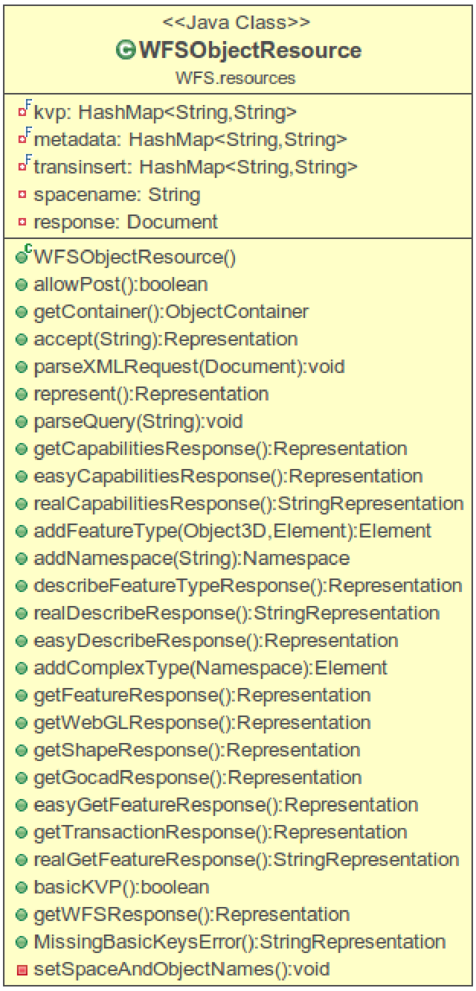


Figure 63 - UML class diagram of WFSObjectResource.

For the handling of POST requests that are used with XML documents, the resource class is registered as follows:

```
router.attach("/{projectname}/wfs",
WFSObjectResource.class);
```

In such a case the requests are processed in the class `WFSObjectResource`. The method `accept` was explicitly developed for the handling of POST requests. The submitted XML document is processed using a SAX parser that prepares the information accordingly. Subsequently, e.g. for a *GetFeature* request, the corresponding spatio-temporal data is queried directly from the 3D/4D geodatabase and is delivered to the client via HTTP. The client can specify an optional request parameter "output format" to choose between the output formats *GOCAD* and *WebGL*.

An overview of the implementation of the WFS for spatio-temporal data based on the Restlet framework is illustrated in Figure 64.

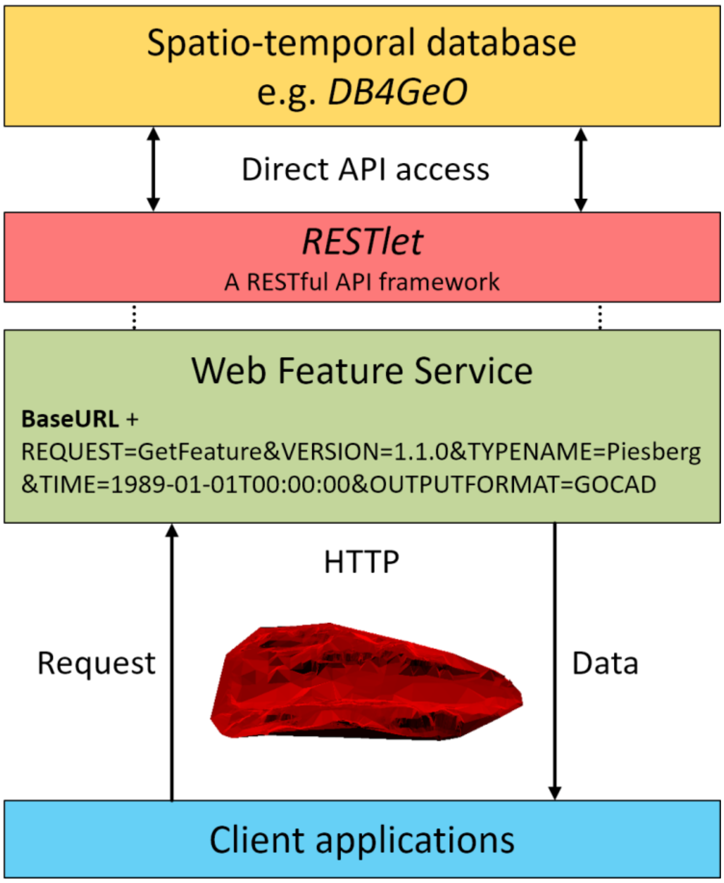


Figure 64 - Implementation of the WFS for spatio-temporal data based on the Restlet framework.

4.3.2 Web Processing Service

The WPS for the provision of spatio-temporal operations and analysis functions is based on first attempts on a WPS for a 3D geodatabase [Breunig et al., 2013b]. The main classes for the development of the WFS are the classes `OGCServer` and `OGCServices`, the same as for the WFS. The resource class `WPSObjectResource` preprocesses all WPS queries and is registered for handling of KVP-based requests as follows:

```
router.attach("/{projectname}/wps?{query}",
WPSObjectResource.class);
```

The class `WPSObjectResource` is a standalone resource class that is derived from the class `BaseResource`. In this class the actual processing of the WPS request takes place. Initially, the request type (*GetCapabilities*, *DescribeProcess* or *Execute*) is determined and the parameters of the request are prepared accordingly. Depending on the request a result object is prepared: e.g. for a DescribeProcess request an XML document is sent to the client that contains the information about the process that was specified in the query. An overview of the class `WPSObjectResource` is illustrated in Figure 65.

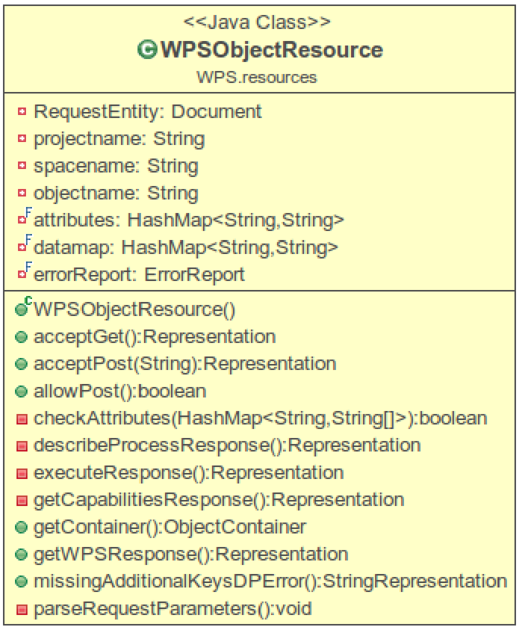


Figure 65 - UML class diagram of WPSObjectResource.

For the handling of POST requests at which XML documents are used for the specification of a query, the appropriate resource class is registered as follows:

```
router.attach("/{projectname}/wps",
WPSObjectResource.class);
```

The requests are processed within the class `WPSObjectResource`. The method `accept` handles such POST requests. The submitted XML document is processed by a SAX parser and the information of the XML document is prepared accordingly. Subsequently, e.g. for an *Execute* request, the corresponding spatio-temporal data is processed directly within a 3D/4D geodatabase. Afterwards the result is delivered to the client via HTTP. There are different results depending on the process. For instance, the process *averagespeed* returns a `double` value.

Spatio-temporal objects that are specified within a request are managed in a 3D/4D geodatabase, e.g. DB4GeO. An overview of the implementation of the WPS for spatio-temporal data based on the Restlet framework is illustrated in Figure 66.

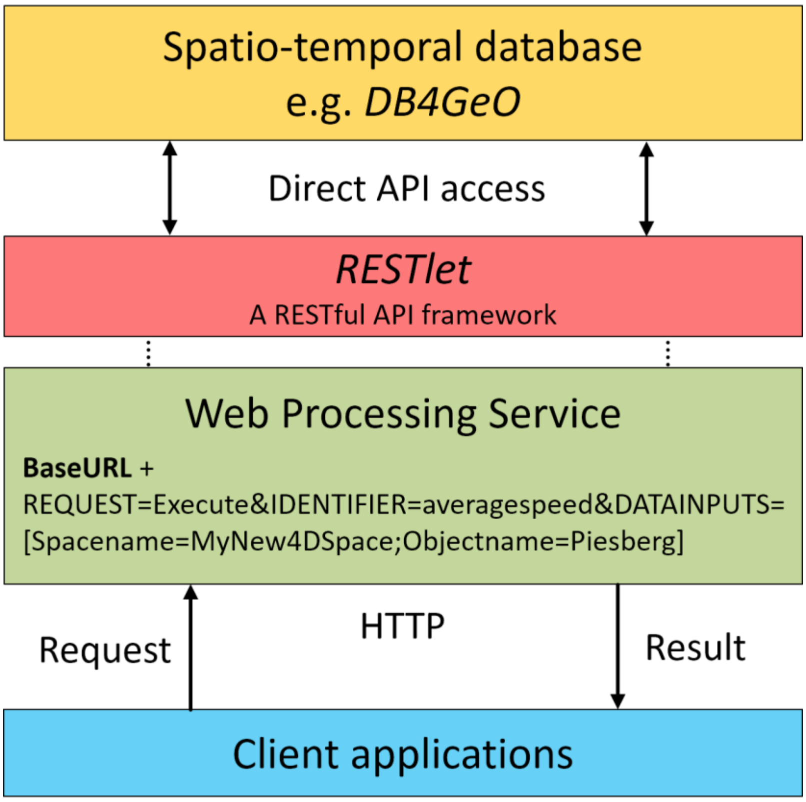


Figure 66 - Implementation of the WPS for spatio-temporal data based on the Restlet framework.

With the development of the two Web services WFS and WPS an interface between a spatio-temporal database and various client applications such as 3D modelling tools and mobile GIS has been established. It was shown that the adaptation of these two OGC standards is a suitable basis for the dissemination of spatio-temporal data and spatio-temporal operations. An implementation of the data management model for spatio-temporal data can be a suitable data source for such Web services, cf. Chapters 5.3 and 6.3.

5 Generic Mobile GIS

There exist two major groups of mobile applications that work with geodata: *location based services* (LBS) and *mobile GIS*. The majority of such applications focus on 2D data, cf. Chapter 2.6. Within this chapter a generic mobile GIS for spatio-temporal data is developed. First, a general concept for a generic and application-independent mobile GIS for professional users is presented in Section 5.1. In Section 5.2 this concept is extended to meet the specific requirements of a generic mobile GIS for spatio-temporal data. Section 5.3 describes the server-side implementation of the comprehensive data management model for the handling of spatio-temporal data that was presented in Chapter 3. Combined with the data infrastructure that was developed in Chapter 4 the preconditions for a realization of a generic mobile GIS for spatio-temporal data are given. The realization of such a generic mobile GIS for spatio-temporal data is presented in Section 5.4.

5.1 Concept for a Generic Mobile GIS

First, the general requirements of a generic mobile GIS application for professional users are defined as follows:

- Visualization of spatial data, including highlighting, generalisation, symbolising, etc.
- Integration into an existing geodata infrastructure to exchange spatial data (with appended attributes and meta-data) bidirectionally, i.e. to be capable of surveying new spatial data.
- Support of a large set of different applications such as public transport, surveying of vegetation, land registry or others that visualize and/or survey geodata, with and without attributes and meta-data.
- Arrangements and precautions to avoid an invalid data input and provide a consistent spatial data stock.

The main requirements of a standardized mobile GIS for professional users are a suitable user interface, an appropriate integration into an existing geodata infrastructure, and the potential to adapt the application to the users' needs. To meet these requirements, it is necessary to have a closer look at the data model, the process of exchanging spatial data and the user interface with its capabilities of interpreting the users' input. Thus the use of the following concepts is proposed:

- Deployment of OGC Web services, e.g. WebMapService (WMS) [OGC WMS] and Web Feature Service (WFS) [OGC WFS] for a standardized and conventional spatial data communication following the INSPIRE guidelines [INSPIRE, 2007].

- Standardized and generic visualisation and symbolization following the rules and concept of OGC Styled Layer Descriptor (SLD) [OGC SLD].
- OGC WFS-T (transactional) concept to provide application dependent templates for the surveying of spatial data including attribute data and meta-data.
- Use of the Simple Feature Access (ISO 19125 standard) [OGC SF].
- OGC WFS-T concept to ensure exclusively predefined and suitable data types (consistency checking). The locking mechanism provide a reliable multi-user support.

The main purpose of professional mobile GIS is to *visualize*, *survey* and *edit* spatial data, with and without attributes or meta-data. When it comes to visualization the user expects a clear focus on the current workflow and the spatial data that is involved.

Regarding the user interface the mobile GIS application has to provide an adapted arrangement of buttons and other suitable input tools for the professional user to support the particular task. Previous work done in this field target the reduction of the cognitive load for the user to finish a certain task [Raubal and Panov, 2009], [Doyle et al., 2010] and [Kadlec et al., 2012].

Regarding the visualization of spatial data, there are two options to achieve an adaptable and generic rendering: either the mobile GIS application itself takes control of a suitable visualisation of the spatial data or the visualisation is handled by using the SLD standard. The second option obviously is more generic, but is also more challenging, due to the fact that the mobile GIS application has to interpret SLD files for the matching geodata. As generic mobile GIS application should be as generic and adaptable as possible SLD files shall be used for the visualisation of spatial data.

A professional and generic mobile GIS application should be capable to communicate with as many data sources as possible. Therefore, one of the most disseminated communication concepts for the exchange of spatial data is used: OGC Web services. These Web services, especially the WMS (capable of delivering already rendered geodata in form of tiles, e.g. for a base map) and the WFS (capable of handling vector based geo data, e.g. *point*, *line*, and *polygon*) are widely used and reviewed as well as tested in many different applications for the exchange of spatial data. Additionally, the coverage of such Web services is still extending due to the realization of the INSPIRE directive within the European Union. To maximise the coverage of possible sources for spatial data the two mentioned Web services shall be implemented within a generic mobile GIS application. Figure 67 illustrates the requirements of a generic mobile GIS for professional users and possible concepts for a realization of such a system.

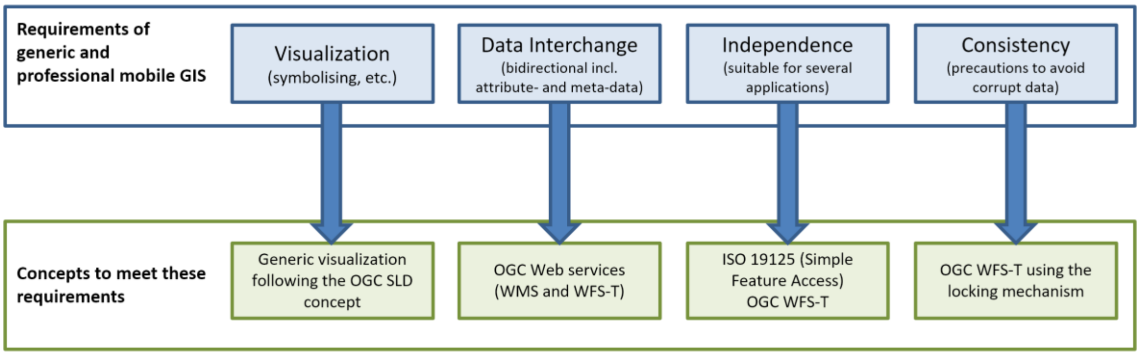


Figure 67 - Requirements of generic mobile GIS for professional users.

Although there is only one main task for most mobile GIS and LBS applications, namely to visualize and/or survey new spatial data, there are countless different highly specialized mobile GIS and LBS applications available on the market. Almost all of these applications are designed to support the user in only one specific task or issue [Han et al., 2015], [Zajíčková, 2015], [Kang et al., 2015]. This implies several applications that often differ only in a nuance. A generic mobile GIS, however, should be capable of dealing with several different tasks. To achieve this goal, it is necessary to have a closer look at different mobile GIS and LBS applications. Whenever the function of a mobile GIS or LBS application is to visualize spatial data in a certain way, it is likely that the use of the SLD concept solves this task. Due to the SLD concept it is possible to affect the appearance of raster and vector based geo data in a mobile GIS or LBS application in several ways, e.g. *colour*, *size*, *symbolization*.

If the mobile GIS needs to deal with additional information, e.g. meta-data or attribute data of a certain spatial object, this information is queried in a standardized way by using the concept of a WFS. For such tasks the *GetFeature* function of the WFS concept is applied to solve queries such as "Is this building a fire station?" or "What type of tree do we have here?". Most of such information is available for objects and is provided by the data infrastructure, i.e. a geoserver. In some cases, it is possible to calculate such information on the server by using a WPS. Following this approach, it is possible to answer queries of the type "How large is the area covered by this certain object?". Due to these Web services it is possible to access additional information about spatial objects in a prescribed way. The mobile GIS application needs to parse the query result and visualize the information for the user in a suitable way.

Another important issue of a mobile GIS application for professional users is the extension of the data pool by surveying new spatial data or by altering already existing data. When it comes to survey new spatial data any kind of application requires the realization of several constraints: usually the user intends to store additional data as meta-data (e.g.

estimated accuracy=5 meters) and attribute data (e.g. type=tree, height=8 meters) linked to spatial data. The concept of the WFS-T is capable of fulfilling these needs for *points*, *lines* and *polygons* (with and without holes), which cover most of the application areas of mobile GIS. The mobile GIS application itself just needs to provide a suitable template for the user to enter such data. Following the WFS-T concept all templates for specific spatial data are predefined before the user goes out in the field to survey new data. This may seem to be a disadvantage at the first sight due to less flexibility for the user, but it is actually a huge advantage to obtain the consistency of spatial data. Quality and consistency of data [Neis et al., 2011] is one of the key challenges when surveying spatial data. Due to the use of the WFS-T concept the data that are surveyed by the user is in a consistent state and input errors (wrong data types, misspelled names, etc.) are reduced significantly. All attribute names are predefined by the indoor services and have an appropriate storage space on the geoserver. New attribute types including the parameter values can be defined in the office instead of creating them in the field. Following these rules, potential errors like spelling errors and invalid parameter values are avoided. Figure 68 illustrates the structure of a workflow for a generic mobile GIS.

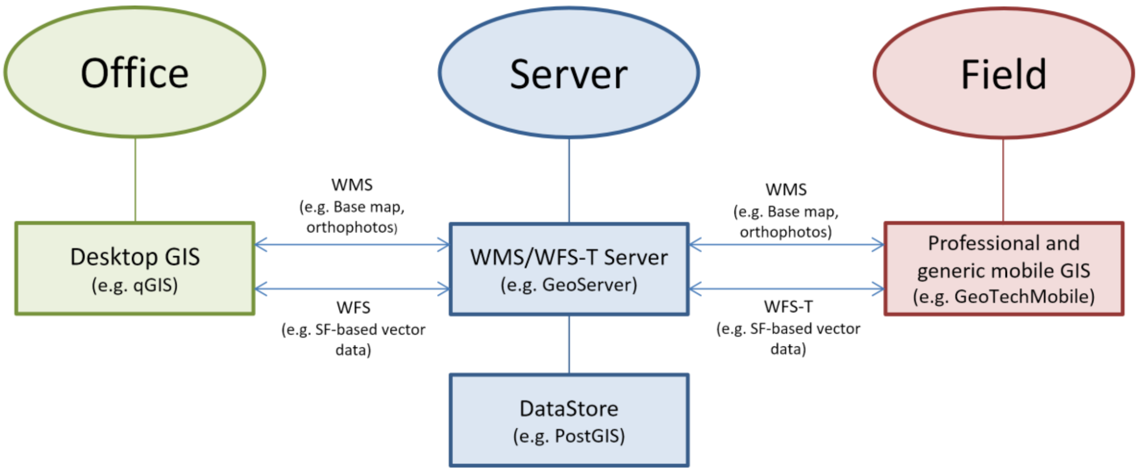


Figure 68 - Structure of a workflow for a generic mobile GIS.

