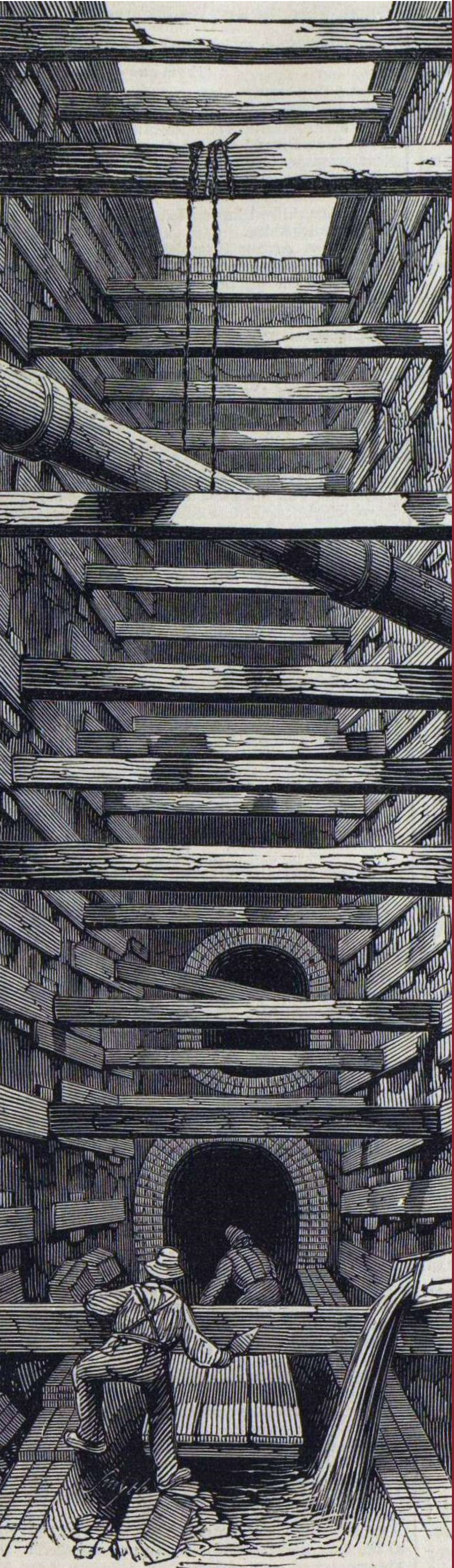


MASTER THESIS

THE OPERATION & MAINTENANCE DOMAIN ONTOLOGY FOR UTILITY NETWORKS

DATABASE ENCODING

Federico Fossatti

FACULTY OF ENGINEERING TECHNOLOGY
DEPARTMENT OF CONSTRUCTION MANAGEMENT & ENGINEERING
MSc CIVIL ENGINEERING AND MANAGEMENT

GRADUATION COMITEE
Prof.dr.ir. A.G. Dorée (University of Twente)
Dr.ir. L.L. olde Scholtenhuis (University of Twente)
PDEng. R.B.A. ter Huurne (University of Twente)
Dr. G. Agugiaro (TU Delft)

JUNE 2020

ENSCHEDE, THE NETHERLANDS

UNIVERSITY OF TWENTE.

UNIVERSITY OF TWENTE.

THE OPERATION & MAINTENANCE DOMAIN ONTOLOGY FOR UTILITY NETWORKS

DATABASE ENCODING

A thesis submitted to the University of Twente in partial fulfilment of the requirements for the degree of

Master's in Science in Civil Engineering and Management

by

Federico Fossatti

June 2020

Enschede, The Netherlands

Summary

In the field of asset management, data modelling is used to design data structures which allow to digitally represent the desired real-world objects, concepts, and the relations between them. This, in turn, supports more structured and analytic decision making in asset management. The Operation & Maintenance Domain Ontology (O&M-model) is such a conceptual data model, and it was originally developed by Ter Huurne (2019). It allows representing utility networks from the perspective of their operation & maintenance, integrating concepts from their whole lifecycle. The O&M-model is a revised version of the Utility Network Application Domain Extension (ADE) of CityGML, an extension of the model for representing utilities. CityGML is, in turn, a digital model to represent cities, including a variety of above-ground city objects, their appearance, geometry, topology, and semantic information (e.g. the number of occupants, age, etc.).

The O&M-model can be used for two main purposes:

- To derive a so-called data transfer format schema, to allow exchanging O&M information between a source and a target system (e.g. a utility owner and a contractor).
- To derive a database schema, that can be used to set up a spatial database. Spatial databases are the most common solution for storage, manipulation, analysis, and presentation of asset data for organizations dealing with assets occupying a large surface (such as utility networks and cities).

The derivation of the database schema of the O&M-model is the focus of this study. *My goal was to create a prototype database that could be used by the department of Campus & Facility Management (C&FM) of the University of Twente to manage their utility networks*. To design it I followed the iterative design cycle proposed by Wieringa (2010). It begins with the problem investigation phase during which I explore the stakeholders' challenges, goals, and the phenomena that define the problem. This phase is followed by a design phase during which I design an artefact to treat the problems and ends with a validation phase to check whether the design meets the stakeholder's requirements.

During the problem investigation phase, I researched the most recent developments in data models for utility networks, data modelling for the design of data schemas, spatial databases as an asset management tool, and the role of encoding data and models in data modelling. I also investigated a non-technological aspect of the problem, namely the stakeholders of the project and their goals, related to data-driven asset management. The findings from the problem analysis were:

- Some stakeholders are interested in improving the way they carry out operation & maintenance by using both the spatial and non-spatial asset data in their decision making. Thus, they need a Geographic Information System (GIS) component that can represent and handle these data. Other stakeholders are interested in the use of the O&M-model as the means to enable interoperable data exchanges.
- A spatial-relational database is a solution that addresses both problems. It is the industry standard for asset management, and can also be used to test the O&M-model in terms of comprehensiveness.
- Deriving such a database implies performing two model transformations, beginning from the O&M-model: conceptual (i.e. the O&M-model) to logical model transformation, and logical to physical model transformation. The physical model can be used to set up a structured database.

The following phase was the design of the treatment. I transformed the goals of the stakeholders into requirements for the database solution. These requirements were the basis for the technical choices that followed. Before committing to the design of an entirely new artefact (i.e. database) I reviewed existing ones

to see if they could be repurposed. I found an existing database implementation of the Utility Network ADE, and later reused it and adapted it to accommodate the O&M-model. I chose as a platform the free database software PostgreSQL/PostGIS. PostGIS is used by C&FM to manage two databases, containing utilities and topographical features of the campus of the university.

Successively, I reviewed the practical documentation of the database implementations of CityGML and the Energy ADE —an Application Domain Extension meant for modelling energy use in cities— to compose a list of rules to be used in the transformations of the O&M-model into a database. I then adapted the original O&M-model to match more closely the structure of the CityGML model by simplifying certain relations and classes. The final step in the design was to use the transformation rules to derive the computer scripts for the installation of the database.

The final phase was the validation of the design. After installing a prototype database, the validation required two preliminary steps. First, I obtained a dataset containing 8 utility networks of the campus of the University of Twente, as well as the streets' footprints. I transformed this dataset to match the physical O&M-model and uploaded it to the database. After adding the data from the campus dataset, I enriched it with dummy attributes to compensate for the lack of data in some categories.

As a test case, I then simulated a street reconstruction at campus containing three phases. I associated each phase with several information requirements. I linked the information requirements to the "competency questions" for which the O&M-model was designed by Ter Huurne and then I formalized these into computer queries. Finally, I connected the database to a GIS application able to show maps based on the output of the queries and to have graphical as well as tabular results.

The validation method I used was a single-case mechanism experiment. I used the computer queries as stimuli on a validation model consisting of the database connected to the GIS application, the data from campus and the street reconstruction task. I showed the use of the model and the results to groups of experts for their assessment.

The database fulfilled the requirement to serve as an asset management support tool. It greatly expands the representation capabilities of the current database used at the campus, and the test showed that it can support routine engineering tasks by providing detailed information on the assets such as location, identification, depth, maintenance and performance history, location of relevant valves, etc. An expert consulted during the validation phase commented on the usefulness of the artefact for planning network extensions, which require 3D clash detections. And an expert in the surveying of gas pipes said that the database could also support the planning and execution of inspection campaigns for leak detection, because of the representation capabilities of the model.

However, the validation of the O&M-model as a means to enable standardized data exchanges still needs to be tested. Even though the database allows in theory to test the comprehensiveness of its underlying O&M-model, a real exchange of information between, for example, a network owner and its contractor should be simulated and tested. This will allow to directly assess the usefulness of the O&M-model as a solution for interoperability, and also to further refine its classes with real operation & maintenance data.

The contribution of this study to data-driven asset management is by showing the information requirements of a utility network owner and a proposed technical solution consisting of a lifecycle-oriented spatial-relational database connected to a GIS application. The practical contribution to the department of Campus & Facility Management of the University of Twente is by providing them with an improved database compared to their current one, with the caveat that it requires more expertise to use. The database also allows organizations such as C&FM to better determine which asset data to collect during surveys and new construction.

Recommendations

The full potential of the database as a support tool can only be exploited if sufficient data is stored within it. With enough data, the database could be used to automatize maintenance decisions by predicting component condition. Thus, organizations should prioritize the collection of data, whenever such possibility presents itself (e.g. during new construction, maintenance, inspections, etc.) and using the classes in the model as a guide. This is however costly, time-consuming and a long-term investment.

The partial implementation of the system (i.e. O&M-database connected to a GIS application for visualization) requires limited expertise in geomatics, programming, and civil engineering. The knowledge requirements for partial implementation and continued testing and improvement of the model are not a high entry barrier. However, the full implementation of the system —including the technical set-up of its web interfaces, database management, set-up of procedures for data update and quality checks, etc.— requires a substantially deeper knowledge of the aforementioned disciplines. This might make the full system too expensive and impractical to implement in-house for organizations such as C&FM with relatively small utility networks, limiting the choice to outsourcing the technology-related tasks.

Testing of the data transfer format schema derived from the O&M-model is still necessary. This will require setting up the technical transformation processes for mapping and encoding the data from a source system to the data transfer format schema and finally to the target system. Further testing of the O&M-model as a means to facilitate data exchange will likely uncover room for improvement in the O&M-model and, by extension, in the database schema.

Finally, the use of the system to map risks and impacts related to utility networks remains untested. Future studies could focus on these aspects of the O&M-model and derive the indicators that are necessary to measure the social, environmental, and economic impacts related to a network and its components.

Acknowledgements

This report and its accompanying products —a GitHub repository that allows to recreate the project, and a paper that I have co-authored and submitted to the 3D GeoInfo 2020 conference— are the fruit of seven months of work and my education at the University of Twente. During this time, I got familiarized with a new perspective on civil engineering other than design, namely management supported by digital tools. I thank my professors for their lessons in this area of engineering.

I would like to thank dr.ir. Léon olde Scholtenhuis for his interest in the topic I chose, and for his countless hours of guidance on, among others, the methodological and management aspects of this research, and also for helping me to quickly restart with this topic after changing projects. Furthermore, I would like to thank prof.dr.ir. André Dorée for agreeing to supervise me and for his advice in the relevant management themes to explore. Special thanks to dr. Giorgio Agugiaro for his knowledgeable guidance on the pandora's box of technologies that I opened by choosing this topic. Your clear explanations and previous work on the Utility Network ADE were invaluable to my research. The final thanks to my supervisory team go to PDEng Ramon ter Huurne, who guided me in choosing and shaping the topic to its final form. Thank you for your time, the fruitful discussions on the ontology, and your help with the validation and contact with experts.

I also thank all of the experts that helped me during the different phases of the research. In particular, ing. Ray Klumpert and André de Brouwer for their time, their input for the validation test case and for trusting me with access to the databases of the campus of the university. Many thanks to Robin Kok of De Landmeetdienst for his help with the data access and his explanation of the current system setup. I am very grateful to the representatives of SIERS Infraconsult, Geonovum, the Kadaster, and to Dr Tatjana Kutzner of TU Munich who helped with the validation by providing advice and feedback on the tests, and on data modelling and the particulars of asset management in practice.

My thanks to my fellow students Henrik, Jeffrey, Ruben and Rob for the continued discussions we had on our projects after parting ways, they did wonders for keeping me motivated and on schedule.

Last but not least, endless thanks to Sonya and my family for their loving support through my studies.

COLOPHON

ORGANIZATION
University of Twente
Faculty of Engineering Technology
Department of Construction Management & Engineering
Drienerlolaan 5, 7522 NB Enschede, The Netherlands

DATE
2020-06-19

VERSION
1.0

STATUS
Submitted

PROJECT
Master's Thesis Civil Engineering & Management

TITLE
The Operation & Maintenance Domain Ontology for Utility Networks

SUBTITLE
Database Encoding

SUPERVISORS

Prof.dr.ir. André Dorée	University of Twente	Thesis supervisor
Dr.ir. Léon olde Scholtenhuis	University of Twente	Daily supervisor
PDEng Ramon ter Huurne	University of Twente	Daily supervisor
Dr. Giorgio Agugiaro	TU Delft	Daily supervisor

AUTHOR
Federico Fossatti

STUDENT NUMER
s2081202

E-MAIL
f.fossatti@student.utwente.nl
federicofossatti@gmail.com

WEBSITE
https://www.utwente.nl/en/cem/

DOCUMENT NAME
Fossatti_CEM_Master's_Thesis

DOCUMENT MANAGEMENT

Version	Date	Section	Alterations
0.1	15/01/2020		Document start
0.1a	20/03/2020	1,2,3,5 & 6	Incomplete draft sent to LoS.
0.2	23/04/2020	1-3	New draft with a different structure
0.3	20/05/2020	1-3	Re-written based on comments by LoS
0.4	1/06/2020	1	Moved problem investigation to 1st chapter as suggested by LoS
		3	Added Treatment Design
		4	Added Treatment Validation
		5	Added final chapter
1.0	19/06/2020	1-5	Addressed comments by RtH, LoS, and GA. Submitted.

CONTENTS

LIST OF FIGURES

LIST OF TABLES

TERMS AND DEFINITIONS

3DCityDB: "[…] a free geodatabase to store, represent, and manage virtual 3D city models on top of a standard spatial relational database. The 3D City Database schema implements the CityGML standard with semantically rich and multi-scale urban objects facilitating complex analysis tasks, far beyond visualization". "The database schema results from a mapping of the object-oriented data model of CityGML 2.0 to the relational structure of a spatially enhanced relational database management system" (3DCityDB, n.d.).

CityGML: "CityGML is an open data model and XML-based format for the storage and exchange of virtual 3D city models. It is an application schema for the Geography Markup Language version 3.1.1 (GML3), the extendible international standard for spatial data exchange issued by the Open Geospatial Consortium (OGC) and the ISO TC211" (OGC, n.d.-a).

Data model: A linguistic description that represents a universe of discourse in the form of text, pictures, and drawings (Kutzner, 2016, pp. 5–8). It can be conceptualized at different levels of abstraction.

- **Informal Conceptual Model**: "a model that defines concepts of a universe of discourse" [ISO 19101]. The definition occurs in a process of abstraction and simplification of the universe of discourse and uses natural languages, such as English.
- **Formal Conceptual Model (or application/conceptual schema)**: "formal description of a [informal] conceptual model … for data required by one or more applications" [ISO 19101]. They are platform-independent, containing no information about their physical implementation (Kutzner, 2016, pp. 14–16). Uses computer interpretable languages, such as UML.
- **Logical Model (or implementation schema)**: description of the computable formal structure (the logic) of a data format (Nyerges, 2010, pp. 3–5), thus making it platform-specific. Uses computer interpretable languages, such as UML. For example, relational, XML or object-oriented logic.
- **Physical Model (or data format schema)**: results from the translation of the logical model into a programming language (script/code) corresponding to specific software and operating system (Nyerges, 2010, pp. 3–6).

