DELFT UNIVERSITY

OF TECHNOLOGY

Master Thesis in Geomatics

Solid CAD Geometries of the Petrochemical Industry in a Spatial DBMS

P2 – Final Graduation Plan

Author:
Myron Ramkisoen
4078691

Graduation Professor:
Prof. dr. ir. Peter van Oosterom

Daily Supervisor:
Drs. Wilko Quak

Company Supervisor:
Ir. Martin Kodde

29th June of 2015
Academic Year 2014-2015
Version 4

Graduation Plan

Confirm the Geomatics graduation template this section briefly summarizes the whole graduation plan. For further explanation one is referred to the rest of this document.

PERSONAL INFORMATION

Name	Myron Ramkisoen
Student number	4078691
Address	Schelpenhof 38
Postal code	2492 ST
Place of residence	The Hague
Telephone number	+31 6 15327465
Email address *Personal* *Internship Company* *TU Delft, student e-mail*	 myron_raj@hotmail.com M.Ramkisoen@fugro.nl M.G.W.Ramkisoen@student.tudelft.nl

GRADUATION RESEARCH

Title of the graduation project	**Solid CAD geometries of the Petrochemical Industry in a Spatial DBMS** *Former title: Solid geometries in a Spatial DBMS*
Mentors	
Graduation Professor (main mentor)	Peter van Oosterom
Daily Supervisor (second mentor)	Wilko Quak
Company Supervisor	Martin Kodde (Fugro)
Research	
Project Location	Fugro GeoServices Leidschendam, The Netherlands.
Problem Statement	There is a need for performing 3D spatial queries with the 3D models of the petrochemical industry, which is stored as a 3D DGN-file (CAD). For this it is necessary to store the objects from the data as a 3D object in a spatial DBMS. For this a conversion is needed in order to save the data in a spatial DBMS (CAD to GIS conversion).
Research Questions and Scope	The emphasis for this graduation project will be on the storage of 3D solids of the petrochemical industry in a

	spatial DBMS and not on the visualization. The visualization will be done basically for validation purposes. The main goal of this research is to have solid geometries of the petrochemical industry stored in a spatial DBMS, with query possibilities. This research project will not focus on b-reps as a solid geometry representation. **Main research question:** How is it possible to store and visualize solid geometries in a spatial DBMS suitable for the petrochemical industry? **Sub research question:** *1.* What is the relationship between CAD and GIS? 2. What are the differences between CAD and GIS ? 3. What is a solid geometry? 4. What are the functional and technical requirements for Fugro's 3D application? *5.* What data does the data set of the petrochemical industry consist of? *6.* Which potential methods exist for the storage of solid 3D geometry in DBMS(s)? *e.g. point clouds in Oracle* *7.* What are the advantages and disadvantages of the potential methods for storing the solid geometry ? 8. Which method is the most efficient storage structure for storing the 3D solid geometries of the petrochemical industry in a spatial DBMS? 9. How is it possible to store the data and incorporate the constrains of Fugro's 3D application? *(DBMS design & processing)* 10. How is the performance of the 3D DBMS for solid geometries?

	11. How does the 3D DBMS fulfill the requirements of Fugro GeoServices? *12.* How can the queries solids be visualized in 3D?

PROCESS

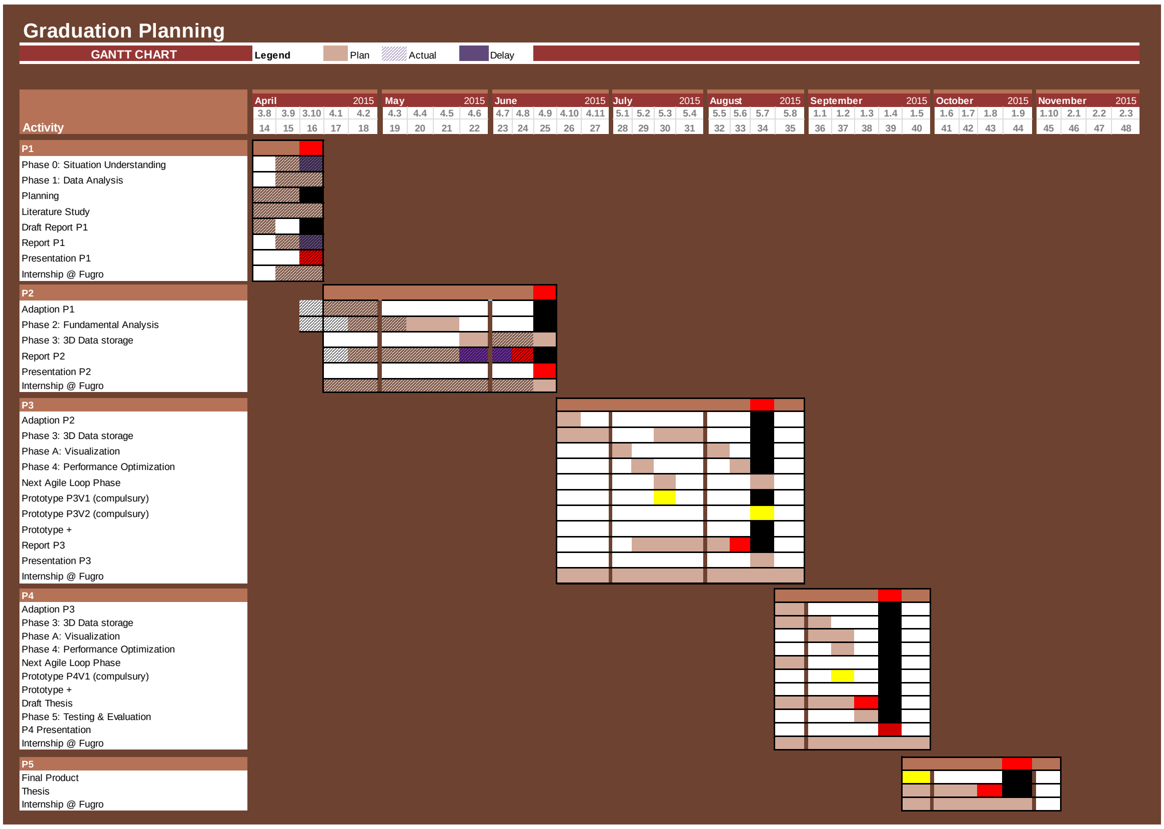
Method Description
First a literature study is required in order to investigate the theory and existing researches, related to this graduation project. After the literature study the data of Fugro GeoServices can be observed and the potency of the DBMSs and methods can be explored for storing solid geometry in a spatial DBMS. This will result into 3 methods, which will be chosen for development. One of these three methods will be chosen for implementation, reflection and evaluation. Another method may be developed, dependent on the strengths and weaknesses of the developed method.
Literature • CAD • GIS • UML diagrams • Spatial DBMSs • 9-IM • Parametric Design • CSG • 3D GIS objects • BIM • IFC • (Spatial) DBMS • Multi Criteria Analysis • Benchmark methodology • CAD & DBMS vendors • Data exchange formats

REFLECTION

Relevance

Within the scientific framework this graduation project will benefit in uniting the CAD world with the GIS world. This will be beneficial, because this graduation research will investigate the current possibilities (including methodologies) for uniting both worlds in a 3D environment.

Time planning



Appendix A Graduation Plan (P2)

Table of Contents

1. Introduction

This graduation plan describes the graduation plan of the graduation thesis (GEO2000) of the Master Geomatics of the Delft University of Technology (TU Delft). This graduation project will be done in cooperation with Fugro GeoServices B.V. in Leidschendam and will be combined with a graduation internship. Fugro GeoServices is the provider of the topic of this graduation project.

Fugro GeoServices operates in The Netherlands in providing services in the field of geoscience, Geo Information and geo-consultancy. Fugro Geoservices is performing their activities worldwide, focusing on the petrochemical industry, construction, mining and governmental issues. The topic for this graduation project is related to the petrochemical industry.

This graduation plan starts with an introduction (chapter 1), followed by the proposed research (chapter 2) and the related work in chapter 3. In chapter 4 the methodology of this graduation project will follow, with the analysis in chapter 5. This graduation plan ends with the graduation plan in chapter 6. This final graduation plan is planned for approximately 6 months.

1.1 Problem Statement

Fugro GeoServices captures installations of the petrochemical industry with point clouds. These point clouds are manually processed to a CSG (Constructive Solid Geometry) in a DGN-file (CAD). This DGN file is processed with a program to a Pipe-JSON. This Pipe-JSON is a JSON (JavaScript Object Notation) structure developed by Fugro GeoServices for saving the attributes of the saved geometry from the DGN-file into a spatial DBMS (Database Management System). As an addition the geometry of the DGN-file is saved as a 2D vector in the spatial DBMS . JSON is an alternative on XML for data exchange between server and external applications in human-readable text with help of attribute-value pairs.

The currently developed Pipe-JSON format is not convenient for advanced queries when having a nested and a single Pipe-JSON combined in one single row. This emphasizes that there is a need for a better DBMS structure. Besides a new DBMS structure there is a demand for performing spatial queries in 3D. For this reason Fugro GeoServices would like to have its CAD-files stored in a 3D DBMS with 3D query possibilities. If the DBMS storage of solids succeeds this DBMS will be the main DBMS, used for future applications.

CAD and GIS both deal with geometry of the same real world objects, but differ in many aspects, such as ontology, size, storage, analysis, semantics, attributes, world projection, etc. For this reason CAD to GIS conversions are done by hand and cannot be resolved automatically. Both worlds take other aspects into account, because CAD and GIS are used in another phase of the life cycle of e.g. a building. According to Van Oosterom, Stoter, and Jansen (2005) there has been much attention for integrating CAD with GIS in the past, but these researches seldom ended how this could be done, nor specifying the fundamental problems for the integration. CAD software is providing many primitives in its semantics, which are not supported in GIS.

The tendency for uniting the worlds of CAD and GIS is a research which exist for a long time. As can be seen in figure 1.1 this tendency started with comparison researches of CAD and GIS, ending with IFC and CityGML conversions. However the IFC standard is not a real CAD standard, it can be concluded that IFC and CityGML conversions succeeded in uniting both worlds. This research project will go one step further in uniting CAD and GIS: to investigate the storing possibilities of CAD in a spatial DBMS (GIS).

1.2 Relevance for the field of Geomatics

Geomatics is the science of geographical information. The Master Geomatics at the TU Delft is focusing on the vital spatial knowledge of the built environment and incorporates many subjects, including Geo DBMS Management Systems and 3D GeoVisualization. This graduation project will be focusing on spatial DBMS management and GeoVisualization within the domains of the master Geomatics.

Within the scientific framework this graduation project will benefit in uniting worlds of CAD and GIS. This will be beneficial, because this graduation research will investigate the current possibilities (including methodologies) for uniting both worlds in a 3D environment. Uniting both worlds will be beneficial, because this will result into more interoperability.

1.3 Author’s Interest

The author’s interest is to gain more knowledge and experience in spatial DBMSs. With help of this graduation project it will be possible to investigate which existing methodologies are the most efficient methodologies for storing solid 3D data. Besides gaining knowledge in spatial DBMSs this graduation project also gives the author the possibility to improve his knowledge in 3D web visualization. The Master Geomatics has only provided limited knowledge regarding web visualization (WebGL, KML), but this graduation project will help in gaining more knowledge and experience. Hopefully this graduation project will result into an usable DBMS for Fugro GeoServices.

1985 **Van Oosterom (1985).** MSc. Thesis in uniting CAD and GIS.

1988 **Cowen (1988).** GIS versus CAD versus DBMS, examining the differences.

1990

Newell & Sancha (1990). Comparison between CAD and GIS

1995 **Movafagh (1995).** GIS/CAD convergence for mapping application enhancement.

1997 **Teklemburg et. al. (1997).** Proposal of CAD and GIS connecting by dynamic interfaces (engines).

2000

2003 **Arens (2003).** MSc. Thesis in modelling 3D primitives in a Geo DBMS (3D city model) .

2004 **Kolbe & Plumer (2004).** Bridging the Gap between GIS and CAAD Geometry, Referencing, Representations, Standards and Semantic Modelling

2005 **Van Oosterom et. al. (2005).** DBMS supports spatial datatypes, not usable for CAD
Pu (2005). MSc. Thesis in saving freeform curves and surfaces in Oracle

2006 **Peachavanish et. al. (2006).** Ontological engineering approach for CAD-GIS integration.

2008 **Dollner & Hagedorn (2008).** CAD-GIS integration possibility by conversion of IFC – CityGML.

2010

2010 **Brugman (2010).** MSc. Thesis in 3D topological structures with validation and conversion functionality in Oracle

2011 **Xu (2011).** MSc. Thesis in Constraints in a 3D spatial DBMS
Hijazi (2011). MSc. Thesis in the use of BIM for geo-analysis purposes
De Laat & Van Berlo (2011). GIS – BIM integration with IFC – CityGML conversion

2012 **Iniguez (2012).** Msc. Thesis in storing 2D objects in a DBMS for 3D object visualization.

2013 **Donkers (2013).** MSc. Thesis in converting IFC to CityGML

Figure 1.1: Sample of the past researches for uniting both worlds of CAD and GIS

2. Proposed Research

Spatial DBMSs allow the storage of geometry data, usually 2D or 2.5D, in a DBMS. This becomes complicated when 3D data has to be stored as a 3D geometry, because there are various methods to represent 3D objects. The difficulty even increases when the data is a CAD file. CAD and GIS have been developed separately, which has resulted into a big difference in e.g. data types and semantics.

Fugro GeoServices is developing a system to store 3D model data of the petrochemical industry, which is generated with a laser scanner out of a surveyed point cloud (figure 2.1). The point cloud is processed by manually fitting 3D solids in the point cloud. This manual processing is visualized in figures 2.2 and 2.3. After combining separate point clouds to one point cloud of the terrain with help of GCP's (Ground Control Points), the terrain can be modelled in 3D with help of solids. This modelling is done with a piping software, that saves the 3D model as a COE-file, what can be exported to a DGN-file (or similar CAD format). Processing the point cloud to a solid is done manually by separating the points, which define a geometry. These separate points can be processed automatically to a solid by the software, or drawn by hand by inserting a solid or extruding a surface to a solid. Every 3D model is inserted back into the object space, until a 3D model is created with the point cloud as reference.

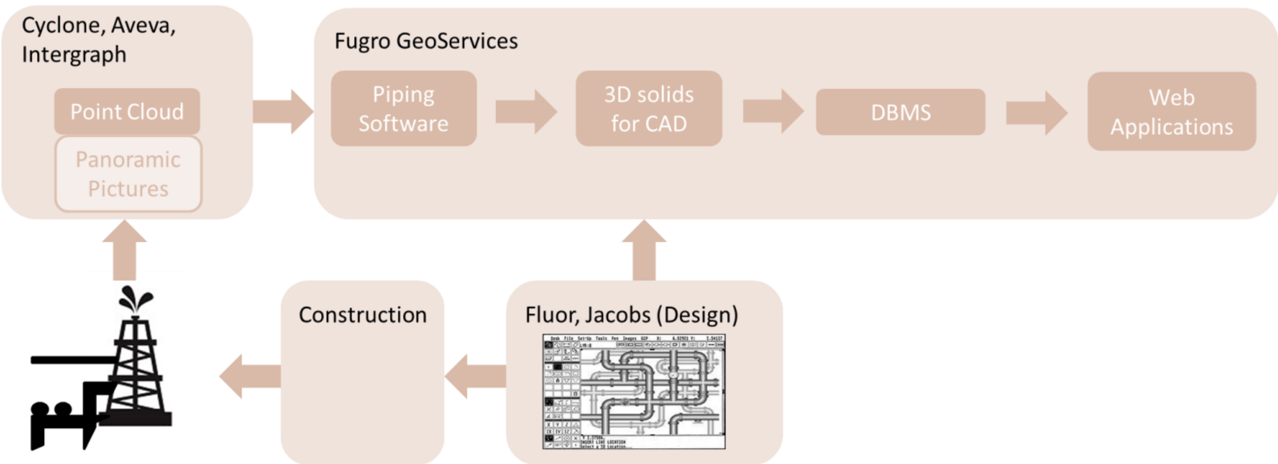


Figure 2.1: Workflow, from point cloud of the petrochemical industry to web application

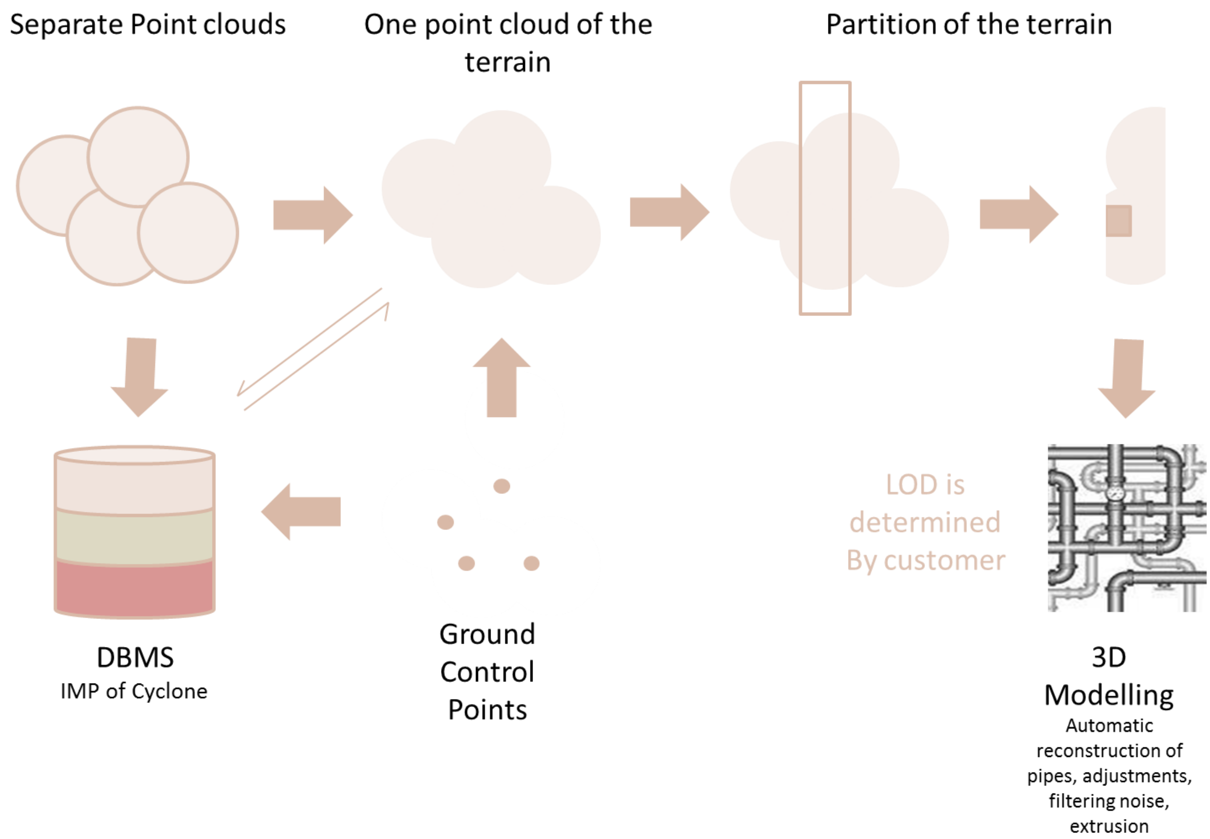


Figure 2.2: Point cloud combining process, from processing separate point clouds to 3D modelling