One of the most advanced functionalities of mobile GIS is to extend or alter already existing data in different ways. This includes the spatial data itself or its attribute and meta-data. One way to achieve this functionality is to follow the rules of the *locking mechanism* of WFS-T. Therefore, it is possible to establish a *lock* on certain spatial data for a certain amount of time. Within this duration only the system that allocated the lock is allowed to alter the data. This concept is used to alter or update either the spatial data itself or the meta-data respectively attribute data [OGC WFS]. It also sorts out multi user issues by prohibiting other users to acquire a lock for the same data for an overlapping time interval.

Suitable for most 2D applications and already frequently used is the *Simple Feature Access standard* (ISO 19125) [OGC SF]. The characteristics and special requirements of a mobile GIS for the handling of three-dimensional or spatio-temporal data are discussed in the following chapter.

5.2 Special Characteristics and Requirements for Spatio-Temporal Data

The approach for a generic GIS for professional users that was described in the previous section is i.a. based on a number of concepts developed by the OGC. These were primarily developed for conventional 2D GIS applications and the processed data is based on the Simple Feature Access data model. There are additional requirements for applications that deal with spatio-temporal data, especially if the data is based on the data model of *d*-simplicial complexes, cf. Chapter 2.2. Applications that use this kind of data are often originated in the research fields of *geothermal energy*, *geophysics* and *geomorphology*. Therefore, surface- and subsurface-structures of the earth are modelled with appropriate thematic and semantic data. Corresponding 4D models are developed i.a. on the basis of bore holes, seismic recordings, terrestrial surveying or by the use of photogrammetric methods. The complexity that is required for the construction of such a model cannot be solitary placed into a mobile GIS application. Therefore, the focus of mobile GIS for such models is on the *visualization* of the data and on the *provision of additional attributes and meta-data* rather than on the construction of such models. In addition, *analysis functions* and *spatio-temporal operations* shall be provided. Thus, the concept of the OGC Web service WFS-T to survey new spatio-temporal data is of no interest for this application area.

The adaption of the concept for a generic mobile GIS presented in the previous section onto the handling of spatio-temporal data is divided into two parts: *data visualization* and *data transfer*.

Requirements for the visualization of spatio-temporal data on a generic mobile GIS

The visualization of spatio-temporal data that is based on the model of *d*-simplicial complexes is delegated to the visualization of a model $m\ (t_x)$ that represents the state of the 4D model m for a specific time $t_x \in [t_{start},\ t_{end}]$. For the visualization of such 3D models particular methods are needed that provide a three-dimensional impression on a two-dimensional output device such as a tablet or smartphone. For this purpose, it is common to work with sources of illumination and shadows [Phong, 1975]. Another important issue is the interaction with these objects. The handling is different compared to the interaction with two-dimensional data. Objects shall be *rotated* and *panned* within the space to be examined from different perspectives. A special feature of the visualization of three-dimensional data is the *augmented reality* (AR) mode, whereas the camera image of the device is superimposed with spatio-temporal data. Therefore, an exact location of the device and also the viewing direction is needed. The use of location sensors such as the *GNSS sensor*, *compass* and the *gyroscope* is essential. In addition, photogrammetric methods can support the correct representation of the spatio-temporal data on the screen [Urban et al., 2013].

In order to visualize the temporal component of a 4D model, the interpolation between different time steps of the model can be realized as a continuous morphing process. Particularly, when handling larger amounts of data high demands on the computing power of the hardware are expected. The consideration of the temporal component by the visualization of individual 3D models, which correspond to specific states within the time interval of the 4D model, significantly reduce the demands on the computing capacity. This approach also corresponds to the use of spatio-temporal data within frequently used modelling tools, cf. Chapter 2.3.

The visualization of the data depends on the application. In many cases it is sufficient to solely illustrate the spatial part of the data. But if attributive or semantic data are available (e.g. a temperature profile) the OGC standard SLD is used to describe an appropriate representation.

Requirements for the transfer of spatio-temporal data to generic mobile GIS

The WFS for spatio-temporal data, realized in Chapter 4.2.1, can be used to transfer spatio-temporal data to a generic mobile GIS. Following this approach, the data transfer is standard compliant and individual components such as the database server remain replaceable. For a practical transfer of data any format can be selected. However, it is advisable to choose a format which uses as little storage as possible to reduce the amount of data that needs to be transferred.

For the invocation of spatio-temporal operations upon the data the WPS OGC standard can be used. The actual process takes place on the server side and afterwards the result can be transmitted to the mobile GIS. Such a WPS was presented and realized in Chapter 4. An overview of a suitable geodata infrastructure for spatio-temporal mobile GIS is illustrated in Figure 69.

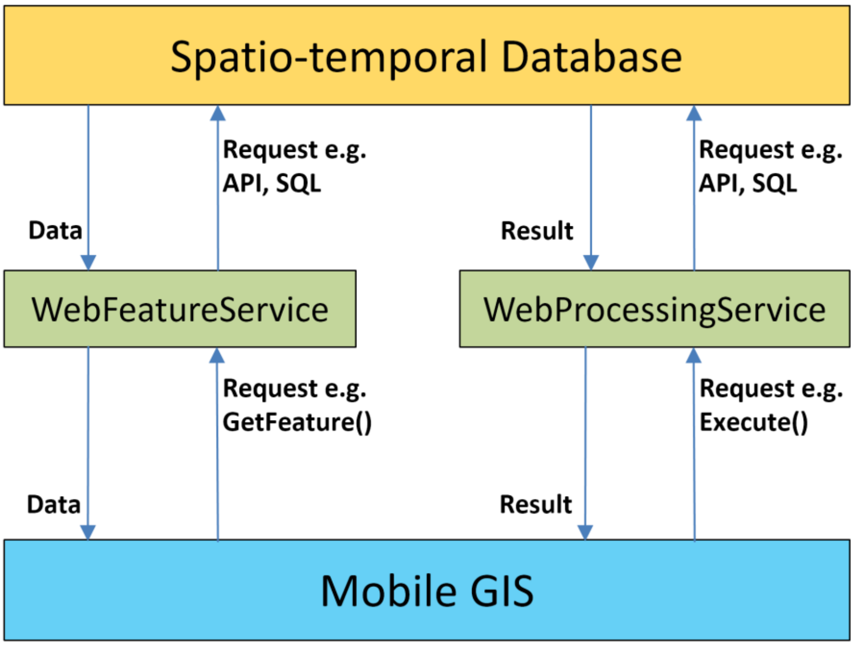


Figure 69 - Communication between a spatio-temporal database and a generic mobile GIS for spatio-temporal data.

For the realization of a generic mobile GIS for spatio-temporal data a geodatabase that implements the proposed data management model for spatio-temporal data has to be implemented, cf. Chapter 3. Such a geodatabase forms an important part of the backend. It handles appropriate data and processes spatio-temporal operations. The implementation of such a geodatabase is subject of the next section.

5.3 Server-Side Implementation

For a proper realization of a generic mobile GIS for spatial-temporal data, the data management model for spatio-temporal data needs to be implemented on the server side as presented in the following. First, it is outlined how the model is integrated into the database architecture *DB4GeO*, which forms the basis of the implementation, cf. Chapter 2.4.1. Subsequently, the implementation of the data management model is described in detail. Typical algorithms are demonstrated in the form of Java code. The software itself is available as free open source software[33].

[33] Db3dcore can be found at https://github.com/geodb/db3dcore

5.3.1 Integration of the Data Management Model into DB4GeO

As pointed out in Chapter 2.4 an object-oriented DBMS is an ideal basic platform for the management of complex geoscientific data. Therefore, the object-oriented geodatabase architecture DB4GeO is used for the implementation of the data management model presented in Chapter 3. In DB4GeO it is currently possible to create and manage spatial models based on d-simplicial complexes with $d \in \{0,1,2,3\}$. In addition, an existing project management and a variety of tools can be used to invoke spatial operations and requests. This preparatory work form an appropriate environment to integrate the comprehensive data management model into the geodatabase architecture.

The majority of the developed classes are integrated into the core of the geodatabase architecture (`db3dcore`), which is published as free and open source software[33]. This includes the basic *geometry classes* of the 4D part and the so-called *builder classes* to compose new 4D objects from 3D models, cf. Chapter 2.3. Figure 70 illustrates the integration of the comprehensive data management model into DB4GeO.

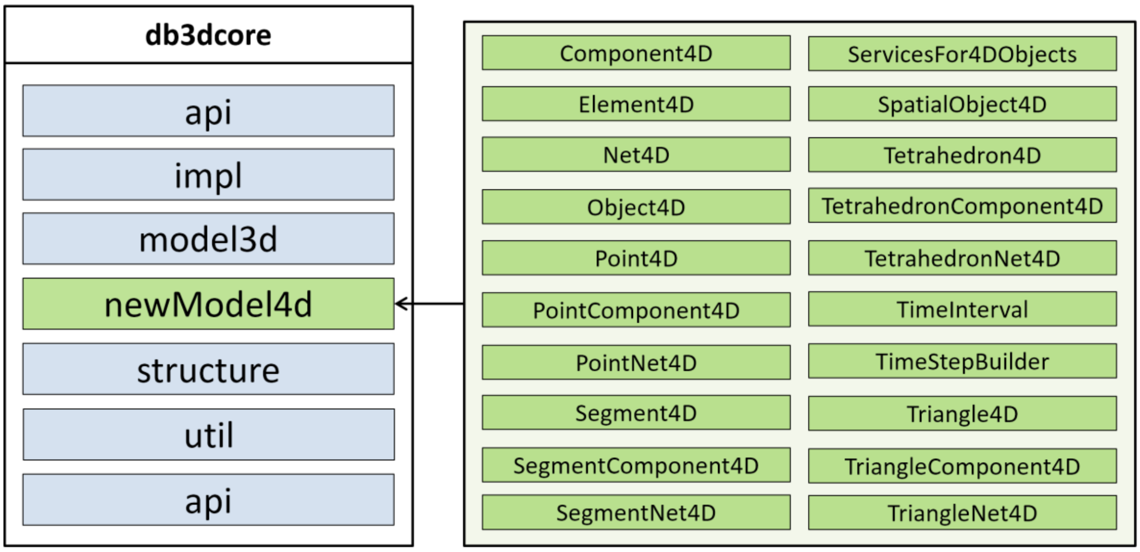


Figure 70 - Integration of the package *newModel4d* into *db3dcore*.

The data management model for the handling of 4D objects is implemented on the same level as the package for the handling of 3D objects to combine both, the geometry classes for 3D and 4D in one module. Furthermore, classes and methods for the processing of spatio-temporal data from ASCII files are realized as *importer classes* and are integrated into the package `db3d` which is based on top of `db3dcore`. In addition, this package includes different *service classes* for managing the net topology structure of 4D models. The test environment, which is part of Chapter 6.1 is also part of this package.

Import routines for a straight forward integration of GOCAD models were integrated into the existing package `operations.io.gocad`. The following Figure 71 illustrates the integration into the `db3d` package.

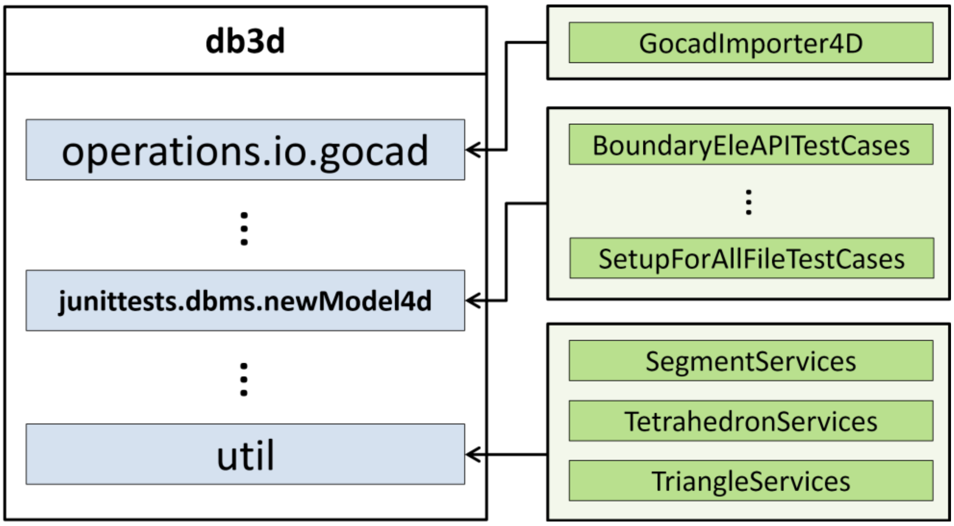


Figure 71 - Overview of the integration into the db3d package structure.

Within the implementation of the data management model the concept of polymorphism is frequently used. Therefore, parts of the implementation can be used for all dimensions $d \in \{0,1,2,3\}$ of a *d*-simplicial complex. Among other things, important functions are abstracted through interfaces. For instance, these are used in various service classes and in several access methods. This approach was used within the entire development process of DB4GeO. The following sections describe the development of key components of the data management model in detail.

5.3.2 Development of a 4D Kernel

The structure of a 4D model that consists of an *n*D-net is represented by interfaces of the geometry core. Such an *n*D-net may consist of multiple net components with different temporal discretizations. These net components are composed of elements, i.e. *n*D-simplices. The correspondent interfaces developed for this procedure are `Net4D`, `Component4D` and `Element4D`. These interfaces form the basis for the implementation of the data management model and define methods that are valid for each correspondent hierarchy level of all net dimensions $d \in \{0,1,2,3\}$. Thus, large parts of the code base for the construction of 4D models are used for all net dimensions, since the appropriate procedures work with methods of the respective interfaces.

General structure of the 4D kernel

The class `Object4D` mainly consists of the `Spatial4D` object that consists of a `Net4D` object. The `Element4D` objects of a net describe the net topology of a net and consist of an amount of *IDs* (1-4, depending on the dimension) that are assigned to individual vertices. The vertices and the implementation of the Point Tube concept are managed in the class `Component4D` by an associative storage of linked lists. Due to the association of the elements within the `Net4D` classes these elements are used across multiple components, cf. Chapter 3.2. The structure of the entry class `Object4D` is illustrated in Figure 72.

An explicit `Net4D` class was developed for all four net-dimensions which implement the interface `Net4D`, i.e. `PointNet`, `SegmentNet`, `TriangleNet` and `TetrahedronNet`. The respective `Net4D` classes are associated with elements of the corresponding simplex dimension, i.e. `Point4D`, `Segment4D`, `Triangle4D` or `Tetrahedron4D`. The concept of the net components with different temporal discretizations is realized with an amount of appropriate components, i.e. `PointComponent4D`, `SegmentComponent4D`, `TriangleComponent4D` or `TetrahedronComponent4D`.

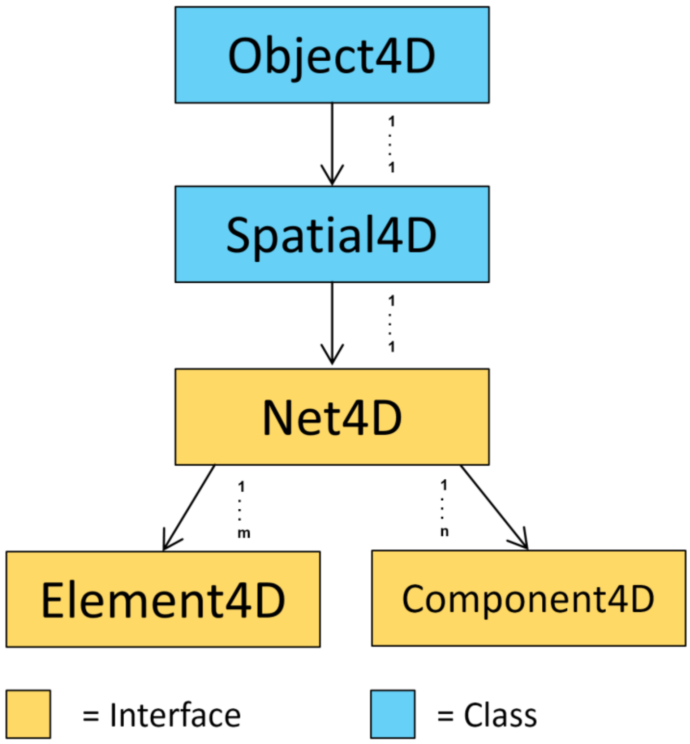


Figure 72 - Overview of the structure of an *Object4D* in DB4GeO.

The data of a 4D model that consists of a time series is always mapped to the appropriate classes. An explicit workflow is presented in Section 5.3.5. Within this process the individual sub-concepts of the data management model are used. In the following the implementation and the interaction of the individual sub-concepts are described based on various classes and methods.

Interface **`Net4D`** and explicit **`Net4D`** classes

Besides *getter*-methods for the data fields of explicit `Net4D` classes, the methods `topologyChange(Date)`, `addChangeTimestep(Date)` and `preparePostObject(Date)` are declared in the interface `Net4D`. These are needed for the composition of a 4D model and for the realization of the topology change concept.

The explicit `Net4D` classes have various data fields. Most important are the associative storages `components(Integer,Component4D)`, `timeIntervals(TimeInterval,List <Integer>)` and `elements`, a list of associative storages (`Integer,Element4D`). `Components` manages the net components with a corresponding unique *ID*. Due to the ability to manage multiple components within a net, the concept of different temporal discretizations is realized.

The time intervals that are managed in `timeIntervals` assign individual time intervals to correspondent net components. An object of the type `TimeInterval` consists of a `start` and `end date`, which describes an interval $[t_{\mathrm{start}}, t_{\mathrm{end}}[$ and forms the scope between a *post*- and a *pre-object*, cf. Chapter 3.3 and 3.6.2.

The list `elements` manages the elements of a net and therefore handles the net topology, which is handled separated from the vertices, cf. Chapter 3.1. Each entry in this list represents a net topology that is valid within a correspondent `TimeInterval` of the list `timeInterval`. The respective `Element4D` objects consists of IDs that are clearly associated with the vertices managed in Point Tube structures within the `Component4D` objects.

In addition, it is possible to optionally store references to construction elements at `TriangleNet4D` and `TetrahedronNet4D` objects, i.e. segments for `TriangleNet4D` objects and segments plus triangles at `TetrahedronNet4D` objects. Therefore, the concept of an explicit storage of construction elements is realized, cf. Chapter 3.4. Such construction elements are also managed in lists of associative storages (`Integer`, `Element4D`) likewise the elements of a net.

To provide access to nets of different dimensions, the interface `Net4D` is implemented by the classes `PointNet`, `SegmentNet`, `TriangleNet` and `TetrahedronNet`.