Extensible Markup Language (XML): "A simple, very flexible text-based Markup language derived from SGML (ISO 8879). It defines a set of rules for encoding documents in a format that is both human-readable and machine-readable" (W3C, 2006).

Encoding (noun): used interchangeably with 'data format schema' (i.e. GML encoding = GML data transfer format).

Encoding (verb): "conversion of data into a series of codes". Transformation of system-dependent data structures (compliant to a given application schema) to produce system-independent data structures suitable for transfer and storage [ISO 19118].

Feature: A term used within the Geographic Markup Language (GML) to refer to real-world objects, which can be concrete or abstract (Lake et al., 2004, p. 3).

Geographic Information System (GIS): "A computer-based system that provides the following sets of capabilities to handle georeferenced data: (1) data capture and preparation, (2) data management, including storage and maintenance, (3) data manipulation and analysis, and (4) data presentation" (Huisman & de By, 2009, p. 32). An example of such a system is the free and open-source Quantum GIS or QGIS (QGIS, n.d.).

Geography Markup Language (GML): "…an XML grammar for expressing geographical features. GML serves as a modelling language for geographic systems as well as an open interchange format for geographic transactions on the Internet" (OGC, n.d.-b).

Infrastructure Asset Management: "Making data-driven infrastructure investment decisions so that life cycle costs are minimized while satisfying performance, risk, tolerance, budget, and other operational requirements" (Brown, 2010, p. 305). Though historically merged into one role, the following functions are becoming increasingly separated, even being carried out by different organizations (Brown & Humphrey, 2005; van der Velde et al., 2013):

- **Asset manager:** Party responsible for translating the criteria of the asset owner into an asset plan (planning of O&M, budgeting, evaluating alternatives, etc.).
- **Asset owner**: Party responsible for strategic thinking, including setting financial, technical and risk criteria.
- **Service provider(s):** Parties responsible for the execution of the asset plan and providing feedback on actual costs and performance to the asset manager (execution of O&M, procurement, construction, inspection & monitoring, etc.).

Model Driven Architecture (MDA): ''…is an approach to software design, development, and implementation. MDA separates business and application logic from underlying platform technology'' (OMG, n.d.). It can be applied to geospatial data modelling, to help structure the translation from high to low levels of abstraction.

- **Computer Independent Model (CIM)**: "A CIM is also often referred to as a business or domain model because it uses a vocabulary that is familiar to the subject matter experts. It presents exactly what the system is expected to do but hides all information technology-related specifications to remain independent of how that system will be (or currently is) implemented" (Truyen, 2006). Corresponds to the informal conceptual model.
- **Platform Independent Model (PIM)**: "A PIM exhibits a sufficient degree of independence so as to enable its mapping to one or more platforms. This is commonly achieved by defining a set of services in a way that abstracts out technical details" (Truyen, 2006). Other models then specify a realization of these services in a platform-specific manner. Corresponds to the formal conceptual model.
- **Platforms Specific Model (PSM)**: "A PSM combines the specifications in the PIM with the details required to stipulate how a system uses a particular type of platform. If the PSM does not include all of the details necessary to produce an implementation of that platform it is considered abstract (meaning that it relies on other explicit or implicit models which do contain the necessary details)" (Truyen, 2006). Corresponds to the logical model.
- **Platform Model (PM)**: Source code of the implemented system. Corresponds to the physical model.

Operation & Maintenance (O&M): It refers to both (1) the execution of all the day-to-day activities necessary for the reliable delivery of a commodity through underground utilities while ensuring that the assets remain serviceable by means of maintenance (Sohail et al., 2005), and (2) the planning of such activities (Brown & Humphrey, 2005).

Relational database: "A relational database is a collection of data organized into a table structure. [..] Within the table structure, the rows are called 'records' or 'tuples' and the columns are called 'attributes'. The structure allows users to identify and access data in relation to another piece of data in the table, or other tables within the database. Tables can be modified, or rows and columns can be added or removed without affecting the rest of the database" (IBM, n.d.). Relational databases can be expanded to support (store and query) spatial objects with complex geometries.

Relational Database Management System (RDBMS): "[…] the software that gives users the ability to update, query and administer a relational database. Structured Query Language (SQL) is typically the standard programming language used to access the database" (IBM, n.d.). An example of such a system is the free and open-source PostgreSQL, which can be spatially enabled using PostGIS (PostGIS Development Team, n.d.; PostgreSQL Development Team, n.d.).

Stereotype: The means by which the basic UML elements can be adapted to specific domains [ISO-TC211].

FeatureType: Classes with this stereotype represent geographic objects.

dataType: Properties without identity, which cannot exist as single instances.

enumeration: Fixed lists of possible values an attribute can take.

CodeList: Extendable lists of possible values an attribute can take.

Transfer format: structured representation of data in a file for transfer between systems (INSPIRE, 2013).

Unified Modelling Language (UML): A general-purpose modelling language that "…helps you specify, visualize, and document models of software systems, including their structure and design" (OMG, 2005).

Universe of discourse: a view of the real or hypothetical world that includes everything of interest' [ISO 19101]

ACRONYMS

ADE: Application Domain Extension

C&FM: Campus & Facility Management

DB: Database

DDL: Data Definition Language

DML: Data Manipulation Language

GIS: Geographic Information System

GML: Geographic Markup Language

M&RE: Maintenance & Real Estate

O&M: Operation & Maintenance

QGIS: Quantum GIS

RDBMS: Relational Database Management System

SQL: Structured Query Language

UML: Unified Modelling Language

XML: eXtensible Markup Language

XSD: XML Schema Definition

Chapter 1

PROBLEM INVESTIGATION, PROBLEM STATEMENT, AND GOALS

In this chapter I first provide the project's background by explaining the efforts of standardization in the field of modelling utility networks, explaining the scope of existing data models, and introducing the O&M Domain Ontology as a possible solution. Second, I analyse the O&M Domain Ontology from the perspective of data modelling. The goal of data modelling is deriving a data structure (or schema) for an application. Thus, two possible applications for the O&M Domain Ontology are explained: as a data transfer format (a solution for interoperability issues), and as a stand-alone system for asset management. Third, I present a possible format for a stand-alone asset management system: spatial-relational databases. Fourth, I explore the role of encoding in data modelling and explain the differences between encoding data and encoding models (such as the ontology). Fifth, after finishing with the technological side of the problem investigation I investigate the stakeholders of the project. Finally, I combine and analyse the knowledge from the previous sections and provide the project's problem statement, goals, objectives, and scope, as well as an explanation on the role of this document and its outline.

1.1. Introduction to the project

The purpose of the 'lifecycle asset management' approach for utility networks is making data-driven investment decisions to minimize lifecycle costs while maintaining organizational and operational requirements. An example of such a decision is to replace, keep or maintain existing pipes in a network based on their maintenance and performance history. These decisions are supported by digital systems such as Geographic Information Systems (GIS). GIS provides the necessary tools to digitally store, manipulate, analyse, and present infrastructure data. However, two interrelated issues present barriers for the GIS-based lifecycle asset management of utilities.

First: interoperability. The international community lacks a standard for the digital representation, storage, and exchange of lifecycle-oriented utility network data. Instead, the fragmented utility sector often relies on organization-specific data models tailored for a single utility for supporting their management decisions. This is because conceptualizations of reality and terms often differ between organisations and between the various types of utility networks. Without the use of such a standard, practitioners often must engage in semantics transformations of their datasets during information exchanges between their own and other organizations. Moreover, data loss has historically occurred due to the lack of comprehensive representation capabilities of information systems, and handover problems due to changing ownership of privatized networks.

Second: the purpose and scope of existing data models. Two developments in the domain of data modelling and standardization have occurred recently, which motivated this research:

- In the Netherlands, the INSPIRE-based IMKL data model (Informatie Model Kabels en Leidingen, 1.2.) was made compulsory for all asset owners when sharing information with the national Cadastre. This data model is a Dutch specific extension of the European INSPIRE Utility Services data specification. It is aimed at reducing excavation damage and reducing the costs of laying new buried infrastructure. In its current version, it does not, however, provide the detailed data representation requirements of lifecycle asset management.
- The second development is that of the utilities data model Utility Network ADE, an Application Domain Extension of CityGML for the domain of utilities. It has been tested in terms of its ability to support applications such as the integrated management of above-ground city objects and their related buried utility networks, as well as the simulation of cascading effects in interdependent networks. However, the support this model offers for lifecycle asset management use-cases is not comprehensive. The model lacks several concepts that are key for the operation and maintenance of utility networks. Maintenance history, performance, related parties, costs, impacts, surrounding soil and groundwater level, for example, are not representable using the base Utility Network ADE 0.9.2, even though they are important for asset owners, managers, and the contractors who execute the work orders.

The Operation and Maintenance Domain Ontology (O&M-model)[1] was designed to both improve the shortcomings in the Utility Network ADE and to address the issue of interoperability. Essentially, the model is an extension of the free and open-source Utility Network ADE data model v0.9.2 (and thus it is an Application Domain Extension of CityGML). The O&M-model keeps the complex topological representation capabilities of the underlying models and adds classes and attributes related to lifecycle asset management. It was designed in collaboration with the department of Campus & Facility Management (C&FM) of the University of Twente, among other stakeholders. C&FM seeks to improve how maintenance of its utilities is supported digitally and for this task, it needs the means to store, manipulate, analyse, and present data. The O&M-model is yet to be tested in practice.

Above, I have introduced two issues with data models and asset data: first interoperability, and next the lack sufficient scope to deal with lifecycle asset management. In this study, I mainly address the issue of lack of model scope by using a comprehensive data model as a starting point for the design of a digital support system for the management of utility networks. I also indirectly address the issue of interoperability by using the support system to test the comprehensiveness of its underlying data model. Before commencing with the problem investigation, I now provide an advance of the object of this study, to facilitate the reading of the remainder of this chapter. This is further elaborated in **Section 1.7**.

The present study aims to design an improved asset management system for C&FM that is compliant to the O&M-model. The system should use a data structure derived from the O&M-model and optimized for data storage, manipulation, analysis, and presentation. The resulting artefact is meant to be used by practitioners —such as C&FM— as a stand-alone GIS tool for supporting O&M.

I also use it to indirectly test how comprehensive the underlying O&M-model is, and by transitivity, to test the comprehensiveness of the O&M-model as a data transfer format to improve interoperability.

[1] The idea to start this project came from reading the future research directions proposed by Ramon ter Huurne in his PDEng project (ter Huurne, 2019, p. 82). He designed an ontology for the domain of operation & maintenance (O&M) of utility networks, the O&M Domain Ontology. A detailed discussion for the need for such an ontology can be found in the project's report (ter Huurne, 2019, pp. 15–26).

Sections 1.2, 1.3, and 1.4 contain the technological introduction to this study, commencing with data modelling. The O&M-model was formally modelled at the conceptual level using a Unified Modelling Language (UML) class diagram. Further, its documentation describes and defines the classes, attributes, and relations in the model. Conceptual models are meant for representing data, but not for storage or exchange. They are used to develop more concrete models —called physical models— that are meant for providing the structure of data to be stored in a specific format, as explained next.

1.2. Data modelling

In this text, data modelling refers to the process of achieving a simplified linguistical representation (i.e. abstraction and conceptualization) of real-world phenomena and domain expertise, from a certain perspective, with a purpose, and using a data modelling (schema) language (Kutzner, 2016; Nyerges, 2010). A data model is, thus, a description of a problem space, also called the universe of discourse. Data models and their respective schema languages exist at different **levels of modelling abstraction**, traditionally called conceptual (high-level), logical (mid-level) and physical (low-level), as shown in FIGURE 1-1. Data models at all levels of abstraction can be thought of as ontologies, as they store domain-specific knowledge about an application (Kutzner, 2016).

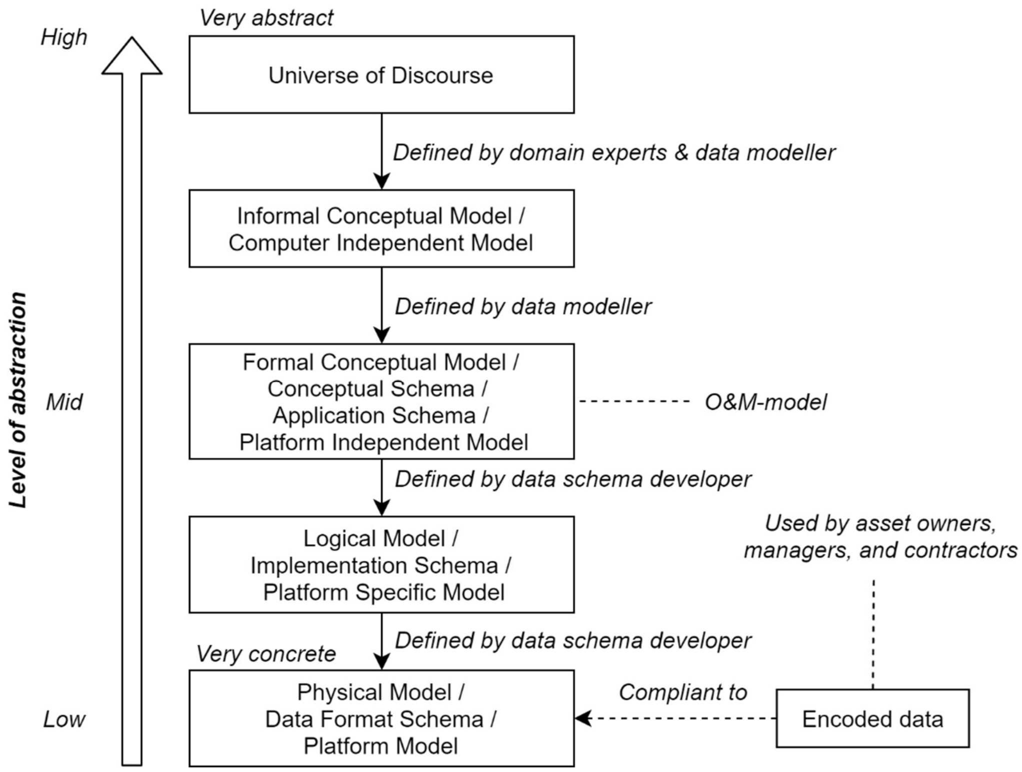


FIGURE 1-1. Data modelling and levels of abstraction.

Data modelling at high-level, **conceptual** abstraction uses natural languages meant for human communication (e.g. English). This is called the **informal** conceptual level, or Computer Independent Model (CIM). On the more concrete **formal** conceptual level, data is modelled using expressive languages such as the entity-relationship model (E-R) or the Unified Modelling Language (UML). At both the informal and formal conceptual levels, the languages are used to both describe and to show schematically data categories, the relationships among them, and the constraints on those relationships. No platform-specific (i.e. software) constraints exist at this level. The representation at the formal conceptual level is also called conceptual schema, application schema and Platform Independent Model (PIM).

Data modelling at the **logical**, mid-level abstraction can also use UML to describe the Universe of Discourse. However, at this level, the data model is based on a specific platform, and thus, the representation is also called Platform Specific Model (PSM) and Implementation Schema. The PSM follows the logical structure of a modelling paradigm, for example, XML/object-oriented paradigm for GML, and relational paradigm for relational databases. It is an intermediate step in the derivation of the Platform Models (PM), which are the actual data format schemas.

Data modelling at the lowest level of abstraction (**physical**) uses Data Description Languages (DDL), specific to a chosen software platform. For example, different DDLs are XML Schema (XSD), JSON Schema, and Structured Query Language DDL (SQL DDL). SQL DDL, in turn, can be used to model different types of databases, also referred to as database models (e.g. hierarchical, network, relational, and object-oriented). These Physical models are also known as Platform Models and **data format schemas**. They can be used to encode and store spatial data. Encoding in this context is changing the syntax of the geospatial data from its original format to the desired one.