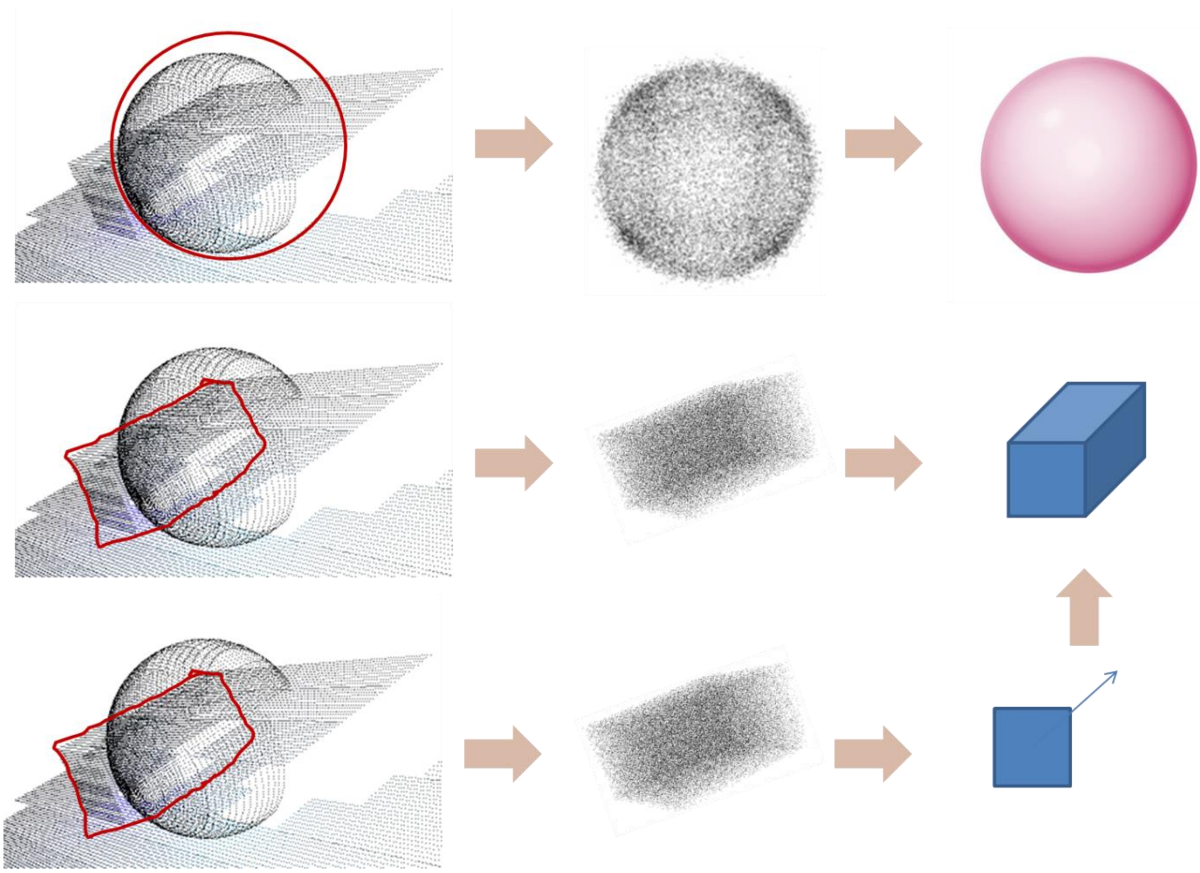


Figure 2.3: Point cloud processing, from point cloud to solids (automatic solid generation, and extruded solid generation)

The accuracy of these 3D models are predefined by Fugro's customers. The customer desires to have a 3D model with a certain LOD (Level of Detail). This LOD defines which pipes (e.g. all pipes with a radius of 8inch or higher) and which attributes (e.g. electricals, architectural elements) are modeled or not. Figure 2.4 shows the different LOD's, which exist in modelling the petrochemical industry. However this is not a universal accepted LOD for modeling the petrochemical industry. Fugro GeoServices does not have a predefined LOD's for 3D modeling.

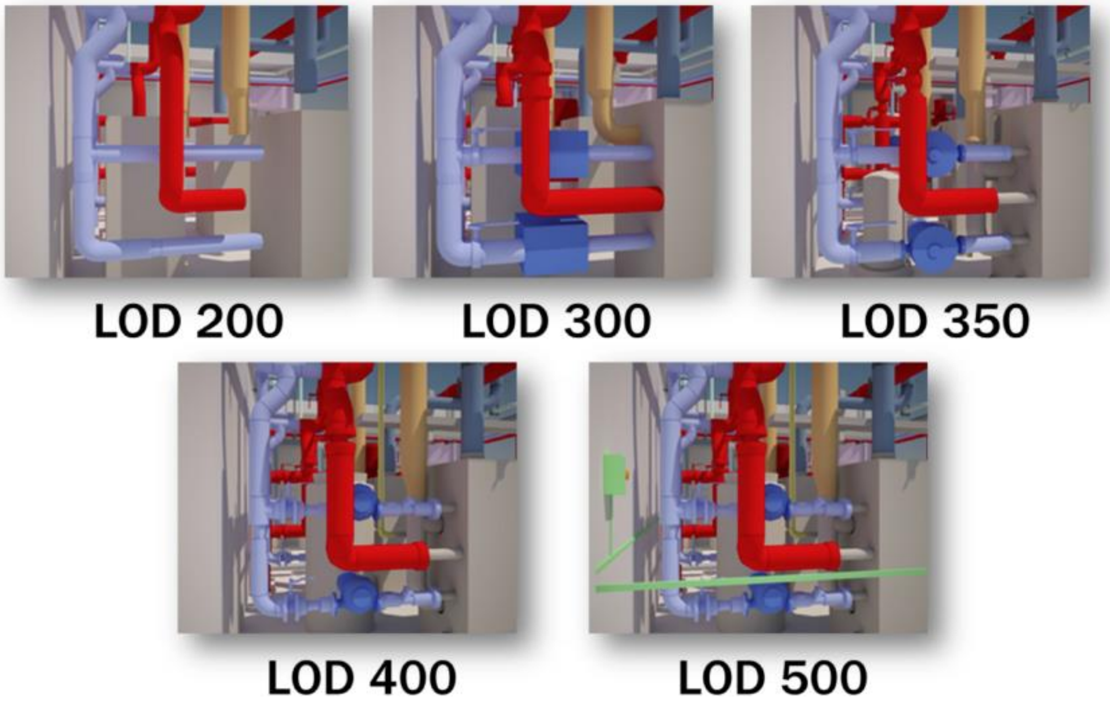


Figure 2.4: Different LOD's in modelling the petrochemical industry (Leonova, 2014). Note: This LOD's only illustrates the existence of different LOD's in the petrochemical industry, These LOD's are not used at Fugro GeoServices.

Newly built models, which will be integrated into the petrochemical industry are also processed by the piping software for DGN generation. The solid model is saved in a 2D spatial DBMS by Fugro GeoServices, usable for web applications. The DGN models is georeferenced in Amersfoort RDNAP CRS (Coordinate Reference System).

Within this graduation project the storage of CAD solids as a true spatial type in a spatial DBMS will be investigated. The goal is to allow spatial queries on the model data. A part of the work is to demonstrate the visualization of these solids in 3D. The following components will be part of this graduation project:

1. Literature study
2. Investigate existing methods and attempts to store solids

3. Investigate functional and technical requirements for Fugro's 3D application
4. Select new storage structures
5. Analyze performance
6. Build demonstrator or visualizer
7. Assessment of solution

2.1 Objectives

2.1.1 Main Objective

The main objective of this graduation project is to save a CSG as a solid geometry in a prototype DBMS. This prototype DBMS will include a basic visualizer for the validation of 3D spatial queries. After this graduation project Fugro GeoServices will be able to save all its CSG of the petrochemical industry in a spatial DBMS with the ability to perform 3D spatial queries. This graduation project will investigate which storing method for 3D solid data of the petrochemical industry in a DBMS is the most efficient storing method for Fugro GeoServices. This graduation project will not be focusing on storing the point cloud data of Fugro.

2.1.2 Project Priorities

Within the time constrains it is important to set the priorities of this project to limit the risk of fail and to guarantee the companies satisfaction. The main objective gives a global description what is wanted, but this can be sharpened by setting priorities. The MoSCoW Rules is a methodology for setting priorities for the project's objectives. The objectives of this project are set by the author with consult of Fugro GeoServices. The MoSCoW Rules divides the objectives of the graduation project into:

- Must haves (compulsory)
    - Functional & Technical requirements of Fugro's 3D application
    - Prototype DBMS
        - Including the storage of Fugro's sample 3D data
    - Example (test) queries in 3D
    - Analysis of current (existing) methods for storing 3D data in a spatial DBMS
    - Analysis of which DBMSs (including extensions, plug ins, modules) supports 3D data
    - Basic visualizer (in e.g. WebGL) for 3D data
    - Evaluation of the prototype DBMS

- Should haves (high priority)
    - Performance optimization of the (prototype) DBMS
    - Store the data (CSG) as efficient as possible
    - The data (CSG) must contain real volume
    - Semantic analysis (CAD versus GIS)
- Could haves (desirable)
    - More advanced visualizer (in WebGL)
    - Store the data (CSG) native
    - Topology storage of the 3D objects
    - Store one whole data set of Fugro in a spatial DBMS
    - Visualizer in both 2D and 3D
- Won't haves (future continuation)
    - Save all 3D Data of Fugro (CSG) in the spatial DBMS

2.2 Research Questions

2.2.1 Main Research Question

How is it possible to store and visualize solid geometries in a spatial DBMS suitable for the petrochemical industry?

The emphasis for this graduation project will be on the storage of 3D solids for the petrochemical industry in a DBMS and not on the visualization. The visualization will be done basically for validation purposes. The main goal of this research is having solid geometries stored in a spatial DBMS, with query possibilities. This research project will not be focusing on b-reps as a solid geometry in the spatial DBMS.

2.2.2 Sub Research Questions

The main research question is divided into the following phases and research questions:

Phase 0: Situation Understanding

1. What is the relationship between CAD and GIS?
2. What are the differences between CAD and GIS?
3. What is a solid geometry?

Phase 1: Data Inventory

4. What are the functional and technical requirements for Fugro's 3D application?

5. What data does the data set of the petrochemical industry consist of?

Phase 2: Potential Exploring

6. Which potential methods exist for the storage of solid 3D geometry in DBMS(s)?
   e.g. point clouds in Oracle

Phase 3: 3D data storage

7. What are the advantages and disadvantages of the potential methods for storing the solid geometry ?
   For this question the 3 most promising methods, resulting from phase 2 will be developed (basically) and analyzed. The advantaged and disadvantages will be highlighted with help of a SWOT analysis.
8. Which method is the most efficient method for storing the 3D solid geometries of the petrochemical industry in a spatial DBMS?
   For this question the best method(s) will be chosen for development from question 7.
9. How is it possible to store the data and incorporate the constrains of Fugro's 3D application?
   (DBMS design & processing)

Phase 4: Performance optimization

10. How is the performance of the 3D DBMS for solid geometries?
   e.g. indexing, validation of calculated attributes by the DBMS such as volume
   Within this sub research question the performance of queries will be improved and validated.

Phase 5: Testing & evaluation

11. How does the 3D DBMS fulfill the requirements of Fugro GeoServices?
   Within this sub research question the performance of the DBMS will be evaluated by testing the DBMS. This sub research question must also incorporate the usability of the 3D DBMS for Fugro GeoServices.

Phase A: Visualization

12. How can the queries solids be visualized in 3D?
   e.g. WebGL

3. Related Work

Many researches (Hassan, Ahmad-Nasruddin, Yaakop, & Abdul-Rahman, 2008; Lee & Koh, 2007; Stoter & Salzmann, 2003) have been done for translating 2D data, which represents 3D objects in the real world, to 3D data. This 2D to 3D translation becomes difficult when data of two different environments must be integrated. CAD and GIS are two technologies which are used in civil infrastructural projects. However both technologies are used in different phases (Hijazi, 2011), there is no unity in standards or interoperability. Till now unity between CAD and GIS has only been achieved with help of BIM (IFC) to GIS (CityGML) conversion (figure 1.1).

In order to understand the integration problem of the worlds of CAD and GIS it is important to dive into both concepts, related to 3D modeling in sections 3.1 and 3.2. Both concepts, CAD and GIS, will be compared in order to see the differences in section 3.3. In section 3.4 the BIM concept will be introduced. Section 3.5 will show some theoretical background regarding DBMSs.

3.1 CAD: Computer-Aided Design

CAD, computer-aided design, is a technology which is used to create, modify, analyze and optimize a design with help of computer systems (Groover & Zimmers, 1983). CAD is based upon parametrical design, boundary representation (Bhanu & Ho, 1987; Monedero, 2000) or voxel representation (Kazar, Kothuri, van Oosterom, & Ravada, 2008) and operates in 2D or 3D in an orthogonal projection of the world (Van Oosterom et al., 2005).

3.1.1 Parametric Design

Parametric design is a form of 3D modeling by defining a form with help of parameters and relations (Monedero, 2000). Parametric design is done with help op parametric models, where some attributes are fixed and some attributes can vary. These variables are the parameters which can be adjusted by the user. The fixed attributes are the constrains within parametric design. Parametric design allow changes without erasing or redrawing (Hernandez, 2006).

According to Hernandez (2006) parametric design can be divided into three modeling types:

1. **Parametric variations models.** Parametric variation models have been the starting point of parametric design. This parametric model is based on the declarative nature for constructing shapes and allows user modelling according to the desired behavior. This kind of modelling

results into a parametrized modeling schema showing the values of the parameters. Changing each value in this parametric schema results into a transformation of the geometry without erasing or redrawing. Changes in the topology[1] are not allowed.

This kind of modelling include NURBS modeling. NURBS, Non Uniform Rational B-Splines, are surfaces based on parametrical curves and is used in computer graphics and CAD for various purposes, from automobile bodies to animated characters (Rogers, 2001). An example of this kind of modelling is Grasshopper-modelling in combination with Rhinoceros 3D (figure 3.1). Rhinoceros 3D is a CAD vendor, based on NURBS.

Figure 3.1: Grasshopper modelling: A parametric variation model (Grasshopper, 2010)

2. **Parametric combination models.** Parametric combination models are the most used type of parametric modeling. This kind of modeling is limited to a number of primitive geometrical components, but offers complexity by combining these primitives and the spatial relations between the geometrical components. This allows the creation of new geometrical shapes.

   An example of a parametric combination model is CAD modeling with CSGs, Constructive Solid Geometries. In CSG complicated solids are represented by adding and subtracting primitive solids by Boolean set operators (union, difference and intersection)(Bhanu & Ho, 1987). The data structure of CSGs is shown in figure 3.2 (Laidlaw, Trumbore, & Hughes, 1986). An example of a primitive is a prismatic volume, having four parameters: location, length, width and height.

[1] Topology: the number of components and their relations (Hernandez, 2006).

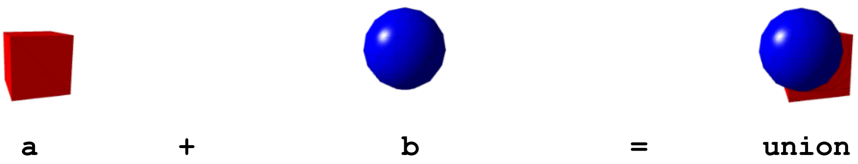
The code in figure 3.2 is an abstract presentation of the CSG data structure. Figure 3.3 illustrates how CSGs are created using JavaScript in WebGL.

```
Object Structure
      array of vertices
      array of polygons
      object extent (minimum and maximum x,y,z)

Vertex Structure
      spatial location (x,y,z)
      array of pointers to adjacent vertices
      status (inside, outside, boundary, unknown)

Polygon Structure
      array of pointers to vertices
      polygon extent (minimum and maximum x,y,z)
      polygon plane equation (x, y, z, d)
```

Figure 3.2: CSG data structures (Laidlaw et al., 1986)




```
var a = CSG.cube({ center: [-0.25, -0.25, -0.25] });
var b = CSG.sphere({ radius: 1.3, center: [0.25, 0.25, 0.25]
});
a.union(b)
```

Figure 3.3: CSG source code in WebGL with visualization as a parametric combination model (Walles, 2011)

3. **Parametric hybrid models.** Parametric hybrid models are less used than the other two types of parametric modelling. Parametric hybrid models is a combination of both parametric variation- and parametric combination models.

   An example of a parametric hybrid model is Autodesk Revit. However Autodesk Revit is a BIM, Building Information Model, it is basically a CAD software which allows parametric hybrid modelling. With Autodesk Revit it is possible to reconstruct buildings with help of basic elements, such as walls, roofs, and windows. By adding all basic elements with their topology a new geometry is created: a house. Each separate basic element in Autodesk Revit is adjustable. For instance a window can be adapted parametrically by changing certain parameters (e.g. height, width), or even the form. The topology of the window as a separate element cannot be changed.

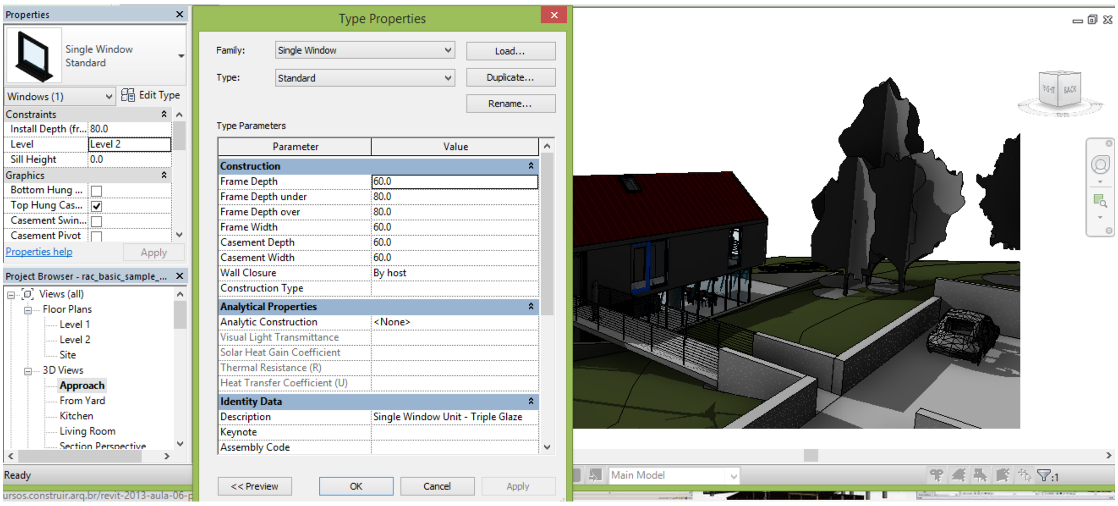


Figure 3.4: Revit parameters of a window. Revit as a parametric hybrid model (image has been made with Revit's Sample Project)

3.1.2 Boundary Representation

Boundary representation (figure 3.5) is a skin representation for solid models and form the boundary between model and non-model. The skin consist out of surfaces or faces, bounded by a sets of edges. These edges are portions of curves, delimited by vertices. At these vertices several faces meet. The data structure can be divided into topology (structure definition of the object) and geometry (form or shape of the object)(figure 3.6)(Stroud, 2006).

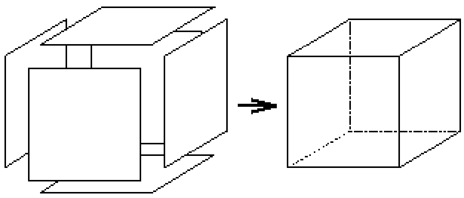


Figure 3.5: Boundary representation (Computer Aided Detector Design, n.d.)

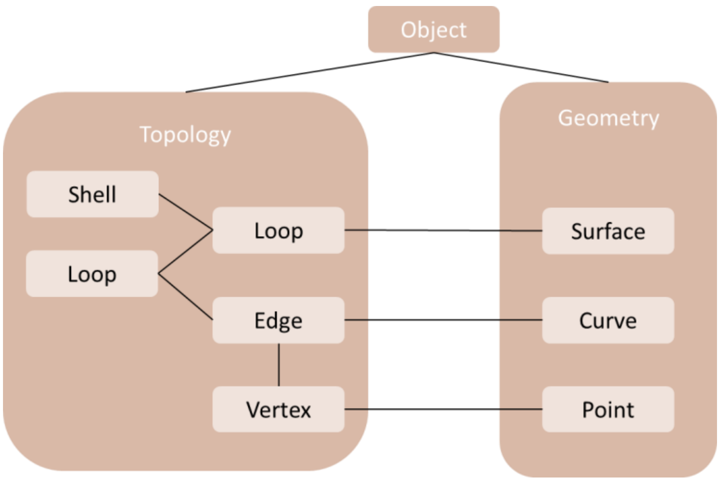


Figure 3.6: Basic data structure boundary representation (Stroud, 2006)

3.1.3 Voxel Representation

Voxels, also known as volumetric graphics or volumetric imaging, are repeated 3D volume elements in order to model a 3D object in a 3D raster representation. Storing 3D objects as a voxel will result into a rough surface and much storage space (Kaufman, 1989; Wesselingh, 2007). As can be seen in figure 3.7 the size of the voxel determines the quality of the 3D object: The smaller the voxel, the more accurate the quality of the 3D object will be.