Figure 73 shows the data field and method declarations using the class `TetrahedronNet4D` as an example:

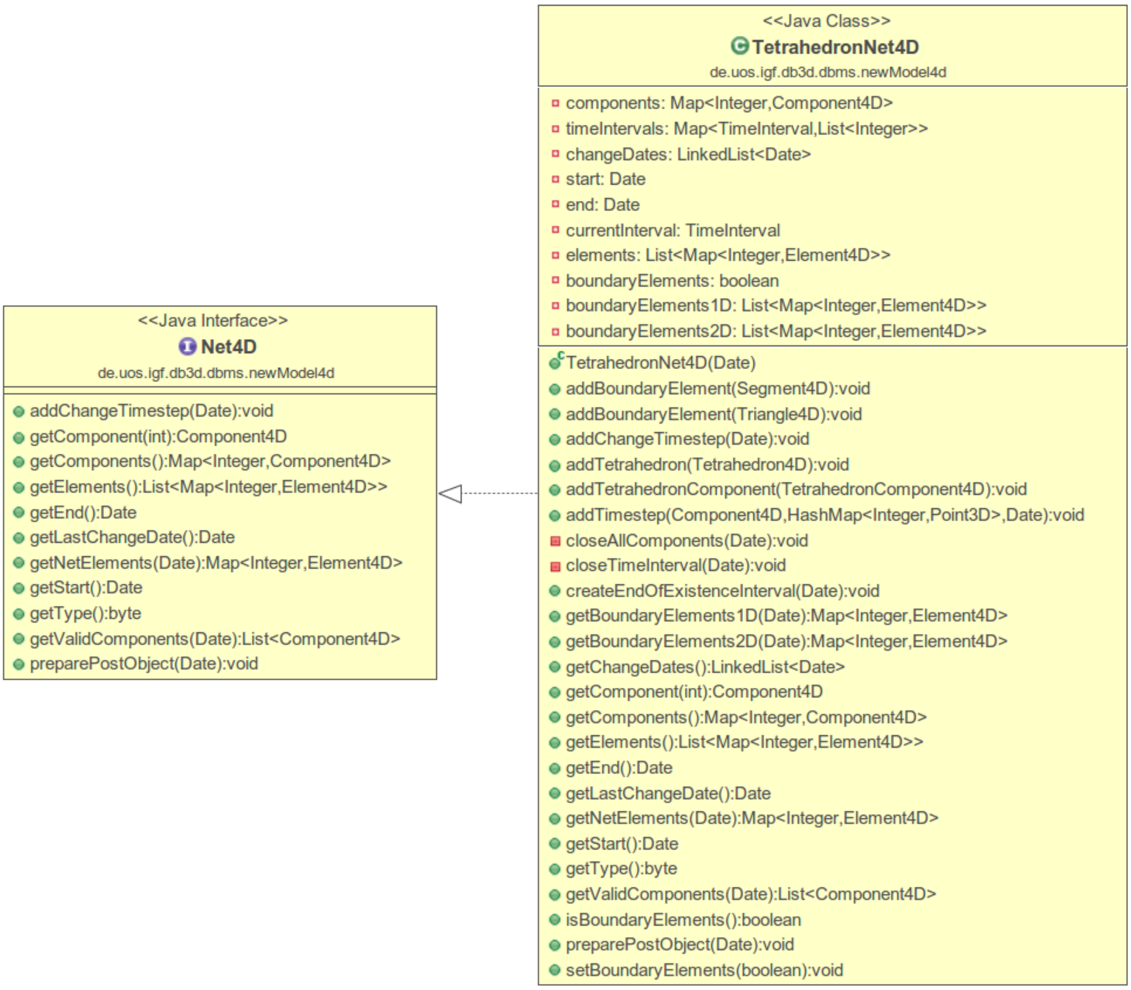


Figure 73 - Methods and data fields of the class *TetrahedronNet4D* and the interface *Net4D*.

Interface **`Component4D`** and explicit **`Component4D`** classes

The interface `Component4D` declares *getter*-methods for the data fields of explicit `Component4D` classes. These are i.a. used by the *service* classes which are presented in Section 5.3.3. The most important data fields are `net`, `timeInterval` the list `timeSteps` and an associative storage `pointTubes`. The field `timeInterval` manages the time interval that is valid for the respective net component. The list `timeSteps` manages a chronological list of explicit time steps of the respective net component cf. Chapter 3.2. The most important data field is the associative storage `pointTubes(Integer, List <Point3D>)`. The key of the storage is a unique identifier of each individual vertex. The list behind a specific `Integer` value that represents the *Point Tube ID* manages a single Point Tube, i.e. a single vertex for different time steps. Such Point Tubes are clearly assigned to the time steps managed in `timeSteps`.

During the construction of these Point Tubes, the individual vertices within the list are checked for similarities between two time steps and are referenced if possible. Thereby the concept of Delta Storage is realized, cf. Chapter 3.5. The composition of these Point Tubes is described below in Algorithm 1 and Algorithm 2, respectively. The ID of a single Point Tube is always associated with an `Element4D` object that is managed in a correspondent `Net4D` object. The interface `Component4D` is implemented by the explicit classes `PointComponent4D`, `SegmentComponent4D`, `TriangleComponent4D` and `TetrahedronComponent4D`. Figure 74 illustrates the data field and method declarations of the class `TetrahedronComponent4D` as an example:

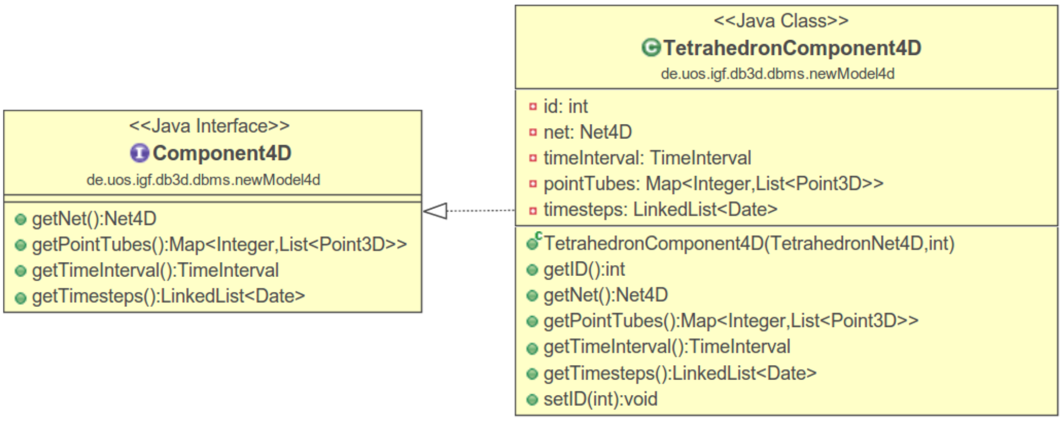


Figure 74 - Methods and data fields of the class *TetrahedronComponent4D*.

Interface **`Element4D`** and explicit **`Element4D`** classes

The interface `Element4D` is a *marker interface* and is implemented by the classes `Point4D`, `Segment4D`, `Triangle4D` and `Tetrahedron4D`. These explicit classes consist of an ID that is unique within a `Net4D` object and references to 1-4 Point Tubes, whereby a single net element is described: i.e. one PointTube for a `Point4D` object, two Point Tubes for a `Segment4D` object, three Point Tubes for a `Triangle4D` object and four Point Tubes for a `Tetrahedron4D` object. The actual Point Tubes are managed in the associative storage `pointTubes` within the corresponding `Component4D` class. Within these element classes the Point Tubes are referenced by the use of `Integer` values (IDs) but actually managed in `Component4D` objects. This enforces the explicit separation of the net topology from the vertices. Following the principles of the Point Tube concept the vertices are reused multiple times by different `Element4D` objects, cf. Chapter 3.1.

IDs for the construction elements are optionally stored for Triangle4D and Tetrahedron4D objects: i.e. segment IDs for Triangle4D objects or segments and triangle IDs for Tetrahedron4D objects. Such IDs refer to the construction element objects managed in the Net4D objects in lists of associative storages: boundaryElements1D and boundaryElements2D, respectively. Figure 75 shows the data field and method declarations of the class Tetrahedron4D as an example:

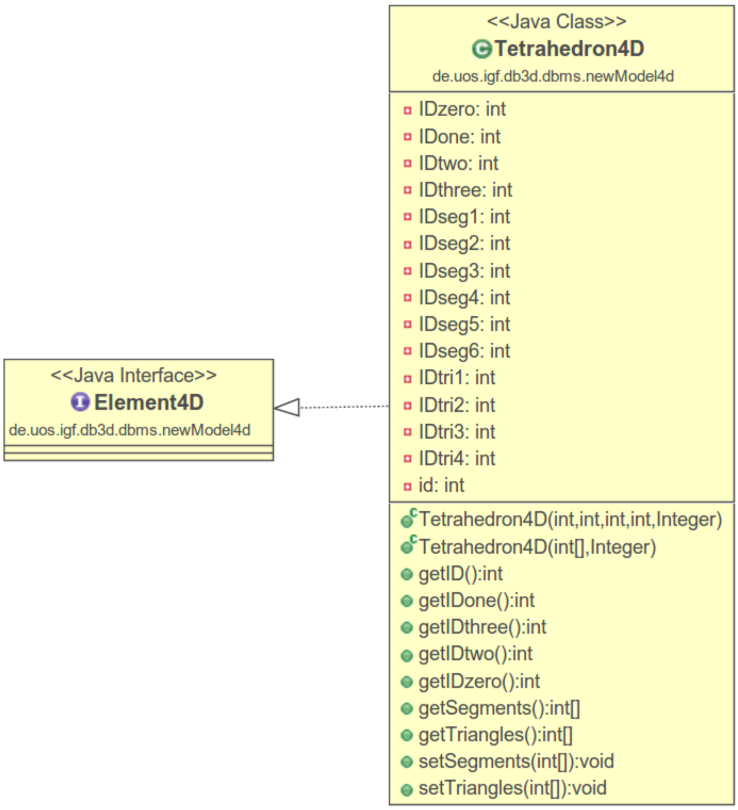


Figure 75 - Methods and data fields of the class *Tetrahedron4D*.

The spatial part of an Object4D object, i.e. a Spatial4D object, consists of a Net4D object that handles the spatio-temporal data. The actual data retains the same net dimension $d \in \{0,1,2,3\}$ over its entire lifetime. Different net dimensions are managed in different Object4D objects. The Net4D object can contain any amount of Component4D objects that exist for different time intervals and may comprise a different temporal discretization. However, certain conditions must be respected, which are summarized in Section 5.3.4. An overview of the internal data management and the allocation of the individual partial concepts to explicit classes is shown in the Figure 76.

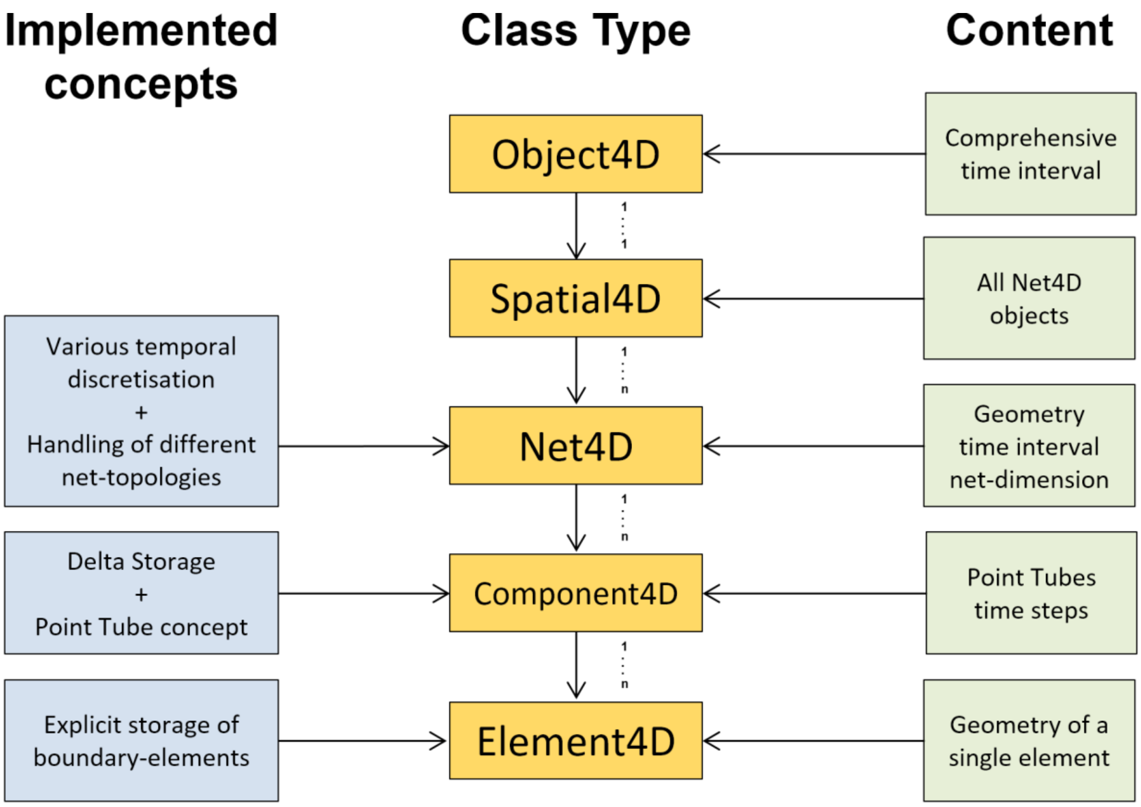


Figure 76 - Interaction of classes with assigned concepts and containing information.

Class `TimeInterval`

In addition to the previously described interfaces `Net4D`, `Component4D`, `Element4D` and their corresponding classes the class `TimeInterval` is especially important for the handling of spatio-temporal data. This class manages a time interval with a *start* and an *end date* that describes a section between a *pre-* and a *post-object*. It is important that the individual components of a net (in the form of `Component4D` objects) are always associated with such a unique time interval. This assignment is realized by using the data field `timeIntervals` of `Net4D` classes. The time interval of the entire `Object4D` is the union of these `TimeInterval` objects. The existence of an explicit time step for the *start* and the *end* point of a time interval is required for all components that are valid within this appropriate time interval, cf. Chapter 3.2.

For the management of 4D models it is essential to initialize the particular objects of the respective classes with the corresponding data in a certain manner. The class `TimeStepBuilder` provides the functionality to compose the internal structure of 4D models correctly. Furthermore, several *service* classes are available that provide important routines for the composition of result objects for various queries on `Object4D` objects. Such classes are also frequently used in multiple import and export functions. Both topics are part of Section 5.3.3 and Chapter 6.

5.3.3 *Service* and *Builder* Classes

The implementation of the data management model is divided into *base* classes that form the *geometry core*, *importer* classes for a file-based import of 4D models, and the *services* and *builder* classes. These are used for import and export operations and for an API-based composition of 4D models.

Class **`TimeStepBuilder`**

For the construction of net components of any dimension the class `TimeStep-Builder` was realized, which processes vertices for new or already existing Point Tubes. To distinguish whether the committed vertices shall create a new time series within a net, i.e. a new `Component4D` object with new Point Tube data structures, or whether an existing time series of a `Component4D` object shall be extended, the `addTimeStep` method has been developed. Subsequently, the appropriate method, i.e. `addNor-malTimeStep` or `firstStep` is used.

Within the method `firstStep` a list of `dates` that handles all *dates* with a corre-sponding time step, is initialized with a given `Date` object. Subsequently, new Point Tube structures are created in form of lists and each is filled with a first `Point3D` object. Along with these lists the ID of the particular point is transmitted to the associative storage `pointTubes` of the correspondent component, cf. Algorithm 1.

```
firstStep(Component4D component, Map<Integer, Point3D> newPoints, Date
          date) {
    Map<Integer, List<Point3D>> pointTubes = component.getPointTubes();
    LinkedList<Date> timesteps = component.getTimesteps();
    // add the effective date as first timestep to the component
    timesteps.add(date);
    // add all Points with their ID to the pointTube Map
    for (Integer id : newPoints.keySet()) {
        // it is the initial step, so create a new List for every Point
        List<Point3D> newTube = new LinkedList<Point3D>();
        newTube.add(newPoints.get(id));
        pointTubes.put(id, newTube);
    }
}
```

Algorithm 1 - *FirstStep* method (simplified) of *TimeStepBuilder* class for handling the initial time step of a 4D object.

Depending whether the construction elements shall be managed explicitly, these are generated for triangular and tetrahedron nets within the methods of the respective *service* classes TriangleServices and TetrahedronServices.

In the method addNormalTimeStep the list of dates is supplemented by the committed Date object. Subsequently, the Point Tubes of the correspondent component are extended by the given vertices. This is done according to the rules of the Point Tube concept, cf. Chapter 3.1. During this procedure it is checked whether the new vertex differs from the vertex of the previous time step. If this is not the case, only a reference to the previous vertex is stored, cf. Chapter 3.5 and Algorithm 2.

```
for (Integer id : newPoints.keySet()) {
    // reference the point if not changed
    if (pointTubes.get(id).get(timesteps.size() - 2)
            .isEqual(newPoints.get(id), sop)) {
        pointTubes.get(id).add(
            pointTubes.get(id).get(timesteps.size() - 2));
    } else {
        // extend the existing pointTube
        pointTubes.get(id).add(newPoints.get(id));
    }
}
```

Algorithm 2 - *AddNormalTimeStep* method (excerpt) of *TimeStepBuilder* class to detect deltas between two time steps.

Determination of the vertices for a specific date

Another important function of the class TimeStepBuilder is the ability to compute the vertices of a net component that are representative for a certain *date*. For this purpose, the method getPointTubeAtInstance is given and is used with the two parameters Component4D and Date. Within this method, the two time steps of the time series are determined that include the parameter Date[34]. Subsequently all points of the component between these two steps are linearly interpolated and returned as the result in the form of an associative storage (<Integer, Point3D>), cf. Algorithm 3. Afterwards this structure is used by the service class ServicesFor4DObjects for further processing.

[34] It is also possible that Date directly corresponds to one of the time steps within the time series.

```
// compute the factor which indicates the position in the interval
double factor = (double) (date.getTime() - intervalStart.getTime())
                / (intervalEnd.getTime() - intervalStart.getTime());
// interpolate a new Point with the help of the computed factor
for (Integer id : allIDs) {
    // check if this ID is active in this timeinterval
    if (pointTubes.get(id).size() >= intervalStartStep) {
        // get the start Point
        Point3D intervalStartPoint = pointTubes.get(id).get(
                                         intervalStartStep);
        double x = intervalStartPoint.getX();
        double y = intervalStartPoint.getY();
        double z = intervalStartPoint.getZ();
        // get the end Point
        Point3D intervalEndPoint = pointTubes.get(id).get(
                                       intervalStartStep + 1);
        // create a new interpolated Point and add it to the map
        points.put(id, new Point3D(x + (intervalEndPoint.getX() - x) *
                   factor, y + (intervalEndPoint.getY() - y) * factor, z +
                   (intervalEndPoint.getZ() - z) * factor));
    }
}
// return the new Map with interpolated points
return points;
```

Algorithm 3 - *GetPointTubesAtInstance* method (excerpt) of *TimeStepBuilder* class for a linear interpolation of a net between two time steps.

The service classes `SegmentServices`, `TriangleServices` and `TetrahedronServices`

For a special preparation of vertices and net topologies of corresponding nets the classes `SegmentServices`, `TriangleServices` and `TetrahedronServices` are available which are primarily used for the *export* and *import* of data. This particularly concerns the identification of vertices that occur multiple times within a net. In addition, different ways to access certain information within a net are provided: e.g. all `Element4D` objects that use a particular vertex can be identified and queried. The results are i.a. used by various *export* classes, e.g. the `ExportWebGLTriangles` class.