1.2.1. Data modelling for the design of data format schemas

Data modelling is used as a paradigm in database design and evolved together with database models (e.g. network, hierarchical, and relational). However, every data modelling process does not necessarily result in a database, only ***data modelling for database design*** does. Alternative physical models (i.e. non-database) of conceptual models are also possible. For example, CityGML is a data model which exists as a UML Class Diagram at the formal conceptual level. This data model has been physically mapped into three different encodings: GML (XSD) and CityJSON (GeoJSON) for data transfer, and SQL via the 3DCityDB (3DCityDB, n.d.; Gröger & Plümer, 2012; Ledoux et al., 2019), which is a spatial relational database for data storage and manipulation (Oracle and PostgreSQL/PostGIS). Thus, in the case of CityGML, the data modelling process has resulted in two non-database physical representations of the real-world phenomena.

Different types of physical models (i.e. data format schemas) have different purposes:

- Some are optimized to **be used as internationally standardized transfer formats**. They consist of a schema definition file —to which instance documents holding the data must comply— and are used to transfer information over the internet. Examples are GML (and its application schemas, like CityGML) and GeoJSON (and its application schemas, like CityJSON).

  The derivation of the transfer format of the O&M-model can be done automatically, using freely available tooling (ShapeChange Development Team, n.d.). It results in an eXtensible Markup Language (XML) Schema Definition (XSD) file that provides the structure for the GML instance documents which store the data. However, GML files are not well-suited for typical Geographic Information Systems (GIS) tasks involving data storage, retrieval, manipulation, analysis, and presentation. Its use is mostly restricted as a standardized means for data transfer over the internet. This type of files can be very large and difficult to parse, so they see little practical use. Thus, it is not easy to use them to formally (i.e. through querying) test the O&M-model.
- Other formats are optimized for **data storage, manipulation, analysis, and presentation**. These formats consist of either a schema definition file or a database script (in case of choosing a database). Examples of databases are Oracle and PostgreSQL/PostGIS. Even though database 'dumps' can be used to transfer information, this is not a standardized exchange mechanism.

For this thesis, I focus exclusively on the preeminent format optimized for storage, manipulation, analysis and presentation of geographic data: spatial databases.

The derivation of the physical model (i.e. data format schema) from the formal conceptual model (i.e. application schema) is called ***model transformation*** in ISO-19118. It can be executed in two ways, as shown in FIGURE 1-2:

- The direct procedure (red arrow in the figure). This is the procedure specified in ISO-19118 and is done directly from an application schema to a data format schema. The transformation is defined by an encoding rule, which includes the relevant *schema conversion rules*. An example of schema conversion rules is the encoding specification of CityGML (OGC, 2012), which specifies how to derive the CityGML transfer format. Unlike the indirect procedure explained next, the encoding rule in this type of transformation needs to be very broad to include the transformation to the logical model too.

- The indirect procedure (green arrow in the figure). This is the common procedure in data modelling for database design (Nyerges, 2010) and the Model Driven Architecture Approach (Kutzner, 2016). The difference with the direct procedure is that the transformation is done in two steps. First, there is a mapping from the formal conceptual model (e.g. UML class diagram) to the logical model of the platform (e.g. relational, object-oriented, etc.). This mapping is defined in a transformation definition. Next, the logical model is transformed into a physical model (e.g. CityGML schema, CityJSON schema, database scripts, etc.). The encoding rule (logical to physical model) in the indirect procedure is simpler than the one in the direct procedure, as it has already been mapped. *This is the approach that I pursue in this thesis.*

In short: A single Platform Independent Model (PIM) can be transformed into several different Platform Specific Models (e.g. GML and GeoJSON models, and relational model), which in turn can be codified into their corresponding Platform Models (GML and GeoJSON schemas, and SQL code respectively) which are the actual data format schemas. The chosen platform model depends on whether the purpose is to exchange or to store data.

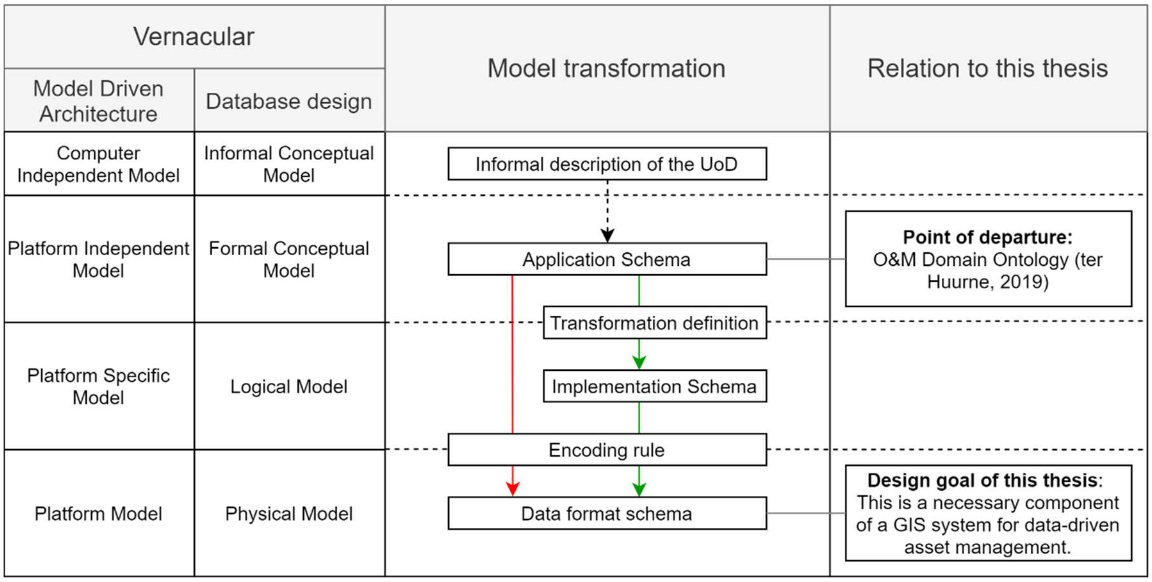


FIGURE 1-2. Comparison of the transformation of models. Direct procedure (red arrow) and Indirect procedure (green arrow). Adapted from Kutzner (2016, p. 38).

The following section explains how data modelling relates to the O&M Domain Ontology project, and which type of data format is of interest to the present study.

1.2.2. Data modelling and the O&M Domain Ontology

FIGURE 1-3 is a visual summary of the O&M Domain Ontology project and the role of data modelling in it. The project can be divided into three parts in the following way:

Part 1: Development of the O&M Domain Ontology. The initial phase of the project was the development of the O&M-model by Ter Huurne (2019). Utility network owners and managers need data models to determine what asset data to capture and how to represent it, with the goals of enabling standardized data exchanges and allowing the storage and manipulation of data. Ter Huurne worked with domain experts to develop an informal conceptual model of the universe of discourse (utility networks and their lifecycle management). He then mapped the informal conceptual model into a formal conceptual model, using the Unified Modelling Language (UML). The O&M-model is a semantic data model. Semantics in the context of data modelling is related to models being able to unambiguously express the meanings of the rich attribute information about geographical objects. The O&M-model was not formally tested at this stage —in terms of comprehensiveness— due to the lack of a suitable encoding for this purpose.

Part 2: Development of a database encoding for the O&M Domain Ontology. The present study begins from the formal conceptual model (O&M-model) and deals with the design of a physical model optimized for data storage, manipulation, analysis, and presentation. It requires mapping the O&M-model (which is object-oriented) to the relational database paradigm and developing the scripts necessary to set up a prototype database. I provide additional explanations to the technologies in:

- **Section 1.3** consists of a review of GIS and spatial databases, which currently are the industry standard for data storage, manipulation, analysis, and presentation. As such, they provide in principle a solid base for developing the desired stand-alone GIS tool for supporting O&M of utility networks.
- **Section 1.4** explains the meaning of encoding in the context of data transfer.

Part 3: Development of data encoding processes to be used in conjunction with the standardized data transfer format (*future research*). This is a line of research that would enable the use of the ontology for data exchange.

1.3. GIS and spatial databases

A Geographic Information System (GIS) is a set of computer-based tools for working with geographic information. The capabilities of a GIS are (1) Data capture and preparation, (2) Data management, including storage and maintenance, (3) Data manipulation and analysis, and (4) Data presentation (Huisman & de By, 2009, pp. 26, 32). Of special interest to this research are GIS components that can handle (store, query and manipulate) both thematic, *attribute* data and *spatial* data.

By far, the most widespread technology for this purpose is spatial relational databases, also known as geo-databases. Even though some GIS applications offer native database support, a common set-up is having a spatial database (e.g. PostgreSQL/PostGIS) managed using a Spatial Relational Database Management System (e.g. pgAdmin), and the data manipulation, analysis and presentation carried out using a GIS application (e.g. QGIS).

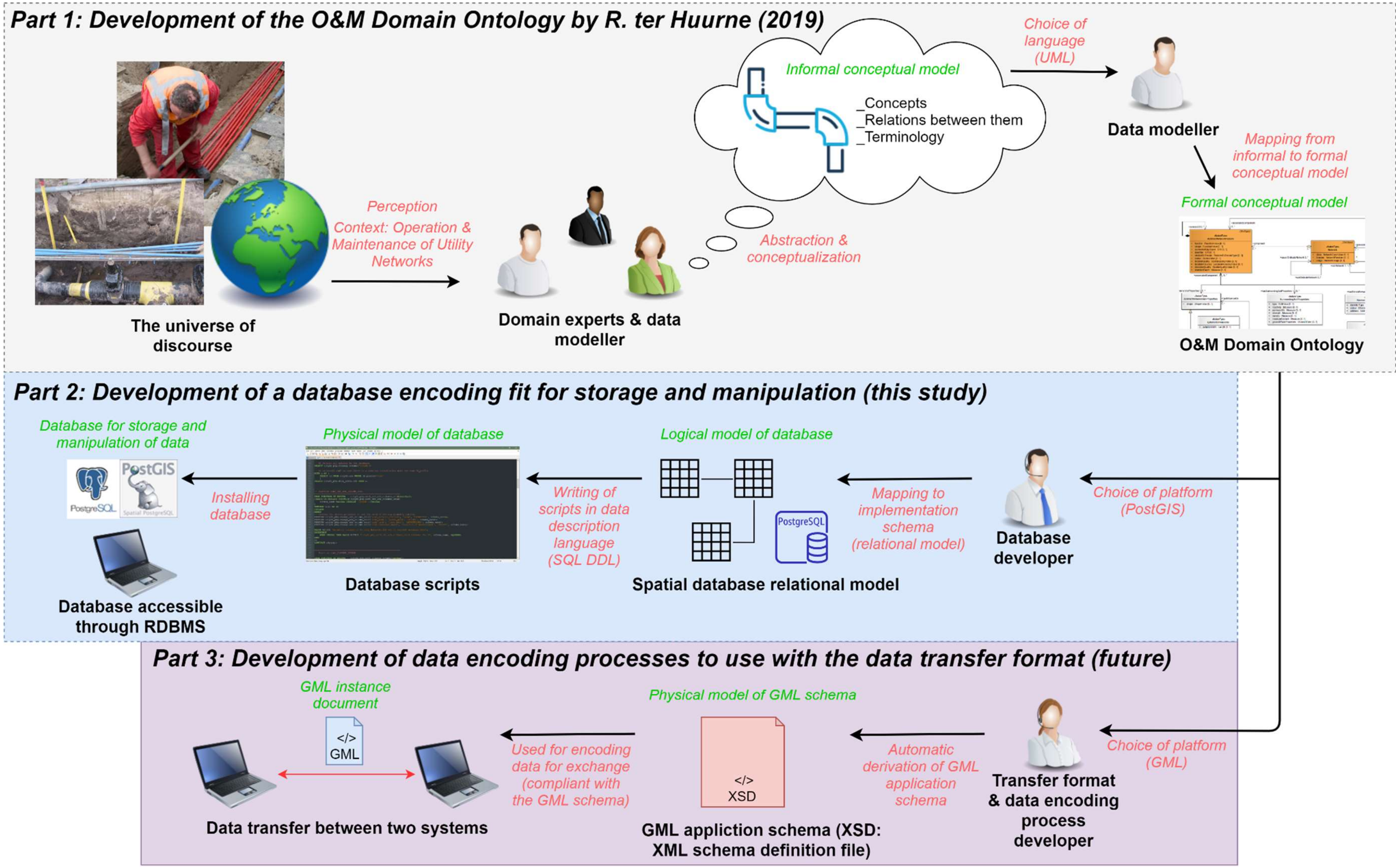


FIGURE 1-3. Data modelling for the design of data formats. Adapted to the scope of this project.

The requirements for spatial databases are (Huisman & de By, 2009, pp. 178–186): (1) possibility of connection to a frontend GIS application, (2) extended query language (SQL) for performing spatial analyses, (3) spatial indexing of the geographic data, (4) a dedicated table for Spatial Reference Systems, and (5) the possibility to store spatial and attribute data together (FIGURE 1-4).

For all of the above, spatial databases are a good match to improve the asset management system at the University of Twente campus. Moreover, using the O&M-model as the basis for database design provides good coverage of lifecycle management concepts.

An existing database is of relevance to this study. The 3D City Database (3DCityDB) is a geo-database that is derived from the application schema of CityGML and used to store the complex city models achievable with CityGML (3DCityDB Development Team, 2016). The 3DCityDB enables the efficient management, analysis and querying of complex city models within a central repository and using SQL.

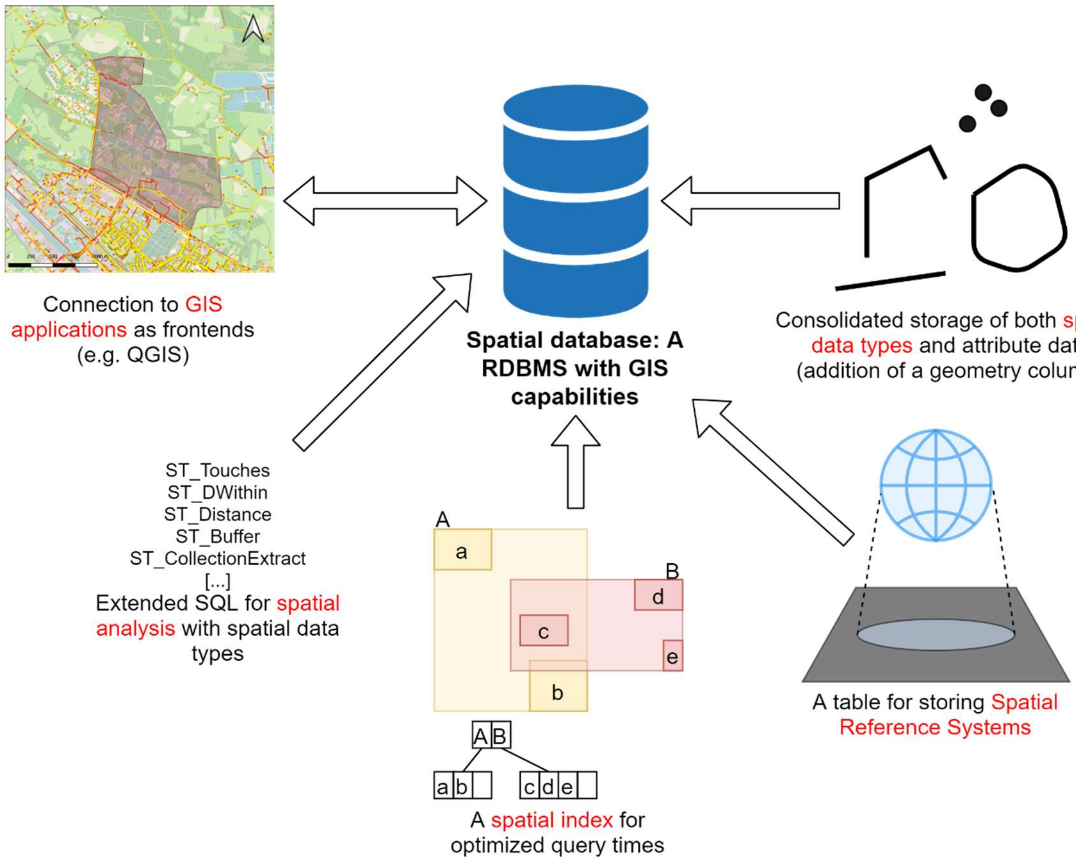


FIGURE 1-4. Spatial databases' requirements

1.4. Encoding of geospatial data and models

To encode data is to transform them from their internal schema to a system-independent schema suitable for transfer. To encode models is to transform them from the conceptual level to the physical. Information integration requires exchanging heterogeneous information in terms of the technical set-up, syntactics, data models used, structure and schematics, and semantics (Kutzner, 2016, p. 33). The process of information integration deals with the transformation of source data (compliant to the structure and semantics of a source model) into data compliant to a target model. Two types of ***data*** (i.e. instance) ***transformations*** can be distinguished: syntactic (mostly related to the format) and semantic (mostly related to the language and vocabulary chosen, also called mapping). The syntactic transformation is defined by the *instance conversion rules* of the *encoding rule*. The semantic transformation is handled using mapping software to translate between the semantics and structure of different data models.