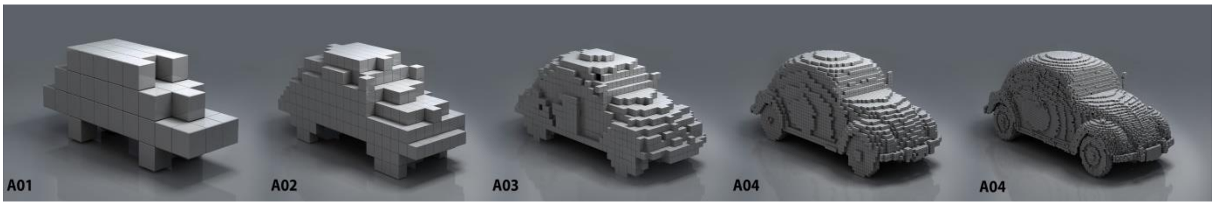


Figure 3.7: Different voxel resolutions, from low quality to high quality (Avin, n.d.)

3.1.4 CAD Vendors

There are different vendors which can be used for CAD design, such as Autodesk AutoCAD and Bentley MicroStation. These CAD vendors are one of the biggest CAD vendors and used by Fugro GeoServices. For this reason a brief introduction will be given of both vendors.

Autodesk AutoCAD

Autodesk AutoCAD is a 2D and 3D technical drawing program for CAD modeling and is known for its open architecture. The user has the flexibility for customizing the program with help of source code files in ASCII format (plain text) and programming languages such as AutoLISP (Autodesk programming language), VBA (Visual Basics for Applications), VB.NET (Visual Basic), C#, C++ and JavaScript. Autodesk AutoCAD allows the following file extensions: DWG, DWF[2], DWFx (DWF based on XML), DGN (Bentley MicroStation format) and PDF (for plotting and drawing). DWF and PDF are used as reference documents in order to protect the original owner's investments, information and integrity in design. DWF and PDF are not that accurate as DWG or DGN (Fane, 2013; Finkelstein, 2014). Autodesk AutoCAD models 3D solids with CSG. These 3D solids can be converted to a mesh object (boundary solid) (Fane, 2013). Another modelling technique what is supported by Autodesk AutoCAD are NURBS surfaces.

Bentley MicroStation

Bentley MicroStation is a 2D and 3D information modeling CAD software, used for e.g. architecture, engineering, construction and operation of utility systems. Bentley MicroStation has several programming environments, including MDL (MicroStation Development Language), JMDL (Java version of MDL) and VBA (Pu, 2005). By default Bentley MicroStation uses DGN, which allows

[2] DWF: Design Web Format. Accurate and compressed vector image representation of drawings.

modeling of 2D and 3D models. 3D modeling in a DGN file can be done via a primitive solid (CSG) or a SmartSolid (boundary representation solid). It is possible to export this DGN file to a DWG file, but then all primitive solids are automatically transformed to a SmartSolid. In contrary to Autodesk AutoCAD, Bentley MicroStation only can save its geometry as a boundary solid as a DWG.

3.2 GIS: Geographical Information Systems

GIS, geographical information systems, have initially been developed for the storage, retrieval and display of geographical information (Fotheringham & Rogerson, 2013) and consist of geographical data, a software package for data processing and a computer system. GIS is aiming for analysis by answering spatial oriented questions (Van Lanen, Bregt, Randen, & Hoosbeek, 1989). One of the main representations of geo-data is vector data. The main types of vector data are points, lines and polygons. Vector data can be stored in both 2D and 3D (Kersting & Döllner, 2002).

In the world of GIS the OGC, The Open GeoSpatial Consortium, desires for standard specifications in GIS in order to support interoperability (Khuan, Abdul-Rahman, & Zlatanova, 2008). The OGC is an international industrial consortium that aims for the participation of companies and universities to develop publicly available standards that will support interoperable solutions.

The OGC WKT, Well Known Text, is a markup language for representing vector geometry data. WKB, Well Known Binary, saves the same information in binary bites, suitable for storing and transferring the data. Table 3.1 shows the difference of the data structure of one identical point (Wang & Wang, 2010). These coding rules are regulated by the OGC in the Simple Feature Specification (OGC, 1999).

Table 3.1: WKT and WKB example (Wang & Wang, 2010)

OGC Service Specification	Formats
WKT	POINT(1 1)
WKB	0101000000000000000000F03F000000000000F03F

Figure 3.8 shows some examples of the data structure of WKT (point, line, polygon (with 1 exterior ring and 0 interior rings) and geometry collection (consisting of 2 points and 1 line string)) (OGC, 1999). Note that figure 3.8 shows a 2D WKT data structure. Figure 3.9 shows an example of a 3D WKT data structure.

```
POINT (10 10)
LINESTRING (10 10, 20 20, 30 40)
POLYGON ((10 10, 10 20, 20 20, 20 15, 10 10))
GEOMCOLLECTION (POINT (10 10), POINT (30 30), LINESTRING (15
15, 20 20))
```

Figure 3.8: WKT data structures (OGC, 1999)

```
LINESTRING (10 10 1, 20 20 1, 30 40 5)
```

Figure 3.9: WKT 3D data structures

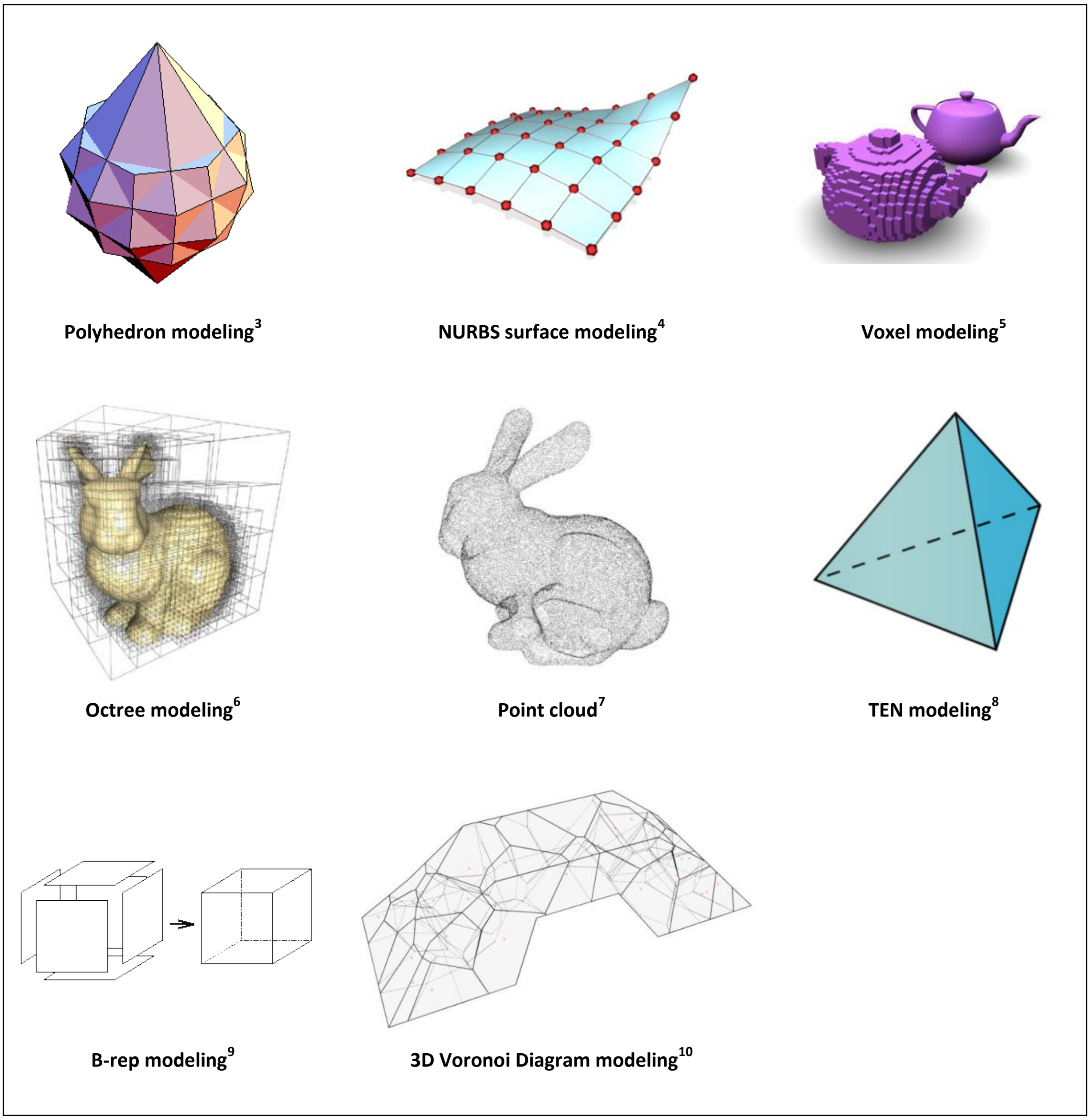


Figure 3.10: 3D GIS modeling methods

[3] Hart (1997)
[4] 3D-max (n.d.)
[5] Mossman (2013)
[6] Pharr and Fernando (2005)
[7] Mederos, Velho, and De Figueiredo (n.d.)
[8] Ctech (n.d.)
[9] Computer Aided Detector Design (n.d.)
[10] Object-e (2009)

GIS operates in a spherical projection of the world. This spherical projection uses the ellipsoid. For large scales GIS uses the orthogonal projection of the world (local coordinate system) (Van Oosterom et al., 2005). Within the world of GIS there are various methods to model 3D data, including polyhedra, NURBS, voxels, octrees, point clouds, Ten, B-reps and a 3D Voronoi Diagram. Figure 3.10 portrays each of these methods.

3.2.1 Polyhedron

A well-known elementary geometry in GIS is the polyhedron as shown in figure 3.10. In this graduation project the polyhedron definition of Cromwell (1997) will be used:

"A polyhedron is the union of a finite set of polygons such that

i. Any pair of polygons meet only at their sides or corners.

ii. Each side of each polygon meets exactly one other polygon along an edge.

iii. It is possible to travel from the interior of any polygon to the interior of any other.

iv. Let V be any vertex and let F_1, F_2, .., F_n the the n polygons which meet at V. It is possible to travel over the polygons F_i from one to any other without passing through V."

Polyhedra consist out of polygons. The relation between the number of faces, edges and vertices of a polyhedron is captured in Euler's famous formula (Eq. 3.1) (Worboys, 1997):

$$f - e + v = 2 \qquad (3.1)$$

where:

f = number of faces

e = number of edges

v = number of vertices.

A polyhedron defines the boundary of a 3D object. A 3D object can also be a single voxel, a single octree component, a single TEN, a B-rep or a single 3D Voronoi component. This 3D object can also be defined with help of NURBS modelling if the NURBS surface is defined as a polygon.

3.2.2 NURBS

NURBS, Non-Uniform Rational B-Spline, are a mathematical representation used for free-form modeling of surfaces. NURBS allow more freeform by parametric modeling with weight, using control points (Pu, 2005; Rogers, 2001). NURBS allow analytic and freeform shape representation and are based on a NURBS curve. The polynomial function of a NURBS curve is shown in Eq. 3.2 (Piegl, 1991; Pu, 2005).

$$C(u) = \frac{\sum_{i=0}^{n} w_i P_i N_{i,k}(u)}{\sum_{i=0}^{n} w_i N_{i,k}(u)} \quad (3.2)$$

where:

w_i = weight

P_i = control points (vector)

$N_{i,k}$ = normalized B-spline basis function of degree k

The normalized B-spline function of degree *k* is shown in Eq. 3.3 (Piegl, 1991).

$$N_{i,0}(u) = \begin{cases} 1 & if\ u_i \leq u < u_i + 1 \\ 0 & \text{otherwise} \end{cases}$$

$$N_{i,k}(u) = \frac{u - u_i}{u_i + k + 1 - u_{i+1}} u_{i,k-1}(u) + \frac{u_{i+k+1} - u}{u_{i+k+1} - u_{i+1}} N_{i+1,p-1}(u) \quad (3.3)$$

where:

u_i= knots forming a knot vector[11]

k = degree

For the NURBS curve the knot vector takes the form of Eq. 3.4, where the end knots *α* and *β* are repeated with multiplicity *k + 1*. The degree, number of knots and number of control points are related to the formula *m = n + k + 1.* In most applications *α=0* and *β=1* (Piegl, 1991).

$$\boldsymbol{U} = \{\alpha, \alpha, \dots, \alpha, u_{k+1}, \dots, u_{m-k-1}, \beta, \beta, \dots, \beta\} \quad (3.4)$$

3.2.3 Voxels

Besides CAD, voxels are also used in GIS for 3D representation (section 3.1.3). Voxels are the 3D counterpart of the 2D pixels, which can represent a material, color, texture, translucency ratio, or other characteristics per voxel (Wesselingh, 2007).

3.2.4 Octrees

In solid modeling octrees are a sub method of the cell decomposition methods. The cell decomposition method divides the object space into unit-sized elements, cubes or spheres, to represent shapes as a collection of these elements. It is not possible to divide the object space into infinitely small elements due computer limitations. The cell decomposition method is also known as voxel modelling. A refinement of this technique is octree modeling.

[11] Knot vectors are the underlying set of breakpoints (Farin, 1982).

With octrees the bounding box of a 3D element is computed which is called the 3D cell. This 3D cell is partitioned into smaller 3D cells until the desired resolution is computed: a 3D mesh. Every partitioning divides the cell into eight children cells. With computations each children cell is classified as filled, partially filled or empty. The partially filled children cells are subdivided until the desired resolution is achieved(Peng & Kuo, 2005; Stroud, 2006).

Both octrees and voxels involve modelling with a 3D cubical element. The main difference between both is that voxels allow modeling with one single defined element and octrees allow partition of the 3D element, what results into smaller elements. This makes octrees more efficient in retrieving a higher resolution. Voxels have to adjust all elements to smaller elements, but octrees only have to adjust the element at necessary places. Another difference between both methods is that octrees always model with cubes, but voxels can model with any created 3D element.

3.2.5 Point Cloud

A point cloud consists of a big amount of points. The amount of points depends on the density of the point cloud. Before modeling a 3D object it is evident to filter out the noise, usually done with a smoothing filter. For 3D modeling this filter will remove all the sharp edges, but these sharp edges can be reconstructed with an algorithm (Mitra, Nguyen, & Guibas, 2004).

A point cloud does not represent a volume, only points. It might be possible to define a boundary volume with help of voxels, by positioning voxels at the position of a point. With help of interpolation the empty spaces between the points or voxels can be filled, so a boundary representation arises consisting of voxels.

3.2.6 TEN

According to Penninga and Van Oosterom (2008) "a TEN is a simplicial complex consisting only of face-connected 3-simplexes that model the full 3D domain." A TEN is a 3D variant of a TIN, Tetrahedral Irregular Network. A tetrahedron is a simple geometrical 3D shape: a pyramid with a triangular ground surface. This tetrahedron consists out of 4 points. The only restriction of these four points is that all the points must not lay on one plane. Multiple tetrahedra can model a 3D model, as shown in figure 3.11 (Wesselingh, 2007).

A TEN can be validated with help of the Euler-Poincaré formula (Eq.3.5) (Penninga & Van Oosterom, 2008):

$$N - E + F - V = 0 \tag{3.5}$$

where:

N = number of nodes

E = number of edges

F = number of faces

V = number of volumes.

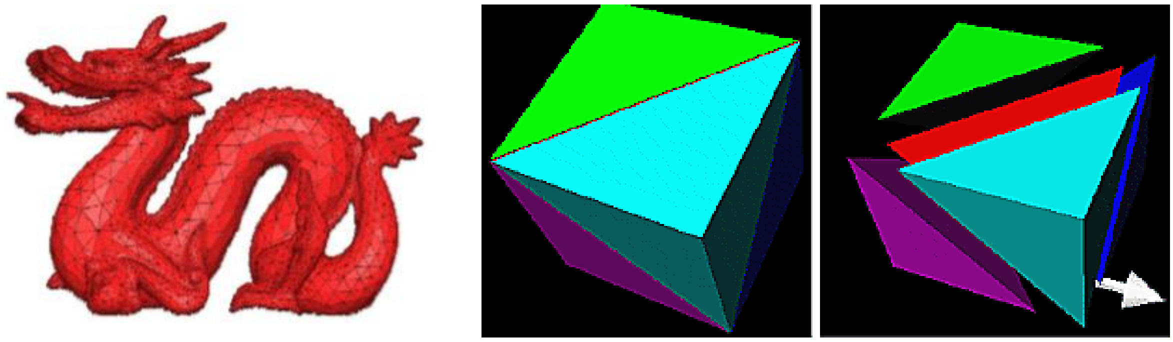


Figure 3.11: A network of tetrahedral, modeling a 3D model (Eckel, n.d.; Wesselingh, 2007)

Relating a TEN to a point cloud it is possible to reconstruct a solid geometry out of the point cloud. By triangulating the point cloud and reconstructing these triangles as such that a tetrahedron exist a solid geometry can be constructed. If it is not possible to reconstruct these triangles as a tetrahedron a B-rep can be created by reconstructing triangles with only 3 points or vertices. This triangle reconstruction with 3 points or vertices is also known as the Delaunay triangulation and can also help in reconstructing voxels with help of boundary boxes, which has been implemented by Sisi Zlatanova and Pirouz Nourian for MonetDB, based on Laine (2013).

3.2.7 B-rep

B-rep, boundary representation, is a collection of connected surfaces which represent a 3D solid model (Wesselingh, 2007). "A boundary representation […] specifies the location of the vertices, their connectivity, and a description of how they should be interpolated or approximated by a simple surface (such as a polyhedron […] or parametric patch)" (Rossignac, 2002).

Both B-rep and NURBS represent the boundary between object and non-object. The main difference between both methods is the parametrical character of the boundary when modeling a NURBS surface. B-rep is a flat surface or one dimensional curved surface (according to the 1^{st} order polynomial) between vertices, but NURBS allow parametrically curved surfaces (of higher order polynomial). This allows more flexibility to NURBS than B-reps.

3.2.8 3D Voronoi Diagram

The basic principle of both the 2D and 3D Voronoi Diagram is based on the distance function between sites (Hoff III, Keyser, Lin, Manocha, & Culver, 1999). The 3D Voronoi Diagram is based on the assumption that a point or node is seen as a centroid, which is the center of a mass with respect to a given density function, of the corresponding Voronoi region (Du, Emelianenko, & Ju, 2006). For the Voronoi region it is important to defines the dominance region. This dominance region determines the Voronoi region. Eq. 3.6 defines the dominance region of a region A_i over A_j (Hoff III et al., 1999).

$$Dom\left(A_i, A_j\right) = \left\{p \,\middle|\, dist(p, A_i) \leq dist(p, A_i)\right\} \quad (3.6)$$

where:

A = site

p = point in space

Dom = Dominance region

dist = distance (from p to A)

With Eq. 3.6 the Voronoi region can be determined of a site in Eq. 3.7 (Hoff III et al., 1999).

$$V\left(A_j\right) = \cap_{i \neq j} Dom(A_{i,}, A_j) \quad (3.7)$$

where:

V = Voronoi region

Dom = Dominance region

A = site

Relating the 3D Voronoi Diagram to a TEN both methods are using point for reconstructing a surface. A TEN uses the points as boundary nodes for triangulation and the 3D Voronoi Diagram uses the points as the center of the reconstructed geometry.

3.2.8 GIS method overview

Table 3.2 gives an overview of all GIS methods, showing which methods have a similarity with which method.