Special treatment of vertices

When managing spatio-temporal data the *index-face-set* procedure shall be used and the *completeness axioms* must be respected, cf. Chapter 2.2. But various data formats such as the GOCAD format allow the redundant storage of vertices. For the identification of points, which are used multiple times within a mesh the `initFor4DPointClouds` method is available. This method is used by import classes, e.g. `GocadImporter4D`. Therefore, only unique vertices are used within an import operation for the construction or extension of a 4D model, cf. Chapter 3.1.

Furthermore, the classes `TriangleServices` and `TetrahedronServices` provide a `createBoundaryElements` method, which is used in the `TimeStep-Builder` class for the composition of construction elements. Similar to the handling of vertices these construction elements are created and handled uniquely within the net structure, cf. Algorithm 4.

```
// collect the unique Segments of this TetrahedronNet
Map<Segment4D, Integer> uniqueSegments = new HashMap<Segment4D, Integer>();
// get all Tetrahedron elements valid for this date
Map<Integer, Element4D> tetraElements = net.getNetElements(date);
for (Integer tetraID : tetraElements.keySet()) {
    Tetrahedron4D tmpTetra = tetraElements.get(tetraID);
    int pointIDs[] = { tmpTetra.getIDzero(), tmpTetra.getIDone(), tmpTetra.
                       getIDtwo(), tmpTetra.getIDthree() };
    /** BUILD SEGMENTS **/
    Segment4D sZero = new Segment4D(pointIDs[0], pointIDs[1], 0);
    Segment4D sOne = new Segment4D(pointIDs[0], pointIDs[2], 0);
    Segment4D sTwo = new Segment4D(pointIDs[0], pointIDs[3], 0);
    Segment4D sThree = new Segment4D(pointIDs[1], pointIDs[2], 0);
    Segment4D sFour = new Segment4D(pointIDs[1], pointIDs[3], 0);
    Segment4D sFive = new Segment4D(pointIDs[2], pointIDs[3], 0);
    Segment4D[] segs = { sZero, sOne, sTwo, sThree, sFour, sFive };
    int[] segmentsForTetra = new int[6];
    // use only unique Segments
    for (int i = 0; i < 6; i++) {
        if (!uniqueSegments.keySet().contains(segs[i])) {
            uniqueSegments.put(segs[i], segID);
            segs[i].setID(segID);
            segmentsForTetra[i] = segID;
            segID++;
        } else segmentsForTetra[i] = uniqueSegments.get(segs[i]);
    }
    tmpTetra.setSegments(segmentsForTetra);
    // register construction elements
    for (Segment4D seg : uniqueSegments.keySet())
        net.addBoundaryElement(seg);
```

Algorithm 4 - *CreateBoundaryElements* method (excerpt) of *TetrahedronServices* class to compose unique construction elements for tetrahedrons.

ServicesFor4DObjects

The `ServicesFor4DObjects` class provides a link to the 3D section of DB4GeO and forms the basis for every export operation, which exports a snapshot of a 4D model for a specific time.

Along with the elements of a `Net4D` object an explicit 3D model for a specific time can be created by using the class `ServicesFor4DObjects`. For this purpose, a `get-InstanceAt` method is available that handles an `Object4D` and a specific `Date` object as parameters. In this method, all components of the spatial part of an `Object4D` that are valid for this date are combined. Subsequently, the positions of the vertices that are valid for this specific date are computed by the method `getPointTubeAtInstance` of the class `TimeStepBuilder`. In a next step, an explicit 3D model for the specified date is created along with the elements of a `Net4D` object that is a member of the spatial part of the `Object4D`. Finally, the 3D model that represents a snapshot of the 4D model for the specific date is returned in form of an `Object3D`.

5.3.4 Construction of 4D-Models / API Overview

As part of the implementation of the comprehensive data management model a *user-friendly* approach to compose 4D models was created. Two procedures are available:

1. The user can use available importer classes that map data in form of ASCII files onto the internal data structure. The most important class here is the class `GocadImporter4D` that imports 3D models developed in GOCAD as single time steps for an integration into a 4D model. GOCAD files can also be used to continuously expand an already existing 4D model. The model can be divided into various components with different temporal discretizations. The order to add new components to a 4D model corresponds to the procedure that was presented in Chapter 3.2.

2. The user can use the API of the geometry core package to create or extend a 4D model. In this case the classes `Object4D` and `Spatial4D`, as well as the implementations of the interfaces `Net4D`, `Component4D` and `Element4D` are used to generate spatio-temporal data. Afterwards the routines that are provided by the class `TimeStepBuilder` are used to set up the internal data structure and a consistent net topology for the 4D model.

Figure 77 represents the composition of a 4D model. The composition starts with the `Component4D` object `comp0(tstart)` for the time t_{start}. Then additional components are added for the first time step t_{start}: `comp1(tstart)` and `comp2(tstart)`. These correspond to regions with a different volatile movement within the model. Based on this additional time steps can be added to the respective components. In this case these are two more steps for `comp0`: t_a and t_{change}, three time steps for the `comp1`: t_b, t_c, t_{change}, and one time step for the `comp2`: t_{change}. A requirement is that all the components of this model have an explicit time step for the time t_{change} due to a change of the net topology at this specific time.

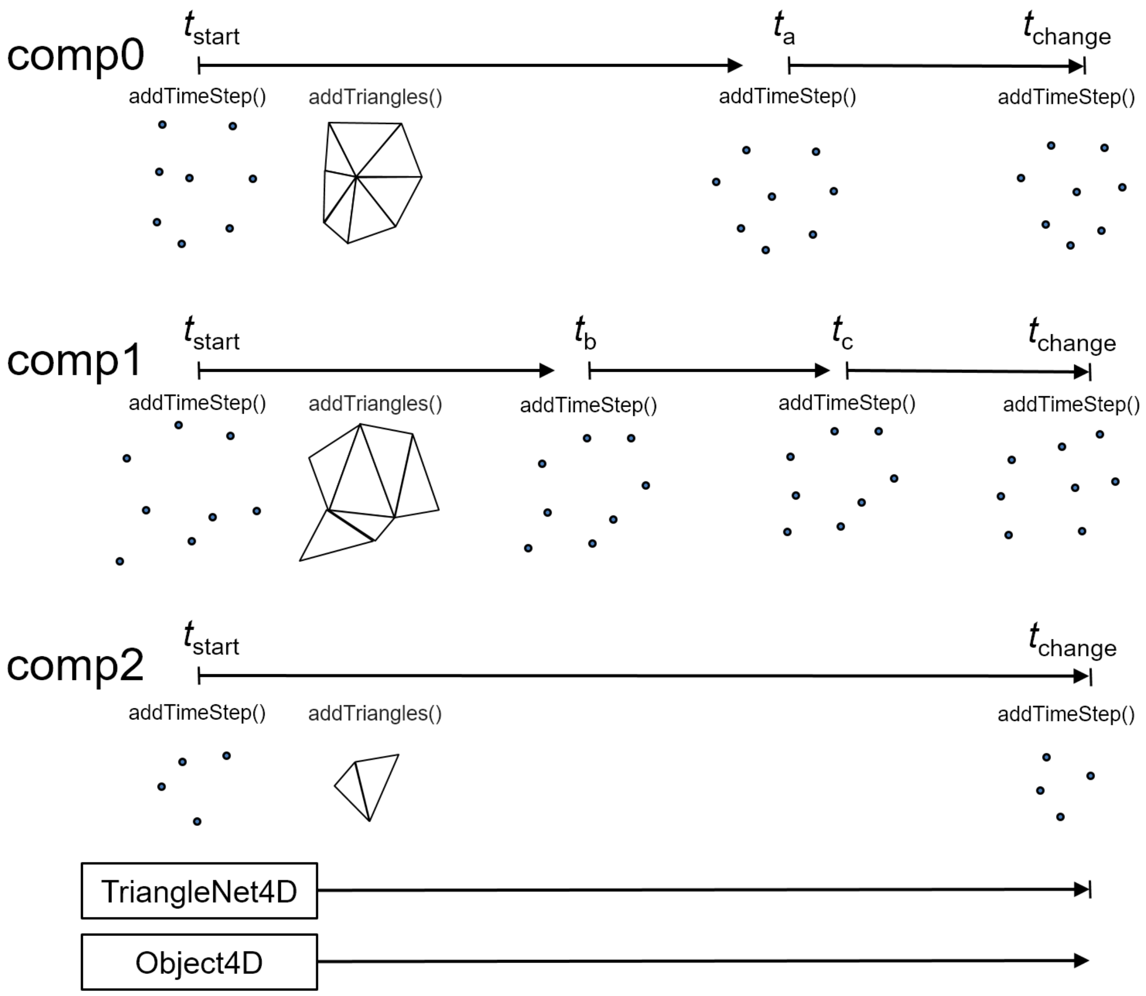


Figure 77 - Construction of a 4D model for time interval [t_{start}, t_{change}[with three net components of different temporal discretizations.

The rules that need to be fulfilled when creating the structure of a 4D model with different volatile subregions (net components) are summarized below, cf. Chapter 3.2:

List 3 - Rules for the development of 4D models in DB4GeO

- All net components must be added at the beginning of the new interval with a common *start* date.
- The components can be extended chronologically but independently from each other.
- All components must have an explicit step for a common *end* date within a valid time interval.
- A change of the net topology must be indicated to finalize a time interval and start a new one.

Combined, the three components form the pre-object of the time series at time t_{change}, cf. Figure 77. Subsequently, new components are created that represent different volatile regions of the 4D model. The amount of components is modified after the transition of the net topology. Therefore, starting with a time step at time t_{change} only two components `comp3` and `comp4` form the 4D model during time interval $[t_{\text{change}}, t_{\text{end}}]$. The construction of these new components also need to follow the specified rules, cf. List 3. Thus, in our example, the two new components `comp3` and `comp4` have to start with a common time step for time t_{change}, cf. Figure 78.

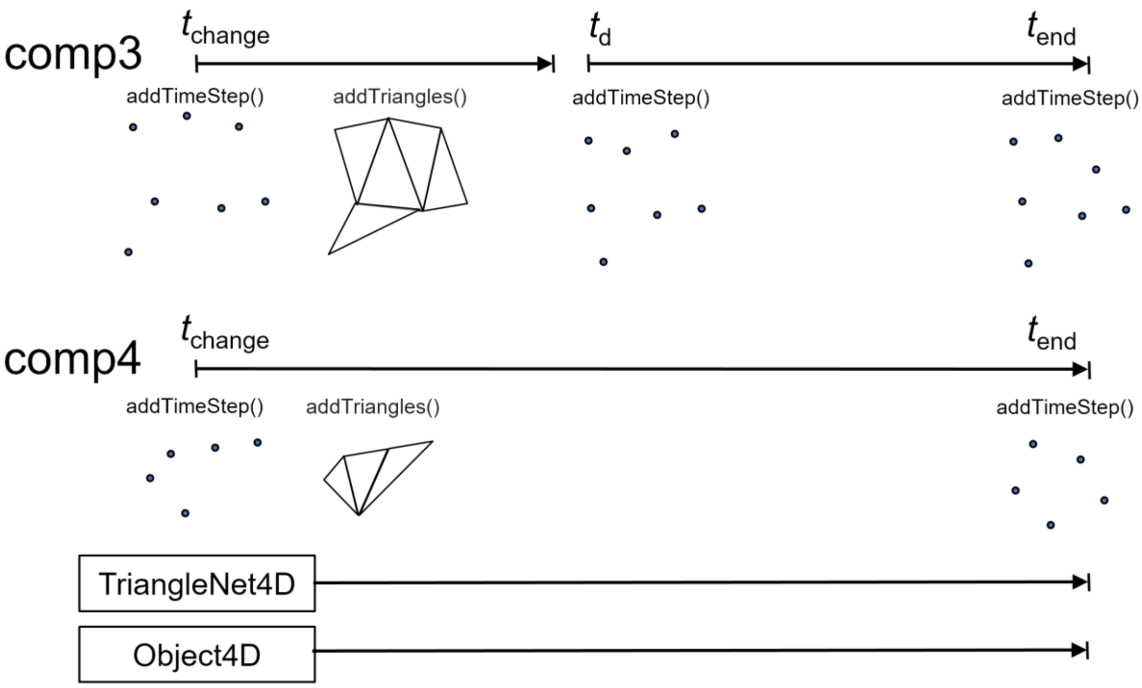


Figure 78 - Extension of a 4D model for time interval [t_{change}, t_{end}] with two net components of different temporal discretizations.

Combined `comp3(tchange)` and `comp4(tchange)` represent the post-object, cf. Chapter 3.3. This procedure can be repeated as often as desired. For a correct finalization of the time series of the 4D model, all involved components need to have a common final time step for time t_{end}. Figure 79 represents the internal structure of a complete 4D model for time interval $[t_{\text{start}}, t_{\text{change}}[$ that is constructed of i different components. The second time interval $[t_{\text{change}}, t_{\text{end}}]$ consists of j different components. All involved components may be discretized differently.

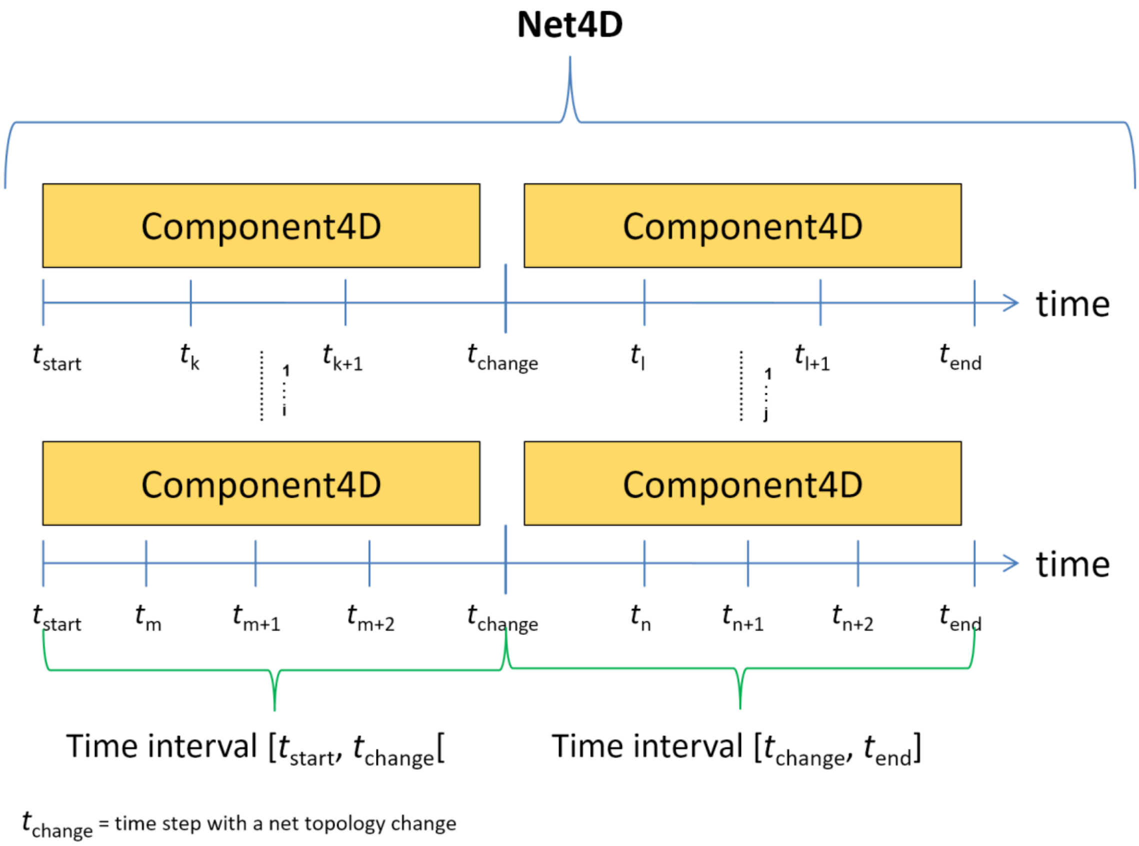


Figure 79 - Internal structure of a 4D model with a various number of net components.

The following section describes both scenarios for the construction of a 4D model within a respective workflow based on test data.

5.3.5 Workflow

This section describes a workflow for the construction of a 4D model based on test data with Java code examples. First the scenario of the *API usage* is described. Subsequently, the scenario of using *importers* is described. In both scenarios, a 4D model m is developed that is defined for the time interval $inter = [t_{start}, t_{end}]$ with a change of the net topology at time $t_{change} \in inter$. Accordingly, the 4D model m is separated into two time intervals $[t_{start}, t_{change}[$ and $[t_{change}, t_{end}]$. The first time interval consists of three components and the second time interval of one component. Every component has a different temporal discretization.

```
// create a new 4D object
Object4D object4D = new Object4D();
// create a new spatial part
SpatialObject4D spatial = new SpatialObject4D();
// Link the spatial part
object4D.setSpatial(spatial);
// create a Triangle4NetD object
TriangleNet4D triangleNet = new TriangleNet4D(dateStart);
// link the net
spatial.setNet(triangleNet);
// set up the net with the tStart date
triangleNet.addChangeTimestep(tStart);
// create some triangles elements
triangleNet.addTriangle(new Triangle4D(0, 1, 3, 0));
triangleNet.addTriangle(new Triangle4D(0, 2, 3, 1));
triangleNet.addTriangle(new Triangle4D(2, 4, 3, 2));
triangleNet.addTriangle(new Triangle4D(3, 4, 1, 3));
triangleNet.addTriangle(new Triangle4D(1, 4, 5, 4));
triangleNet.addTriangle(new Triangle4D(0, 2, 6, 5));
// create a first component
TriangleComponent4D component1 = new TriangleComponent4D(triangleNet,
                                 compID1);
// link the component
triangleNet.addTriangleComponent(component1);
// add the 1. timestep for the 1. component with points and date tStart
TimeStepBuilder.addTimestep(component1, points, tStart);
// add the 2. component
triangleNet.addTriangleComponent(component2);
// add the 1. timestep for the 2. component with points and date tStart
TimeStepBuilder.addTimestep(component2, pointsComp2, tStart);
// add the 2. timestep for the 2. component
TimeStepBuilder.addTimestep(component2, pointsComp2STEP2, dateStep2);
// add the 3. timestep for the 2. component
TimeStepBuilder.addTimestep(component2, pointsComp2STEP3, dateStep3);
// add the 3. component
triangleNet.addTriangleComponent(component3);
// add the 1. timestep for the 3. component with points and date tStart
TimeStepBuilder.addTimestep(component3, pointsComp3, tStart);
// add the 2. timestep for the 3. component
TimeStepBuilder.addTimestep(component3, pointsComp3STEP2, dateStep4);
// close the 1. time interval for all three components
TimeStepBuilder.addTimestep(component1, pointsComp1STEPChange, dateChange);
TimeStepBuilder.addTimestep(component2, pointsComp2STEPChange, dateChange);
TimeStepBuilder.addTimestep(component3, pointsComp3STEPChange, dateChange);

/** CHANGE OF NET-TOPOLOGY - CREATION OF A NEW POST OBJECT **/
// prepare the net
triangleNet.preparePostObject(dateChange);
// add a new mesh
triangleNet.addTriangle(new Triangle4D(0, 1, 2, 0));
triangleNet.addTriangle(new Triangle4D(0, 1, 3, 1));
// add the 1. timestep of the new Component
TimeStepBuilder.addTimestep(compPost, pointsPostComp1Change, dateChange);
// close the 2. time interval
TimeStepBuilder.addTimestep(compPost, pointsPostComp1End, dateEnd);
```

Algorithm 5 - Code example for creating a 3D model by using the API.