Model transformations (defined by the *schema conversion rules* of the *encoding rule*) are also a necessary task in information integration. These were explained in **Section 1.2.1**, which discusses data modelling for the design of data format schemas.

FIGURE 1-5 shows all the necessary transformations (**data** and **model**) during data exchange between a source and a target system according to ISO-19118.

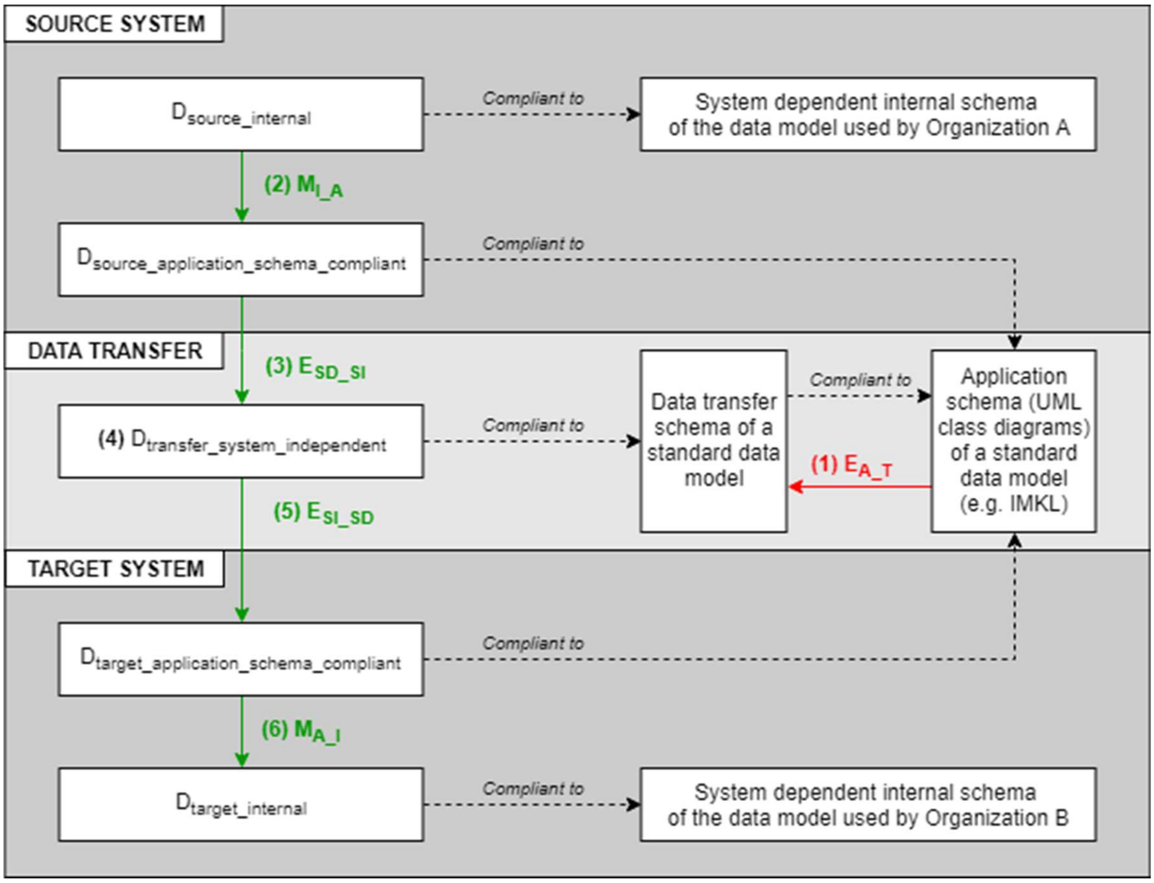


FIGURE 1-5. Transformations during data interchange between systems [ISO-19118]. Based on Kutzner (2016, p. 37)

These are the steps shown in the figure:

1. Model transformation E_{A_T} (**E**ncoding from **A**pplication Schema to **T**ransfer Schema) of the application schema into the data transfer schema. This schema defines the structure of the data transfer format. The data transfer format is system-independent because it can be interpreted by any computer[2].
2. Data transformation M_{I_A} (**M**apping **I**nternal Schema to **A**pplication Schema). Semantic mapping of the internal schema of organization A to the external application schema.
3. Data transformation E_{SD_SI} (**E**ncoding **S**ystem **D**ependent to **S**ystem **I**ndependent). Syntactic encoding of the system-dependent source data into system-independent transfer data.

[2] Please note, system-independent should not be confused with Platform Independent Model as explained in **Section 1.2**.

4. Data transfer from the source to the target system.
5. Data transformation **E_{SI_SD}** (**E**ncoding **S**ystem **I**ndependent to **S**ystem **D**ependent). Syntactic encoding (decoding) of the system-independent transfer data into the system-dependent target data.
6. Data transformation **M_{A_I}** (**M**apping **A**pplication Schema to **I**nternal Schema). Semantic mapping of the external application schema to the internal schema of the target system.

1.5. Stakeholder analysis

The starting point for the stakeholder analysis is the one done by Ter Huurne (2019, pp. 15–16). More detailed analysis of the department of Campus & Facility Management and their current GIS is in **Appendix B**. The stakeholders of interest to this research thus are (***Table 1-1***):

#	Name	Description	Goal
1	Encoding designer	Master's student at the University of Twente responsible for the development of the database	Acquiring a Master's degree
2	Ontology designer	PDEng from the University of Twente who designed the O&M Domain Ontology	Increasing the use of the ontology (diffusion)
3	Department of Campus & Facility Management of the University of Twente	Owner of the utilities (circa 300 km) at the campus. Has a simple spatial database with utility networks. Lacks complex data representation capabilities	Improving operation & maintenance of utilities at the campus
4	Standardization agencies (Geonovum)	Organizations that design standards for the transfer of geographic information	Facilitating the sharing and updating of maintenance information between parties
5	Authorities (Kadaster)	The organization that maintains the IMKL model in the Netherlands	Keeping a similar use of terminology with IMKL to avoid semantic differences
6	Utility owners	Networks owners	Sharing and updating maintenance information, e.g. with contractors
7	Utility operators	Specialized organizations that manage the utilities of other organizations	
8	Contractors	Organizations executing on-site activities	Executing tasks safely and at the expected costs

Table 1-1. Stakeholders of the project

1.6. Conclusions from the problem investigation phase

The study of the stakeholders shows two broad categories of needs:

- *Stand-alone GIS system*: The need for improved **management and performance** of the assets (owners and managers, like C&FM, and users).
- *Interoperability*: The need for establishing **standardized exchanges** of operation & maintenance-related information (contractors, owners, and managers).

The problem investigation showed that spatial databases are a good artefact to be used as a treatment for both kinds of needs, as explained next.

In terms of **management and performance**, a spatial database based on the right application schema enables the storage, management, analysis, and presentation of the spatial and non-spatial asset data simultaneously. Thus, it can be used as a stand-alone product to improve the management of assets.

In terms of the establishment of **standardized exchanges**, a spatial database is not a system-independent transfer format, and thus, it is not meant for enabling the data interchange defined by ISO-19118. However, the database can be used to test the comprehensiveness of the formal conceptual model from which the data transfer format schema is derived. In this way, the database helps to bring closer the required standardized exchanges of operation & maintenance-related information.

1.7. Problem statement, project goal, and scope

Owners and managers of utility networks —like Campus & Facility Management— lack an open and free system to comprehensively support lifecycle asset management. Without such a system they may not have the capability to record the data they need and to use the data they own to make informed decisions. Furthermore, the international community lacks a data exchange standard for lifecycle data of utility networks. The O&M Domain Ontology could potentially address both issues, but its practical effectiveness has not been demonstrated. Thus, to demonstrate its effectiveness I (a) develop a spatial database based on the O&M-model fit for data storage, manipulation, analysis, and presentation, and (b) test it with a use case.

The main design problem is the database design for the O&M Domain Ontology, addressed in **Section 3.5**. There are, however, nested subproblems involved in the creation of the database. The first is to determine if the O&M Domain Ontology conceptual model requires changes (at the class diagram level) to facilitate its transformation from a high to a low level of abstraction (**Section 3.5.1**). The second subproblem is about mapping an existing dataset to comply with the O&M-model and loading these data to the database (**Section 4.2**). The final design problem is that of the test case for validating the database, shown in **Section 4.3**.

Furthermore, the design cycle goes through a validation phase, which in this research serves the double purpose of also validating the comprehensiveness of the underlying O&M-model and transfer formats.

Thus, the **goal** of this study is:

“To design an encoding of the O&M Domain Ontology that can be used by practitioners as a Geographic Information System component for the management of utility networks and to *indirectly* validate the usefulness of the underlying O&M-model”

This design research goal contributes to higher hierarchy, social context goals. These types of goals can be enunciated abstractly as “to improve a problem context with the final purpose of contributing to achieving stakeholder goals”. Framed from the perspective of C&FM: *To improve the way that the campus infrastructure is managed, which, in turn, contributes to reduce costs, and increase service reliability*. From a wider perspective considering network operators: *To improve the way that utility network data is shared, to achieve comprehensive standardized exchanges of O&M data*.

To find a solution to the design problem, I have set a series of intermediate **objectives**:

1. Determine an encoding for the O&M-model that is suitable for storage, manipulation, analysis, and presentation of data

Since each different encoding would require a different workflow, the first objective is investigating the suitability of spatial relational databases as a solution for handling utility network data to provide improved asset management capabilities.

2. Define the methodology for designing the encoding

The second objective is to select a methodology for deriving a suitable encoding from the O&M Domain Ontology class diagram (application schema / formal conceptual model). After choosing a platform (i.e. software) the methodology can be obtained through a literature review.

3. Define the requirements for the encoding

The functional requirements for the encoding are tied to the goals of the stakeholders. Thus, a stakeholder analysis is carried out as part of the problem investigation. However, at this stage of the data modelling process, there is far less room for accommodating stakeholder requirements. The most important decisions in terms of functionality happened at the design phase of the O&M Domain Ontology.

4. Design the encoding

This objective is the design of the artefact by following the methodology and covering the requirements. Furthermore, this step involves reviewing the existing application schema and determining if simplifications and other changes are desirable. The result is the computer scripts that allow to set-up the encoding.

5. Create and populate a prototype of the encoding

The creation of a physical instance of the encoding is necessary for its validation. Furthermore, the nested subproblem of populating the otherwise empty prototype encoding with real data is addressed.

6. Validate the prototype encoding (and indirectly the O&M-model)

The validation of the prototype encoding involves performing tests to determine if the requirements are satisfied. By analysing the comprehensiveness of this encoding, the O&M-model is indirectly covered too (and thus, the transfer format too).

In terms of **scope**:

- The project is focused on the design of a convenient encoding for the O&M-model to obtain an open, free, and stand-alone GIS support tool for asset managers.
- The transfer format (XSD file format) is not tested directly, especially its practicality (i.e. the complexity of GML compared to other transfer formats). However, the comprehensiveness of the transfer format is tested indirectly by testing the alternative encoding of the O&M-model designed in this research. This is because both encodings have a transitive relation, as they are derived from the same data model.
- The project ends with a prototype of the encoding which is necessary for its validation. No actual technological transfer to the stakeholders is planned at this stage.
- Furthermore, it includes only 2 of the 8 utility networks located at the campus of the University of Twente, due to time concerns. These are the gas low pressure and district heating networks. In turn, district heating consists of two subnetworks: supply and return.

1.8. Role and outline of this document

The document aims to explain the design process and outcomes for the encoding of the O&M Domain Ontology. Since a thesis report alone is not very suitable for showing the encoding itself, I created a GitHub repository (Fossatti, 2020b) that contains the modelling part of the research and allows to interact with the software components. Moreover, during the development of this project I have co-authored a paper that has been submitted for publication at a geoinformation conference.

I followed wherever possible the structure of the O&M Domain Ontology' report (ter Huurne, 2019) to increase the readability of the document as a pair. The remainder of this document is thus structured as follows. Chapter 2 contains the methodology followed to design the project. Chapter 3 shows the design of the treatment. Chapter 4 shows the validation of the treatment. Chapter 5 contains a discussion on the results, the conclusions, recommendations, and future research ideas.

Chapter 2
METHODOLOGY

2.1. General approach: the design cycle

A practical encoding of the O&M Domain Ontology formal conceptual model is missing. This encoding should follow as closely as possible the structure of its formal conceptual model. In the previous chapter, I presented the development of the encoding as a design problem. As such, it can be addressed following the engineering cycle as sketched by (Wieringa, 2010, p. 28).

In the vernacular of the design cycle, the proposed encoding is called an ***artefact***. Its use is meant to improve a ***problem context:*** (a) the current GIS at UT campus does not sufficiently support asset management, and (b) the O&M-model is formally untested in terms of comprehensiveness. Thus, the artefact is evaluated in terms of its ***usefulness*** for these improvement goals. Therefore, the usefulness of the encoding cannot be evaluated outside of its context (or a prototype of both). Moreover, the encoding does not by itself solve the problem. The interaction of the artefact with the problem context —called the ***treatment***— is what contributes to solving the problem.

Since technology transfer to the client is excluded from the scope of the thesis, I will only use a subset of the engineering cycle called the design cycle. This smaller cycle focuses on design tasks —problem investigation, treatment design, and treatment validation— and excludes the implementation (technology transfer and use by the client) and the evaluation of the implementation. In this framework, there is a nesting of cycles. For example, to carry out the treatment validation phase it is necessary to make designs and to investigate problems.

The application of the design cycle to the design of the alternative encoding of the O&M Domain Ontology results in the following phases:

1. Problem investigation. Define what the problem is, and the parties involved. Determine what artefact can potentially treat the problem.
2. Treatment design. Design an encoding that satisfies the requirements of the client using the formal conceptual model of the O&M-model as a basis.
3. Treatment validation. Implement a prototype of the encoding and test it in (a prototype of) its context. Evaluate its usefulness.

FIGURE 2-1 shows a graphic overview of the three phases of the design cycle as applied to the present research. The cyclical nature of the design implies iterating through these phases. For the sake of brevity, the iterations are not explained, instead, the whole work to complete each phase is shown. The remainder of this chapter explains the phases of the design cycle in detail, as well as the activities outlined in FIGURE 2-1.