Table 3.2: GIS method similarity overview

	Polyhedron	NURBS	Voxels	Octrees	Point cloud	TEN	B-rep	3D Voronoi Diagram
Polyhedron		●	●	●	○	●	●	●
NURBS	●		○	○	○	○	●	○
Voxels	●	○		●	●	●	○	○
Octrees	●	○	●		○	○	○	○
Point cloud	○	○	○	○		●	●	○
TEN	●	○	●	○	●		○	●
B-rep	●	●	○	○	●	○		○
3D Voronoi Diagram	●	○	○	○	○	●	○	

3.3 CAD versus GIS

Both CAD and GIS are founded in the early computer graphics. CAD applications are focusing on design and often lack a robust attribute model. GIS applications are focusing spatial analysis and are relying on a DBMS attribute model (Karimi & Akinci, 2009). However CAD and GIS have been developed separately both worlds meet at some places.

3.3.1 Modeling methods

Table 3.3 compares both modeling methods of CAD and GIS in similarity.

Table 3.3: Modeling method comparison

GIS	**CAD**				
	Parametric design			Boundary representation	Voxels
	Parametric variant modeling	Parametric combination modeling	Parametric hybrid modeling		
Polyhedron	○	○	○	●	○
NURBS	●	○	○	●	○
Voxels	○	○	○	○	●
Octrees	○	○	○	○	○
Point cloud	○	○	○	○	○
TEN	○	○	○	○	○
B-rep	○	○	○	●	○
3D Voronoi Diagram	○	○	○	○	○

Table 3.3 illustrates that the CAD modeling techniques meet the GIS modeling techniques with parametric variant modeling, boundary representations and voxels. The parametric variant modeling uses NURBS as modeling technique, which is also used as a GIS modeling technique. Boundary representation equals polyhedron, NURBS and B-rep.

3.3.2 Solid Geometry

An important term within this graduation project is the term solid geometry. Both worlds, CAD and GIS, have another perception of a 3D solid. Natively a solid geometry was defined as a geometry in the three dimensional Euclidean space. In the sense of computer modeling there are mainly four representation techniques for solid modelling (Stroud, 2006):

1. **Cell decomposition.** Dividing the object space into unit-sized elements.
2. **General sweeping.** Extruding 2D shapes along general curves.
3. **Set theoretic.** This technique for solid modeling involves two kinds of solid representation:
    i. Representing solids as a combination of primitive shapes with Boolean operations.
    ii. Representing solids as object boundaries, which allow intersections, based on Boolean operations.
4. **Boundary representation.** Representing the boundary between object and non-object.

With the separate development of CAD and GIS both worlds have been using other semantic interpretations for solid modelling. The leading definition in the world of GIS for solid geometry is the definition of the OGC: a boundary surface. The OGC defines a 3D solid object as a GM_Solid, which is a subclass of GM_Primitive (Khuan et al., 2008). This notion complies with the fourth solid modelling technique: boundary representation. However GIS also uses the first solid modeling technique (octrees and voxels) and second modelling technique (NURBS). As written in section 3.1 CAD refers to a parametric modelling (CSG and NURBS) or boundary representation for solid modelling. This notion complies with the second, third and fourth solid modeling technique: CSG for the third modelling technique, boundary representation for fourth solid modelling technique and NURBS for the second modelling technique. Table 3.4 summarizes the division of all CAD and GIS methods according to the solid modelling representation techniques. Note: TEN and 3D Voronoi Diagram are using a similar solid representation technique as a cell decomposition.

Note that set theoretic and boundary representation both involve boundary representations. The difference between both is only in the modelling technique. Both CAD and GIS share general sweeping (NURBS) and boundary representation (Mesh versus B-rep or polyhedron) as a similar modelling technique.

Table 3.4: Division of solid modelling representation techniques of the world of CAD and GIS

	Solid modelling			
	Cell decomposition	General sweeping	Set theoretic	Boundary representation
GIS	●	●	○	●
CAD	○	●	●	●
GIS				
Polyhedron	○	○	○	●
NURBS	○	●	○	○
Voxels	●	○	○	○
Octrees	●	○	○	○
Point cloud	○	○	○	○
TEN	◐	○	○	○
B-rep	○	○	○	●
3D Voronoi Diagram	◐	○	○	○
CAD				
Parametric design	○	●	●	○
Boundary represent.	○	○	●	●
Voxels	●	○	○	○

For this graduation project it is evident to make a distinction between two kinds of solid modelling:

- **True solid:** A true solid is defined as a solid with volume.
- **Boundary solid:** A boundary solid is defined as a boundary surface representing the boundary between object and non-object, with a missing volume concept.

3.3.3 Semantical differences

However both worlds meet at some modeling techniques, they both differ in data and take other aspects into account. In order to get both worlds closer it is evident to have a framework which incorporate both the geometry as the corresponding theoretical characteristics: a semantical analysis (Jansen, Van Oosterom, & Stoter, 2004; Van Oosterom et al., 2005). This semantical analysis has been done by Karimi and Akinci (2009), as shown in table 3.5. Table 3.5 illustrates that both CAD and GIS are different in semantic.

Table 3.5: Semantic Comparison between GIS and CAD Platforms (Karimi & Akinci, 2009)

Platform	Class	Interfaces	Semantic interchange potential
CAD	Geometry	Vectors accessible by layer or sheet	Limited, elemental geometry
	Data	Attributes can be tagged to certain features, no attribute / RDBMS model	High, if entities are tagged at functional level of abstraction
	Metadata	Limited to project or file	Limited
GIS	Geometry	Attribute and spatial query possible	Moderate, geometry is topologically integrated, linked through abstract class relationships
	Data	Attribute and spatial query possible	High, attribute model is accessible for semantic discovery
	Metadata	Descriptive text, XML, standards-based	High, data is coded with source and parameter definitions

3.3.4 Theoretical differences

Besides the semantical comparison it is also important to have a comparison overview regarding the most important theoretical characteristics, such as ontologies, world projection and the saving structure of CAD and GIS. This comparison may help in understanding the differences between CAD and GIS. This comparison overview may provide important information for closing the gap between CAD and GIS. In table 3.6 the comparison overview of CAD and GIS is shown (excluding the semantical differences of table 3.5 and saving structure comparison of section 3.3.1).

Table 3.6: Comparison between GIS and CAD Platforms

Platform	Class	Interfaces
CAD	Ontology	Ontology operations overlap[12]
	World projection	Orthogonal projection[13]
GIS	Ontology	Ontology operations overlap[9]
	World projection	Spherical projection (ellipsoid) and orthogonal projection*[10]

* GIS is using a spherical projection, but meets the world of CAD at larger scale: using an orthogonal projection (the local coordinate system).

[12] Akinci, Karimi, Pradhan, Wu, and Fichtl (2010)
[13] Van Oosterom et al. (2005)

Another important theoretical difference between CAD and GIS is the focus. CAD focusses more on the shape of the object, than the validness (Kazar et al., 2008).

3.4 BIM: Building Information Model

CAD has had a steady evolution, which has resulted into BIM: Building Information Model. BIM includes some intelligent behavior of GIS, such as associated data and rules, non-redundant geometry (topology), automatic modification of associated geometries, and others. BIM is a 3D design tool with parametric objects that behave or interact with other objects. Within a BIM software it is possible for vendors to provide objects, which differ from the standard objects from the BIM software. BIM allows data sharing with other project members and serves as a design communication medium. BIM is collaboratively developed with all stakeholders (input) (Karimi & Akinci, 2009).

IFC, Industrial Foundation Classes , is the international standard data format to describe, exchange and share information within the building and facility management industry sector. IFC is an open standard developed for BIM by the international organization buildingSMART and is based on the EXPRESS language as part of the STEP (Standard for the Exchange of Product model data) standard[14], which enables to define IFC models using XML (Hijazi, 2011).

According to Hijazi (2011) IFC include the following key contents:

- Object-oriented and semantic model
- Re-use of building information through the whole building lifecycle.
- 3D representation of building models
- Spatially data model (spatial relationships between building elements is maintained hierarchically)

IFC is a share data model (figure 3.12), and has been accepted by the Dutch 'Forum en College Standaardisatie' as an open standard. This obligates governmental organizations to use the IFC standard when working with BIM (Donkers, 2013).

Besides creating CSG and B-rep models, the IFC standard also includes swept solids and half-space solids. Swept solids are solids created by revolution and linear extrusion of a solid, which are obtained by sweeping or extruding a planar face (with may contain holes). Half space solids are semi-infinite solids (on one side of a surface) which can be limited by a box domain (BuildingSMART, n.d.-a). IFC is parametric design, because the IFC standard for BIM is based on CAD. Looking at e.g. a

[14] ISO 103030 (Hijazi, 2011).

sphere, the IFC standard saves the following attributes for constructing this sphere: center and radius (BuildingSMART, n.d.-b).

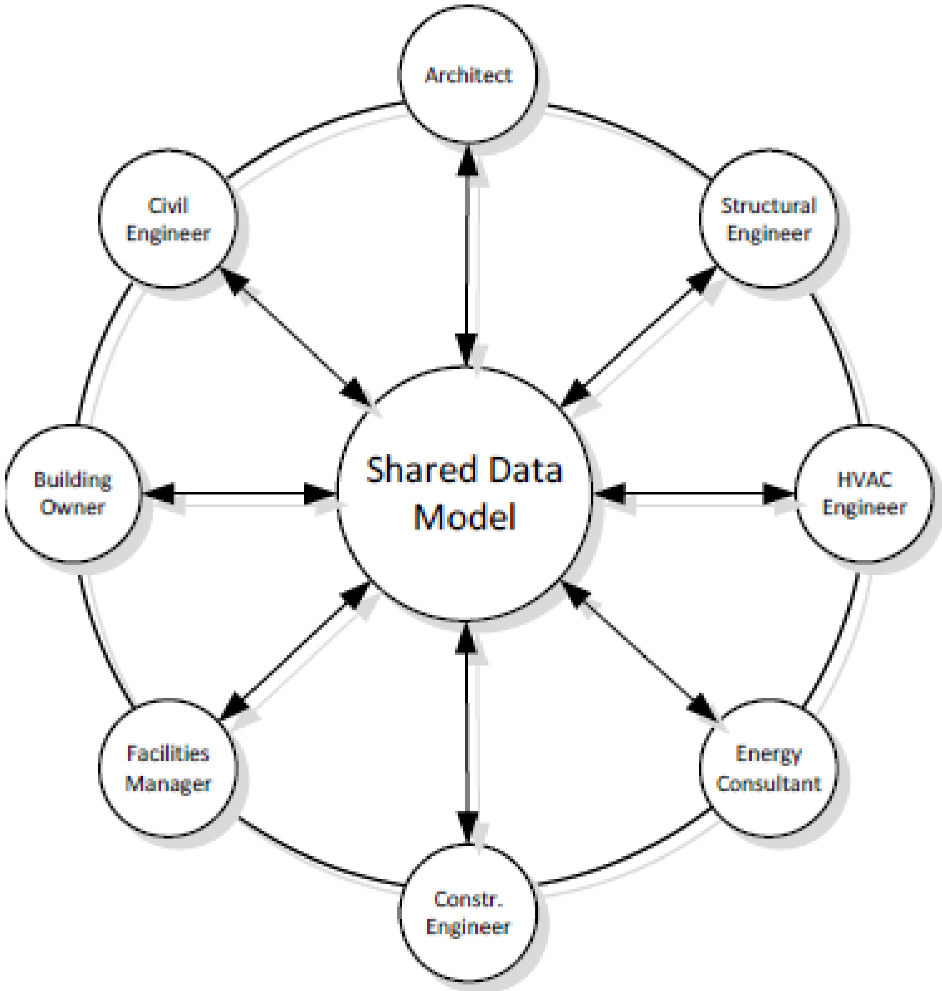


Figure 3.12: IFC as a shared data model (Donkers, 2013)

Recently much research has been done for uniting the worlds of CAD and GIS with help of the IFC standard (Benner, Geiger, & Leinemann, 2005; Donkers, 2013; I-Chen & Shang-Hsien, 2007; Nagel, Stadler, & Kolbe, 2009). However these researches were mainly focusing on the transformation of architectural elements, such as walls, spaces and doors into CityGML and GML (Hijazi, 2011).

3.5 DBMS: Database Management System

A DBMS, Database Management System, is a software which handles all access to the database (Date, 1975) and controls the storage & retrieval, addition/deletion, data definition and journaling of data, which consists out of a kernel code (for managing memory and storage), repository data (data dictionary) and a query language (enables to access the data) (Ashdown & Kyte, 2014; Chorafas, 1983). Figure 3.13 visualizes the four parts of a DBMS. In order to access and manage the data a DBMS application is needed (Ashdown & Kyte, 2014).

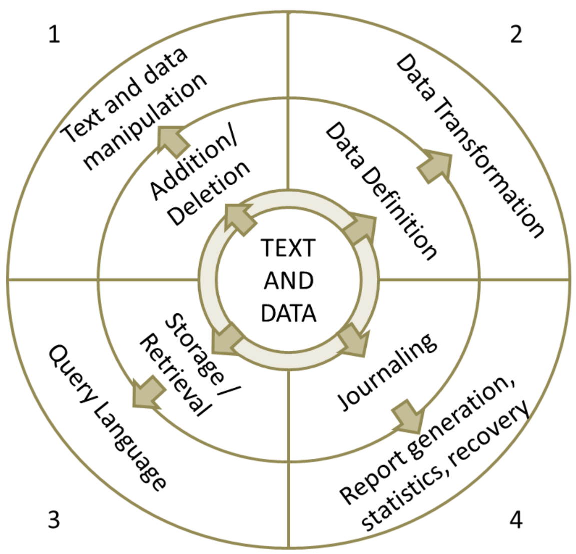


Figure 3.13: The four parts which a DBMS involves (Chorafas, 1983)

A DBMS has traditionally been used to handle large volumes of data and ensures consistency and integrity of data. The development of the DBMS went from managing administrative data to managing spatial data. Spatial data used to be organized by having the administrative data in a Relational DBMS (RDBMS) separately from the spatial data in a GIS (Zlatanova, 2006). A RDBMS has the relational model (structure, operations and integrity rules) as basis and move, store and retrieve data from or into a DBMS, so it can be manipulated by applications. A RDBMS contains logical operations[15] and physical operations[16] (Ashdown & Kyte, 2014). The separate spatial data was managed in a separate single file in a proprietary format. Now the dual architecture is replaced by a layered architecture and spatial data is supported by DBMSs (Zlatanova, 2006), it is possible to manage spatial data into a single spatial DBMS.

3.5.1 Spatial DBMS

A spatial DBMS is able to maintain spatial data types (such as point, lines, polygons) in contrary to traditional DBMSs. With a spatial DBMS it is possible to perform functions on spatial data types, like returning geometric information, doing basic geometric transformation, maintaining valid geometry, etc. However spatial DBMSs can be 3D, the functions are based on 2D, which means that the z-value is not considered (Pu, 2005).

There are many DBMS which support spatial data. The biggest DBMSs with spatial support are Oracle, PostgreSQL, MonetDB, Microsoft SQL server and MySQL.

[15] Logical operations specifies what content is required (Ashdown & Kyte, 2014).
[16] Physical operations specifies how things should be done (Ashdown & Kyte, 2014).

Oracle

Oracle is a RDBMS that implements object-oriented features (user-defined types, inheritance, polymorphism), which is called an object-relational database management system (ORDBMS). An ORDBMS makes it possible to store complex business models in Oracle DBMS (Ashdown & Kyte, 2014). The spatial extension of Oracle DBMS is called Oracle Spatial. Oracle Spatial enables the storage, retrieval, update and query of spatial features in an Oracle DBMS using SQL schema and functions. 3D spatial data is supported by Oracle Spatial (Ashdown & Kyte, 2014).

PostgreSQL

PostgreSQL is an open source object-relational database management system (ORDBMS), which supports the SQL standard to a large extend. PostgreSQL can be extended by the user by adding new data types, functions, operators, aggregate functions, index methods and procedural languages (PostgreSQL, n.d.). PostgreSQL supports spatial data with the PostGIS extension, which allows PostgreSQL to be used as a back-end spatial DBMS for GIS applications. PostGIS follows the OpenGIS Simple Feature Specification for SQL and supports 3D spatial data (The PostGIS Development Group, n.d.).

A new development for 3D operations is the SFCGAL extension of PostGIS. SFCGAL is a C++ wrapper around CGAL for advanced 2D and 3D functions (The PostGIS Development Group, n.d.). SFCGAL includes a solid geometry type, but this solid geometry is based on the ISO 19107. This amplifies that the SFCGAL Solid follows the solid concept of the GIS world: A boundary solid defined as a B-rep with a missing volume concept (SFCGAL, 2013).

In order to store large field values it is possible to use TOAST (The Oversized-Attribute Storage Technique). TOAST limits the logical size of any value of a TOAST-able data type to 1 GB (PostgreSQL, n.d.).

MonetDB

MonetDB is an open source column-oriented DBMS designed for multi-core parallel executions on desktops for reducing complex query processing. MonetDB supports many programming interfaces, such as JDBS, ODBC, PHP, Python, RoR,C/C++ and Perl (MonetDB, n.d.-a) and stores data in a binary structure, which are called a BAT (Binary Association Table). BAT represents a mapping from an OID[17] to a base type value (Goncalves & Kersten, 2011). MonetDB/SQL has an interface which supports 2D spatial data (OGC compliant), but has no support for 3D spatial data (MonetDB, n.d.-c).

[17] OID: object identifier. Unique value in a database of an object (Ramakrishnnan, 2003).

Microsoft SQL server

Microsoft SQL server is a RDBMS, based on SQL with XML support for web applications (Microsoft, n.d.-c). Microsoft SQL server 2014 supports spatial data in a Euclidian (flat) coordinate system and in a round-earth coordinate system. Microsoft SQL server only supports 2D data (Microsoft, n.d.-d).

MySQL

MySQL is a popular open source RDBMS, which is based on SQL and developed, distributed and supported by Oracle Corporation. The source code of MySQL is available to study and to adapt according to the user's preferences. With MySQL it is possible to access DBMSs via the Internet. MySQL supports spatial data only in 2D (MySQL, 2015).

3.5.2 Spatial Queries

A spatial DBMS can perform spatial queries on spatial data. 2D spatial queries are based on the 2D topological relationships as shown in figure 3.14. Figure 3.14 shows the following topological relationships: disjoint, contains, touch, equals, inside and overlap (Clementini, Di Felice, & Van Oosterom, 1993). In order to perform spatial queries in 3D the topological relationships of figure 3.14 have to be converted to 3D topological relationships. The conversion of 2D topological relationships to 3D topological relationships is called the volume and volume relationship, which has been proposed by Egenhofer (1995). With the conversion of the 2D topological relationships to the 3D topological relationships two new relationships are proposed: covers and cover by. Figure 3.15 shows an overview of all possible volume and volume relationships in relation to figure 3.14, highlighting the new proposed volume and volume relationships of Egenhofer (1995).