Afterwards, the 4D model is fully constructed in form of an `Object4D`. This workflow can be processed without any restrictions also with data of other net dimensions. In the GOCAD ASCII file based workflow the explicit API calls are processed within the methods of the importer. Accordingly, the workflow that works with GOCAD files is more compact compared to the direct use of the API.

```
// create a new object
Object4D object4D = new Object4D();
// create an importer instance
GocadImporter4D importer = new GocadImporter4D();
// add the first step of the three components
FileReader reader = new FileReader(gocadFile_Comp1_Step_1);
importer.importObject(reader, object4D, tStart);
// add the second step of the three components
reader = new FileReader(gocadFile_Comp2_Step_1);
importer.importObject(reader, object4D, tStart);
// add the third step of the three components
reader = new FileReader(gocadFile_Comp3_Step_1);
importer.importObject(reader, object4D, tStart);
// add the second for the second component
reader = new FileReader(gocadFile_Comp2_Step_2);
importer.importObject(reader, object4D, dateStep2);
// add the third step for the second component
reader = new FileReader(gocadFile_Comp3_Step_3);
importer.importObject(reader, object4D, dateStep3);
// add the second step for the thrid component
reader = new FileReader(gocadFile_Comp3_Step_2);
importer.importObject(reader, object4D, dateStep4);
// close the first time interval for all three components
reader = new FileReader(gocadFile_Comp1_Step_Change);
importer.importObject(reader, object4D, dateChange);
reader = new FileReader(gocadFile_Comp2_Step_Change);
importer.importObject(reader, object4D, dateChange);
reader = new FileReader(gocadFile_Comp3_Step_Change);
importer.importObject(reader, object4D, dateChange);
// add the first timestep of the new Component
reader = new FileReader(gocadFile_Comp4_Step_Change);
importer.importObject(reader, object4D, dateChange);
// close the second time interval
reader = new FileReader(gocadFile_Comp4_Step_End);
importer.importObject(reader, object4D, dateEnd);
```

Algorithm 6 - Code example for importing a 4D model.

Within the individual methods of the class `GocadImporter4D` the API of the 4D model geometry core is frequently used.

5.3.6 Ad Hoc Visualization

For a productive work with spatio-temporal data an appropriate visualization is essential. This allows the user to verify the data and to interpret it visually. Additionally, changes in spatio-temporal data might be better recognised by a visual comparison of the particular time steps. For a quick and convenient visualization of spatio-temporal data with the

DB4GeO-WebGL-Viewer a solution was developed that is based on the Web standard WebGL[35].

By the DB4GeO-WebGL-Viewer spatio-temporal data is visualized directly from the database and also platform independent in any modern browser. Therefore, no additional tools such as plug-ins need to be installed. A visualization of the data is achieved without the use of a specific modelling software. For this purpose, the corresponding data is processed according to the rules of the WebGL library *Three.js*[36] within an HTML file.

The user of the viewer has the possibility to visualize a *snapshot* of a 4D model that corresponds to a specific time within the valid time interval of such a model. This snapshot, i.e. a 3D model, is generated within DB4GeO by a linear interpolation. Within the viewer the user can use typical graphical operations such as *panning*, *zooming* and *rotating* to interact with the visualization of the model. This is realized for spatial and spatio-temporal data based on *d*-simplicial complexes with $d \in \{0,2,3\}$. The viewer provides a simple visualization of the geometry but can also consider colour values (if available) and offers an optional grid view. The viewer is implemented as an export class `ExportWebGLMain`, which handles an `Object3D` as a parameter. Such an object can be the result of an interpolation of a 4D object. Subsequently, depending on the dimension of the net a corresponding WebGL export class finishes the export process, e.g. `ExportWebGLTriangles`.

The Viewer is also available for 3D models that are managed by DB4GeO. Figure 80 shows two models being visualized with the DB4GeO-WebGL-Viewer.

[35] WebGL – Web Graphics Library, a JavaScript API for the rendering of 3D objects within a web browser. https://www.khronos.org/webgl
[36] Three.js – an open source JavaScript based framework for the development of WebGL-based 3D scenes. https://github.com/mrdoob/three.js

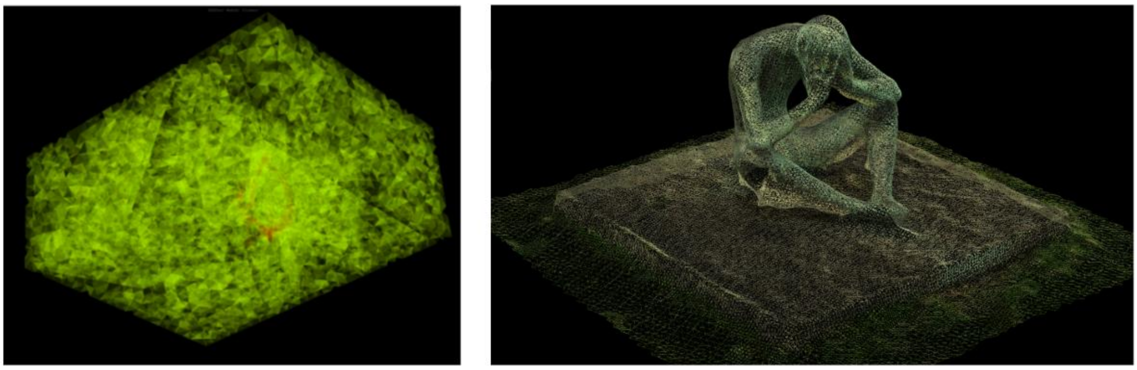


Figure 80 - Left: 3-simplicial complex with temperature data of a geothermal subsurface model. Right: 2-simplicial complex of statue "the thinker"[37] at KIT campus.

Though, the viewer does not reflect the capabilities of a professional visualization within complex modelling tools, it provides an ad hoc solution to obtain a quick overview of spatial and spatio-temporal data managed in the geodatabase. Within this thesis, the viewer was frequently used for a quick visual check of the results that were produced by the implementation of the data management model for spatio-temporal data.

5.4 Realization of a Generic Mobile GIS for Spatio-Temporal Data

This section describes the realization of a generic mobile GIS for spatio-temporal data. The realization is based on the general concept for generic mobile GIS and the particular requirements for spatio-temporal data. First, the general structure of such a mobile GIS is described and divided into interrelated modules. The functionality and the interaction between these modules are explained in detail. Subsequently, an implementation based on the mobile operation system Android is presented.

5.4.1 Structure

The realization of a generic mobile GIS for spatio-temporal data is structured into four different modules:

1. *MainUI* – User interface and guided workflow.
2. *Transaction* – Handling of queries for spatio-temporal data and operations.
3. *Data handling* – Handling of spatio-temporal data on the device itself.

[37] 3D-Model by Simon Schuffert (KIT), All rights reserved.

4. *Visualisation* – Mobile visualisation of the data.

The general design of the user interface is specified in the module *MainUI*. This includes the following features:

- A guided workflow for the specification of appropriate Web services.
- Buttons and touch gestures for the interaction with the data, i.e. zooming, panning, rotating.
- Capabilities to select spatio-temporal operations.

A WFS and WPS to query spatio-temporal data and/or spatio-temporal operations has been designed and implemented in Chapter 4. In most cases such queries result in 3D data. The interaction with the data needs to be as intuitive as possible and is based on rules for an appropriate handling of 3D data, cf. Chapter 5.3.6. Therefore, intuitive multi-touch gestures are used. The menu displays all available spatio-temporal operations including their input and output parameters according to the specified WPS. Additional options to specify input parameters and to invoke such operations are available.

The queries for spatio-temporal data and/or operations that are directed to appropriate Web services are processed and handled within the module *Transaction*. This module requires a clear distinction of WFS and WPS based transactions:

WFS

Within this sub-module WFS-requests are composed and the appropriate methods are called, i.e. *GetCapabilities*, *GetFeatureInfo*, *GetFeature*. The results are forwarded to the modules *MainUI* and *Data handling*.

WPS

Within this sub-module WPS-requests are composed and the appropriate methods are called, i.e. *GetCapabilities*, *DescribeProcess*, *Execute*. The results are forwarded to the module *MainUI* for alphanumeric data and *Data handling* for 3D/4D data.

For the preparation of request and results it is sufficient to use an SAX parser. Alternatively, requests can be composed based on KVP encoded URLs.

The results of requests for spatio-temporal data consist of 3D data that reflect a specific snapshot of a 4D model. Such data may not only be available on the mobile GIS transiently, but can also be stored permanently. Therefore, it is possible to use the data without an active network connection. Such a data storage on a mobile device can be accomplished with an appropriate database, e.g. SQLite[38].

A major challenge for the realization of a mobile GIS for spatio-temporal data is the visualization of the data. This requires the visualization of specific snapshots of a 4D model in the form of 3D data. Furthermore, *thematic* or *semantic data* need to be processed and visualized, e.g. representation of temperature values within a solid or the visualization of different types of rock strata. A *Visualization* module combines this functionality, i.e. the interpretation and visual processing of relevant thematic and geometric data. Such data is transferred directly from the *Data handling* module. An overview of the interaction between the individual modules of a prototypical implementation illustrates the following Figure 81.

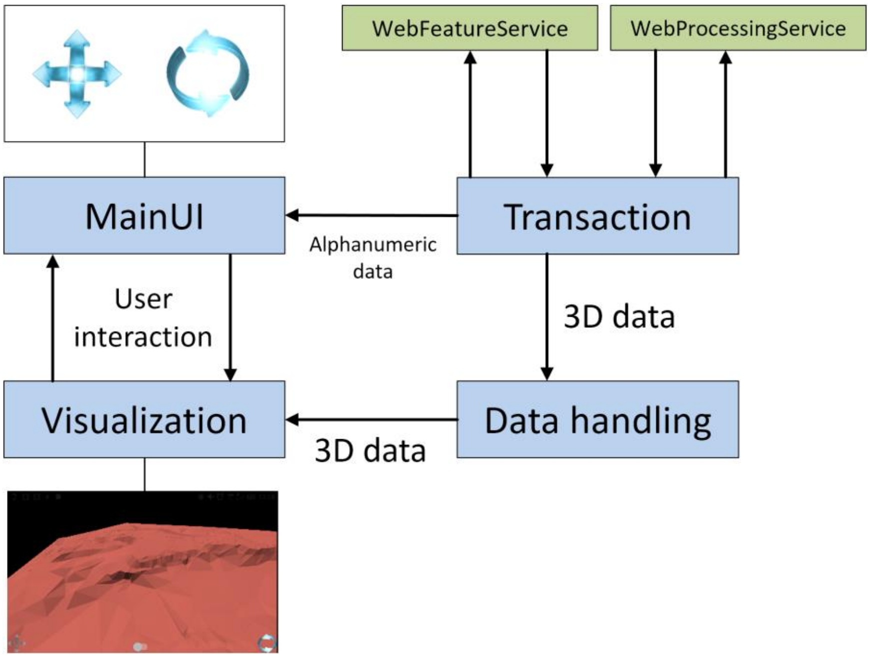


Figure 81 - Interaction between individual modules of a mobile GIS for spatio-temporal data.

The definition and interaction of the individual modules form the foundation for a prototypical implementation of a generic mobile GIS for spatio-temporal data based on the Android platform, which is part of the following section.

[38] SQLite is a popular open source relational DBMS. https://www.sqlite.org

5.4.2 Implementation

The implementation of a mobile GIS for spatio-temporal data is realized as an application based on the mobile operation system Android. It is based on the prior work on a mobile AR-viewer for 3D data that was developed by [Dittrich, 2012]. This viewer offers the functionality to visualize 3D models based on 2-simplicial complexes.

As part of the implementation appropriate software packages are assigned to the modules discussed in Section 5.4.1. The following list describes fundamental packages and their key functions:

> **`coordtrafo`**: responsible for the transformation of coordinates.
>
> **`database`**: responsible for the storage of server information (e.g. WFS/WPS address) in an SQLite database.
>
> **`dataresource`**: responsible for the selection and processing of information about appropriate Web services, i.e. WFS and WPS. Preparation of information about existing data and/or available operations of a Web Service.
>
> **`importer`**: processing the results of a corresponding request to a Web service. Pre-processing of the 3D data into a format that is processed further.
>
> **`opengles`**: visualization of 3D data with the help of OpenGL ES[39]. Corresponding shader and sources of illumination are used for an appropriate visualization of the data. In addition, the optional camera image is processed within this package.
>
> **`view3d`**: composition of the GUI including menu items and corresponding tools for an appropriate movement within a 3D scene.

These specific packages are responsible for the control and visualization of the spatio-temporal data within the mobile GIS. Each query session starts with a *getCapabilities* request to retrieve important information about a specific server and available data sets. By using the `Activity` class `WFSSelectionActivity` a suitable WFS that distributes appropriate data is selected. Subsequently, a specific spatio-temporal data set is selected by the use of the class `QueryActivity`. Here, a specific date is picked by a `DatePickerDialog` object and used for the request, cf. Figure 82. The request itself is handled by the `dataresource` package. Once the server responses with a 3D object,

[39] OpenGL for embedded systems, a subset of OpenGL, developed by Khronos Group.

i.e. a specific snapshot of a 4D model, the data is pre-processed in the `GOCADConnector` class. Subsequently, the data is rendered in the classes `InteractiveRenderer` and `InteractiveSurfaceView` and visualized in the `Activity` class `InteractiveActivity`.

The visualization in the augmented reality mode (AR mode) is handled by the classes `ARRenderer` and `ARSurfaceView` or `ARActivity`, respectively. In the AR-mode a position and orientation of the device is required, which is determined from the sensor information of the device within the classes `ARActivity` via a `LocationManager` and a `SensorManager` object. A visualization of a specific snapshot of the 4D model "Piesberg" – a landfill site near the city of Osnabrück, Germany – for the date 10.02.1984:00:00:00 is shown in Figure 82.

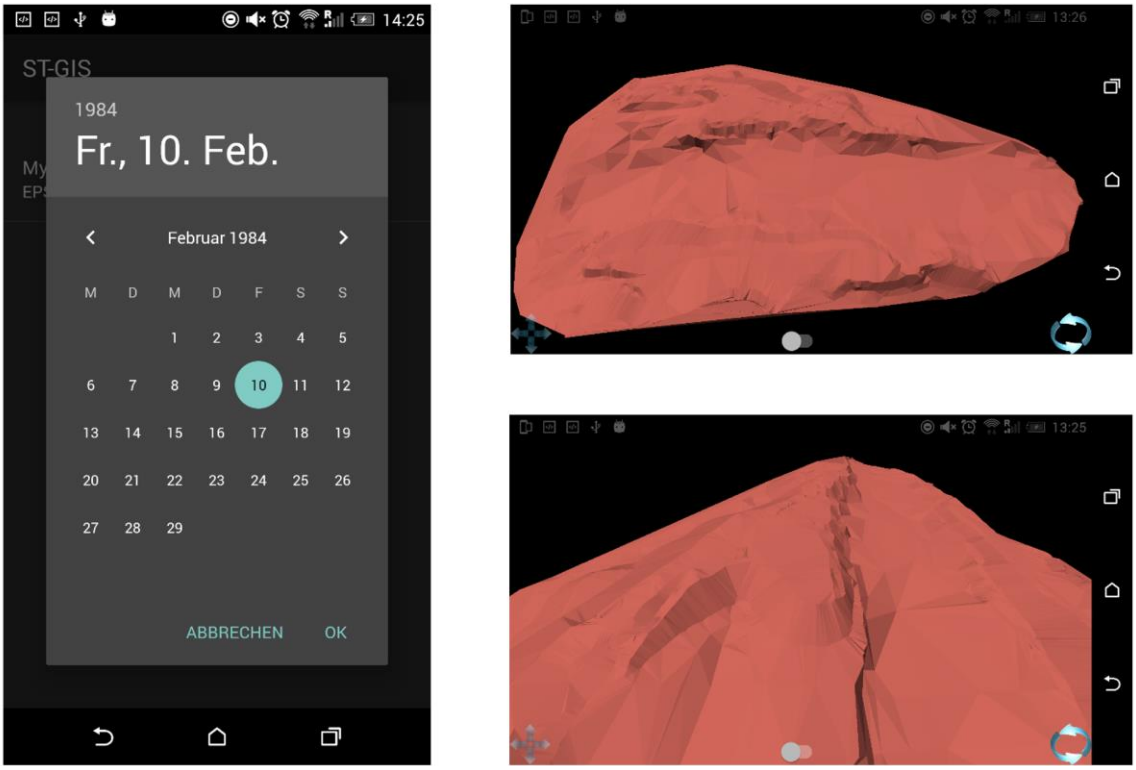


Figure 82 - Visualization of a specific snapshot of the 4D model Piesberg on the mobile GIS. *Left*: Selection of a specific date. *Right*: The Piesberg viewed from different angles.

Available processes of a WPS are started via *Execute*-queries, e.g. an *averagespeed* query that determines and returns the average speed of a spatio-temporal object. The final prototype runs on a Sony Xperia Z® tablet, containing a GNSS (GPS + GLONASS) sensor, WiFi and LTE capabilities. Thus the user can self locate easily in almost any environment and send or query spatial and spatio-temporal data if WiFi or a mobile network is available. The software was published under the name *ST-GIS* as free and open source software[40].

With the development of a generic mobile GIS for spatio-temporal data the third part of the workflow for the handling of spatio-temporal data has been established. It was shown that the intensive use of spatial data standards is an appropriate basis for such a system. A summarized overview of the entire workflow based on a typical test data set is presented in Chapter 6.3.

[40] ST-GIS can be found at https://github.com/PaulVincent/ST-GIS

6 Validation and Evaluation

The development of sustainable software requires an appropriate test environment. Thus, the implementation of the data management model presented in Chapter 5.3 is validated by a *unit test framework* which is introduced in Section 6.1.

The implementation of the comprehensive data management model offers advanced query facilities. A selection of those and possible application areas are presented in Section 6.2.

A summarized overview of the workflow for the handling of spatio-temporal data is presented in Section 6.3. This includes a typical test data set and a visual overview of the workflow in action.

6.1 Software Validation

This section discusses the process of evaluation and the composition of an appropriate test environment in which various *test scenarios* are systematically processed. For this procedure, the approach of *unit testing* is frequently used for the professional development of software [Osherove, 2013]. Following this approach *test environments* and *test scenarios* are composed in a systematic and structured way. In the programming language Java unit testing is realized within the *JUnit framework* [Cheon and Leavens, 2002], which also forms the basis of the test environment in this thesis.

Initially suitable test data is developed for each test scenario. These scenarios validate the individual sub-concepts of the data management model. For this purpose, both an *API*- and a *file-based test environment* are developed. To test both options for the construction of 4D models it is necessary to distinguish between *API*- and *file-based* test scenarios. The individual test scenarios are presented and described in the following.

Construction of suitable test data

In the following two different ways to generate test data for suitable test scenarios are presented. The first one is API-based. Therefore, the API of the implemented data management model is used, e.g. to create a `TriangleNet4D` object with multiple `TriangleComponent4D` objects, each consisting of multiple `Triangle4D` objects. These objects are created directly in Java source code. A suitable place for such operations are JUnit testcase files.