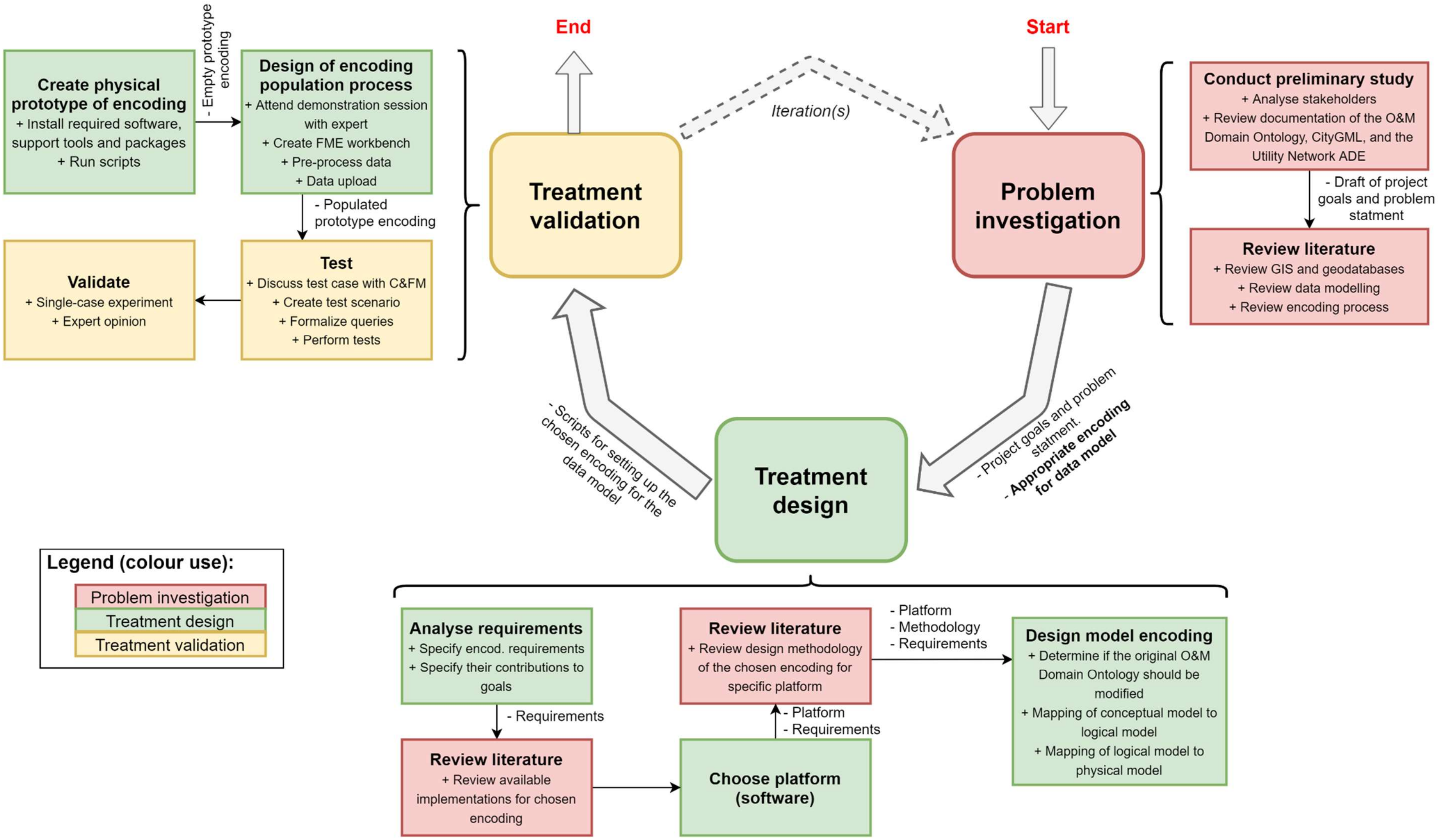


FIGURE 2-1. Research methodology.

2.2. Problem investigation (Chapter 1)

The problem investigation consisted of two parts: a preliminary study on the problem context (**Section 1.1**), and a literature review on GIS and geodatabases, data modelling, and encoding (**Sections 1.2, 1.3, and 1.4**). The combined review of both the problem context and the broader technological analysis of this section was aimed at pin-pointing the appropriate treatment useful for improving the stakeholder's standing, the stakeholder's goals, and defining the problem statement.

The first step was reviewing the documentation of CityGML, the Utility Network ADE and their related UML class diagrams and database encodings. This study also included reviewing the standardization efforts in the field of utility networks, such as the INSPIRE program in Europe, and the IMKL standard in the Netherlands. The study concludes with the review of the O&M Domain Ontology and the design process that gave origin to it. Please note that to keep the length of this document in check this section of the problem investigation is in **Appendix A** and summarized as **Section 1.1**.

With regards to the literature review (**Sections 1.2, 1.3, and 1.4**), three broad topics contained the most relevant information for this research: (1) Geographic Information Systems (GIS) and spatial databases, (2) data modelling for database design, and (3) the encoding of models and data. The study of GIS was meant to review available spatial data formats and system setups and to compare them with the system at the campus. It also provided basic knowledge for moving into the more specific topic of data modelling in the context of spatial databases.

The final investigative step involved analysing the stakeholders (**Section 1.5**). The starting point for the stakeholder analysis was the one done by Ter Huurne (2019, pp. 15–16). The new analysis involved re-evaluating the stakeholders and their goals from the perspective of this study.

Finally, I compared the stakeholders' goals and current system with the documentation related to the O&M Domain Ontology and the phenomena from the literature review to justify the choice of encoding as an artefact (**Section 1.6**). This lead to the formulation of the problem statement, goals, and objectives (**Section 1.7**).

2.3. Treatment design (Chapter 3)

The goal of this phase was to design an encoding that satisfies the requirements of the client using the formal conceptual model of the O&M Domain Ontology as a basis. The methodology for the design of the treatment consisted of five steps.

In **Section 3.1**, I re-examined the analysis of the goals and requirements of the department of Campus & Facility Management of UT (C&FM) and other stakeholders performed by Ter Huurne (2019, pp. 28–32). I also examined the informal competency questions that the O&M conceptual model was originally designed to answer, as they are linked to the user requirements. I presented new functional requirements linked to the proposed encoding, as opposed to the more abstract conceptual model.

In **Section 3.2**, I reviewed available treatments (i.e. existing database solutions). I did this to determine whether a new artefact is *actually* needed or an existing one could be (re)used. This step overlapped in part with the technological study in the problem investigation, during which existing encodings are reviewed.

In **Section 3.3**, I chose a software platform that fits with the chosen encoding and its intended purpose. At this point, all platform choices needed to be fully defined because the transformation of the model requires knowing to which platform it will be transformed.

In **Section 3.4**, I reviewed the methodological steps to design the chosen encoding, which also partly overlapped with the more general study on the encoding of data and models (**Section 1.4**). The goal was to determine the necessary components of such an artefact and elucidating the process of mapping from the formal conceptual model of the O&M Domain Ontology to a logical and physical model.

The final step was the artefact design, shown in **Section 3.5**. I started by examining in detail the UML class diagrams of the O&M-model and compared them with available documentation of similar systems to determine whether it is convenient to alter the model before mapping it from the conceptual to the logical level of abstraction. The systems used for comparison were CityGML, the Energy ADE, the Utility Network ADE v0.9.2, and all their databases. The last part of the design consisted of deriving the logical and physical models from the revised UML class diagrams that formally describe the conceptual model.

2.4. Treatment validation (Chapter 4)

To validate the use of the database encoding as a GIS component means to justify that if this treatment were implemented in the problem context it would contribute to the goals of the stakeholders. I did this by showing that the treatment requirements —linked to the stakeholder goals— were satisfied., as shown in **FIGURE 2-2**. Because there was no actual technological transfer to a stakeholder, I did this by studying the interaction between a model of the artefact and a model of its context (prototype artefact X simulated use case) after applying stimuli (model inputs).

Model of the artefact. To study the effects of the interaction between artefact and context, the database first needed to be physically implemented in a computer, to be used as a prototype of a real implementation. Several packages had to be pre-installed, and then the computer scripts derived during the design phase had to be executed. This is addressed in **Section 4.1.** Finally, the model of the artefact was finished by connecting the prototype database to QGIS, a GIS application to act as frontend to the system.

Model of context. Next, the prototype artefact needed to be populated with a dataset with real utility networks which (partially) contained the necessary data to perform tests. First, the dataset was downloaded and checked for digitization issues which had to be fixed before transforming it. Following, I designed a transformation process that uses the pre-processed campus dataset as input, maps the data in it to comply with the O&M Domain Ontology model, and outputs the data into the artefact directly. This is shown in **Section 4.2**.

In **Section 4.3**, I determined what type of test was appropriate by discussing the needs of C&FM with a representative. Then I broke down the test into phases and linked the different phases with the competency questions for which the O&M Domain Ontology was designed (ter Huurne, 2019, pp. 28–29). Next, I formalized these competency questions into computer queries for formally testing the model, and finally, I executed the queries.

For validation techniques, I used a single-case experiment combined with expert opinion. The result of the case experiment were the effects of the interaction between the model of the artefact (database + GIS application) and the model of the context (campus dataset + street reconstruction test). The final step of the validation (**Section 4.4**) consisted of (a) showing the tests to experts to get their feedback, and (b) contrasting their feedback with the requirements from **Section 3.1**.

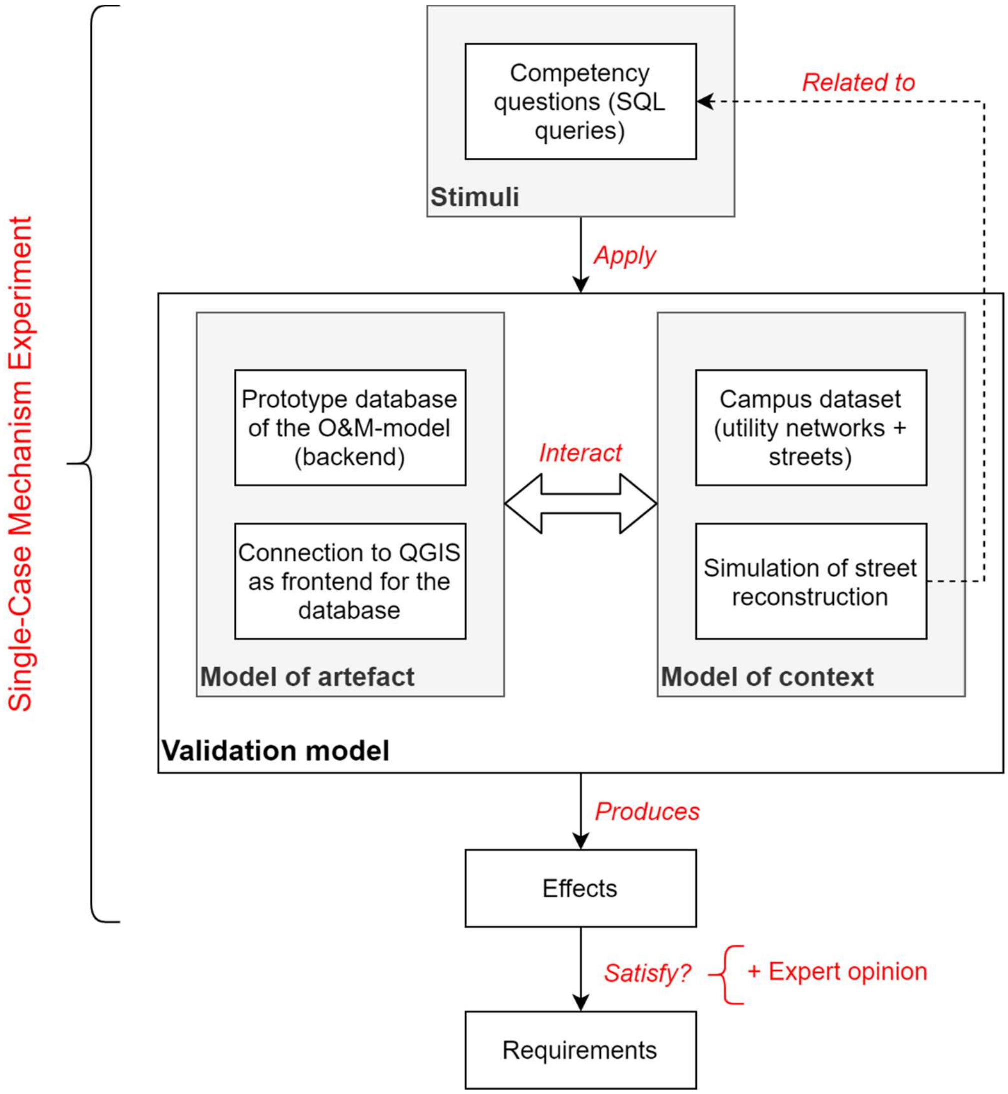


FIGURE 2-2. Validation method

Chapter 3
TREATMENT DESIGN

This chapter consists of five sequential sections. First, I link the goals of the stakeholders to requirements for the encoding. Second, I explore the availability of existing treatments to deal with the stakeholder requirements, as reusing one would allow skipping the design of a new one. Third, I choose a platform (software and modelling paradigm) because the formal conceptual model must be mapped to a specific platform. Fourth, I review the literature on the design of spatial-relational databases, with specific attention on CityGML and its ecosystem. Finally, with the combined knowledge from the previous sections, I design the database schema of the O&M Domain Ontology.

3.1. Requirements engineering

The requirements for the treatment are the properties desired by the stakeholders, or in other words, goals that the artefact should meet in a certain context (Wieringa, 2010, p. 51). The artefact of this research is an encoding derived from the O&M Domain Ontology, which in turn is the artefact of previous research by Ter Huurne (2019). As such, it directly inherits some of the requirements which the data model should fulfil. A functional requirement thus inherited is:

***RO** The encoding must allow the representation and handling of the data that are necessary to answer the competency questions for which the ontology was designed by Ter Huurne (2019, pp. 28–29)*

The choice of the requirements in this study is justified by using a contribution argument (Wieringa, 2010, p. 52) which can be abstractly expressed as (Artefact Requirement) x (Context Assumptions) contribute to (Stakeholder Goal). In the case of **R0**:

- If the encoding enables the representation of the data necessary to answer the original competency questions for the domain ontology,
- and assuming that the users of the encoding use it to actually store and query operation & maintenance data as part of their asset management strategy,
- then the encoding **contributes to the user's goals of** 'improving O&M of utilities at the campus' and 'executing tasks safely and at the expected costs' (see stakeholder goals in section **1.5**)

***R1** The encoding must be useful for C&FM and other organizations that manage utility networks, i.e. the stakeholder must have a favourable opinion about its utility*

- If the encoding is considered useful by experts in asset management and utility networks,
- and assuming that the reality of asset management of utility networks conforms to the prototype validation test shown to the experts,
- then the encoding **contributes to the user's goals of** 'improving O&M of utilities at the campus'

As explained in **Section 1.1**, some data formats are optimized for data transfer and others for data storage and handling. One of the main goals of the O&M Domain Ontology is to use it as the data model from which a data transfer format is derived. With such encoding of the data model, organizations can exchange their O&M data according to ISO-19118 (see **Figure 1-5**). Even though the derivation of the data transfer format from the data model is relatively simple, it remains formally untested in terms of comprehensiveness of representation. The most common data transfer format is GML and it is not well suited for performing queries. Thus, a requirement for the chosen encoded can be stated as:

R3 *The encoding must allow testing the comprehensiveness of the O&M-model from which the data transfer format is derived*

- If the encoding enables testing the comprehensiveness of the O&M-model (and transitively tests the comprehensiveness of the data transfer format),
- and assuming that data transfer using the transfer format becomes widespread in the sector,
- then the encoding **contributes to the stakeholder's goals of** 'sharing and updating of maintenance information between parties'

3.2. Available treatments

Even though there is no spatial database encoding of the O&M Domain Ontology, there are three other encodings which are helpful to the design of the artefact. First, as mentioned in **Section 1.3,** CityGML has a spatial database encoding named the 3DCityDB, which uses both the Oracle and PostGIS platforms. Second, the Energy ADE has a database encoding built on top of 3DCityDB (Agugiaro & Holcik, 2017b). Both encodings have good documentation to help guide the design of similar encodings. Finally, —and most importantly— the Utility Network ADE version 0.9.2 also has been encoded into a spatial database on top of 3DCityDB (Agugiaro, n.d.-a). However, no official documentation exists for either the data model or the database encoding of the Utility Network ADE.