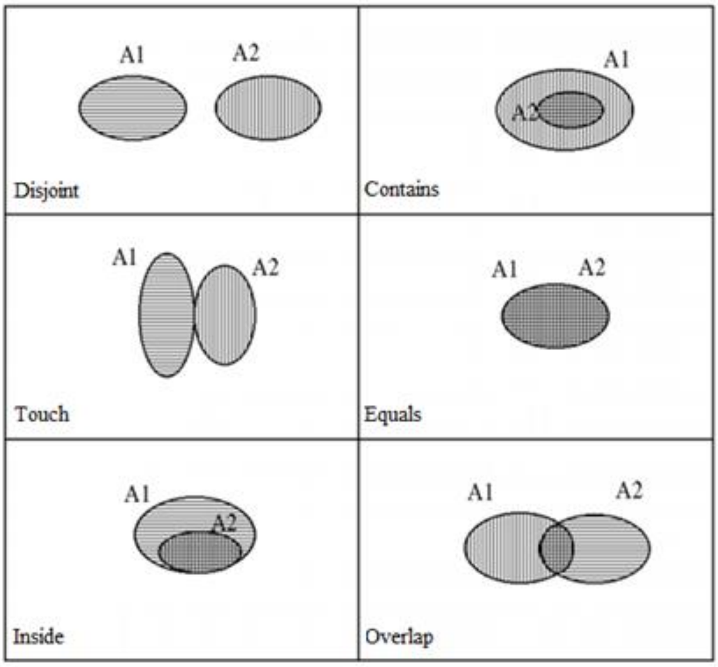


Figure 3.14: Visualization of six different 2D topological relationships (based on Clementini et al. (1993))

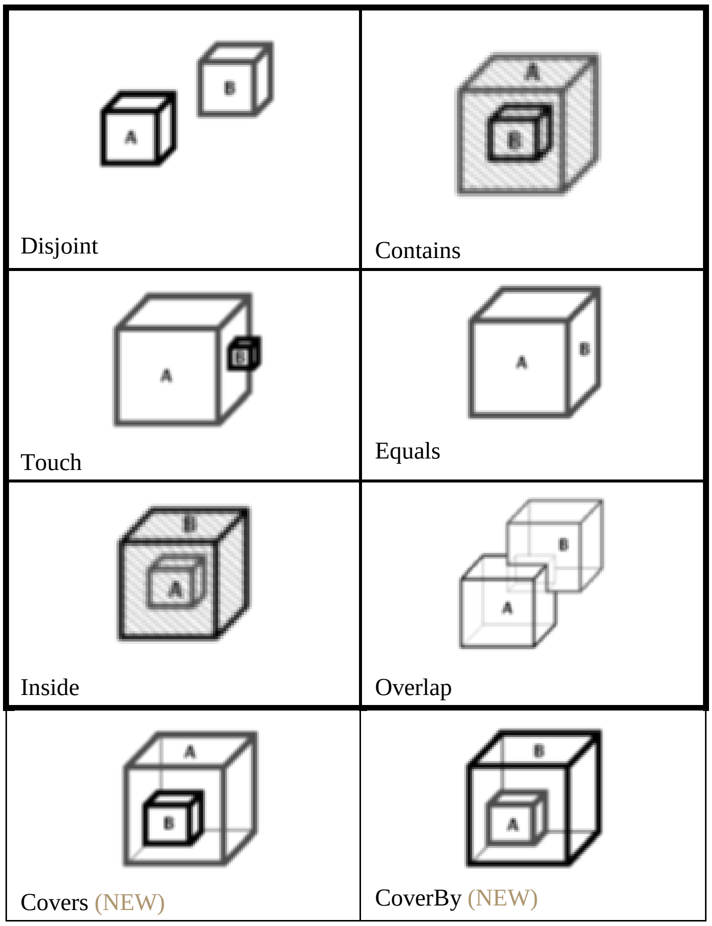


Figure 3.15 Visualization of the proposed volume and volume relationships by Egenhofer (1995) (Image based on Zlatanova (2000))

The distinction between all the volume and volume relationships have been made with help of the 9-IM. The 9-IM (9-intersection model) is a 3x3 matrix as shown in Eq. 3.8 (Egenhofer, 1995).

$$\widetilde{\varsigma_9}(A,B) = \begin{pmatrix} A \cap B & A \cap \delta B & A \cap B^- \\ \delta A \cap B & \delta A \cap \delta B & \delta A \cap B^- \\ A^- \cap B & A^- \cap \delta B & A^- \cap B^- \end{pmatrix} \tag{3.8}$$

where:

A = interior of volume A

B = interior of volume B

δA = boundary of volume A

δB = boundary of volume B

A^- = exterior of volume A

B^- = exterior of volume B

The 9IM or Egenhofer- Matrix is an extension of the 4IM (Four Intersection Model) (Strobl, 2008) and incorporates six object parts of two volumes: interior, boundary and exterior (Eq. 3.8). These six object parts show a topological relation of two volumes. In figure 3.16 the volume and volume relationships of figure 3.15 are shown with the corresponding 9IM. The 9IM is a binary topological relationship model, so true is indicated with 1 and false is indicated with 0 (Egenhofer, 1995).

$\begin{array}{c c} & B \quad \partial B \quad B^- \\ \begin{array}{c} A \\ \partial A \\ A^- \end{array} & \begin{pmatrix} 0 & 0 & 1 \\ 0 & 0 & 1 \\ 1 & 1 & 1 \end{pmatrix} \end{array}$ Disjoint	$\begin{array}{c c} & B \quad \partial B \quad B^- \\ \begin{array}{c} A \\ \partial A \\ A^- \end{array} & \begin{pmatrix} 1 & 1 & 1 \\ 0 & 0 & 1 \\ 0 & 0 & 1 \end{pmatrix} \end{array}$ Contains
$\begin{array}{c c} & B \quad \partial B \quad B^- \\ \begin{array}{c} A \\ \partial A \\ A^- \end{array} & \begin{pmatrix} 0 & 0 & 1 \\ 0 & 1 & 1 \\ 1 & 1 & 1 \end{pmatrix} \end{array}$ Meet	$\begin{array}{c c} & B \quad \partial B \quad B^- \\ \begin{array}{c} A \\ \partial A \\ A^- \end{array} & \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \end{array}$ Equals
$\begin{array}{c c} & B \quad \partial B \quad B^- \\ \begin{array}{c} A \\ \partial A \\ A^- \end{array} & \begin{pmatrix} 1 & 1 & 1 \\ 0 & 1 & 1 \\ 0 & 0 & 1 \end{pmatrix} \end{array}$ Inside	$\begin{array}{c c} & B \quad \partial B \quad B^- \\ \begin{array}{c} A \\ \partial A \\ A^- \end{array} & \begin{pmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{pmatrix} \end{array}$ Overlap
$\begin{array}{c c} & B \quad \partial B \quad B^- \\ \begin{array}{c} A \\ \partial A \\ A^- \end{array} & \begin{pmatrix} 1 & 0 & 0 \\ 1 & 1 & 0 \\ 1 & 1 & 1 \end{pmatrix} \end{array}$ Covers	$\begin{array}{c c} & B \quad \partial B \quad B^- \\ \begin{array}{c} A \\ \partial A \\ A^- \end{array} & \begin{pmatrix} 1 & 0 & 0 \\ 1 & 0 & 0 \\ 1 & 1 & 1 \end{pmatrix} \end{array}$ CoverBy

Figure 3.16: The volume and volume relationships with their 9IM (Egenhofer, 1995)

3.5.3 Data Exchange Formats

“Data exchange is the problem of taking data structured under a source schema and creating an instance of a target schema that reflects the source data as accurately as possible” (Fagin, Kolaitis, Miller, & Popa, 2005). Data exchange is used for data transfer between existing, independently created applications (Fagin et al., 2005). The most used languages for data exchange are RDF, XML, JSON, YAML, Gellish and CSV. These languages are compared with the biggest DBMSs from section 3.5.1 in table 3.6. Table 3.6 gives an overview which DBMS is compatible with which data exchange formats. This table only incorporates existing extensions. Formats which are marked as incompatible might be supported by DBMSs if the DBMS allows user defined types.

Table 3.6: DBMS format ready comparison for data exchange[18]

	RDF*	XML	JSON	YAML	Gellish**	CSV
Oracle	●	●	●	○	◍	●
PostgreSQL	◍	●	●	●	◍	●
MonetDB	●	●	●	○	◍	●
MS SQL	●	●	●	○	◍	●
MySQL	◍	●	●	○	◍	●

[18] Black marked are natively supported, grey are supported with external implementation. This table is based on several sources (Adams, 2014; Boncz, Manegold, & Rittinger, 2005; Gellish, n.d.; Intellidimension, n.d.; Microsoft, n.d.-a, n.d.-b, n.d.-e; Minh Duc, 2013; MonetDB, n.d.-d; MySQL, 2015; PostgreSQL, n.d.; W3C, 2003).

* RDF script of W3C (W3C, 2003) works with all SQL DBMSs.

** Gellish implementation is possible in any DBMS (Gellish, n.d.).

4. Methodology

This chapter focusses on the methodology of this graduation project. During this graduation project the most suitable solution for storing 3D solids of the petrochemical industry in a spatial DBMS will result into a new DBMS design for Fugro GeoServices. The DBMS redesign of the current DBMS of Fugro GeoServices contains two subdivisions: 1. Observation (section 4.1) and 2. DBMS Design (section 4.2). After redesigning the DBMS of Fugro GeoServices it is important to evaluate the new designed DBMS (section 4.3).

4.1 Observation Methodology

For this graduation project Fugro GeoServices has provided the data set of the petrochemical industry of Pernis for a pilot. This data must be analyzed in order to understand the context and current situation.

4.1.1 Data Inventory

For the Data inventory it is important to analyze the current DBMS in order to see how Fugro GeoServices is dividing the 3D components. For this an UML (Unified Modeling Language) diagram has to be made. This must result into technical and fundamental requirements, which are the constrains of the saving method of a 3D solid.

While analyzing the data set of Fugro GeoServices it also must be considered if extending the existing volume and volume relationships of Egenhofer (1995) is needed. If a situation in the data set does not fit within the volume and volume relationships of Egenhofer (1995), or it is desirable to emphasize a situation an additional volume and volume relationship must be defined in order to see if extending figure 3.15 in section 3.5 is needed. This additional relationship will be compared with the closest similar volume and volume relationship of Egenhofer (1995) with help of the 9-IM of Egenhofer (1995). This comparison is needed in order to verify if the additional volume and volume relationship is not the same as a defined volume and volume relationship of Egenhofer (1995).

4.1.2 Potential Exploring

After defining the constrains (resulting from the Data inventory) it is important to explore the potency of the DBMSs and methods. The DBMSs of section 3.5.1 will be assigned as 2D and/or 3D ready. These DBMSs are compared with the methods for storing 3D solids into a DBMS. These methods are CAD and GIS saving methods, which were treated in sections 3.1 and 3.2. All these methods are compared with the CAD and BIM formats (section 3.1 and 3.4). This comparison will show which method(s) can be used with which format. All methods will also be assigned as a true solid or boundary solid (terms are explained in section 3.3.2). Exploring the potency of the DBMSs and methods will result into a summary, showing which methods allow the storage of a true solid in a DBMS.

4.2 DBMS Design Methodology

After observing the data set of Fugro GeoServices and exploring the methods for storing a true solid the DBMS of Fugro GeoServices will be redesigned. The DBMS redesign will happen with an iterative approach.

4.2.1 Data Storage

The 3 most promising methods of section 4.1 have to be chosen for storing the petrochemical industry as a 3D solid into a spatial DBMS. Each method should be defined in a flow chart, and supported by literature. The flow chart will be made confirm the ECMA-4 standard (ECMA, 1966). The most important symbols for the program flowchart confirm ECMA-4 can be found in Appendix B.

After defining every method the pros and cons have to be noted for each method in the form of a SWOT analysis. For choosing the most efficient and promising method an adapted version of the multi criteria analysis framework of Kaklauskas, Zavadskas, and Raslanas (2005) will be used. The multi-criteria analysis is a framework for ranking and scoring (multiple) variants against multiple criteria (Hajkowicz & Collins, 2007; Hajkowicz & Higgins, 2008). The value of these criteria will be displayed in the evaluation framework with a weight. This weight gives importance to a criterion in relation to the other criteria. With this evaluation framework it will be possible to either compare different variants with each other as the total concept or to focus on an isolated criterion (Kaklauskas et al., 2005).

The multi-criteria analysis is suitable for comparing the 3 methods, because every method is a variant. The criteria for this framework will be the technical and functional requirements, which will result from the data observation (section 4.1). As an addition on these requirements the pros and cons will be added as five separate criteria (strength, weaknesses, opportunities and threats) in order to give weight to the pros and cons of the methods. The method which will be ranked as the highest will be chosen for development. If the chosen method cannot fulfill the requirements, another potential method will be chosen for development. The choice of the other method depends on the development and weaknesses of the method. It might be possible to choose another method for exploring the possibilities of choosing another method, even if the developed method fulfills all criteria.

During this graduation project each iteration will lead to a new variant. Every new variant needs to be analyzed and evaluated against former variants. This analysis will also be done with an adapted version of the multi criteria analysis framework of Kaklauskas et al. (2005).

4.2.2 Performance Analysis

This analysis will focus on the performance of spatial queries (topological relationships and attribute calculation) and optimization of the data storage. The criteria for this analysis will be the following:

1. Spatial relationship calculation (in terms of precision)
2. Attribute calculation (in terms of precision)
    a. 2D
       e.g. length
    b. 3D
       e.g. volume

All data of every variant must be saved in terms of e.g. query time, 2D attribute calculation, 3D volume computation, etc. This may provide interesting statistical information for the evaluation.

For analyzing the data storage it is important to investigate the influence of indexing on the DBMS. For this a performance analysis framework will be used, which is based on Bing Yao and Hevner (1984), for evaluating the performance of DBMSs. 4 scenarios regarding indexing will be evaluated (Bing Yao & Hevner, 1984):

0. No indexing on DBMS
1. Clustered index on key attributes in the DBMS
2. Nonclustered indices on secondary key attributes in the DBMS
3. Complete indexing on all attributes in the DBMS

The response time (time-to-last-record) of this scenarios will be recorded in time for the following 4 query conditions (Bing Yao & Hevner, 1984):

1. Atomic condition

   Simple selection on a relation, e.g.:

   ```
   Relation.Attibute= '10'
   ```

2. Item condition

   Disjunction (OR) of two atomic conditions, e.g.:

```
Relation.Attibute= '10' OR Relation.Attibute= '20'
```

3. Record condition

   Conjunction (AND) of two atomic conditions, e.g.:

   ```
   Relation.Attibute= '10' AND Relation.Attibute2= 'ABC'
   ```

4. Query condition

   Disjunction (OR) of record conditions, e.g.:

   ```
   Relation.Attibute= '10' OR Relation.Attibute2= 'ABC'
   ```

This analysis must end with an overall conclusion, concluding how the data storage can be improved. Besides indexing other considerations must be taken into account, such as clustering and object adaption.

4.3 Evaluation Methodology

For evaluating DBMSs it is necessary to analyze the overall performance of a DBMS. The overall performance evaluation of the 3D DBMS will be based on the benchmark methodology of Bing Yao and Hevner (1984). The benchmark methodology considers a variety of variables in order to evaluate the performance of a DBMS. Each variable must be isolated as much as possible for the evaluation. As shown in figure 4.1, the benchmark methodology consists out of three stages (Bing Yao & Hevner, 1984):

1. Benchmark Design

   In this stage the experimental design will be developed after designing the DBMS. This stage includes setting the running environment (loading the test data into the DBMS) and setting the benchmark workload. The benchmark workload provides general information about the usage of the DBMS in practice, such as types of queries (single-relation queries, multi-relation queries, updates), number of users, accessibility of users at the same time, etc. (Bing Yao & Hevner, 1984).

   In the experimental design parameters are selected for testing the DBMS. These parameters must push the system to its performance limitations. Parameters to be considered are DBMS size, background load, number of indices, query complexity and number of simultaneous users (Bing Yao & Hevner, 1984).

2. Benchmark Execution

   In the benchmark execution stage the experiment of phase 1 is executed and the performance data of this experiment is collected (Bing Yao & Hevner, 1984).

3. Benchmark Analysis

   In the benchmark analysis stage the performance data is analyzed. If more than one system (DBMS) is benchmarked the performance of the different systems can be compared. This phase result into a summary, pointing to interesting results and explaining the reasons behind the results. The benchmark analysis can provide two kinds of analyses: I.) individual system analysis (based on one DBMS) and II.) comparative system analysis (only if multiple DBMSs are considered) (Bing Yao & Hevner, 1984).

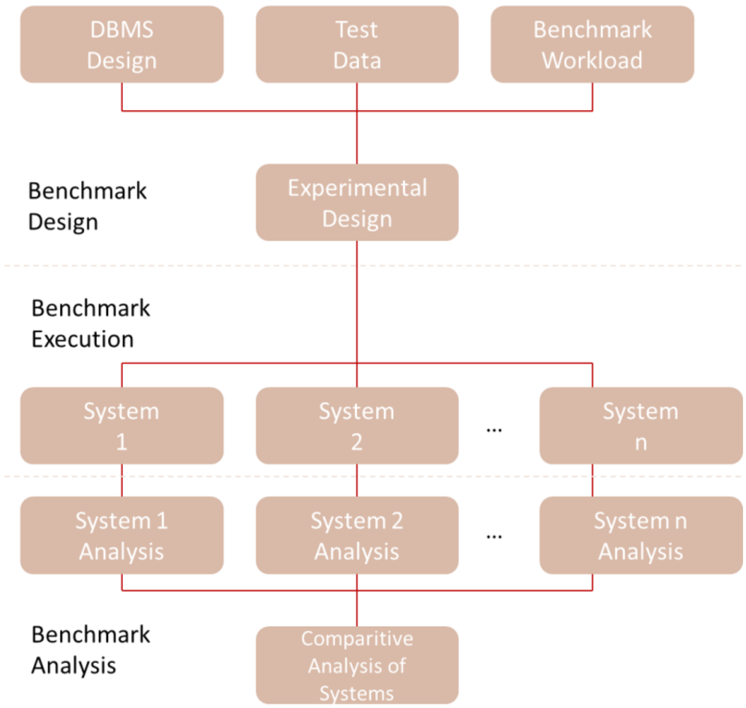


Figure 4.1: DBMS benchmark methodology flow chart (based on Bing Yao and Hevner (1984))

For the experimental design (Stage 1: Benchmark Design) response time will be chosen as the performance measurement, because response time is the most readily available measure and most chosen measurement utilized in benchmarks. Response time can be considered as time-to-first-record and time-to-last-record. The difference between both is the query time stops when one result has been found (time-to-first-record) or continues until all records are found (time-to-last-record) (Bing Yao & Hevner, 1984). For this research time-to-last-record will be used as measure unit. These result must also incorporate whether the queries are executed in a 'cold' of 'hot' environment, because this may affect the results. The difference between a ‘hot’ and ‘cold’ environment is whether the query already has been executed and is present in the memory/OS of DB cache (Van Oosterom et al., 2015).

The following experimental variables are compulsory for the experimental design (Bing Yao & Hevner, 1984):

1. Query complexity (section 4.4)
    a. Atomic condition
    b. Item condition
    c. Record condition
    d. Query condition
2. Records retrieved
3. Order of query execution
4. Sorting
   e.g. ‘ORDER BY’
5. Aggregation functions
   e.g. ‘COUNT’ or ‘MAX’

In contrary to Bing Yao and Hevner (1984), 2D and 3D spatial queries also have to be incorporated as compelled variables for this research. It is allowed to add additional experimental values, according to the designed DBMS, such as DBMS storage size. The result of all these comparisons and tables must result into a conclusion if the 3D DBMS is usable for Fugro GeoServices and which improvements have to be made. As has been done by Van Oosterom et al. (2015), the benchmark design stage (experimental design) will incorporate the following three phases:

1. Mini-benchmark
   The mini-benchmark phase is used as a test for the benchmark design phase. This phase of benchmark design is used to decide the test options and system configuration for the next phase (medium-benchmark phase).

2. Medium-benchmark
   The medium-benchmark phase performs the benchmark design phase with a pilot data set. This phase eliminates most of the systems, based on performance, for the upscaled-benchmark phase. This scale for performing the benchmark execution will be the highest

scale of this graduation project. For this reason this scale will be used for the benchmark execution stage.

3. Upscaled-benchmark

   The upscaled-benchmark phase performs the benchmark execution phase with the whole data set of Fugro GeoServices. This phase will not be incorporated within this graduation project, because this gradiation project only focusses on a pilot data set of Fugro GeoServices.

5. Analysis

Within this chapter various analysis's have been made in order to see where the opportunities and innovations are for integrating CAD with GIS and storing the petrochemical data model in a 3D spatial DBMS. First it is important to observe the data and explore the opportunities for storing the data in a 3D DBMS in section 5.1.