The second way for constructing test data is based on explicit ASCII files that contain appropriate spatio-temporal data. For this, the file format of the 3D modelling and visualization tool GOCAD was used. Different files were created for corresponding test cases. Subsequently, the `GocadImporter4D` class is used to read and process these files. In this thesis both ways for the composition of new 4D models are tested. Therefore, the test environment is divided into the following two main sections:

API-based tests

> When using the API, the test data is created directly by the use of explicit classes of the data management model. The starting point of this procedure in this case is an `Object4D` that owns a `Spatial4D` object. This `Spatial4D` object represents the spatial part of a 4D model, cf. Figure 76. This spatial part consists of a `Net4D` object. Once the dimension of the net has been established, it cannot be changed, cf. Chapter 5.3.4. The actual test data sets are generated at this point. The corresponding time steps of a model along with the geometry are created directly in the source code. API-based test data sets are available for all four net dimensions.
>
> Multiple JUnit tests verify the constructed 4D models. Each particular sub-concept is tested individually for each of the four appropriate net dimensions, cf. Chapter 3. The combination of these concepts is tested with additional test cases.

ASCII-file-based tests

> Multiple test data sets with different characteristics were developed to verify the ASCII-file-based workflow. The data within the ASCII-files follow the GOCAD file format guidelines [Cheng, 1997]. A 4D model is created from individual time steps. These time steps are assigned to individual GOCAD files which are imported using the `GocadImporter4D` class.
>
> The various sub-concepts of the data management model are validated and verified individually in multiple JUnit test cases. Additionally, the entire spectrum of all sub-concepts is tested within appropriate test cases.

The test scenarios are standardized according to the rules of unit testing and can be used on various occasions, e.g. the development of a new software version. An overview of the test scenarios is shown in Figure 83. In addition to the scenarios visible in the figure, further test scenarios cover all sub-concepts simultaneously.

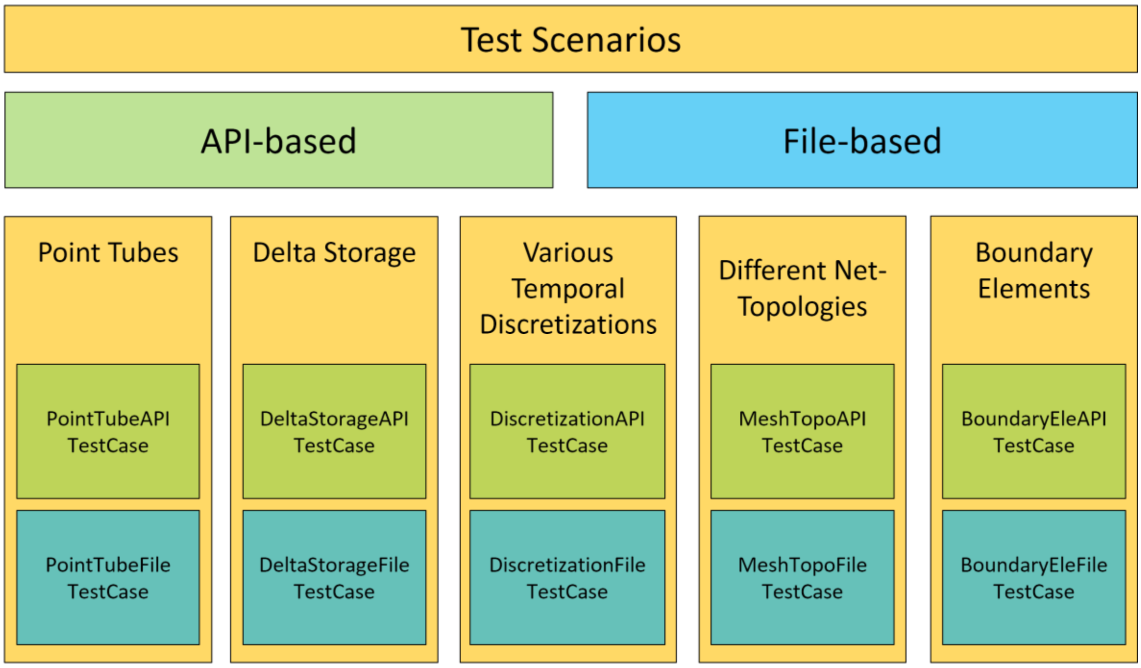


Figure 83 - Overview of available JUnit testcase scenarios.

A specific procedure of a particular test passes through the following three stages:

1. *Setup*: composition of the 4D model including all time steps and explicit data (API or file-based).
2. *Verification of the data structure*: comparison of the data structure with reference values. The verification is proceeded by the use of JUnit methods, e.g. *assertTrue*, *assertNull*, *assertEquals*, etc.
3. *Verification of behaviour*: processing of operations and methods with a subsequent comparison of the results with appropriate reference data.

The composition of the model structure, i.e. the actual 4D model, is processed within the classes `SetupForAllAPITestCases` and `SetupForAllFileTestCases` in which 4D models are created from GOCAD files or by using the API, respectively. Each of the test cases specified in Figure 83 is performed for *d*-simplicial complexes with $d \in \{0,1,2,3\}$. Accordingly, every test case consists of four parts. If the data structure and the results correspond with the reference data, the test is considered as a success.

The test scenarios are executed after each change of the actual implementation. Due to this test environment the development of the software components of this thesis was permanently monitored and evaluated. Thus the implementation of sustainable software was supported.

6.2 User Validation

This section reflects multiple advanced query facilities that were not explicitly mentioned within Chapter 3 and 5.3. These can be useful for solving tasks within various applications. In this context the *service* classes (cf. Chapter 5.3.3) and a particular use of different sub-concepts have the following added value:

Interpolation

Since the vertices of 4D models are managed in Point Tube data structures and are collectively accessible for the entire model, interpolation operations can be realized in a simple manner. Within the comprehensive data management model for handling spatio-temporal data interpolations are always performed between two vertices that are managed in Point Tubes. There is always a 1 to 1 connection, which describes a trajectory in 3D space, i.e. the movement over time. The type of interpolation is determined at a central location in the software and is applied to all appropriate vertices. This offers the possibility to easily change the type of interpolation. A linear interpolation is not always an appropriate approximation of the real situation. For instance, a landslide may be a very abrupt event. If the corresponding region is modelled in large time intervals, the movement of the landslide would occur unnaturally slow. In such a case a different set of interpolation rules may improve the outcome. Due to the data structure of the data management model such rules can be deployed easily.

Boundary queries

Queries for the boundary of a 4D object that consists of a *d*-simplicial complex for a specific time t_x result in an accumulation of (*d*-1)-simplices. The net topology can be used for the determination of these simplices. The positions of the particular vertices are calculated by the use of the Point Tube data structures. Therefore, it is possible to calculate an appropriate representation of the boundary for a specific time t_x.

Merge and split operations

The use of the concept for handling net topology changes can be used to merge or split multiple net components over time. For instance, a 4D model for a time interval $[t_{n-1}, t_n[$ might consist of one net component nc_1 and is split into two separated net components nc_2 and nc_3 at time step t_n. Afterwards the 4D model consists of two net components for a time interval $[t_n, t_{n+1}[$. This can also be performed in the other direction, i.e. a merge operation at which two net components nc_1 and nc_2 are combined to one nc_3 that is valid within the second time interval.

Extension of time series

The time series of an existing 4D object can be extended in a simple manner. There exist two different procedures for this task:

1. The net topology remains the same: only the appropriate Point Tube data structures get extended by another time step.
2. The net topology of the 4D object changes: a new time interval is established. New Point Tube data structures are created to handle the vertices. The new net topology is stored and linked to the new time interval.

In both cases the user starts the file importer and the data management model determines the appropriate steps to handle the request. Alternatively, the user can use the API to invoke the appropriate operations directly.

Extensive use of point clouds

The vertices of a 4D model are managed in Point Tube data structures and therefore available for the processing of various spatio-temporal questions. Several applications and operations do not require a net topology and are solved solely by the use of the vertices, e.g. the calculation of a bounding box. Due to the data model the access of the underlying point cloud of *d*-simplicial complexes with $d \in \{1,2,3\}$ is significantly simplified and can be queried for every specific time $t_x \in [t_{\text{start}},\ t_{\text{end}}]$.

Export of data

There are multiple options available for the user to export data that is stored in the 3D/4D geodatabase architecture DB4GeO. One important format is the GOCAD format that can be used for a further processing of the data in the 3D modelling tool GOCAD. Another export format is the preparation of the data within a WebGL file. Thus, the data can be viewed within a conventional web browser, cf. Chapter 5.3.6. For both options the data is processed accordingly. GOCAD and the WebGL viewer have not been developed to visualize a 4D model in its entire form, but snapshots of the model, e.g. $m\ (t_x)$ with $t_x \in [t_{\text{start}},\ t_{\text{end}}]$. Therefore, the model m is computed for the specific time t_x. The class `TimeStepBuilder` is internally used for the processing of such a snapshot $m\ (t_x)$ of the 4D model m, cf. Chapter 5.3.3.

Import of data

For the import and composition of a 4D model either the API or GOCAD files can be used. The latter represent single time steps of corresponding net components. Both approaches have been presented in Chapter 5.3.5.

Topological information

Data models and formats that handle *d*-simplicial complexes with $d \in \{2,3\}$ usually handle the net elements in a way that the *d*-simplices refer to *d*+1 vertices, e.g. a triangle refers to its three corner points. Within a *d*-simplicial complex these vertices are used by multiple *d*-simplices, cf. Chapter 2.3.1. Usually, when a topological data structure such as g-maps (cf. Chapter 2.1) is not explicitly used, a specific vertex cannot be queried for all *d*-simplices it is part of. However, the *service* classes that were developed within this thesis offer the capability to compute such information directly based on the net structure. For some objectives it is necessary to know such information. For instance, they are used by export classes and are available to the user for other applications.

Advanced data preparation

The data preparation within the data management model requires a particular treatment. Several tasks need a data structure that meets specific constraints. One important constraint is that vertices within a net are unique, although they are referenced by multiple net elements. This also applies to construction elements such as segments in triangle nets or segments and triangles in tetrahedra nets. Under certain circumstances these can exist multiple times within the raw data. This conflicts with the *completeness axioms*, cf. Chapter 2.2.

The *services* classes were designed to ensure a valid processing of such data. Construction elements that are used by multiple net elements within a net exist uniquely and are referenced by the appropriate net elements, cf. Algorithm 4. These algorithms are internally used during the construction of a 4D model for the composition of construction elements to ensure the consistence of the net topology. However, the functionality can be used for any other application as well, e.g. to validate the consistence of 3D-nets.

6.3 Typical Application of the Workflow

In the previous chapters, three components of a workflow for an effective handling of spatio-temporal data have been developed: *data management*, *data dissemination* and a *client application*.

Within this section the workflow is demonstrated based on a typical data set. Therefore, appropriate spatio-temporal data is required. One suitable scenario for such data is the modelling of the continuous change of a region over a certain period of time. The data set of the Piesberg landfill site in Osnabrück offers such a data base [Lautenbach and Berlekamp, 2002], [Butwilowski, 2015]. From 1976 to 1993 in total 12 models of the Piesberg were developed. They describe the continuous transformation of this region, see Figure 84.

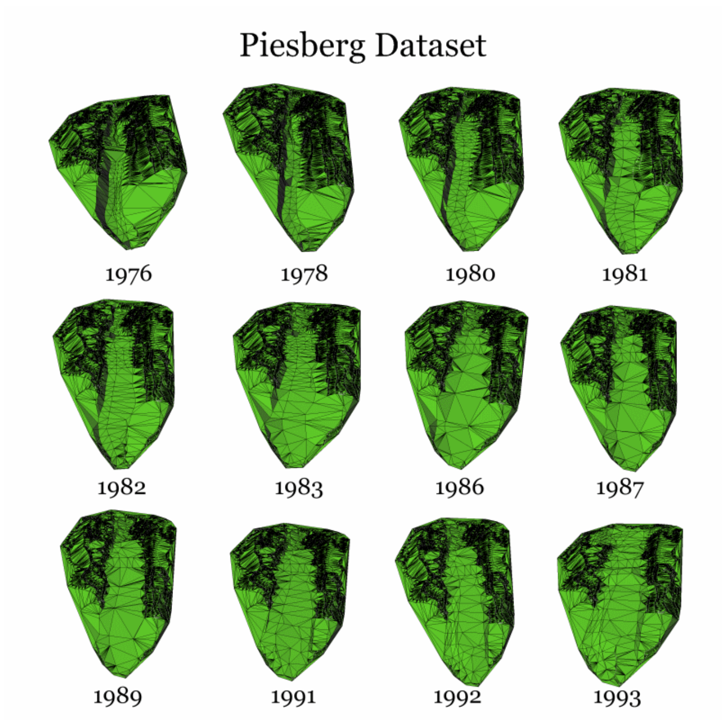


Figure 84 - Piesberg dataset with 12 time steps from 1976 to 1993.
Source: [Butwilowski, 2015]

First, the data, which is represented in the GOCAD file format, is integrated into the internal structure of the spatio-temporal database presented in Chapter 5.3. For this purpose, a corresponding 4D model is constructed by the use of the GOCAD file-based workflow, cf. Chapter 5.3.5. During this process the particular concepts of the comprehensive data management model are used, cf. Chapter 3.

Subsequently, the data and corresponding spatio-temporal operations are provided by the Web services described in Chapter 4. This includes WFS queries for single time steps of the model as well as requests of snapshots that represent the model for a specific date. The latter are created by an interpolation between two consecutive time steps. In addition, spatio-temporal operations such as the average speed of the geometry of a 4D model are queried by the use of the WPS.

In a next step, such requests are specified and performed by a generic mobile GIS for spatio-temporal data as presented in Chapter 5. Snapshots that describe the state of the Piesberg for a specific time $t_x \in [t_{start}, t_{end}]$ are visualized directly within the mobile GIS. Alternatively, the data is displayed on top of the camera image by the use of the AR-mode, cf. Chapter 5.4. This must be realized on site. An overview of the workflow and the corresponding data flows is illustrated in Figure 85.

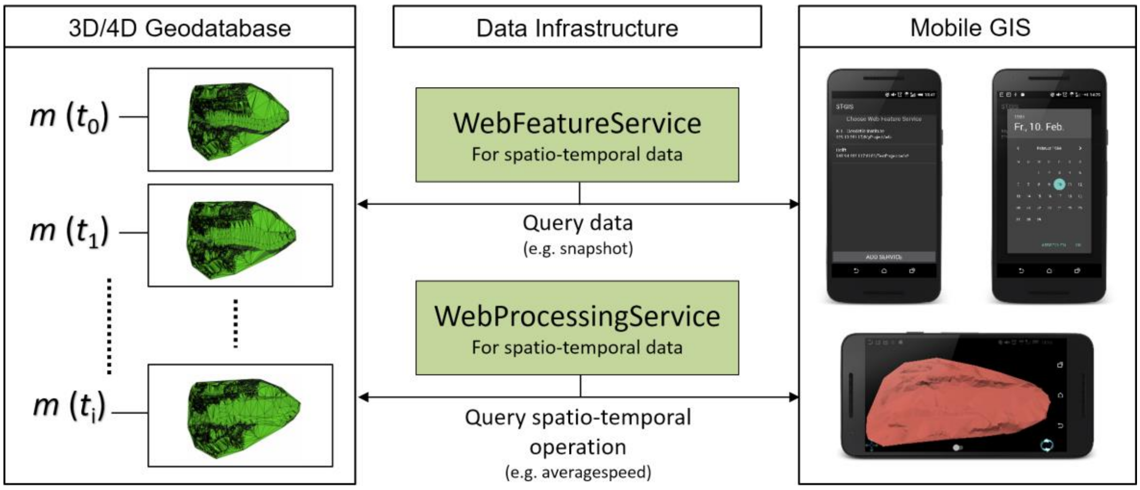


Figure 85 - Overview of the workflow for the handling of spatio-temporal data.

This completes the workflow for processing spatio-temporal data. In a further step, the 4D model of the Piesberg as well as other 4D models can be easily extended with additional time steps.

7 Summary and Outlook

Geoscientific disciplines such as geology, geophysics and geodesy generate various kinds of spatial and spatio-temporal data. Several modelling tools are involved in this process using multiple spatial representations such as voxel, g-maps or simplicial complexes. This thesis focused on the model of simplicial complexes that is used for the geometric modelling of natural structures, e.g. the earth's surface and subsurface. The corresponding data include *point clouds*, *segment-*, *triangle-* and *tetrahedron-nets*, i.e. *d-simplicial complexes* with $d \in \{0,1,2,3\}$.

Once time is introduced to model the continuous process of spatial structures, an efficient handling of vast amounts of data with an appropriate data management model is required. Additionally, this thesis treats the dissemination of such data to establish a collaborative and comfortable access to the latest information for various client applications. Finally, mobile GIS are addressed as a typical client application, which provide access to data storage and analysis functions on site.

In this thesis three parts of an efficient and practical workflow for the handling of spatio-temporal data were presented.

The first part is a *data management model for spatio-temporal data*, which is based on the five concepts Point Tubes, Delta Storage, Topology Management, Handling of Thematic Data, and Net Components. First, the adjustments of these concepts along with their impact on different net dimensions were elaborated. Based on this, the concepts were merged into a comprehensive data management model and the internal data structure of this model was described in detail. With this model, it is possible to handle various kinds of spatio-temporal applications. These include massive changes between consecutive time steps and a differentiated handling of thematic and semantic data. Variable temporal discretizations of partial regions of a 4D model are managed accordingly. Critical issues such as the preservation of the net topology between time steps and the handling of boundary regions of adjacent net components could be solved. The data management model reduces the storage requirements for the handling of 4D models and accelerates spatial and spatio-temporal operations significantly.

The model was implemented based on the database architecture DB4GeO. Subsequently, the implementation was validated by the use of multiple test scenarios within a JUnit framework. API- and file-based data sets were used to check the behaviour each concept individually and also in interaction with the other concepts. As a result of this examination it was discovered that the developed data management model fulfils the postulated requirements and that the individual concepts work as expected. The implementation is provided to the community as free open source software.

The second part of the workflow is a *generic approach for the dissemination of spatio-temporal data via a network interface*. First, the particular requirements for the dissemination of spatio-temporal data and operations were identified and presented. With the OGC standards WFS and WPS two widely used standards were selected as a foundation for the realization of two corresponding Web services. One for the dissemination of spatio-temporal data that is based on *d*-simplicial complexes and a second for spatio-temporal operations. Therefore, it is now possible to query specific time steps, to compute interpolated snapshots, and to determine the average speed of a 4D model via a network interface, e.g. the Internet. These Web services were developed and realized on the basis of the previously developed data management model. The implementation is based on a RESTful web application framework. The approach follows the rules of the OGC Web service standards and a communication with various clients can easily be established.

Finally, as third part of the workflow a *generic mobile GIS for spatial-temporal data* was designed and developed as a typical application. First, the definition of a universal generic mobile GIS was specified based on the requirements of mobile GIS in general. It was shown how such a generic mobile GIS can be used in various applications. Subsequently, additional specific requirements for spatio-temporal data were identified and described. Based on this, a generic mobile GIS for spatio-temporal data was designed. An implementation of a prototype of such a mobile GIS application, based on the mobile OS Android, was developed. Specific snapshots of 4D models are retrieved via a WFS and are visualized directly on the mobile device. A link to a WPS was established and spatio-temporal operations were queried. The implementation is provided to the community as free open source software.

Each part of the workflow individually improves the handling and management of spatio-temporal data:

- The data management model improves the performance of computations and the processing of spatio-temporal data. Therefore, the response time of requests and storage requirements are reduced.
- The two Web services enable the dissemination and processing of spatio-temporal data via a network interface. Thus, a standard based solution for a simultaneous use by multiple clients is now available.
- The mobile GIS application offers an examination of spatio-temporal data on site and supports the collaboration between office and field work.