Thus, no existing treatment can be directly reused. However, the existing Utility Network ADE database physical model can serve as a strong base for the physical model of the O&M Domain Ontology database. Moreover, the documentation of the 3DCityDB and the Energy ADE can help to determine the design methodology for this type of encoding.

3.3. Choice of platform

As explained in **Appendix B**, the department of Campus & Facility Management (C&FM) currently uses two PostGIS databases for data storage and management of its assets, and QGIS as a frontend for analysis and presentation. Both are simple, with no associations between tables. Moreover, the 3DCityDB and the Utility Network ADE database both are built using PostGIS. Furthermore, PostGIS is free to use and meets all the functionalities of spatial databases (see **Section 1.3**). Significantly, it can be managed directly from the GIS application currently used by C&FM, namely QGIS. For these three reasons, PostGIS is the most convenient platform on which to encode the database.

3.4. Design methodology of spatial-relational databases

As explained in **Section 1.2.1**, in this study data modelling for database design implies deriving a spatial-relational database schema for a specific platform (physical model), based on a logical model (implementation schema, with relational database paradigm), based on a formal conceptual model (application schema, with object-oriented paradigm), based on an informal conceptual model that represents a universe of discourse. In this study, the process begins with the application schema of the O&M Domain Ontology, so the previous steps are not explained.

Three approaches are possible to derive a spatial database schema starting from the formal conceptual model of this thesis: **(a)** a fully automatic approach facilitated by 3DCityDB tooling, **(b)** the Model Driven Architecture semi-automatic approach, and **(c)** the fully manual approach. They are explained next.

The fully automatic approach **(a)** is possible in versions 4.0 and onwards of the 3DCityDB Import/Export tool. The user first needs to transform the conceptual model available as a UML class diagram into a GML data format schema (XSD file). This can also be done automatically with free tooling (ShapeChange Development Team, n.d.). The Import/Export tool takes as input the GML data format schema and returns a fully installed database following the structure of the conceptual model and running on top of 3DCityDB.

The Model Driven Architecture approach **(b)** consists of predefining a transformation rule to convert the conceptual model into a logical one. Next, one converts the logical model into a physical one using predefined encoding rules. The user's input is in the definition of the transformation and encoding rules.

The manual approach **(c)** is the one that I followed. This method is the one followed by the developers of the 3DCityDB, the developers of the Energy ADE database (Agugiaro & Holcik, 2017a), and the developers of the Utility Network ADE database (Agugiaro, n.d.-a). As explained in the previous section, it also involves two steps: deriving the logical model and then the physical model. Unlike in MDA, both the mapping from the conceptual to the logical model and the encoding from the logical to the physical model are handled manually.

3.4.1. Logical Model

The data models of CityGML and its ADEs follow an *object-oriented paradigm*. Thus, the main problem of transforming the conceptual model to the logical model of a relational database is the changing of paradigm to the *relational one*, which results in different representation capabilities. A first step to prepare the data model for transformation is to manually examine the original UML diagrams and simplify them according to the capabilities of the relational model (3DCityDB Development Team, 2016; Yao & Kolbe, 2017). The next steps involve the manual mapping of the classes onto tables. The main tasks are shown next:

- Multiplicities of attributes with an unlimited number of occurrences (*) are substituted by a data type (like string) that allows storing multiple values in a record by using a predefined separator.
- Cardinalities of the type 'm:n' are stored in a separate table, and wherever possible checked for a more restrictive definition, allowing to reduce the number of tables.
- Other class associations than those with 'm:n' cardinalities are handled with foreign keys.
- Data types defined in the class model are adapted to the corresponding data types permitted by the database.
- The database may add metadata tables/columns for representing project-specific metadata, which may be lacking in the original diagrams. A metadata module containing 5 tables should be installed previous to the setup of any ADE (Agugiaro & Holcik, 2017b). One such table in this module is called objectclass, which acts as a guide for all class names in the model. Another table permits to 'register'

an ADE, as well as the prefix chosen for the ADE. The tables of different ADEs have different prefixes so that the database can handle multiple ADEs concurrently.

- Wherever convenient one or more classes with an inheritance relationship are mapped to a single table, with the table taking the name of the class of highest hierarchy (plus the ADEs prefix). This is done manually by colour coding the UML class diagram to schematically represent the proposed table structure. Each instanced object of a class should be represented by a single table row. The reduced number of tables implies improved querying efficiency.
- Each table shall have at least one column as primary key, with unique identifiers for instanced objects (within the table).
- The attributes of the classes are mapped to the columns of the corresponding tables.
- For attributes containing measures an additional column with the unit of measurement is added
- Code lists and enumerations are mapped to static look-up tables.
- For ADEs it is important not to alter the underlying 3DCityDB. Thus, extensions of classes of CityGML are handled in separate tables. Also, relations between new ADE classes and the existing CityGML classes shall be handled so that a table derived from CityGML is not altered (for example, adding an association table if required).

3.4.2. Physical Model

The previous step defined the changes to the Platform Independent Model to obtain a Platform Specific Model (relational schema). The next step is manually writing the SQL script that contains the instructions to create the database. It involves a systematic transformation of the PSM into code. It is done on a class-by-class, attribute-by-attribute basis until all classes have been coded.

After finishing the transformation of the classes into tables, four more steps are carried out to complete the physical model. First, several Data Manipulation Language (DML) stored procedures (i.e. functions) are created to handle the insertion, deletion and updating of data systematically. Second, triggers are created (if necessary). Third, custom table views are created to simplify access to the data by the user. Finally, static table data is prepared in a separate script for its insertion into lookup tables and the object class table. The insertion happens automatically when the scripts are run, and the database is installed.

3.5. Spatial-relational database design

I designed the spatial-relational database by first addressing the necessary simplifications and enhancements to the conceptual data model. After re-working the data model I manually derived the database schema.

3.5.1. Simplification and enhancement of the data model

I simultaneously simplified and enhanced the original data model of the O&M Domain Ontology (ter Huurne, 2019), to have a more manageable spatial-relational database in terms of complexity. **FIGURE *3-1*** and **FIGURE 3-2** show an example of the changes in the UML diagrams, by comparing the maintenance module of version 4.1 of the ontology with the simplified version called 0.5.8. These changes are listed next:

- Renaming the versioning of the model to a lower level to acknowledge that it has not been released yet, from version 4.x to 0.5.x.
- The class *AbstractExtraInformation* was renamed *IdentificationProperties*. Extra information was deemed ambiguous as it can refer to anything, while the attributes in the class were compatible a more specific naming.

- The class *MaintenanceProperties* was renamed *MaintenanceActivity* and connected to the class *RelatedParty*, to allow storing which contractor performed maintenance.
- Making changes in dataTypes, featureTypes and relations. Several classes were lowered in the hierarchy from featureTypes to dataTypes, which in principle only means that these classes behave more like properties than as real elements and cannot have an id. This, in turn, implied rethinking the multiplicities of the relations involving these classes.
  - The classes rebranded as dataTypes were: *Depth*, *DimensionalProperties*, *SurroundingSoilProperties*, *AbstractExtraInformation*, *CostProperties*, and *DateProperties*.
  - The multiplicity of the relation between *AbstractNetworkFeature* and *AbstractDimensionalProperties* was reduced from [0..*] to [0..1].
  - The multiplicity of the relation between *AbstractNetworkFeature* and *SurroundingSoilProperties* was reduced from [0..*] to [0..1].
  - The relation between *AbstractNetworkFeature* and *CostProperties* was reduced from [0..*] to [0..1].
  - The relation between *AbstractNetworkFeature* and *DateProperties* was reduced from [0..*] to [0..1].
  - The relation between *MaintenanceProperties* and *CostProperties* was reduced from [0..*] to [0..1].
  - The relation between *MaintenanceProperties* and *DateProperties* was reduced from [0..*] to [0..1].
  - The relation between *AbstractNetworkFeature* and *ImpactProperties* was reduced from [0..*] to [0..1].
  - The relation between Network and *ImpactProperties* was reduced from [0..*] to [0..1].

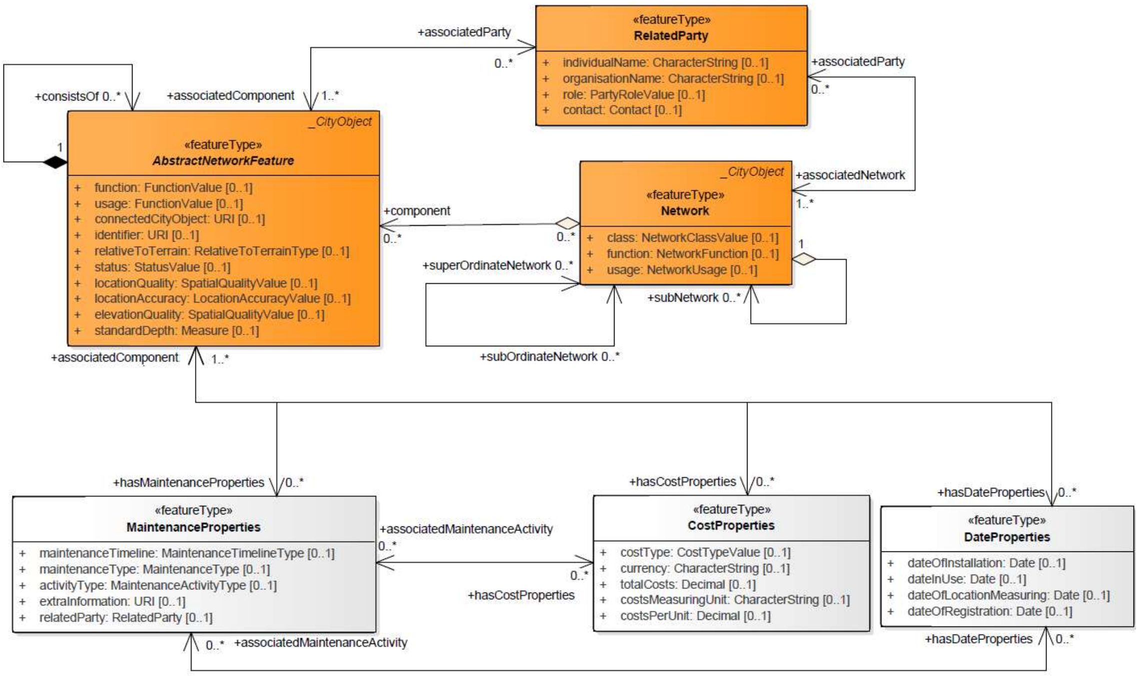


FIGURE 3-1. Data model version 4.1 (original). Maintenance module.

- Deleting attributes *WaterPipeType*, *ThermalPipeType*, *OilGasChemicalsPipeType* because they did not add new information to the model.
- Replacing the attribute 'Angle of rotation' of 'Network Feature' for 'slope', typically used in utility network design and construction.
- The new attribute 'Relative to terrain' proposed in version 4.1 of the O&M ontology was replaced with the existing CityGML attribute ''Relative to terrain' corresponding to 'city object'.

- The new class '*Address'* proposed in version 4.1 of the O&M ontology was replaced with the existing CityGML class '*Address'*.
- The Hollow space module was deleted since it is no longer part of UtilityNetwork ADE 0.9.4 and it was deemed unnecessary for this research.

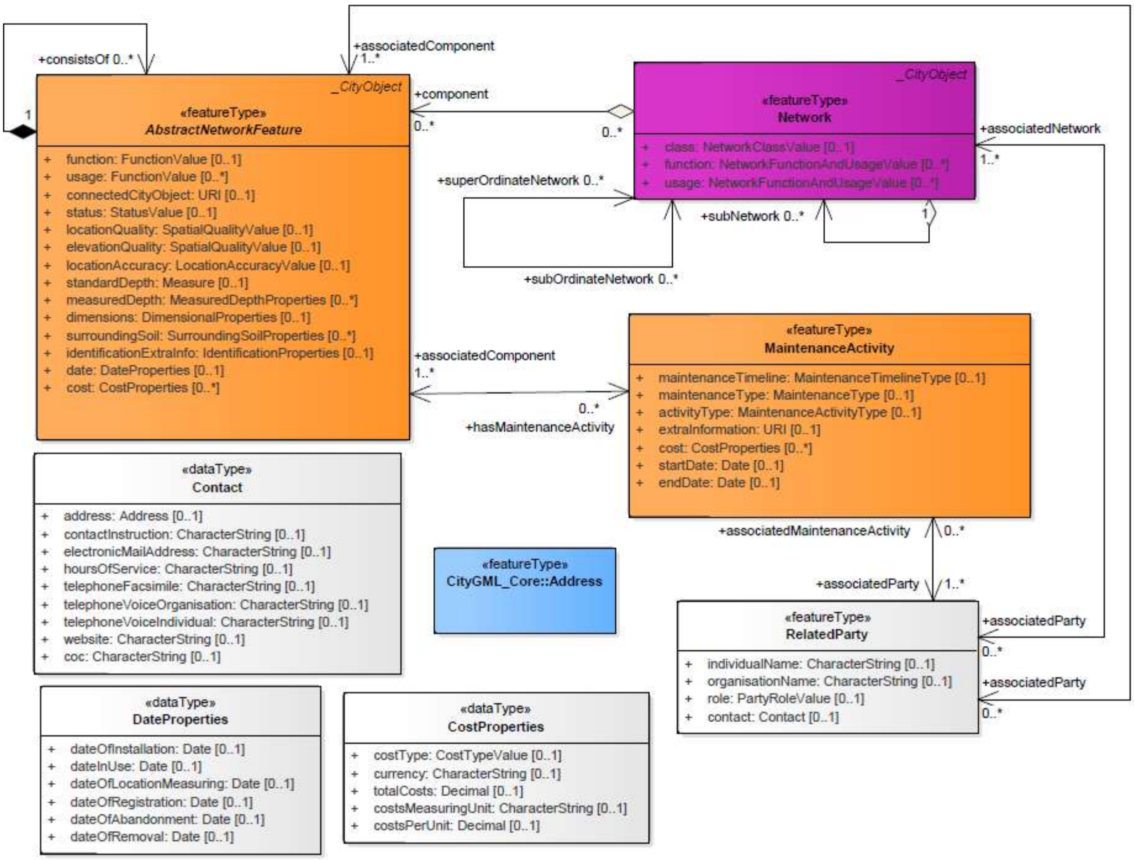


FIGURE 3-2. Data model version 0.5.8 (modified). Maintenance module.

3.5.2. Manual derivation of the database schema

The O&M Domain Ontology data model is fairly similar to the Utility Network ADE model v0.9.2 and the latter already has a database encoding (Agugiaro, n.d.-a). Therefore, I reused as much as possible of this database in my design.