5.1 Observation

5.1.1 Data Inventory

While analyzing the dataset of the petrochemical industry (figure 5.1) it becomes visible that the data set contains CSGs. The data set is saved as a DGN-file in Bentley MicroStation V8. Each element in the DGN-file is a solid geometry (primitive solid) in a cell. Looking at the dataset and DBMS it becomes visible that the dataset consists of the following components:

- Pipes, with the following primitives:
    - Cylinder
    - Elbow
    - Cone
    - Dome
    - Belt
    - Not implemented
    - Grouped geometry (combination of above)
- Mechanicals, with the following primitives:
    - Cylinder
    - Shape
    - Dome
    - Cone
    - Grouped geometry (combination of above)
- Basemap
    - Cylinder
- Structures
    - Shape

However this information has been retrieved by looking at a sample of the model and DBMS. It might be possible that some geometry types have not been noticed. From the converting program (DGN to JSON) the following geometry types are present in the DBMS and models:

- Line (non-solid)
- Dome
- Cone
- Ellipse (with and without orientation) (non-solid)
- Arc (non-solid)
- Elbow
- Grouped
- Shape
- EllipseArc (non-solid)
- SolidSphere
- SolidShape
- NotImplementedGeometry
- Belt

From this geometry types the non-solid geometries are highlighted. These non-solid geometries are used for grouping and assigning these grouped non-solids to a true CAD solid. Comparing the found geometry types in the DBMS with the geometry types of the conversion program it becomes visible that the SolidSphere has not been found, and could be a geometry in one of the classes.

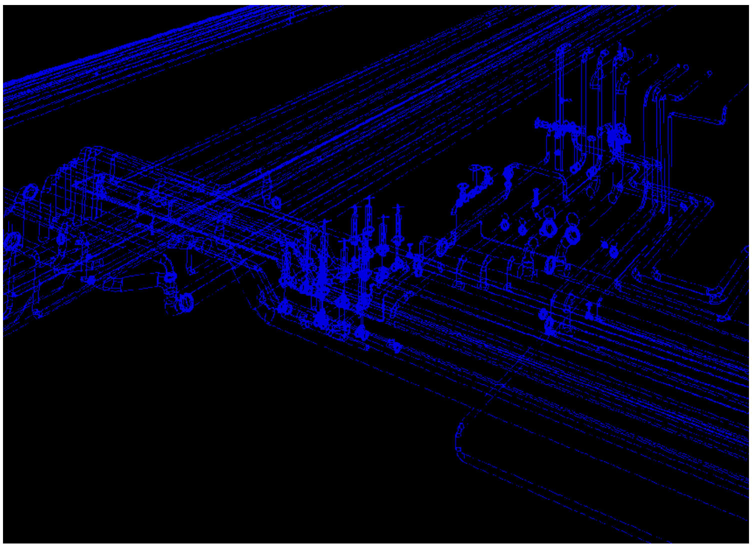


Figure 5.1: Petrochemical industry as CSG (dataset Fugro GeoServices)

This information (including parametric sizes) is currently stored as characteristics of a 2D line in a spatial DBMS. With help of the current DBMS structure it is not possible to extract all the different component sizes to determine if the components have standard component sizes. Because of the steel construction it may be assumed that the profile of the components (width) have default sizes.

Current DBMS structure

The current DBMS structure is shown in figure 5.2. The spatial data types (pipes, mechanicals, basemap and structures) are stored as a 2D line in the DBMS. The attributes of these 2D lines are stored in a Pipe-JSON. This Pipe-JSON includes the id, type of geometry and their parametrical attributes, such as radius or center point, normal vector (in case of a dome), height (in case of a dome), starting point and ending point, etc. The points are stored with their x, y, and z coordinates. In case of a Shape only the vertices are stored as a point. The electricals-, civils- and control system classes are empty in the current DBMS. The line size and name of the lines (representing the solids in 2D) are also stored in the DBMS.

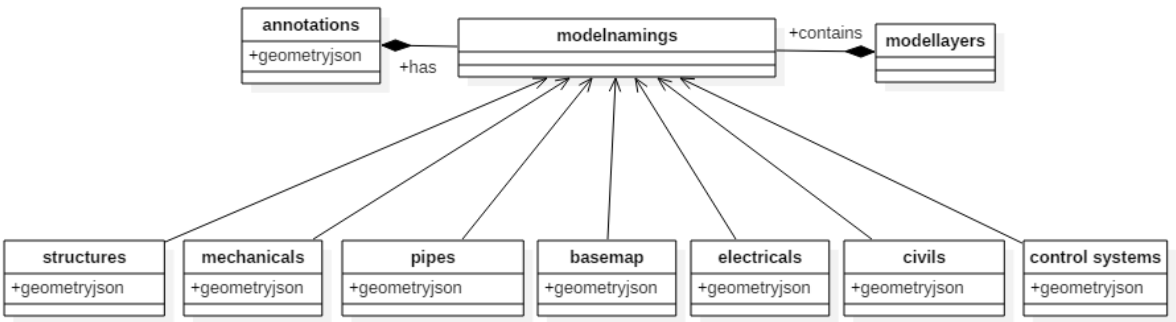


Figure 5.2: UML diagram of the current DBMS. The extensive version of this UML diagram can be found in Appendix C.

It might be possible that a geometry has a nested Pipe-JSON as an attribute: the geometry can exist e.g. of a cylinder and a cone. Figure 5.3 shows an example of a single and nested Pipe-JSON data. As can be seen from figure 5.3 the topology of the data set is not stored. Within the DBMS there is a schema, called “topology,” but this schema is unpopulated.


```
"{"Id":"P1","Type":"Cylinder","Radius":0.024,"PointBegin":{"X"
:-449.612,"Y":115.702,"Z":0.402},"PointEnd":{"X":-
449.612,"Y":115.702,"Z":0.404}}"

"[{"Id":"P9","Type":"Cone","RadiusBegin":0.033,"RadiusEnd":0.0
24,"PointBegin":{"X":-
286.857,"Y":160.273,"Z":1.936},"PointEnd":{"X":-
286.857,"Y":160.273,"Z":1.981}},{"Type":"Cylinder","Radius":0.
064,"PointBegin":{"X":-
286.857,"Y":160.273,"Z":1.919},"PointEnd":{"X":-
286.857,"Y":160.273,"Z":1.936}}]"
```


Figure 5.3: Single and Nested Pipe-JSON data

Technical and Functional requirements

With the new DBMS for Fugro GeoServices the following requirements have to be fulfilled:

- 3D query possibility
- Result must be visualized in 2D and 3D
- Volume calculation possibility

  Note: volume can be defined as material volume or object volume.
- Buffer calculation (of certain area)
- Volume and volume relationships
- Length calculation
- Distinguish properties of the components, e.g. current state.
- Topology determination (no high priority)

Additional volume and volume relationships

Comparing figure 3.15 in section 3.5 with the data set one volume and volume relationship seems to be missing: 'crossing.' However this volume and volume relationship is hard to define. This notion is supported in figure 5.4.

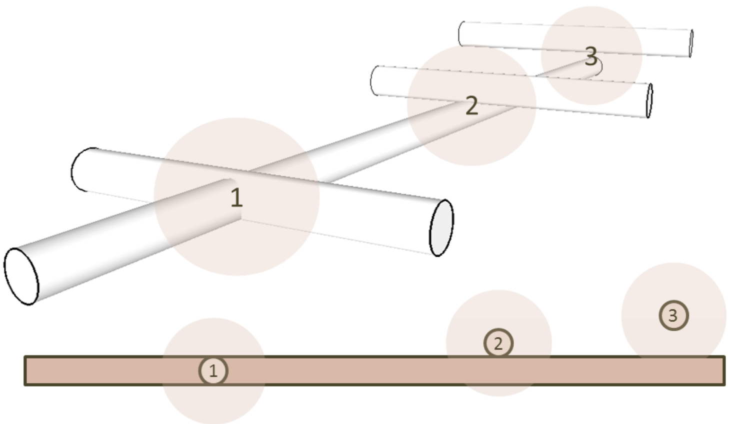


Figure 5.4: Crossing example situation 1, 2 and 3

Figure 5.4 shows three kinds of crossing: 1. crossed overlap, 2. crossed touching and 3. distance crossing. These three kinds of crossing are similar to the following volume and volume relationships of Egenhofer (1995): I.) crossed overlap versus overlap, II.) crossed touching versus touch and III.) distance crossing versus disjoint.

Table 5.1 compares the proposed volume and volume relationships with the existing volume and volume relationships of Egenhofer (1995). From table 5.1 can be concluded that the proposed- and the existing volume and volume relationship are identical. However figure 5.5 shows some examples that this is not the case.

Table 5.1: Relationship verification

Relationships	1. Crossing overlap 1 1 1 1 1 1 1 1 1	2. Crossed touching 0 0 1 0 1 1 1 1 1	3. Distance crossing 0 0 1 0 0 1 1 1 1
Overlap 1 1 1 1 1 1 1 1 1	1	0	0
Touch 0 0 1 0 1 1 1 1 1	0	1	0
Disjoint 0 0 1 0 0 1 1 1 1	0	0	1

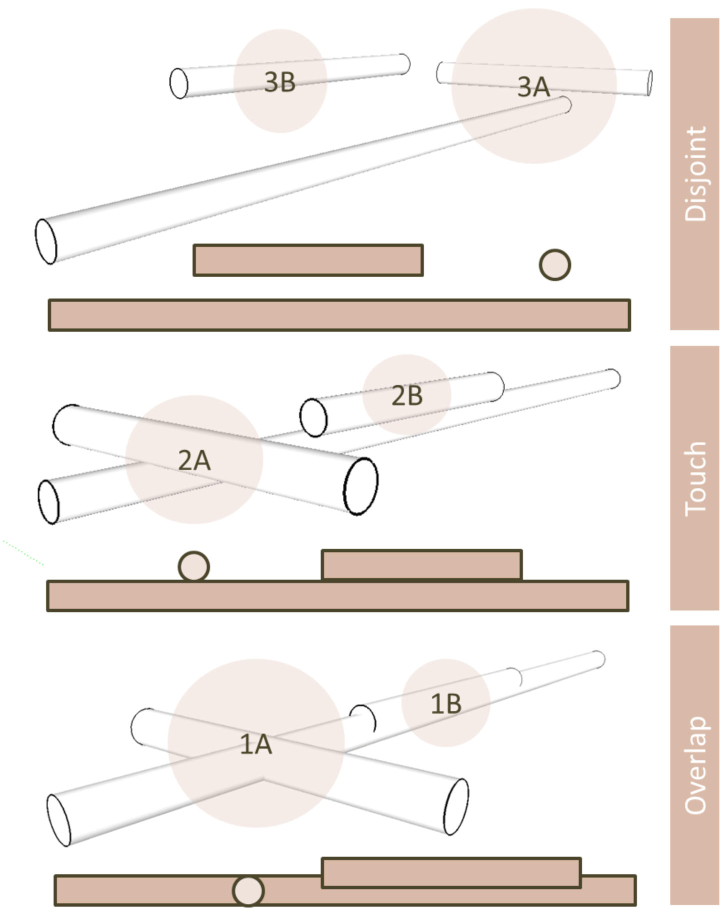


Figure 5.5: Situation examples overlap, meet & disjoint

Figure 5.5 shows the overlap, touch and disjoint volume and volume relationship. If one wishes to introduce the volume and volume relation ‘crossing’ all the A-situations are crossing situations, which cannot be distinguished from the B-situations (which are no crossing situations) with help of the 9IM. In order to make this distinction the following definition of ‘crossing’ has to be maintained:

> *Crossing: If a volume A overlaps, touches or is disjoint with a volume B and both volumes (A and B) are not parallel to each other, these volumes (A and B) are crossing.*

Conclusion

Figure 5.2 shows the UML diagram of the current DBMS structure. Looking at the documentation of Fugro GeoServices it becomes visible that this UML diagram is not complete. During this graduation project it will be necessary to adapt and/or extend this UML diagram according to the provided documentation of Fugro GeoServices for the new DBMS of Fugro GeoServices. The new DBMS has to fulfill the following new requirements:

- 3D query possibility
- Result must be visualized in 2D and 3D
- Volume calculation possibility
  Note: volume can be defined as material volume or object volume.
- Buffer calculation (of certain area)
- Volume and volume relationships
- Length calculation
- Distinguish properties of the components, e.g. current state.
- Topology determination (no high priority)

In order to make a distinction of pipes crossing other pipes a new volume and volume relationship has to be introduced. The current Pipe-JSON is not convenient for advanced queries and has to be improved.

5.1.2 Potential Exploring

As a result of sections 3.1, 3.2 and 3.4 the DBMS & method potential table can be made, which is shown in table 5.2.

Table 5.2: DBMS & method potential table

	Dimension ready		Method ready								
DBMS	2D	3D	Polyhedron	NURBS	Voxels	Octrees	Point Cloud	B-rep	TEN	3D Voronoi Diagram	Parametric Design
Oracle	●	●	●	●	○	○	●	●	▨	○	●
PostgreSQL	●	●	●	●	○	○	●	●	▨	○	●
MonetDB	●	◉	○	○	◉	○	◉	○	○	○	○
MS SQL	●	○	○	○	○	○	○	○	○	○	○
MySQL	●	○	○	○	○	○	○	○	○	○	○
Format											
DGN			○	○	○	○	●	●	○	○	●
DWG*			○	●	○	○	●	●	○	○	●
IFC			○	○	○	○	○	●	○	○	●
Solid type											
True			●	○	●	●	○	○	●	●	●
Boundary			●	●	○	○	●	●	●	○	●

* Bentley MicroStation only saves boundary solids as a DWG.

Legend:

● Ready

○ Not Ready / Uknown

◉ Ready in future / Ready, but not as existing plug in

▨ Combination of the three above (color of background and color of stripes)

DBMS ready

Section 3.5 explains which DBMS is 2D and/or 3D ready, which is summarized in table 5.2. Table 5.2 shows that Oracle and PostgreSQL are the only DBMSs which support 3D geometry. MonetDB is currently not supporting 3D data, but MonetDB will support 3D point clouds and voxels in the future (MonetDB, n.d.-b) and this 3D extension is available for this graduation project via Sisi Zlatanova.

B-reps (including polyhedra), NURBS, point clouds and parametric design are natively supported by DBMSs. TEN storage is possible in Oracle, because Wesselingh (2007) has used TEN in Oracle during his MSc. Thesis. However this is not an implemented extension and is marked grey. Both Oracle and PostgreSQL can support TENs when the boundary surface is triangulated and the geometry is saved as a boundary solid. Parametric design is possible, because Oracle and PostgreSQL both allow users-defined types.

Table 5.2 only mentions what is currently possible. The other methods and standards also might be possible with Oracle and PostgreSQL, but has not been done for so far. For this reason this has been marked as false.

Formats versus methods

Comparing the formats with the methods in table 5.2 shows which methods are supported by which formats. With this comparison it becomes visible that these formats are based on boundary solids and parametric design.

True solids versus boundary solids

In table 5.2 a distinction has been made: true solids versus boundary solids. Polyhedra, NURBS, point clouds and B-reps are noted as boundary solid, because these methods only describe the boundary of a 3D object. However point clouds only described the boundary with points, these points can be interpolated for describing the whole boundary. Voxels, octrees, TEN and 3D Voronoi Diagram are noted as a true solid. Parametric design supports both.

Besides boundary solid, Polyhedra can be seen as a true solid when polyhedra are modelled with the constraint that holes in the faces of the polyhedron are only accepted when these holes are also filled with inner walls, as proposed by Kazar et al. (2008) (figure 5.6). By preserving this constrains of Kazar et al. (2008) this kind of solid modeling with polyhedrons seems similar to solid modeling with CSG (figure 5.7). Another opportunity for converting polyhedra to a true solid is by represent the interior of the polyhedron as a set of TENs, as proposed by Penninga and Van Oosterom (2008).

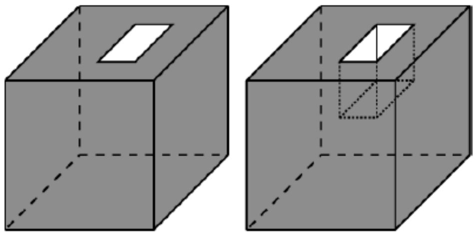


Figure 5.6: Invalid solid polyhedron (left) versus valid solid polyhedron (right) (Kazar et al., 2008)

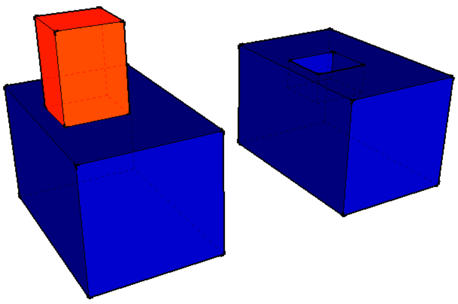


Figure 5.7: CSG similar to figure 5.6 by differencing two geometries (red and blue) with difference Boolean operator.

Voxels, octrees, TEN and 3D Voronoi Diagrams (section 3.2) are 3D repeated elements, which together form a volume: a true solid. However each separate element is a boundary solid this has been considered as a true solid, because the volume concept can be retrieved by assuming all boundary solids together form a true solid. Parametric design (section 3.1) can be both true solid and boundary solid. This depends on the method of parametric design.

All formats, which are CAD and BIM standards, support true solids. This notion is supported in sections 3.1 and 3.4.

Conclusion

From the potential analysis can be concluded that the following methods are suitable for storing true solids in a spatial DBMS:

- Polyhedra
- Voxels
- Octrees
- TEN
- 3D Voronoi Diagram
- Parametric design

Voxels and octrees will not be chosen for storing method, because both methods are not efficient in data storage and visualization. 3D Voronoi Diagram as a method will be considered as not suitable, because this method requires a big amount of nodes and nodes negatively effects the performance (Guerrero Iñiguez, 2012). TEN storage has already been done by Wesselingh (2007).

Parametric design and solid polyhedron modeling (figure 5.8) are the most obvious method for storing a true solid CAD geometry in a spatial DBMS, because CAD is based on parametric design and solid polyhedron modeling. Parametric design is a sufficient method for data storage. For this method it will be necessary to define a parametric library to reconstruct these solid geometries with solid polyhedron modeling and to write functions for querying the 3D geometries. Saving some topological attributes may improve the reconstruction. The attributes of all geometry objects have to be stored in a single cell (per object) in a suitable data exchange format. It also could be possible to triangulate the interior of the polyhedron, as has been proposed by Kazar et al. (2008). Looking at the data set and the type of solid modeling (CSG) it is important to set constrains to the geometries (no hole allowance for this specific type of data sets).

Besides CAD based parametric design it also could be possible to extend this parametric design with BIM. The CAD geometry can be converted to a BIM geometry, according to BIM semantics. These BIM semantics can be used for a DBMS structure.

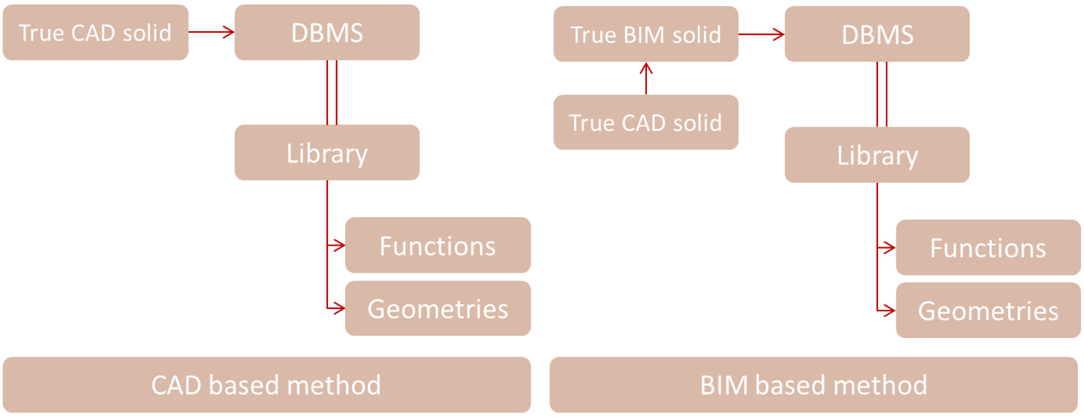


Figure 5.8: Parametric design solution based on CAD and BIM

Parametric design could also be possible with help of a point cloud (figure 5.9). With this method it could be possible to reconstruct geometry out of a highly tinned point cloud. The points of this point cloud based parametric design are stored as a 3,5D point: Having characteristics as attributes (e.g. type of geometry and radius) of a 3D point.