The combination of these parts form an efficient and state of the art workflow that is beneficial for research fields such as *geothermal energy*, *geophysics* and *geomorphology* in terms of *data management*, *data dissemination* and *field work*.

During the entire development process one key challenge was that individual parts of the workflow had to remain replaceable. To achieve this, the respective realizations were based on an extensive use of well established standards, so that new parts can still be easily integrated into the workflow.

The integration of further components into the workflow could, for instance, include 3D/4D modelling tools such as GOCAD as a conventional client application. Another promising group are virtual reality (VR) devices such as Ocultus Rift[41] or HTC Vive[42]. Appropriate data can be prepared within a 3D/4D geodatabase architecture such as DB4GeO. Due to the capabilities of VR devices the user can experience a real 3D impression. A real time interpolation between individual time steps of 4D models could improve the visualization of particular changes within a 4D model. Due to a cable connection to a conventional desktop PC the required computation power should be available.

One direction of further developments is the realization of advanced spatio-temporal analysis functions. With the data management model developed in the thesis at hand, the preconditions for a development of such analysis functions are now established. Due to the use of well known software design standards the integration of additional operations and analysis functions can be handled in a simple manner. Such functions can include, for instance, the calculation of differences of a geometric model between two time steps of a 4D model or the prediction of a transformation for future dates.

The data management model was implemented on top of an object-oriented DBMS due to the complex and widely ramified structure of spatio-temporal data. Nevertheless, the model can be implemented in any type of DBMS since the implementation is independent of the model itself. Therefore, it would be interesting to have an additional realization based on an object-relational DBMS and to compare the results of both approaches.

As pointed out, there are several advanced query facilities which can be used in different application fields. Further investigations can focus on the detection of additional use cases. The access to these facilities, e.g. based on the implementation in DB4GeO, should be designed in a user-friendly way, i.e. a simple access should be provided.

[41] Oculus Rift - a virtual reality device by Oculus VR. Released in March 2016.
[42] HTC Vive Rift - a virtual reality device by HTC. Released in April 2016.

The concept of a general generic mobile GIS was used as a foundation for a realization for spatio-temporal data. Parallel to this thesis the concept has been used for the realization of a generic mobile GIS for 2D data, which led to the development of the commercial product *GeoTechMobile*. This proves the practicability of the concept for 2D data and should be further investigated. One specific direction of further developments could be the human-computer interaction.

Another aspect is the importance of mobile GIS in smart city environments. In a smart city the integration and handling of spatial data is an essential aspect. Therefore, professional mobile GIS can be used for complex workflows like the organisation of help in case of emergency.

The preconditions for respective future research are accomplished with this thesis.

8 References

[Abraham and Roddick, 1999] Abraham T., Roddick J. F. (1999) "Survey of Spatio-Temporal Databases", GeoInformatica, vol. 3, no. 1, pp. 61–99.

[Allen, 1984] Allen J. F. (1984) "Towards a general theory of action and time", Artificial Intelligence, vol. 23, no. 2, pp. 123–154.

[Alms et al., 1998] Alms R., Balovnev O., Breunig M., Cremers A.B., Jentzsch T., Siehl A. (1998) "Space- time modelling of the Lower Rhine Basin supported by an object-oriented database". Physics and Chemistry of the Earth, Vol. 23(3), pp. 251-260.

[Alonso et al., 2004] Alonso G., Casati F., Kuno H., Machiraju V. (2004) "Web Services: Concepts, Architectures and Applications", Berlin, Heidelberg: Springer Berlin Heidelberg.

[Apel, 2004] Apel M. (2004) "A 3d geoscience information system framework", PhD Thesis, Faculty of Geosciences, Geotechnology and Mining, University of Technology Bergakademie Freiberg, 94p.

[Balovnev et al., 2004] Balovnev O., Bode Th., Breunig M., Cremers A.B., Müller W., Pogodaev, G., Shumilov, S., Siebeck, J., Siehl, A., Thomsen, A. (2004) "The Story of GeoToolKit – an Object-oriented Geo-Database Kernel System", Geoinformatica 8:1, Kluwer Academic Publishers, pp. 5-47.

[Bär, 2007] Bär W. (2007) "Management of geo-scientific 3D data in mobile database systems", Ph.D. Thesis, University of Osnabrück, Germany, 166p.

[Bartelme, 2005] Bartelme N. (2005) "Geoinformatik: Modelle, Strukturen, Funktionen", 4. vollständig überarb. Auflage, Berlin: Springer, 466p.

[Bast et al., 2014] Bast H., Delling D., Goldberg A., Müller-Hannemann M., Pajor T., Sanders P., Wagner D., Werneck R. (2014) "Route Planning in Transportation Networks ", Microsoft Research, TechReport, MSR-TR-2014-4, 57p.

[Beckmann et al., 1990] Beckmann N., Kriegel H.-P., Schneider R., Seeger B. (1990) "The R*-tree: An Efficient and Robust Access Method for Points and Rectangles", Proceedings of the 1990 ACM SIGMOD International Conference on Management of Data, New York, NY, USA, pp. 322–331.

[Biljecki et al., 2015] Biljecki F., Stoter J., Ledoux H., Zlatanova S., Çöltekin A. (2015) "Applications of 3D City Models: State of the Art Review", ISPRS International Journal of Geo-Information 4(4), December 2015, pp. 2842–2889.

[Bode et al., 1994] Bode T., Breunig M., Cremers A.B. (1994) "First Experiences with GEOSTORE, an Information System for Geologically Defined Geometries", IGIS1994, Proceedings of the Intern. Workshop on Advanced Research in Geographic Information Systems, Monte Verita, Ascona, Switzerland, Lecture Notes in Computer Science 884, Springer, Heidelberg, pp. 35-44.

[Borrmann et al., 2012] Borrmann A., Ji Y., Jubierre J. R. (2012) "Multi-scale geometry in civil engineering models: Consistency preservation through procedural representations", Proc. of the 14th Int. Conf. on Computing in Civil and Building Engineering, Moscow, Russia, 8p.

[Bottero et al., 2013] Bottero M., Dalla Chiara B., Deflorio F.P. (2013) "Wireless sensor networks for traffic monitoring in a logistic centre", Transportation Research Part C: Emerging Technologies, Volume 26, pp. 99-124.

[Breunig and Zlatanova, 2011] Breunig M., Zlatanova S. (2011) "3D geo-database research: retrospective and future directions". Computers & Geosciences, 37, pp. 791-803.

[Breunig et al., 1999] Breunig M., Cremers A.B., Götze H.-J., Schmidt S., Seidemann R., Shumilov S., Siehl A. (1999) "First steps towards an interoperable 3D GIS - an example form southern Lower Saxony, Germany", Physics and Chemistry of the Earth, Part A, Vol. 24(3), pp. 179-190.

[Breunig et al., 2000] Breunig M., Cremers A.B., Götze H.-J., Schmidt S., Seidemann R., Shumilov S., Siehl A. (2000) "Geological Mapping based on 3D models using an Interoperable GIS", GIS – Journal for Spatial Information and Decision Making, Vol. 13, No.2/2000, pp. 12-18.

[Breunig et al., 2009] Breunig M., Schilberg B., Thomsen A., Kuper P.V., Jahn M., Butwilowski E. (2009) "Towards a 3D/4D geo-database supporting the analysis and early warning of landslides", Proceedings of Cartography and Geoinformatics in Early Warning and Crises Management - towards better solutions, Prag, 17p.

[Breunig et al., 2010] Breunig M., Schilberg B., Thomsen A., Kuper P.V., Jahn M., Butwilowski E. (2010) "DB4GeO, a 3D/4D geodatabase and its application for the analysis of landslides", Geographic nformation and Cartography for Risk and Crisis Management. Lecture Notes in Geoinformation and Cartography, Springer, Heidelberg, pp. 83-102.

[Breunig et al., 2011] Breunig M., Rank E., Schilcher M., Borrmann A., Hinz S., Mundani R.-P., Ji Y., Menninghaus M., Donaubauer A., Steuer H., Vögtle T. (2011) "Towards computer-aided collaborative subway track planning in multi-scale 3D city and building models", Proc. of the 6th International ISPRS Conference on 3D Geoinformation, Wuhan, China, 8p.

[Breunig et al., 2013a] Breunig M., Butwilowski E., Golovko D., Kuper P.V., Menninghaus M., Thomsen A. (2013) "Advancing DB4GeO", Progress and New Trends in 3D Geoinformation Sciences, Lecture Notes in Geoinformation and Cartography, Springer, pp. 193-210.

[Breunig et al., 2013b] Breunig M., Kuper P. V., Dittrich A., Wild P., Butwilowski E., Al-Doori M. (2013) "Design, Implementation and Applications of 3D Web-Services in DB4GeO", ISPRS Ann. Photogramm. Remote Sens. Spatial Inf. Sci., II-2/W1, 87-94, doi:10.5194/isprsannals-II-2-W1-87-2013, 8p.

[Breunig et al., 2016] Breunig M., Kuper P.V., Butwilowski E., Thomsen A., Jahn M., Dittrich A., Al-Doori M., Golovko D., Menninghaus M. (2016) "The Story of DB4GeO - A Service-Based Geodatabase Architec-ture to Support Multi-Dimensional Data Analysis and Visualization", ISPRS Journal of Photogrammetry and Remote Sensing, Volume 117, pp. 187-205.

[Breunig, 2001] Breunig M. (2001) "On the Way to Component-Based 3D/4D Geoinformation Systems", Lecture Notes in Earth Sciences vol. 94, Springer Berlin Heidelberg, 199p.

[Butwilowski, 2015] Butwilowski E. (2015) "3D data and model management for the geosciences with particular emphasis on topology and time", PhD thesis, Karlsruhe Institute of Technology.

[Carey and Bell, 1997] Carey R., Bell G. (1997) "The Annotated VRML 2.0 Reference Manual", Essex, UK, Addison-Wesley Longman Ltd., 1997.

[Chen and Guinness, 2014] Chen R., Guinness R. (2014) "Geospatial Computing in Mobile Devices", Artech House, 211p.

[Chen, 1976] Chen P. P.-S. (1976) "The Entity-relationship Model—Toward a Unified View of Data", ACM Trans. Database Syst., vol. 1, no. 1, pp. 9–36.

[Cheng, 1997] Cheng, A. (1997) "GOCAD in a Hurry", 4th Edition, ASGA Nancy, France, 184p.

[Cheon and Leavens, 2002] Cheon Y., Leavens G. T. (2002) "A Simple and Practical Approach to Unit Testing: The JML and JUnit Way", ECOOP 2002 — Object-Oriented Programming, B. Magnusson, Ed. Springer Berlin Heidelberg, pp. 231–255.

[Dadam et al., 1984] Dadam P., Lum V. Y., Werner H.-D. (1984) "Integration of Time Versions into a Relational Database System", Proceedings of the 10th International Conference on Very Large Data Bases, San Francisco, CA, USA, pp. 509–522.

[Daigneau, 2011] Daigneau R. (2011) "Service Design Patterns: Fundamental Design Solutions for SOAP/WSDL and RESTful Web Services", 1 edition. Upper Saddle River, NJ: Addison-Wesley Professional.

[Dao et al., 2015] Dao T. T., Pouletaut P., Charleux F., Lazáry Á., Eltes P., Varga P. P., Ho Ba Tho M. C. (2015) "Multimodal medical imaging (CT and dynamic MRI) data and computer-graphics multi-physical model for the estimation of patient specific lumbar spine muscle forces", Data & Knowledge Engineering, vol. 96–97, pp. 3–18.

[Deininger, 2012] Deininger K. (2012) "Rechnergestützte Geodatenerfassung mit Android", Diploma thesis, Geodetic Institute, Karlsruhe Institute of Technology, 154p.

[Dittrich, 2012] Dittrich A.P. (2012) "Entwicklung eines 3D-Reality-Viewers für geowissenschaftliche Daten auf Android Plattformen", MSc. thesis, Geodetic Institute, Karlsruhe Institute of Technology, 74p.

[Doyle et al., 2010] Doyle J., Bertolotto M., Wilson D. (2010) "Evaluating the benefits of multi-modal interface design for CoMPASS—a mobile GIS", Geoinformatica, vol. 14, no. 2, pp. 135–162.

[Egenhofer et al., 1990] Egenhofer M. J., Frank A. U., Jackson J. P. (1990) "A Topological Data Model for Spatial Databases", Proceedings of the First Symposium on Design and Implementation of Large Spatial Databases, New York, NY, USA, pp. 271–286.

[Fielding, 2000] Fielding R.Th. (2000) "Architectural styles and the design of network-based software architectures", Ph.D. thesis, University of California, Irvine, 180p.

[Förstner, 1995] Förstner W. (1995) "GIS - The Third Dimension", In Proc. IUSM WG on GIS/LIS Workshop "Current Status and Challenges of Geoinformation Systems". Hannover, pp. 65-72.

[Frank et al., 2004] Frank C., Caduff D., Wuersch M. (2004) "From GIS to LBS–An Intelligent Mobile GIS", IfGI prints, vol. 22, pp. 261–274.

[Gabriel et al., 2015] Gabriel P., Gietzel J., Le H.H., Schaeben H. (2015) "GST: A Network Based Datastore for Geoscience Data and Geomodels and Its Implementation—ProMine's Contribution Towards Interoperability", 3D, 4D and Predictive Modelling of Major Mineral Belts in Europe, P. Weihed, Ed. Springer International Publishing, pp. 59–71.

[Götze and Lahmeyer, 1988] Götze H.-J., Lahmeyer B. (1988) "Application of three-dimensional interactive modeling in gravity and magnetics", Geophysics Vol. 53, No. 8, pp. 1096-1108.

[Gröger and Plümer, 2005] Gröger G., Plümer L. (2005) "How to Get 3-D for the Price of 2-D—Topology and Consistency of 3-D Urban GIS", Geoinformatica, vol. 9, no. 2, pp. 139–158.

[Güting and Schneider, 2005] Güting R. H., Schneider M. (2005) "Moving Objects Databases", Amsterdam: Boston: Elsevier Science & Technology, 416p.

[Güting, 1994] Güting R.H. (1994) "An Introduction to Spatial Database Systems", VLDB Journal 3 (4), pp. 357-399.

[Han et al., 2015] Han W., Hu Y., Zhang J. (2015) "Mobile Data Acquisition and Management System Design Based on GIS and GPRS", Metallurgical and Mining Industry, No. 2, p. 243-249.

[Hoppe, 1996] Hoppe H. (1996) "Progressive meshes", ACM SIGGRAPH Conference Proceedings, pp. 99-108.

[Hu et al., 2010] Hu L., Sander P. V., Hoppe H. (2010) "Parallel view-dependent level-of-detail control", IEEE Trans Vis Comput Graph, vol. 16, no. 5, pp. 718–728.

[INSPIRE, 2007] Inspire, Infrastructure for Spatial Information in the European Community, 2007: http://inspire.ec.europa.eu, last accessed 2016-04-15

[Isenburg, 2002] Isenburg M. (2002) "Compressing Polygon Mesh Connectivity with Degree Duality Prediction", Proceedings of Graphics Interface 2002, pp. 161-170.

[Jensen et al., 1992] Jensen C. S., Clifford J., Gadia S. K., Segev A., Snodgrass R.T. (1992) "A glossary of temporal database concepts.", SIGMOD Rec. 21, pp. 35-43.

[Kadlec et al., 2012] Kadlec J., Ames P. D., Nelson J. (2012) "User Interface Design Considerations for a Time-Space GIS", International Congress on Environmental Modelling and Software Managing Resources of a Limited Planet, Sixth Biennial Meeting, Leipzig, Germany, 8p.

[Kälberer et al., 2005] Kälberer F., Polthier K., Reitebuch U., Wardetzky M., (2005) "FreeLence - Coding with Free Valences", Computer Graphics Forum, vol. 24, no. 3, pp. 469–478.

[Kang et al., 2015] Kang B., Lee J., Kissinger J., Lee R. Y. (2015) "A Procedure for the Development of Mobile Applications Software", Software Engineering Research, Management and Applications, R. Lee, Ed. Springer International Publishing, pp. 141–150.

[Kolbe, 2009] Kolbe T. H. (2009) "Representing and Exchanging 3D City Models with CityGML", 3D Geo-Information Sciences, J. Lee and S. Zlatanova, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, pp. 15–31.

[Kuper et al., 2014] Kuper P.V., Butwilowski E., Breunig M., Wild P., Hoch T., Wiedemann S., Klauser B. (2014) "Mobile Geodaten-Erfassung mit GeoTab", Publikationen der Deutschen Gesellschaft für Photogrammetrie, Fernerkundung und Geoinformation e.V. Band 23, Potsdam, 10p.

[Kuper, 2010] Kuper, P.V. (2010) "Entwicklung einer 4D Objekt-Verwaltung für die Geodatenbank DB4GeO", Diploma thesis, University of Osnabrück, 64p.

[Langran and Chrisman, 1988] Langran G., Chrisman N.R. (1988) "A Framework for Temporal Geographic Information", CARTOGRAPHICA Vol 25 No. 3, pp. 1-14.

[Lautenbach and Berlekamp, 2002] Lautenbach S., Berlekamp J. (2002) "Datensatz zur Visualisierung der Zentraldeponie Piesberg in Osnabrück", University of Osnabrück.

[Le et al., 2013] Le H.H., Gabriel P., Gietzel J., Schaeben H. (2013) "An object-relational spatio-temporal geoscience data model", Computers & Geosciences, Vol. 57, 104-115.

[Le et al., 2014] Le H.H., Schaeben H., Jasper H., Görz I. (2014) "Database Versioning and Its Implementation in Geoscience Information Systems", Computers and Geosciences, vol. 70, no. C, pp. 44–54.

[Le, 2014] Le H.H. (2014) "Spatio-temporal Information System for the Geosciences: Concepts, Data models, Software, and Applications", PhD thesis, Technische Universität Bergakademie Freiberg, 169p.

[Lévy, 2000] Lévy B. (2000) "Topologie Algorithmique - Combinatoire et Plongement", PhD Thesis, INPL Nancy, 202p.

[Li and Jiang, 2011] Li C., Jiang Y. (2011) "Development of mobile GIS system for forest resources second-class inventory", Journal of Forestry Research, vol. 22, no. 2, pp. 263–268.

[Li, 2015] Li Z. (2015) "CAD, 3D Modeling, Engineering Analysis, and Prototype Experimentation: Industrial and Research Applications", Springer, 255p.

[Lienhardt, 1991] Lienhardt P. (1991) "Topological models for boundary representation: a comparison with n-dimensional generalized maps", Computer Aided Design 23(1), pp. 59-82.

[Mallet, 1992] Mallet J.L. (1992) "GOCAD: a Computer Aided Design Programme for Geological Applications", Turner A.K. (Ed.), Three-Dimensional Modelling with Geoscientific Information Systems. Proceedings of NATO ASI 354, Kluwer Academic Publishers, Dordrecht, pp. 123-142.

[Mallet, 2002] Mallet J.L. (2002) "Geomodeling", Oxford University Press, 599 p.

[Meek et al., 2013] Meek S., Priestnall G., Sharples M., Goulding J. (2013) "Mobile capture of remote points of interest using line of sight modelling", Computers & Geosciences, vol. 52, pp. 334–344.

[Mejia et al., 2015] Mejia P., Royer J. J., Fraboulet J. G., Zielińska A. (2015) "4D Geomodelling A Tool for Exploration—The Kupferschiefer in The Lubin Region, Poland", 3D, 4D and Predictive Modelling of Major Mineral Belts in Europe, Mineral Resource Reviews, pp 163-187.