3.5.2.1. Partial derivation of Logical Model

I sketched the logical structure of the database by colour coding the formal conceptual model of the simplified version (as per section **3.5.1**) of the O&M-model. The UML diagram of the O&M-model has 77 classes (featureTypes, dataTypes and types) divided into 7 modules. Moreover, it has 40 codeLists and enumerations. I mapped the 77 classes into 26 database tables. Additionally, 10 'association' tables were necessary to store the many-to-many relations in the model. The 40 codeLists were encoded into 29 'lookup' tables, which do not store information specific to a single network and are only used as a reference.

The next two examples show the mapping between the Platform Independent Formal Conceptual Model and the Platform Specific Database Logical Model. In ***FIGURE 3-3***, the class *Network* and the relation *subnetwork* are mapped to a single table called *uom5_network* (light pink background), whereas the m:n relation between *subOrdinateNetwork* and *superOrdinateNetwork* is represented in a separate table called *uom5_network_to_network* (light green background).

In **FIGURE 3-4**, the classes *SurroundingSoilProperties* and *GroundWaterProperties* are mapped to a single database table called '*uom5_soil_and_groundwater*' (light pink background).

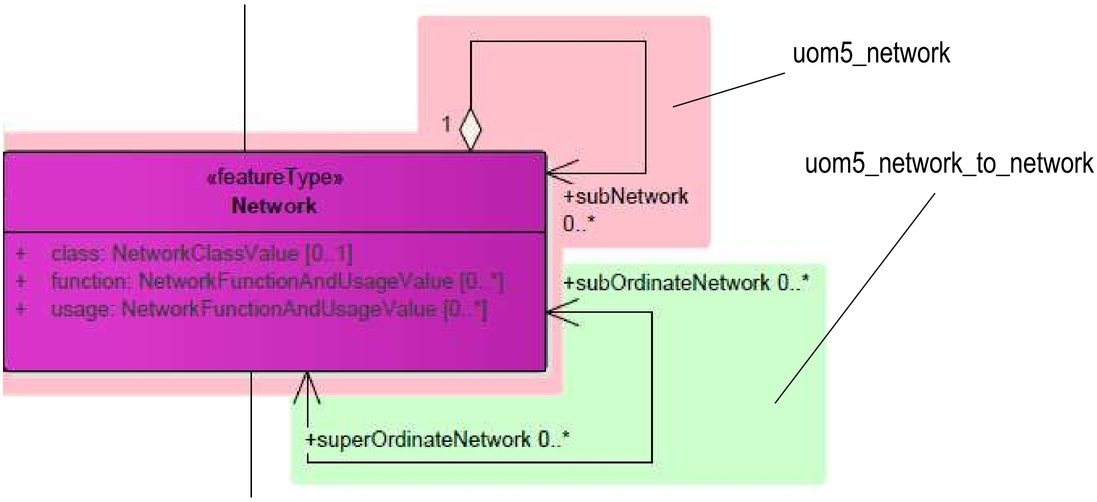


FIGURE 3-3. Example of mapping between (a) the classes '*Network' and the relation 'subNetowrk'* to the database table *'uom5_network'* (schematically shown in light pink), and (b) the relation '*subOrdinateNetwork'* and '*superOrdinateNetwork'* to the database table '*uom5_network_to_network*' (light green)

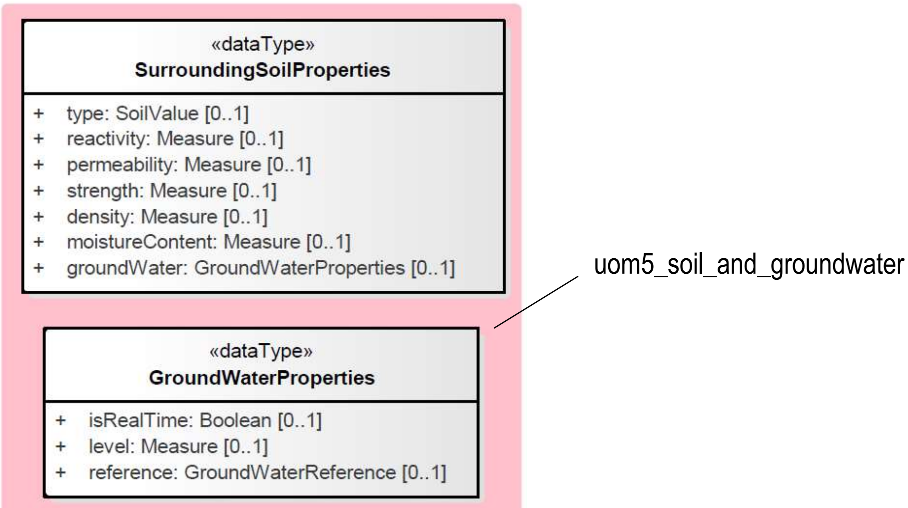


FIGURE 3-4. Example of mapping between the classes '*SurroundingSoilProperties'* and '*GroundWaterProperties'* to the database table '*uom5_soil_and_groundwater*', represented in light pink.

Note that so far I have only followed a few of the necessary steps from the manual method's methodology to derive a logical model. Namely, (a) using separate association tables for m:n type relations; (b) combining classes into single tables wherever possible, and (c) using separate tables for code lists and enumerations. Moreover, I have done so with simple colour boxes as opposed to actually merging the classes using, for example, UML. This means that the result from this step is not a computer-interpretable Logical Model, but instead a guide to show the structure of the database and to facilitate the work in the next phase.

The output of this phase is also a schematic view of the structure of the proposed database, as shown in *FIGURE 3-5*. Such a sketch is used as a guide to facilitate the translation of the classes into code during the next step. Specifically, to identify the associations between tables and the foreign keys necessary.

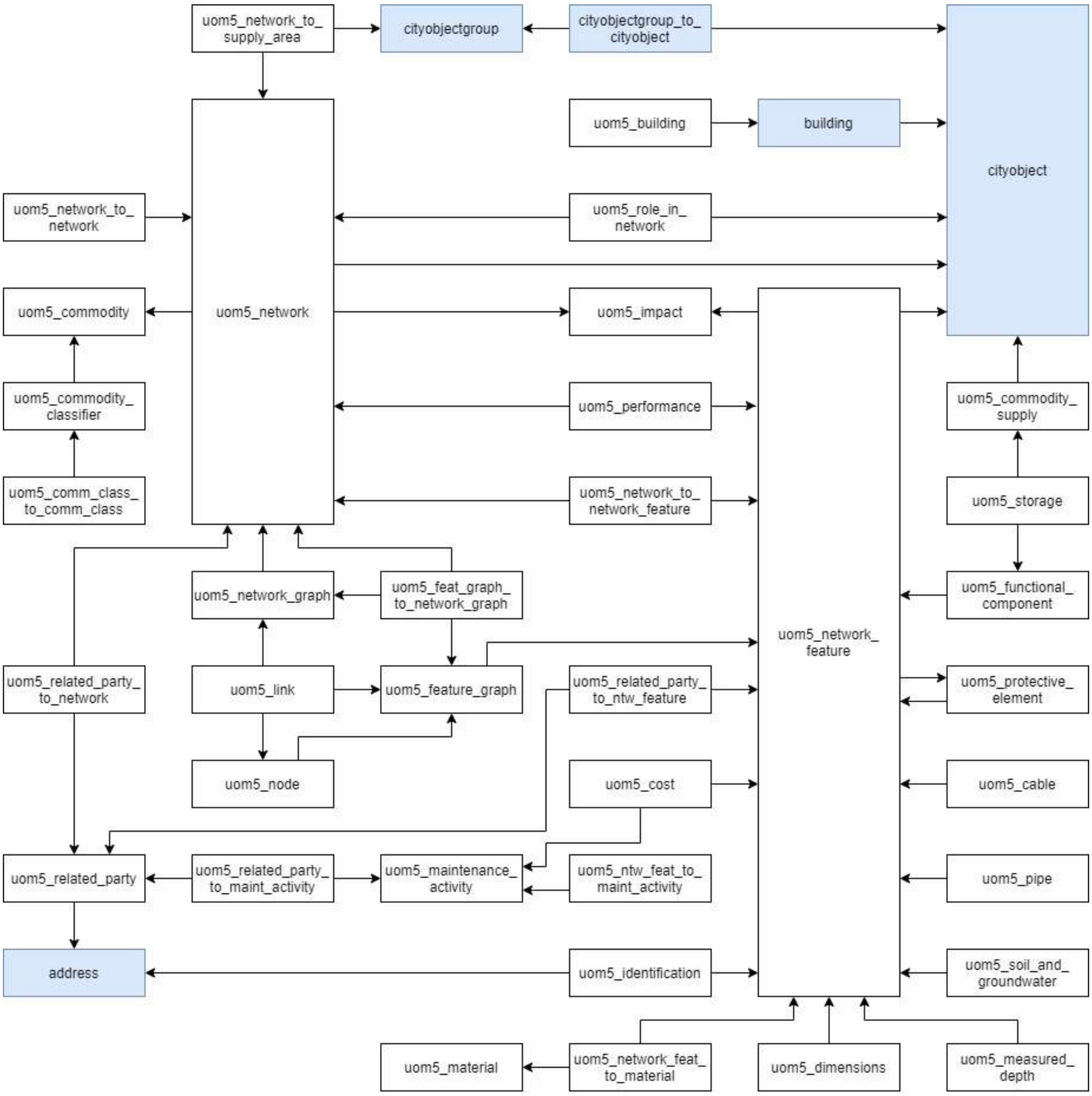


FIGURE 3-5. The table structure of the proposed database. Result of mapping from the object-oriented to the relational paradigm. *In light blue* are shown the tables from the 3DCityDB that the proposed database reuses.

3.5.2.2. Physical model

All the remaining mapping rules from **Section 3.4.1** need to be applied during the transformation of the output from the previous phase into the SQL code that is the physical model of the database. The O&M-model and the Utility Network ADE v0.9.2 are similar. Moreover, the Utility Network ADE already has an existing database implementation and its SQL scripts are freely available. Thus, I have adapted and extended those SQL scripts to match the structure of the O&M-model.

Continuing with the example from *FIGURE 3-3*, *FIGURE 3-6* shows the encoding of the partial logical model of the class '*Network*' and the relation '*subNetwork*' into the table '*uom5_network*'. The transformation adds the columns 'id' and 'objectclass_id' as explained in **Section 3.4.1.** Furthermore, it adds the columns ‘*commodity*_id’ and ‘*impact*_id’, which contain foreign keys to the tables ‘*uom5_commodity*’ and ‘*uom5_impact*’ (see FIGURE 3-5).

FIGURE 3-7 shows the encoding of the partial logical model of the relation *‘subOrdinateNetwork’ and ‘superOrdinateNetwork’* to the database table *‘uom5_network_to_network’.*

```
-----------------------------------------------------------------
-- Table NETWORK (_CityObject)
-----------------------------------------------------------------
DROP TABLE IF EXISTS citydb.uom5_network CASCADE;
CREATE TABLE         citydb.uom5_network (
    id                   integer PRIMARY KEY,    -- FK citydb.cityobject (id)
    objectclass_id       integer NOT NULL,       -- FK citydb.objectclass (id)
    network_parent_id    integer,                -- FK uom5_network (id)
    network_root_id      integer,                -- FK uom5_network (id)
    class                varchar,                -- FK uom5_lu_EN_network_class (id)
    function             varchar,                -- No FK. Use delimiters.
    usage                varchar,                -- No FK. Use delimiters.
    commodity_id         integer,                -- FK uom5_commodity (id)
    impact_id            integer                 -- FK uom5_impact (id)
);
CREATE INDEX uom5_ntw_objclass_id_fkx      ON citydb.uom5_network USING btree (objectclass_id);
CREATE INDEX uom5_ntw_ntw_parent_id_fkx    ON citydb.uom5_network USING btree (network_parent_id);
CREATE INDEX uom5_ntw_ntw_root_id_fkx      ON citydb.uom5_network USING btree (network_root_id);
CREATE INDEX uom5_ntw_comm_id_fkx          ON citydb.uom5_network USING btree (commodity_id);
CREATE INDEX uom5_ntw_impact_id_fkx        ON citydb.uom5_network USING btree (impact_id);
```

FIGURE 3-6. Extract from the script to set up the database, showing the table uom5_network (screen capture from Notepad ++).

```
-----------------------------------------------------------------
-- Table NETWORK_TO_NETWORK
-----------------------------------------------------------------
DROP TABLE IF EXISTS citydb.uom5_network_to_network CASCADE;
CREATE TABLE         citydb.uom5_network_to_network(
    superordinate_network_id    integer NOT NULL,        -- FK uom5_network (id)
    subordinate_network_id      integer NOT NULL,        -- FK uom5_network (id)
    CONSTRAINT uom5_network_to_network_pk
    PRIMARY KEY (superordinate_network_id, subordinate_network_id)
);
CREATE INDEX uom5_ntw_to_ntw_superntw_id_fkx ON citydb.uom5_network_to_network
             USING btree (superordinate_network_id);
CREATE INDEX uom5_ntw_to_ntw_subntw_id_fkx   ON citydb.uom5_network_to_network
             USING btree (subordinate_network_id);
```

FIGURE 3-7. Extract from the script to set up the database, showing the table uom5_network_to_network (screen capture from Notepad ++).

FIGURE 3-6 and FIGURE 3-7 show only a small part of one out of five scripts that together form the physical model of the database. These scripts are:

- 01_uom5_FUNCTIONS.SQL. Contains the code to set up general stored procedures for the management of the data.
- 02_uom5_DML_FUNCTIONS.sql. Contains the code to set up specific stored procedures aimed at inserting and deleting data exclusive to the ADE.
- 03_uom5_TABLES.SQL. Contains the code to set up the database tables.
- 05_uom5_TABLE_DATA.sql. Contains the code to update the tables with the values from code lists and enumerations, and to fill the values of the object class table.
- 06_uom5_VIEWS.sql. Contains the code to set up custom views that are simply table JOINs.

Chapter 4
TREATMENT VALIDATION

In this thesis, the validation phase of the design cycle begins with the creation of the model of the artefact, which includes the prototype database and the connection to a GIS application (**Section 4.1**). Next, I design the model of the context, which consists of two parts. The first one is the input data for validation, which is a dataset of utilities and street that belongs to the University of Twente. Having this data in the database requires the processing shown in **Section 4.2**. The second part of the model of the context is the design of the test case, which I show in **Section 4.3**. Finally, in **Section 4.4** I show the validation method and results.

4.1. Model of artefact

4.1.1. Installation of the prototype database

A simple but necessary step is setting up the prototype database. It requires performing the following preliminary steps in the specified order. First, downloading and installing the RDBMS PostgreSQL. Second, creating a new empty database with its name preferably in lowercase. Then, adding four extensions to the database with the query tool, namely '*postgis*', '*postgis_raster*', '*uuid_oosp*', and '*pgrouting*'. Third, downloading and installing the 3DCityDB version 3.3.1. Fourth, downloading and installing both the 3DCityDB utilities package and the metadata module (Agugiaro, n.d.-b).

After these preliminary steps have been completed the database can be installed in the computer by using the SQL scripts from section **3.5.2.2**. However, the database is empty at this stage, and thus, the available dataset needs to be transformed and uploaded to it.

4.1.2. Connection to GIS application

QGIS is a GIS application that, unlike the RDBMS, allows visualizing the results of queries in the form of maps with layers. Moreover, QGIS is the GIS application used by C&FM. I created a connection to the database from QGIS and used the plugin 'DB Manager' to interact with the database via a query tool and load in the map the results of the queries (***FIGURE 4-1***).