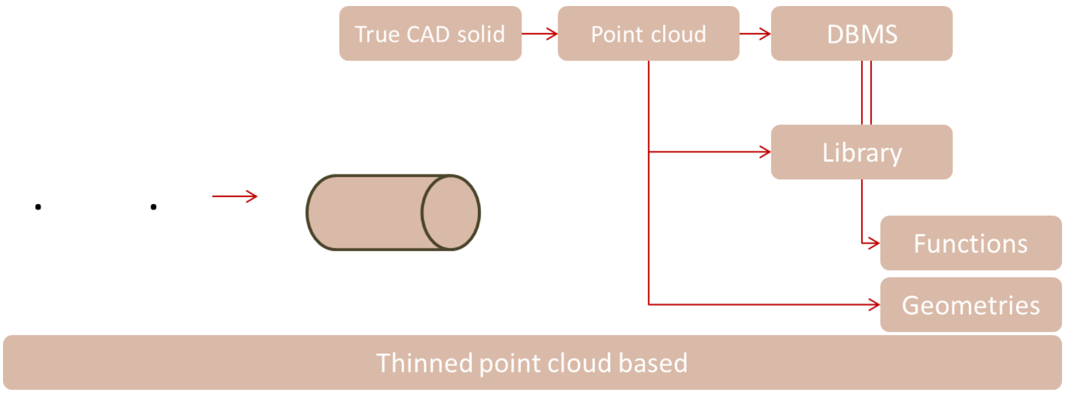


Figure 5.9: Parametric design solution based on a thinned point cloud

Another form of parametric design could be based on the voxelization method. The main problem with the current voxelization method is the high amount of voxels to represent a 3D geometry. This aspect lowers the performance and increases the storage space of voxels. Another weakness of voxels is the usage of one single repeated element (voxel) for all geometry types, which can result into a non-smooth object with a large amount of voxels. By combining parametric design with voxels a hybrid voxelization can be introduced (figure 5.10).

This hybrid voxelization uses different voxel geometries for a specific object type, which is best suitable for reconstructing one specific geometry of this object type. By repeating this specific voxel for a specific object type a 3D model can be created. For instance a cylinder can be represented by repeating one cylinder-voxel. This repeated cylinder voxel is constructed parametrically based on the radius.

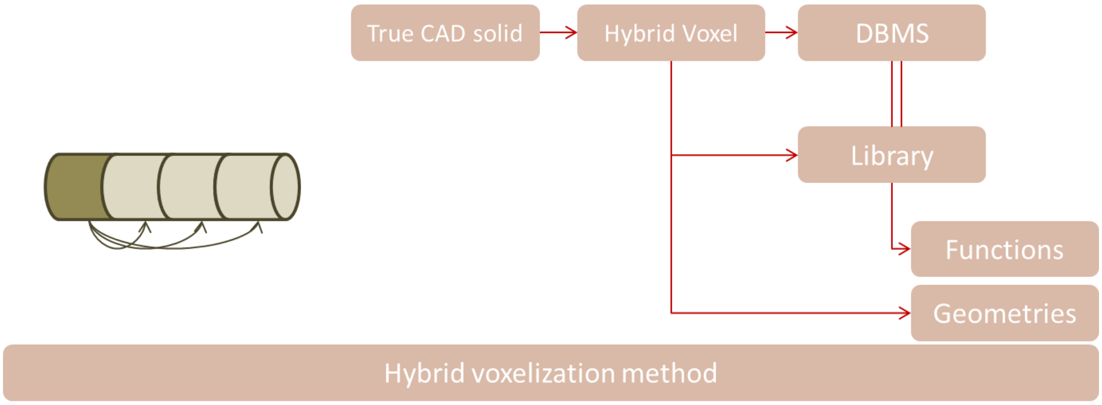


Figure 5.10: Parametric design solution based on a hybrid voxelization method

According to table 5.2 Oracle and PostgreSQL are the most suitable DBMSs for storing 3D objects as a true solid, because these DBMSs are 3D ready.

6. Graduation Plan

This graduation project, GEO2000, is divided into 5 phases (P1-P5) and has to be planned in order to monitor the overall progress. The important dates and deadlines for this graduation thesis are shown in table 6.1.

Table 6.1: Important dates graduation

	Week		Date	Description
P1	17	4.1	April 16th, 2015	Presentation Progress review Graduation plan
	24	4.8		<1 week before P2> Submit final graduation plan to all
P2	25	4.9	June 19th, 2015	Presentation Go / no go assessment Graduation plan
	33	5.6		<1 week before P3> Submit draft thesis to all
P3	34	5.7	August 2015*	Presentation Colloquium midterm
	36	1.1	September 1st , 2015	Final Application date P4 Go/No Go
	38-39	1.3-1.4		<1 week before P4> Submit draft thesis to all
P4	39-40	1.4-1.5	September 2015*	Presentation Go / no go assessment Process
	40 43-44	1.5 1.8-1.9	September 2015	Final Application date P5 Go/No Go <1 week before P5> Submit thesis to all
P5	44-45	1.9-1.10	November 2015*	Final Public Presentation & assessment

* Date is not specified. Must be set yet.

Table 6.1 is set as a framework to plan this graduation project. In order to make a more detailed planning it is first necessary to define the research methodology.

6.1 Research Methodology

The chosen research methodology for this project is the iterative methodology approach (figure 6.1), which is the core of the agile approach. “Iterative development is an approach to building software (or anything) in which the overall lifecycle is composed of several iterations in sequence” (Larman & Basili, 2003). With the iterative development it is recommended that the length of an iteration is between one and six weeks. Changes from external stakeholders are only possible at the end of each iteration (Larman & Basili, 2003). The iterative methodology approach enables to attack and react on time in an efficient manner. With having an iterative development it is possible to foresee misunderstandings and to encourage active feedback. This approach gives stakeholder evidence of the project's status during the lifecycle (Kruchten, 2001). During this research it is important to iterate in order to evaluate the current process of the project. These iterations will ensure that the end product fulfills the companies need. During each iteration it is possible to redefine the requirements of the spatial DBMS. This approach will ensure the companies satisfaction.

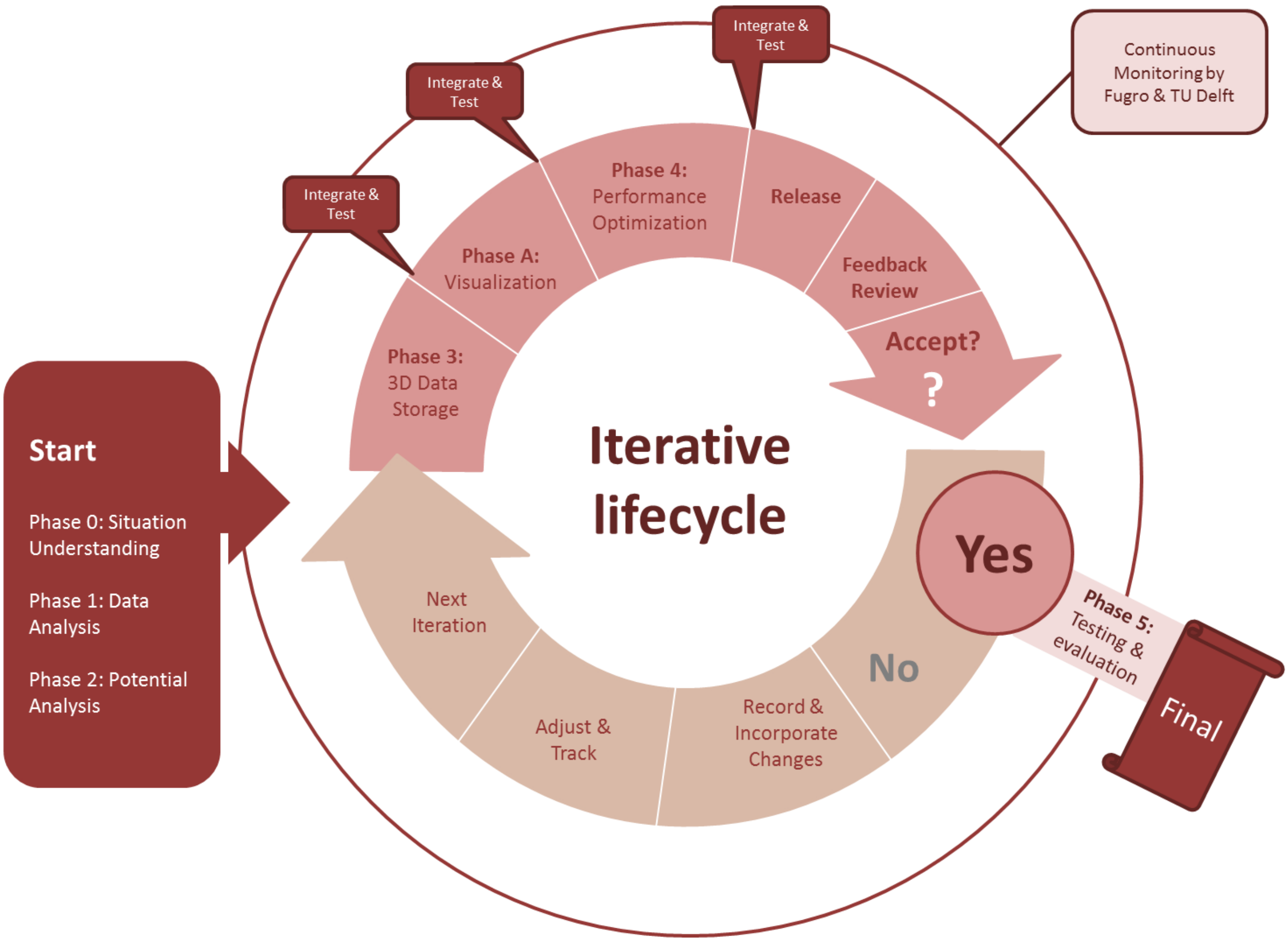


Figure 6.1: Iterative methodology approach

Within this graduation project the first alternative of the three chosen methods from phase 2 will be the most uncertain method for development. This unsure aspect is based on the agile approach, which has been the former research methodology of this graduation process. This aspect of the agile approach may lead to surprising results and new insights for future development.

6.2 Planning

Within these phases (P1-P5) the phases of this graduation are planned, aiming for a graduation in November 2015. This planning is shown in figure 6.2 and appendix A.

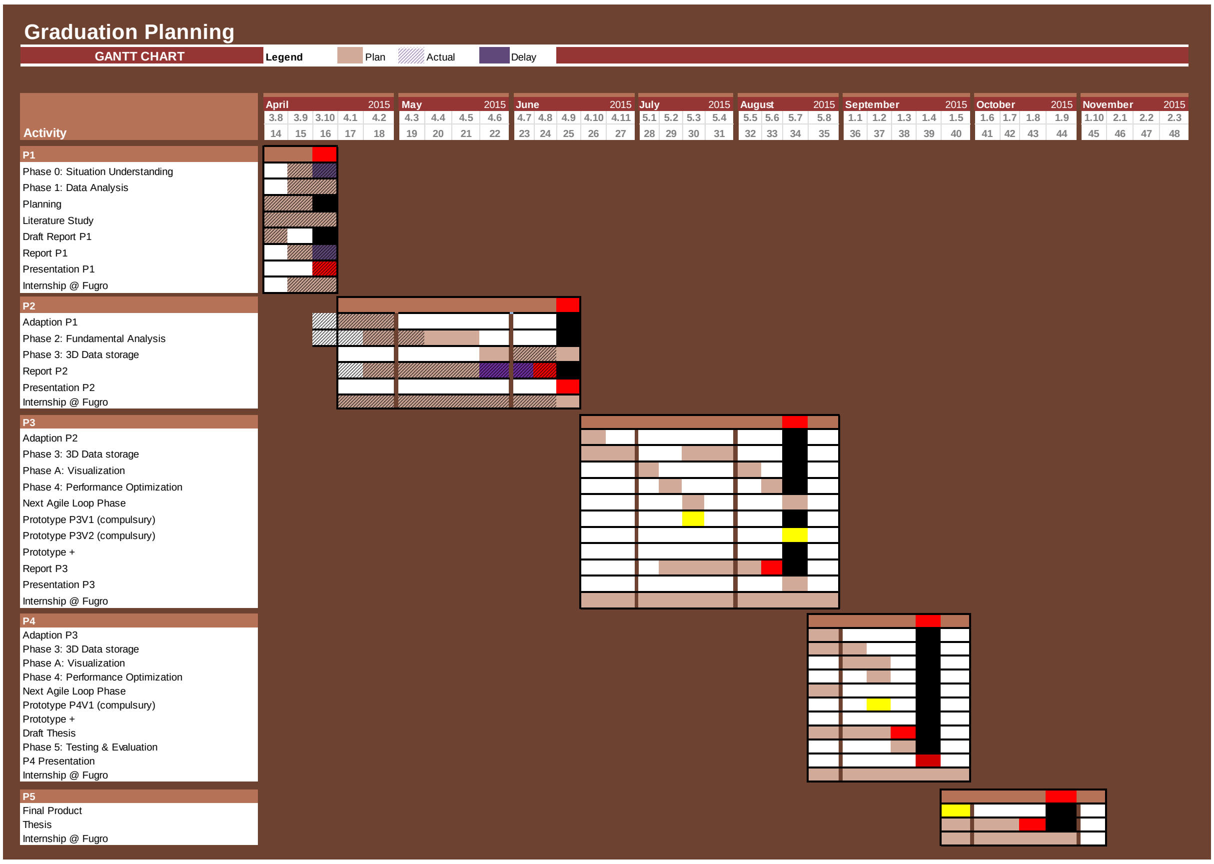


Figure 6.2: Gantt planning graduation (appendix A)

6.3 Supervisors

During this graduation project Wilko Quak and Martin Kodde will be my supervisors. Wilko Quak will support me via the TU Delft as my daily supervisor and Martin Kodde will support me via the company (Fugro GeoServices) as my company supervisor. Peter van Oosterom will be my graduation professor.

During this graduation project I will have a weekly meeting with my supervisors. Every Tuesday I will have my company meeting and every Friday I will have my weekly university meetings. From time to time all supervisors, including my graduation professor will have a plenary meeting in order to keep everyone on the same track. Every meeting will be reported in minutes and are sent to all supervisors.

The graduation professor will be present during every P-presentation at the TU Delft, and on the additional plenary meetings.

6.4 Equipment

During this graduation project the following equipment will be used or are available (marked grey):

- Company Desktop
  - Dell Precision T1700
  - Windows 7
  - Intel Xeon 3.3 GHz
- Laptop
  - Acer V3-571G
  - Windows 8
  - Intel Core i7 2.2 GHz
- PostgreSQL 9.3
  - DBMS
  - current company PostgreSQL version
  - Including the following extension(s):
    - PostGIS
- PostgreSQL 9.4
  - DBMS
  - Laptop PostgreSQL version
  - Including the following extension(s):
    - PostGIS
    - SFCGAL
- Oracle
  - DBMS
- FME
  - Data coversion program
- Autodesk AutoCAD
  - CAD tool
- Bentley MicroStation
  - CAD tool
- StarUML
  - UML tool
- Python
  - Scripting program
- MinGW

    - Development environment for Windows applications
- MSYS
    - Tool for building applications and programs, which where traditional developed for UNIX.
- Modelnamings
    - Program developed by Fugro GeoServices which extracts parameters of CAD files into DBMS INSERT scripts.

Other software applications are not set till now.

References

3D-max. (n.d.). Point Surface. Retrieved 22th April, 2015, from http://www.3dmax-tutorials.com/Point_Surface.html

Adams, D. (2014). *Oracle Database Online Documentation 12c Release 1 (12.1). Application Development*

Akinci, B., Karimi, H., Pradhan, A., Wu, C.-C., & Fichtl, G. (2010). CAD and GIS interoperability through semantic web services. *CAD and GIS Integration*, 199.

Arens, C. (2003). *Maintaining reality: modelling 3D spatial objects in a Geo-DBMS using a 3D primitive.* (MSc. Thesis), TU Delft, Delft University of Technology.

Ashdown, L., & Kyte, T. (2014). *Oracle Database Online Documentation 11g Release 2 (11.2): Database Administration* Retrieved from https://docs.oracle.com/cd/E11882_01/server.112/e40540/intro.htm#CNCPT88782

Avin. (n.d.). Wilkommen in der Soup von Avin Fathulla. Retrieved April 22th, 2015, from http://avin.soup.io/since/298207348?mode=own

Benner, J., Geiger, A., & Leinemann, K. (2005). Flexible generation of semantic 3D building modeld. *Proceedings of the 1st Intern. Workshop in Next Generation 3D City Models*, 17-22.

Bhanu, B., & Ho, C.-C. (1987). CAD-based 3D object representation for robot vision. *Computer;(United States), 20*(8).

Bing Yao, S., & Hevner, A. R. (1984). *A Guid to Performance Evaluation of Database Systems*. Washington, USA: USGPO.

Boncz, P. A., Manegold, S., & Rittinger, J. (2005). *Updating the Pre/Post Plane in MonetDB/XQuery*: CWI. Information Systems [INS].

Brugman, B. (2010). *3D topological structure management within a DBMS, validating a topological volume.* (MSc. Thesis), TU Delft, Delft University of Technology.

BuildingSMART. (n.d.-a). IFC Geometric Model Resource. Retrieved April 29th, 2015, from http://www.buildingsmart-tech.org/ifc/IFC2x3/TC1/html/ifcgeometricmodelresource/ifcgeometricmodelresource.htm

BuildingSMART. (n.d.-b). IfcSphere. Retrieved April 30th, 2015, from http://iaiweb.lbl.gov/Resources/IFC_Releases/R2x3_final/ifcgeometricmodelresource/lexical/ifcsphere.htm

Chorafas, D. N. (1983). *DBMS for distributed computers and networks*. New York: Petrocelli.

Clementini, E., Di Felice, P., & Van Oosterom, P. (1993). *A small set of formal topological relationships suitable for end-user interaction.* Paper presented at the Advances in Spatial Databases.

Computer Aided Detector Design. (n.d.). Boundary representation. Retrieved April 22th, 2015, from http://cadd.web.cern.ch/cadd/cad_geant_int/thesis/node23.html

Cowen, D. J. (1988). GIS versus CAD versus DBMS: what are the differences? *Photogrammatric Engineering and Remote Sensing, 54*(11), 1551-1555.

Cromwell, P. R. (1997). *Polyhedra; 'one of the most charming chapters of geometry'*. Cambridge, UK: Cambridge University Press.

Ctech. (n.d.). Triangular Networks. Retrieved April 22th, 2015, from http://evshelp.ctech.com/Content/workbooks/fundamentals_of_visualizing_environmental_data/triangular_networks.htm

Date, C. J. (1975). *An introduction to database systems*. Reading: Addison-Wesley.

De Laat, R., & Van Berlo, L. (2011). Integration of BIM and GIS: The development of the CityGML GeoBIM extension *Advances in 3D geo-information sciences* (pp. 211-225): Springer.

Dollne, J., & Hagedorn, B. (2008). Integrating urban GIS, CAD, and BIM data by service-based virtual 3D city models. In M. Rumor, E. Fendel & S. Zlatanova (Eds.), *Urban and Regional Data Management*. London, UK: Taylor & Francis Group.

Donkers, S. (2013). *Automatic generation of CityGML LoD3 building models from IFC models.* (MSc. Thesis), TU Delft, Delft University of Technology.