[Menninghaus et al., 2016] Menninghaus M., Breunig M., Pulvermüller E. (2016) "High-Dimensional Spatio-Temporal Indexing", Open Journal of Databases (OJDB), 3 (1), pp. 1-20.

[Menninghaus, 2010] Menninghaus M. (2010) "Mobile Geodatenvisualisierung mit Android", Proceedings 5. MMS Conference, Münster, pp. 15-28.

[Milanes et al., 2014] Milanes A., Stevens M., Alford D. (2014) "Mobile GIS and Real-Time Data Display in a Common Operating Picture", International Oil Spill Conference Proceedings, vol. 2014, no. 1, pp. 1607–1620.

[Neis et al., 2011] Neis P., Zielstra D., Zipf A. (2011) "The Street Network Evolution of Crowdsourced Maps: OpenStreetMap in Germany 2007–2011", Future Internet, vol. 4, no. 1, pp. 1–21.

[Neteler et al., 2012] Neteler M., Bowman M. H., Landa M., Metz M. (2012) "GRASS GIS: A multi-purpose open source GIS", Environmental Modelling & Software, vol. 31, pp. 124–130.

[Nogueras-Iso, 2005] Nogueras-Iso J., Zarazaga-Soria F. J., Béjar R., Álvarez P. J., Muro-Medrano P. R. (2005) "OGC Catalog Services: a key element for the development of Spatial Data Infrastructures", Computers & Geosciences, vol. 31, no. 2, pp. 199–209.

[OGC SF] OGC® Inc., Open Geospatial Consortium, 2011: Simple Feature Access Specification 1.2.1., http://www.opengeospatial.org/standards/sfa

[OGC SLD] OGC® Inc., Open Geospatial Consortium, 2007: Styled Layer Descriptor profile of the Web Map Service Implementation Specification 1.1.0., http://www.opengeospatial.org/standards/sld

[OGC Web Service Common] OGC® Inc., Open Geospatial Consortium, 2010: Web Service Common Implementation Specification 2.0.0., http://www.opengeospatial.org/standards/common

[OGC WFS] OGC® Inc., Open Geospatial Consortium, 2005: Web Feature Service Implementation Specification 1.1.0., http://www.opengeospatial.org/standards/wfs

[OGC WMS] OGC® Inc., Open Geospatial Consortium, 2006: Web Map Service Implementation Specification 1.3.0., http://www.opengeospatial.org/standards/wfs

[OGC WPS] OGC® Inc., Open Geospatial Consortium, 2007: Web Processing Service 1.0.0., http://www.opengeospatial.org/standards/wps

[OGC, 2011] OGC® Inc. (ed.) (2011) "OpenGIS® Implementation Standard for Geographic information - Simple feature access - Part 1: Common architecture".

[Ohori et al., 2015] Ohori K. A., Ledoux H., Stoter J. (2015) "An evaluation and classification of nD topological data structures for the representation of objects in a higher-dimensional GIS", International Journal of Geographical Information Science, vol. 29, no. 5, pp. 825–849.

[Ohori, 2016] Ohori K.A. (2016) "Higher-dimensional modelling of geographic information", PhD thesis, Delft University of Technology, 236p.

[Oosterom and Stoter, 2010] van Oosterom P., Stoter J. (2010) "5D Data Modelling: Full Integration of 2D/3D Space, Time and Scale Dimensions", Proceedings of the 6th International Conference on Geographic Information Science, Berlin, Heidelberg, pp. 310–324.

[Osherove, 2013] Osherove, R. (2013) "The Art of Unit Testing", 2nd edition, Manning, 266p.

[Paterson et al., 2006] Paterson J., Edlich S., Hörning H., Hörning R. (2006) "The Definitive Guide to db4o", Apress, 484p.

[Pelekis et al., 2004] Pelekis N., Theodoulidis B., Kopanakis I., Theodoridis Y. (2004) "Literature Review of Spatio-temporal Database Models", Knowl. Eng. Rev., vol. 19, no. 3, pp. 235–274.

[Penninga, 2008] Penninga F. (2008) "3D Topography. A Simplicial Complex-based Solution in a Spatial DBMS", PhD thesis, Publications on Geodesy 66, TU Delft, Netherlands, 204p.

[Peuquet, 1984] Peuquet D. J. (1984) "A conceptual framework and comparison of spatial data models", Cartographica: The International Journal for Geographic Information and Geovisualization, vol. 21, no. 4, pp. 66–113.

[Phong, 1975] Phong B. T. (1975) "Illumination for Computer Generated Pictures", Commun. ACM, vol. 18, no. 6, pp. 311–317.

[Polthier and Rumpf, 1994] Polthier K., Rumpf M. (1994) "A concept for Time-Dependent Processes", Goebel et al., (Eds), Visualization in Scientific Computing, Springer, Vienna, pp. 137-153.

[Pouliot and Bédard, 2007] Pouliot, D., Bédard, T. (2007) "Development of a Web Geological Feature Server (WGFS) for sharing and querying of 3D objects", 2nd International Workshop on 3D Geo-Information, Delft, Netherlands, 15p.

[Pouliot et al., 2013] Pouliot J., Daniel S., Hubert F., Zamyadi A. Eds. (2013) "Progress and New Trends in 3D Geoinformation Sciences", 2013 edition. Heidelberg, New York: Springer, 314p.

[Ramakrishnan and Gehrke, 2002] Ramakrishnan R., Gehrke J. (2002) "Database Management Systems", 3rd Edition. Boston: McGraw-Hill, 1104p.

[Raubal and Panov, 2009] Raubal M., Panov I. (2009) "A Formal Model for Mobile Map Adaptation", Location Based Services and TeleCartography II, G. Gartner and K. Rehrl, Eds. Springer Berlin Heidelberg, pp. 11–34.

[Raubal, 2011] Raubal M. (2011) "Cogito Ergo Mobilis Sum: The Impact of Location-based Services on Our Mobile Lives", In The SAGE Handbook of GIS and Society, edited by Timothy L. Nyerges, Helen Couclelis and Robert McMaster, London: SAGE Publications Ltd., pp. 159-175.

[Renolen, 1997] Renolen A. (1997) "Temporal Maps and Temporal Geographical Information Systems (Review of Research)", Department of Surveying and Mapping, The Norwegian Institute of Technology, 12p.

[Richardson and Ruby, 2007] Richardson L., Ruby S. (2007) "RESTful Web Services", 1st ed. Farnham: O'Reilly & Associates, 454p.

[Rinner et al., 2005] Rinner C., Raubal M., Spigel B. (2005) "User Interface Design for Location-Based Decision Services", Proceedings of the 13th International Conference on GeoInformatics, 6p.

[Rinner, 2008] Rinner C. (2008) "Mobile Maps and More – Extending Location-Based Services with Multi-Criteria Decision Analysis", Map-based Mobile Services, L. Meng, A. Zipf, and S. Winter, Eds. Springer Berlin Heidelberg, pp. 335–352.

[Roddick et al., 2004] Roddick J. F., Hoel E., Egenhofer M. J., Papadias D., Salzberg B. (2004) "Spatial, Temporal and Spatio-temporal Databases - Hot Issues and Directions for Phd Research", SIGMOD Rec., vol. 33, no. 2, pp. 126–131.

[Rolfs, 2005] Rolfs C. (2005) "Konzeption und Implementierung eines Datenmodells zur Verwaltung von zeitabhängigen 3D-Modellen in geowissenschaftlichen Anwendungen", Diploma thesis, 90p.

[Royer et al., 2015] Royer J. J., Mejia P., Caumon G., Collon P. (2015) "3D and 4D Geomodelling Applied to Mineral Resources Exploration - An Introduction", 3D, 4D and Predictive Modelling of Major Mineral Belts in Europe, Mineral Resource Reviews, pp. 73-89.

[Runder Tisch GIS e.V., 2014] Runder Tisch GIS e.V. (2014) "Leitfaden Mobiles GIS und standortbezogene Dienst", 133p.

[Schmidt and Götze, 1998] Schmidt S., Götze H.-J. (1998) "Interactive visualization and modification of 3D-models using GIS-functions", Physics and Chemistry of the Earth, Elsevier Science Ltd. Vol. 23, pp. 289-295.

[SFB350] Sonderforschungsbereich 350, Report (2000) "Wechselwirkungen Kontinentaler Soffysteme und ihre Modellierung", University of Bonn, Germany, 507p.

[Shekhar et al., 1997] Shekhar S., Coyle M., Goyal B., Liu D.-R., Sarkar S. (1997) "Data Models in Geographic Information Systems", Commun. ACM, vol. 40, no. 4, pp. 103–111.

[Shi et al., 2009] Shi W., Kwan K., Shea G., Cao J. (2009) "A dynamic data model for mobile GIS", Computers and Geosciences 35, 11, pp. 2210-2221.

[Shitkova et al., 2015] Shitkova M., Holler J., Heide T., Clever N., Becker J. (2015) "Towards Usability Guidelines for Mobile Websites and Applications", Proceedings 12. Internationalen Tagung Wirtschaftsinformatik, Osnabrück, Germany, pp. 1603–1617.

[Shumilov and Siebeck, 2001] Shumilov S., Siebeck J. (2001) "Database Support for Temporal 3D Data: Extending The GeoToolKit", Proc. of the 7th EC-GI and GIS Workshop, Potsdam, Germany, 14p.

[Shumilov et al., 2002] Shumilov S., Thomsen A., Cremers A.B., Koos, B. (2002) "Management and visualisation of large, complex and time-dependent 3D objects in distributed GIS", ACM-GIS 2002, 113-118.

[Siebeck, 2003] Siebeck J. (2003) "Concepts for the Representation, Storage, and Retrieval of Spatio-Temporal Objects in 3D/4D Geo-Information-Systems", PhD Thesis, Faculty of Mathematics and Naturlal Sciences at the Rheinischen Friedrich-Wilhelms-Universit, Bonn, 143p.

[Snodgrass, 1987] Snodgrass R. (1987) "The temporal query language TQuel.", ACM Transactions on Database Systems (TODS) 12, pp. 247-298.

[Strathoff, 1999] Strathoff F. (1999) "Speichereffiziente Verwaltung zeitabhängiger Geometrien für ein GeoToolKit", Diploma thesis.

[Thomsen et al., 2008] Thomsen A., Breunig M., Butwilowski E., Broscheit B. (2008) "Modelling and managing topology in 3D geoinformation systems", Proceedings of 3D Geoinfo 07, Delft, Advances in 3DGeoinformation Systems, Lecture Notes in Geoinformation and Cartograhphy, Springer, Heidelberg, pp. 229-246.

[Tilkov, 2009] Tilkov S. (2009) "REST und HTTP: Einsatz der Architektur des Web für Integrationsszenarien", 1., Aufl. Heidelberg: dpunkt Verlag, 238p.

[Tomlinson, 1967] Tomlinson R.F. (1967) "An Introduction to the Geo-Information System of the Canada Land Inventory", Canada Land Inventory ARDA, 25p.

[Torge, 2003] Torge, W. (2003) "Geodäsie", de Gruyter, 369p.

[Urban et al., 2013] Urban S., Leitloff J., Wursthorn S., Hinz S. (2013) "Self-localization of a multi-fisheye Camera based augmented reality system in textureless 3D building models", ISPRS Ann. Photogramm. Remote Sens. Spatial Inf. Sci., II-3/W2, 43-48. doi:10.5194/isprsannals-II-3-W2-43-2013, 6p.

[W3C SOAP] World Wide Web Consortium (W3C), 2007: SOAP Version 1.2 Part 1: Messaging Framework (Second Edition), https://www.w3.org/TR/soap12

[Weihed, 2015] Weihed P. (2015) "3D, 4D and Predictive Modelling of Major Mineral Belts in Europe", 1st ed. 2015 edition. Springer, 330p.

[Wild, 2012] Wild P.A. (2012) "Entwicklung einer Service-Infrastruktur der Geodatenbank DB4GeO nach Richtlinien des OGC®-Standards", Diploma thesis, Karlsruhe Institute of Technology, 86p.

[Williams, 1976] Williams C. D. (1976) "GEOQUEL: A Geographical Data Retrieval and Display System", Research Project. University of California, Berkeley.

[Wolfson et al., 2002] Wolfson O., Cao H., Lin H., Trajcevski G., Zhang F., Rishe N. (2002) "Management of Dynamic Location Information in DOMINO", Advances in Database Technology — EDBT 2002, C. S. Jensen, S. Šaltenis, K. G. Jeffery, J. Pokorny, E. Bertino, K. Böhn, and M. Jarke, Eds. Springer Berlin Heidelberg, 2002, pp. 769–771.

[Worboys and Duckham, 2004] Worboys M., Duckham M. (2004) "GIS: A Computing Perspective", 2nd Edition. Boca Raton, FL, USA: CRC Press, Inc, 448p.

[Worboys, 1992] Worboys M. (1992) "A generic model for planar geographical objects", International Journal of Geographical Information Systems, vol. 6, no. 5, pp. 353–372.

[Worboys, 1994a] Worboys M. (1994) "A Unified Model for Spatial and Temporal Information", The Computer Journal, 37(1), pp. 25–34.

[Worboys, 1994b] Worboys, M. F. (1994) "Object-oriented approaches to geo-referenced information", International Journal of Geographical Information Systems, 8(4), pp. 385-399.

[Xing et al., 2015] Xing H., Liu Y., Gao J., Chen S. (2015) "Recent development in numerical simulation of enhanced geothermal reservoirs", J. Earth Sci., vol. 26, no. 1, pp. 28–36.

[Zajíčková, 2015] Zajíčková L. (2015) "Mobile Application for Acquiring Geodata on Public Transport Network", Geoinformatics for Intelligent Transportation, I. Ivan, I. Benenson, B. Jiang, J. Horák, J. Haworth, and T. Inspektor, Eds. Springer International Publishing, pp. 257–269.

List of Publications

Kuper, P.V., Breunig, M., Al-Doori, M., Thomsen, A. (2016) "Application of 3D Geospatial Plus Temporal Data Modeling, Management, and Analysis in DB4GeO", ISPRS Ann. Photogramm. Remote Sens. Spatial Inf. Sci., IV-2/W1, 163-170, doi:10.5194/isprs-annals-IV-2-W1-163-2016, 8p.

Breunig M., Kuper P.V., Butwilowski E., Thomsen A., Jahn M., Dittrich A., Al-Doori M., Golovko D., Menninghaus M. (2016) "The Story of DB4GeO - A Service-Based Geodatabase Architecture to Support Multi-Dimensional Data Analysis and Visualization.", ISPRS Journal of Photogrammetry and Remote Sensing, Volume 117, pp. 187-205.

Kuper P. (2016) "Management of spatio-temporal data with a link to mobile geographic information systems." Middle East Geospatial Forum, Proc. GRASF 2016 Conference, Dubai, abstract.

Breunig M., Al-Doori M., Butwilowski E., Kuper P.V., Benner J., Haefele K.-H., eds. (2015) "3D Geoinformation Science: The Selected Papers of the 3D GeoInfo 2014.", Lecture Notes in Geoinformation and Cartography. Heidelberg: Springer Science+Business Media, pp. 258.

Butwilowski E., Thomsen A., Breunig M., Kuper P.V., Al-Doori M. (2015) "Modeling and Managing Topology for 3-D Track Planning Applications.", 3D Geoinformation Science: The Selected Papers of the 3D GeoInfo 2014, Lecture Notes in Geoinformation and Cartography, Heidelberg: Springer, pp. 37-55.

Breunig M., Kuper P., Butwilowski E., Al-Doori M. (2015) "Mobile Spatial Data Acquisition with GeoTechMobile", Proc. GRASF 2015 Conference, abstract.

Kuper P., Butwilowski E., Breunig M., Al-Doori M. (2015) "The effective handling of geospatial and time-dependent data in the geodatabase DB4GeO", Proc. GRASF 2015 Conference, abstract.

Kuper P.V., Butwilowski E., Breunig M., Wild P., Hoch T., Wiedemann S., Klauser B. (2014) "Mobile Geodaten-Erfassung mit GeoTab", Publikationen der Deutschen Gesellschaft für Photogrammetrie, Fernerkundung und Geoinformation e.V. Band 23, 10p., Potsdam.

Breunig M., Kuper P.V., Dittrich A., Wild P., Butwilowski E., Al-Doori M. (2013) "Design, Implementation and Applications of 3D Web-Services in DB4GeO", ISPRS Ann. Photogramm. Remote Sens. Spatial Inf. Sci., II-2/W1, pp. 87-94, doi:10.5194/isprsannals-II-2-W1-87-2013, 8p.

Breunig M., Butwilowski E., Kuper P.V., Golovko D., Thomsen A. (2013) "Topological and Geometric Data Handling for Time-Dependent Geo-Objects Realized in DB4GeO.", Advances in Spatial Data Handling, ed. Sabine Timpf and Patrick Laube, Advances in Geographic Information Science. Berlin, Heidelberg: Springer, pp. 1–13.

Breunig M., Butwilowski E., Golovko D., Kuper P.V., Menninghaus M., Thomsen A. (2013) "Advancing DB4GeO." In Progress and New Trends in 3D Geoinformation Sciences, ed. Jacynthe Pouliot, Sylvie Daniel, Frédéric Hubert, and Alborz Zamyadi, Lecture Notes in Geoinformation and Cartography. Berlin, Heidelberg: Springer, pp. 193–210.

Butwilowski E., Golovko D., Kuper P.V., Breunig M. (2012) "Distributed geo-databases for the management of geo-scientific data.", Proceedings of the 32nd gOcad Meeting. Nancy, France, 11p.

Breunig M., Butwilowski E., Kuper P.V., Paul N., Thomsen A., Schmidt S., Götze H.-J. (2011) "Handling of spatial data for complex geo-scientific modelling and 3D landfill applications with DB4GeO", Geoinformatik 2011, Muenster, 6p.

Thomsen A., Schmidt S., Götze H.-J., Breunig M., Schilberg B., Kuper P.V. (2010) "On the way to synoptic interpretation of geoscientific data in joint CCS project CO2-MoPa.", Geoinformatik 2010, Kiel, 8p.

Breunig M., Schilberg B., Thomsen A., Kuper P.V., Jahn M., Butwilowski E. (2010) "DB4GeO, a 3D/4D Geodatabase and its Application for the analysis of landslides.", Geographic Information and Cartography for Risk and Crisis Management. Lecture Notes in Geoinformation and Cartography, Springer, pp. 83-102.

Breunig M., Schilberg B., Thomsen A., Kuper P.V., Jahn M., Butwilowski E. (2009) "Towards a 3D/4D geo-database supporting the analysis and early warning of landslides.", Proceedings of Cartography and Geoinformatics in Early Warning and Crises Management - towards better solutions, Prag, 17p.

Breunig M., Schilberg B., Thomsen A., Kuper P.V., Jahn M., Butwilowski E. (2009) "DB4GeO: Developing 3D geo-database services", 4th International 3DGeoInfo Workshop, Ghent, Belgium, 6p.

Breunig M., Schilberg B., Kuper P.V., Jahn M., Reinhardt W., Nuhn E., Mäs S., Boley C., Trauner F.X., Wiesel J., Richter D., Abecker A., Gallus D., Kazakos W., Bartels A. (2009) "EGIFF - Developing advanced GI methods for early warning in mass movement scenarios", Geotechnologien Science Report, Statusseminar "Frühwarnsysteme gegen Naturgefahren", Geotechnologien Programm, München, 24p.