Du, Q., Emelianenko, M., & Ju, L. (2006). Convergence of the Lloyd algorithm for computing centroidal Voronoi tessellations. *SIAM journal on numerical analysis, 44*(1), 102-119.
Eckel, G. (n.d.). *OpenGL Volumizer Programmer's Guide* C. Cary (Ed.) *Document Number: 007-3720-002*
ECMA. (1966). Standard ECMA-4 *Flow Charts*. Geneva, Switzerland: ECMA.
Egenhofer, M. J. (1995). Topological relations in 3D: Technical report, University of Maine, Orono, USA.
Fagin, R., Kolaitis, P. G., Miller, R. J., & Popa, L. (2005). Data exchange: semantics and query answering. *Theoretical Computer Science, 336*(1), 89-124.
Fane, B. (2013). *AutoCAD 2014 for dummies*. Hoboken, NJ: John Wiley & Sons.
Farin, G. (1982). Visually C 2 cubic splines. *Computer-Aided Design, 14*(3), 137-139.
Finkelstein, E. (2014). *AutoCAD 2015 and AutoCAD LT 2015 Bible* (pp. 1299).
Fotheringham, S., & Rogerson, P. (2013). *Spatial analysis and GIS*: CRC Press.
Gellish. (n.d.). System Requirements for using Gellish. Retrieved May 18th, 2015, from http://www.gellish.net/index.php/knowledge-base/12-knowledge-base/16-faq-2.html
Goncalves, R., & Kersten, M. (2011). The data cyclotoon query processing scheme. *ACM Transactions n Database Systems (TODS), 36*(4).
Grasshopper. (2010). Grasshopper discussion forum: weaire phelan using galapagos. Retrieved May 20th, 2015, from http://www.grasshopper3d.com/forum/topics/weaire-phelan-using-galapagos
Groover, M., & Zimmers, E. (1983). *CAD/CAM: computer-aided design and manufacturing*: Pearson Education.
Guerrero Iñiguez, J. (2012). *Three-dimensional reconstruction of underground utilities for real-time visualization.* (MSc. Thesis), TU Delft, Delft University of Technology.
Hajkowicz, S., & Collins, K. (2007). A review of multiple criteria analysis for water resource planning and management. *Water Resources Management, 21*(9), 1553-1566.
Hajkowicz, S., & Higgins, A. (2008). A comparison of multiple criteria analysis techniques for water resource management. *European journal of operational research, 184*(1), 255-265.
Hart, G. (1997). Calculating Canonical Polyhedra. Retrieved 22th April, 2015, from http://www.georgehart.com/canonical/canonical-supplement.html
Hassan, M., Ahmad-Nasruddin, M., Yaakop, I., & Abdul-Rahman, A. (2008). An integrated 3D cadastre–Malaysia as an example. *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences, 37*(B4), 121-126.
Hernandez, C. R. B. (2006). Thinking parametric design: introducing parametric Gaudi. *Design Studies, 27*(3), 309-324.
Hijazi, I. (2011). *Integrated management of indoor and outdoor utilities by utilizing BIM and 3DGIS.* (Doctoral dissertion), University of Osnabrueck.
Hoff III, K. E., Keyser, J., Lin, M., Manocha, D., & Culver, T. (1999). *Fast computation of generalized Voronoi diagrams using graphics hardware.* Paper presented at the Proceedings of the 26th annual conference on Computer graphics and interactive techniques.
I-Chen, W., & Shang-Hsien, H. (2007). Trandormation from IFC data model fot GML data model: Methodology and tool development. *Journal of the Chinese Institute of Engineers, 30*(6), 1085-1090.
Intellidimension. (n.d.). Semantics.Server 2.0. Retrieved May 18th, 2015, from http://www.intellidimension.com/products/semantics-server/
Jansen, E., Van Oosterom, P., & Stoter, J. (2004). Het overbruggen van de kloof tussen CAD en GIS (deel 2). *Geo-info, 2004*(12), 518-523.
Kaklauskas, A., Zavadskas, E. K., & Raslanas, S. (2005). Multivariant design and multiple criteria analysis of building refurbishments. *Energy and Buildings, 37*(4), 361-372.
Karimi, H. A., & Akinci, B. (2009). *CAD and GIS integration*: CRC Press.
Kaufman, A. (1989). *3D voxel-based graphics*. New York: Pergamon.

Kazar, B. M., Kothuri, R., van Oosterom, P., & Ravada, S. (2008). On valid and invalid three-dimensional geometries *Advances in 3D geoinformation systems* (pp. 19-46): Springer.

Kersting, O., & Döllner, J. (2002). *Interactive 3D visualization of vector data in GIS.* Paper presented at the Proceedings of the 10th ACM international symposium on Advances in geographic information systems.

Khuan, C. T., Abdul-Rahman, A., & Zlatanova, S. (2008). 3D solids and their management in DBMS *Advances in 3D geoinformation systems* (pp. 279-311): Springer.

Kolbe, T., & Plumer, L. (2004). Bridging the Gap between GIS and CAAD Geometry, Referencing, Representations, Standards and Semantic Modelling. *GIM international, 18*, 12-38.

Laidlaw, D. H., Trumbore, W. B., & Hughes, J. F. (1986). *Constructive solid geometry for polyhedral objects.* Paper presented at the ACM SIGGRAPH computer graphics.

Laine, S. (2013). *A topological approach to voxelization.* Paper presented at the Computer Graphics Forum.

Lee, J.-Y., & Koh, J.-H. (2007). A conceptual Data Model for a 3D Cadastre in Korea. *Journal of the Korean Society of Surveying, Geodesy, Photogrammetry and Cartography, 25*(6_1), 565-574.

Leonova, M. (2014). LOD - Development or Detail & Why it Matters. Retrieved May 29th, 2015, from http://lanmarservices.com/2014/05/14/lod-in-scan-to-bim/

Mederos, B., Velho, L., & De Figueiredo, L. (n.d.). *Moving Least Squares Multiresolution Surface Approximation*. IMPA–Instituto de Matematica Pura e Aplicada. Unpublished.

Microsoft. (n.d.-a). BULK INSERT (Transac SQL). Retrieved May 26th, 2015, from https://msdn.microsoft.com/en-us/library/ms188365.aspx

Microsoft. (n.d.-b). Import Data from a Data Feed. Retrieved May 18th, 2015, from https://msdn.microsoft.com/en-us/library/gg413490(v=sql.110).aspx

Microsoft. (n.d.-c). Relational Database Components. Retrieved April 24th, 2015, from https://technet.microsoft.com/en-us/library/aa174501(v=sql.80).aspx

Microsoft. (n.d.-d). Spatial Data (SQL Server). Retrieved April 24th, 2015, from https://msdn.microsoft.com/en-us/library/bb933790.aspx

Microsoft. (n.d.-e). XML Data (SQL Server). Retrieved May 18th, 2015, from https://msdn.microsoft.com/en-us/bb522446.aspx

Minh Duc, P. (2013). Self-organizing Structured RDF in MonetDB.

Mitra, N. J., Nguyen, A., & Guibas, L. (2004). Estimating surface normals in noisy point cloud data. *International Journal of Computational Geometry & Applications, 14*(04n05), 261-276.

Monedero, J. (2000). Parametric design: a review and some experiences. *Automation in Construction, 9*(4), 369-377.

MonetDB. (n.d.-a). Column store feature. Retrieved April 24th, 2015, from https://www.monetdb.org/Home/Features

MonetDB. (n.d.-b). Extending MonetDB with support for 3D Point Cloud and Voxels. Retrieved April 29th, 2015, from https://www.monetdbsolutions.com/extending-monetdb-support-3d-point-cloud-and-voxels

MonetDB. (n.d.-c). GeoSpatial. Retrieved April 24th, 2015, from https://www.monetdb.org/Documentation/Extensions/GIS

MonetDB. (n.d.-d). *MonetDB SQL Reference Manual*

Mossman. (2013). Videogames based in voxels. Retrieved April 22th, 2015, from http://mossman.es/videogames-based-in-voxels/

Movafagh, S. (1995). GIS/CAD Convergence Enhances Mapping Applications. *GIS World, 8*(5), 44-47.

MySQL. (2015). *MySQL 5.7 Reference Manual*

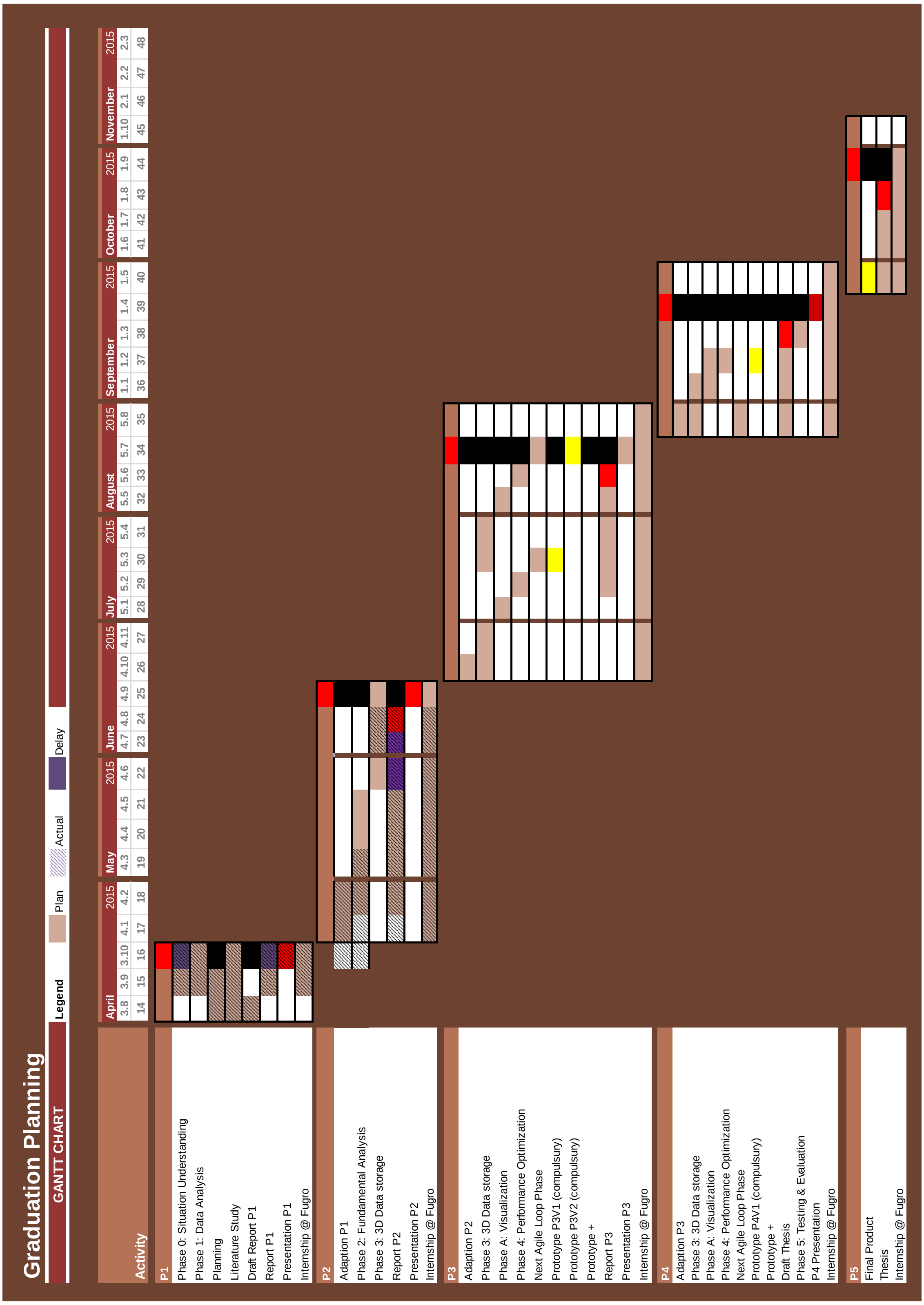
Nagel, C., Stadler, A., & Kolbe, T. (2009). Conceptual requirements for the automatic reconstruction of building information models from uninterpreted 3D models. *Proceedings of the International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences*, 46-53.

Newell, R. G., & Sancha, T. L. (1990). The difference between CAD and GIS. *Computer-Aided Design, 22*(3), 131-135.

Object-e. (2009). Voronoi 3D v0.1.ms. Retrieved May 13th, 2015, from http://object-e.net/tools/voronoi-3d-v0-1-ms
OGC. (1999). Consortium Inc: OpenGIS Simple Feature Specification For SQL Version 1.1. *Open GIS project document*, 99-049.
Peng, J., & Kuo, C.-C. J. (2005). *Geometry-guided progressive lossless 3D mesh coding with octree (OT) decomposition.* Paper presented at the ACM Transactions on Graphics (TOG).
Penninga, F., & Van Oosterom, P. (2008). A simplicial complex-based DBMS approach to 3D topographic data modelling. *International Journal of Geographical Information Science, 22*(7), 751-779.
Pharr, M., & Fernando, R. (2005). *Gpu gems 2: programming techniques for high-performance graphics and general-purpose computation*: Addison-Wesley Professional.
Piegl, L. (1991). On NURBS: a survey. *IEEE Computer Graphics and Applications, 11*(1), 55-71.
PostgreSQL. (n.d.). *PostgreSQL 9.4.1 Documentation*
Pu, S. (2005). *Managing freeform curves and surfaces in a spatial DBMS.* (MSc. Thesis), TU Delft, Delft University of Technology.
Ramakrishnnan, R. (2003). *Database management systems* (3rd ed.). Boston, USA: McGraw Hill Publisher.
Rogers, D. (2001). *An introduction to NURBS with historical perspective*. San Francisco: Morgan Kaufmann.
Rossignac, J. (2002). *CSG-BRep duality and compression.* Paper presented at the Proceedings of the seventh ACM symposium on Solid modeling and applications.
SFCGAL. (2013). SFCGAL::Solid Class Reference. Retrieved April 29th, 2015, from http://oslandia.github.io/SFCGAL/doxygen/class_s_f_c_g_a_l_1_1_solid.html
Stoter, J., & Salzmann, M. (2003). Towards a 3D cadastre: where do cadastral needs and technical possibilities meet? *Computers, environment and urban systems, 27*(4), 395-410.
Strobl, C. (2008). *Dimensionally Extended Nine-Intersection Model (DE-9IM)*: Springer US.
Stroud, I. (2006). *Boundary Representation Modelling Techniques*. London: Springer.
Teklemburg, J., Timmermans, H., & Borges, A. (1997). Design tools in a integrated CAD-GIS environment: space syntax as an example. *Decision support systems in urban planning. London: E & FN Spon*, 261-276.
The PostGIS Development Group. (n.d.). *PostGIS 2.1.8dev Manual SVN Revision (13470)* Retrieved from http://postgis.net/docs/manual-2.1/index.html
Van Lanen, H. A. J., Bregt, A. K., Randen, Y. v., & Hoosbeek, M. R. (1989). *Use of GIS, EC soil map, and land evaluation procedures to explore crop growth potential*. Wageningen: Winand Staring Centre.
Van Oosterom, P. (1985). *Visualiseren van aerodynamische gegevens (in Dutch).* (MSc. Thesis), TU Delft, Delft University of Technology.
Van Oosterom, P., Martinez-Rubi, O., Ivanova, M., Horhammer, M., Geringer, D., Ravada, S., . . . Gonçalves, R. (2015). Massive point cloud data management: design, implementation and execution of a point cloud benchmark. *Computers & Graphics*.
Van Oosterom, P., Stoter, J., & Jansen, E. (2005). 1 Bridging the Worlds of CAD and GIS. *Large-scale 3D data integration: challenges and opportunities*, 9-36.
W3C. (2003). RDF SQL HOWTO. Retrieved May 18th, 2015, from http://www.w3.org/1999/02/26-modules/User/RdfSQL-HOWTO.html
Walles, E. (2011). Csg.js. Retrieved 10th April, 2015, from https://github.com/evanw/csg.js/
Wang, Y., & Wang, S. (2010). *Research and implementation on spatial data storage and operation based on Hadoop platform.* Paper presented at the Geoscience and Remote Sensing (IITA-GRS), 2010 Second IITA International Conference on.
Wesselingh, S. (2007). *Visualization of a TEN (Tetrahedral Irregular Network) in a web client.* (MSc. Thesis), TU Delft, Delft University of Technology.
Worboys, M. (1997). GIS: A Computing Perspective. herdruk. *Londen: Taylor & Francis*.

Xu, D. (2011). *Design and Implementation of Constraints for 3D Spatial Database: Using Climate City Campus Database as an Example* (MSc. Thesis), TU Delft, Delft University of Technology.
Zlatanova, S. (2000). *3D GIS for urban development.* (PhD. Thesis), ITC, The Netherlands.
Zlatanova, S. (2006). 3D geometries in spatial DBMS *Innovations in 3D geo information systems* (pp. 1-14): Springer.

A. Appendix: Gantt-chart



B. Appendix: Flowchart Symbols

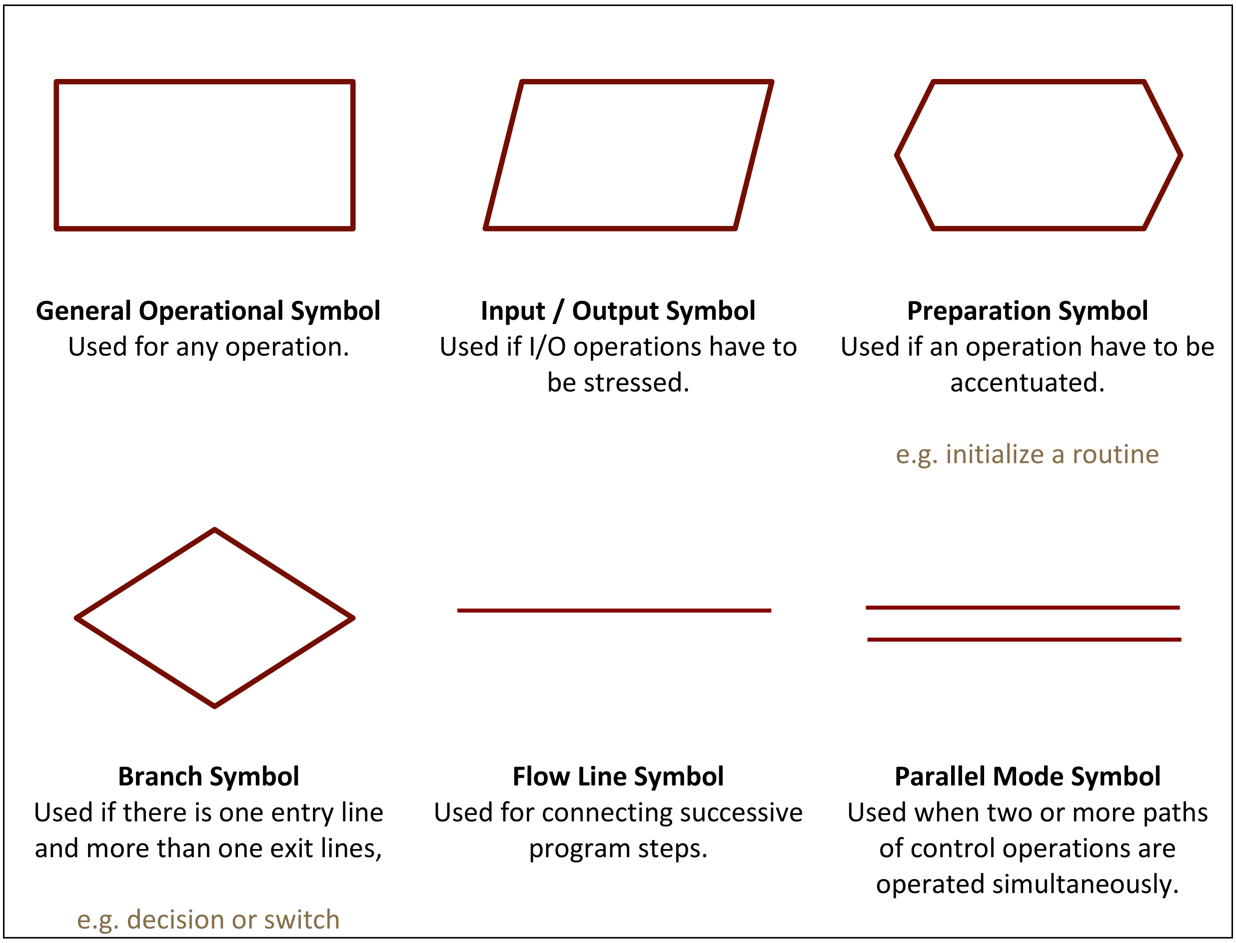


Figure B.1: Program flowchart symbols, based on ECMA (1966) confirm ECMA-4 standard

C. Appendix: UML Diagram Fugro DBMS