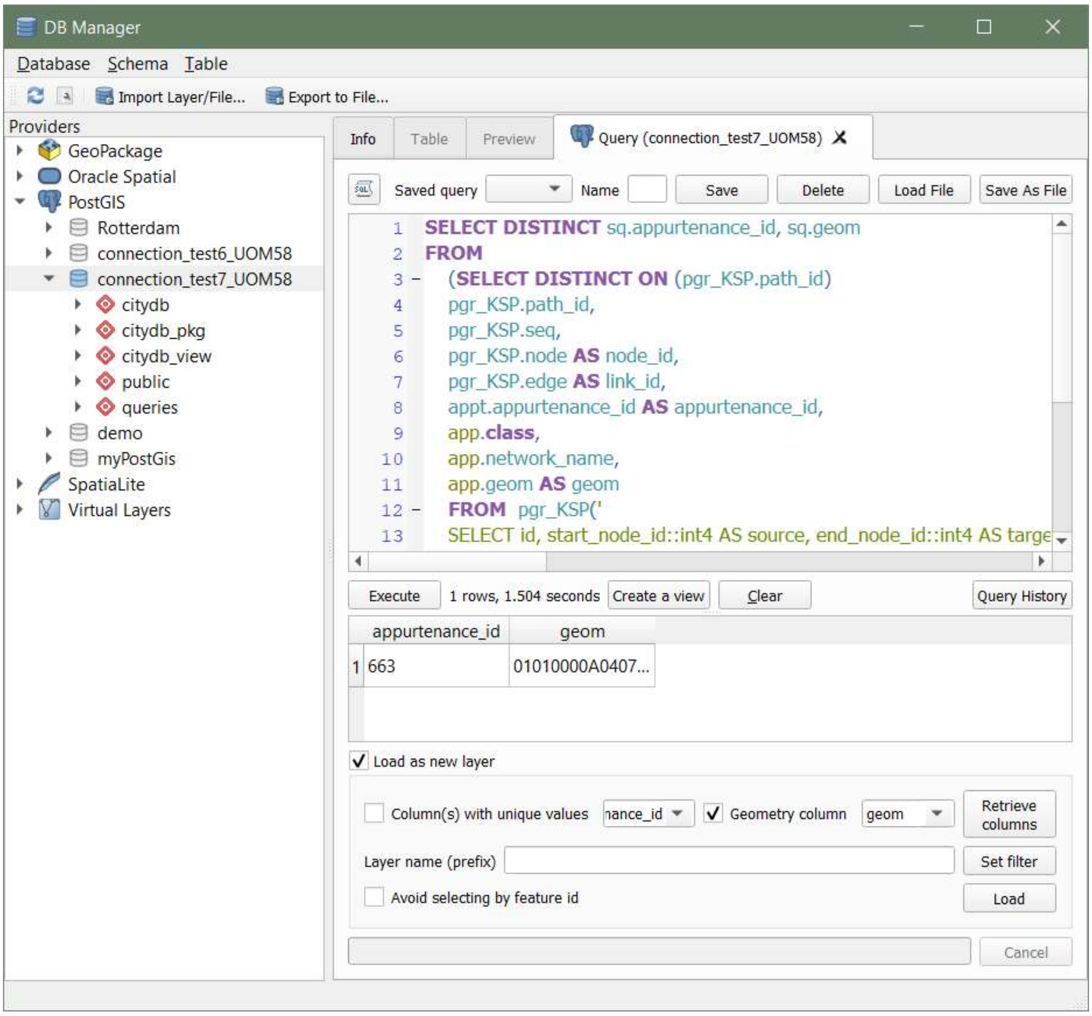


FIGURE 4-1. DB Manager from QGIS showing a query.

4.2. Data gathering, transformation and uploading

I designed a workflow that permits to transform the spatial and semantic data obtained from the campus of UT (in a spatial data format) to conform to the O&M-model and then to upload it to the prototype database. This step is explained in section 4.2.2.

However, before being able to perform the transformation I had to obtain a dataset and pre-process it to correct digitization issues. This is shown in section 4.2.1.

4.2.1. Data gathering and pre-processing

4.2.1.1. Utilities

The dataset from this research was provided by C&FM via a Web Feature Service (WFS). The data was in a simple PostGIS spatial database consisting of only two tables: one for linear elements such as pipes and cables, and the other for nodal elements, such as appurtenances.

I first used the WFS to load as two separate map layers the two full tables from the database. Next, I downloaded to my computer the two map layers as two shapefiles, because of the relative simplicity to handle (i.e. edit) those kinds of files. Next, I loaded the resulting shapefiles to QGIS to explore the networks visually. After assessing the relative difficulty to map each of the eight utility networks contained in the dataset, I chose the **gas low pressure** and **district heating networks (supply and return)**.

Next, I used the filtering functionalities of QGIS to create a single shapefile per element of interest (pipe, protective element, and appurtenance) for each network (gas, district heating supply, and district heating return). This facilitates the input for the software used in the data mapping and upload process.

Finally, I used the error detecting tools provided in QGIS to explore the chosen networks and correct the typical digitization issues found in datasets containing utility networks. A detailed account on the acquisition of the dataset and its pre-processing is available in **Appendix C**.

4.2.1.2. Topographic features

The department of C&FM owns a second database that contains relief features of the campus such as flowerbeds, streets, pavements, building footprints, trees, etc. I also got access to this database via a WFS and downloaded two of the topographic features as shapefiles, namely the streets' footprints and the areas at the campus with tree cover.

4.2.2. Data mapping and upload process

The first step involved the choice of a software able to both transform the data and to upload it to a PostGIS database. Python can handle reading shapefiles, modifying the data contained, and writing to PostGIS. However, the Python script approach was estimated to be too complicated because of the lack of visual aid when writing the necessary code. On the other hand, the Feature Manipulation Engine (FME) by SAFE Software (n.d.) offers a visual way of presenting transformations. FME is used extensively to perform Extract, Transform and Load (ETL) operations in the CityGML ecosystem (Biljecki et al., 2018; Kutzner, 2016; Yao et al., 2018).

Thus, I chose FME as a platform for the transformation and upload process. I composed an FME workbench that reads the pre-processed shapefiles (section **4.2.1**), transforms the data contained in them to match the structure and semantics of the database schema (section **3.5.2**), and uploads the transformed data into the corresponding database table in the prototype spatial database (section **4.1**).

The main challenge of this step is to digitally reconstruct the chosen utility networks starting from the simple data structure in the University of Twente campus database and ending in a much more complex structure (compliant to the O&M-model). A detailed explanation of selected transformations is available in **Appendix D**. The full workflow with explanatory comments is freely available at a GitHub repository (Fossatti, 2020b).

I also transformed and uploaded the selected topographic features at the campus (section **4.2.1.2**). This was possible because the prototype database for utility networks is built on top of 3DCityDB, the database of CityGML. CityGML is meant for representing cities, and thus, it allows to store the topographic features as well. The purpose of uploading these types of data to the prototype data is to have more realistic tests, as explained in the next section.

4.2.2.1. Results

As shown in **Appendix C.3**, the original number of features received from campus related to gas low pressure and district heating, and including all pipes, appurtenances and protective elements was 1294. Due to the simplicity of data model from the University, these **1294 features** were originally represented with an equal number of database tuples. After the transformation and population of the database, the table *uom5_network_feature* —which is comparable in its purpose and in what attributes it stores to the original campus database— now holds **2084 features**. However, this is just 1 out of 16 tables of the new database written with data. Thus, the new structure of the data is far more complex, as evidenced by the summary of tuples written per database relation, presented in **Table 4-1**.

Database table (relation)	# of tuples
uom5_node	3,064
uom5_link	3,042
uom5_network_to_network_feature	2,610
uom5_feature_graph_to_network_graph	2,564
cityobject	2,088
uom5_network_feature	2,084
uom5_feature_graph	2,050
uom5_network_feat_to_material	1,048
uom5_dimensions	1,048
uom5_functional_component	1,036
uom5_pipe	1,014
uom5_protective_element	34
uom5_material	8
uom5_network	4
uom5_network_graph	4
uom5_commodity	2
total	**21,700**

Table 4-1. Database populated with campus dataset. The number of tuples per database table.

4.2.3. Artificial attributes for tests

Most of the 31 attribute values that came with the original datasets were empty. Thus, to have meaningful tests I had to generate artificial (or 'dummy') attributes with more information. I created them in an Excel file that mimicked the structure of the database tables. I then used a custom Python script to read these data and upload them to the database using its stored procedures. The test case and its information needs (see **Section 4.3**) determined which tables and attributes had to be written, namely:

- CityGML tables:
    - Address
- Utility Network O&M tables:
    - Maintenance activity
    - Related party
    - Related party to maintenance activity
    - Related party to network
    - Network feature to maintenance activity
    - Performance
    - Cost
    - Identification
    - Measured depth
    - Soil and groundwater

4.3. Development of a test case

This section explains the process to develop the test, as well as its execution.

4.3.1. Finding a use case and linking it to the competency questions

I reviewed the competency questions together with a representative of C&FM and he proposed a realistic test case for the database. Street reconstructions at the campus are frequent and the database could potentially support their planning by reducing uncertainty in key aspects. These aspects are the identification of buried assets in the streets' vicinity, detailing their maintenance and performance history, and locating important valves before commencing the reconstruction.

I then decomposed the street reconstruction into three sequential phases:

1. **Phase 1**. Identification. After determining which street will be reconstructed it is necessary to identify all components of the utility networks that lie beneath it.
2. **Phase 2**. Decision. Following the identification of the affected components, a management decision must be made regarding whether to keep the components as they are, perform maintenance or replace them entirely taking advantage of their uncovering during the reconstruction. The decision is supported by information about the maintenance and performance history of the components.
3. **Phase 3**. Execution. Finally, the phase when the on-site activities will be carried out requires knowing the location of the nearest valves to shut off the flow to the affected area, as well as the site conditions to estimate the cost and safety measures for the trenching (shoring support).

Next, I linked the information needs of each phase to the competency questions of the O&M Domain Ontology, as shown in ***Table 4-2***.

#	Use case / competency question of the O&M-model
1	**Assessment of the functioning and usage of a utility network and / or its individual components.**
1.1	What kind of commodity does the utility network distribute?
1.2	What is the function of the utility network and / or its components?
1.3	What is the actual usage of the utility network and / or its components?
1.4	What is the state of operation of a utility network and / or its components?
2	**Identification of a utility network and / or its individual components.**
2.1	What is the location of a utility network and / or its components in the (1) x-, (2) y-, and / or (3) z-coordinate?
2.2	What is the spatial accuracy of the utility network and / or its components?
2.3	What are the interior and exterior dimensions of the utility component?
2.4	What is the shape of the utility component?
2.5	What is the (main) colour of the utility component?
2.6	What exterior, interior and filling material does the utility component exist of?
2.7	What is the unique ID of the utility network and / or its components?
3	**Assessment of dependencies and redundancies between (1) utility networks, (2) a utility network and its components, and (3) components.**
3.1	How are utility networks related to each other?

#	Use case / competency question of the O&M-model
3.2	How are utility networks and their components related to each other?
3.3	How are utility components within a single network related to each other?
3.4	How are utility components within distinctive networks related each other?
4	**Mapping the impact of a utility network and / or its individual components.**
4.1	What is the social impact of the utility network and / or its components?
4.2	What is the environmental impact of the utility network and / or its components?
4.3	What is the economic impact of the utility network and / or its components?
5	**Mapping the risks involved within a network and its individual components.**
5.1	What is the risk to damages due to excavation?
5.2	What is the risk to damages due to vegetation?
5.3	What is the risk to damages due to settlement of the subsurface?
5.4	What is the risk to function loss due to deterioration or any other reason?
6	**Assessment of the performance of a utility network or its individual components.**
6.1	Which performance requirements are set for the utility network and its components?
6.2	What is the nominal flow of a commodity through a distribution line?
6.3	What is the operational flow of a commodity through a distribution line?
6.4	What is the maximum flow of a commodity through a distribution line?
6.5	What is the age of the utility component?
6.6	For how long has the utility component been in use?
6.7	What is the expected lifespan of the utility component?
7	**Planning of operations and maintenance related activities.**
7.1	Who is the owner of the utility network and / or its components?
7.2	Who is the operator of the utility network and / or its components?
7.3	When was the last maintenance activity performed?
7.4	What has been done during the last maintenance activity?
7.5	How many utility components are enclosed within a particular protective component?
(new)	What are the surrounding soil and groundwater properties in the study area?
(new)	What is the measured depth of the components?

Table 4-2. Relation of competency questions to phases of the street reconstruction. **References**: Phase 1 (identification) in red, Phase 2 (decision) in blue, and Phase 3 (execution) in green.

4.3.2. Translation of competency questions into SQL queries

After identifying the competency questions that would be part of the test (***Table 4-2***) I used Structured Query Language to compose 9 queries, as follows:

1. Phase 1. Identification.
    a. Query 1: Selecting the street to be reconstructed.
    b. Query 2: Identifying all pipes affected (showing relevant attributes)
    c. Query 3: Identifying all appurtenances affected (showing relevant attributes)
2. Phase 2. Decision.
    a. Query 4: Showing maintenance records of affected pipes
    b. Query 5: Showing performance history of affected pipes

3. Phase 3. Execution.
    a. Query 6: Characterizing the soil and groundwater around each pipe
    b. Query 7: Finding relevant gas valves
    c. Query 8: Finding relevant district heating supply valves
    d. Query 9: Finding relevant district heating return valves

The scripts are shown in **Appendix E**

4.4. Validation

To study the validation model, I applied a combination of research methods. The first method that I applied was a ***'single-case mechanism experiment'***, as was shown in **FIGURE 2-2**. I applied stimuli (computer queries) to the validation model (model of artefact + model of context), which produced responses. In turn, the usefulness can be assessed by its intended users by studying the responses/outputs and how the system works.

This leads to the second method that I applied, which is '***expert opinion'***. I showed the tests of the database to experts in utility networks and data modelling. Practically, I organized this in two ways:

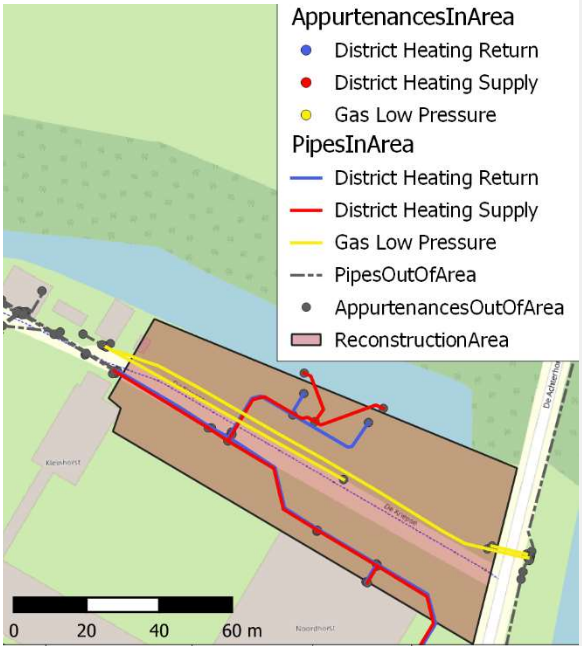
- First, I arranged a virtual meeting with two experts in data modelling from the Dutch Cadastre, a representative from C&FM, an expert in data modelling from TU Munich, and three of my supervisors. During the meeting, I explained the project background and did a live demonstration of the use of QGIS in combination with the database derived from the O&M-model. After the live demonstration, the participants were invited to share their feedback.
- To expedite the following meetings, I pre-recorded a video (Fossatti, 2020a) with the explanation and live demonstration part. After having watched the video, I asked their opinion on the scope of the database and the greater context of interoperability, according to their area of work. I consulted in this way an expert in data modelling from a Dutch standardization agency, and three experts in asset management from a consulting firm.

Since the experts whom I consulted came from different backgrounds —including network management, maintenance, standardization, and data modelling— there was sufficient coverage of the two main uses for the database (as a stand-alone GIS tool, and as a means to formally test the comprehensiveness of the O&M-model).

4.4.1. Validation model responses (i.e. effects)

In ***FIGURE 4-2*** I present two examples of the results of executing the queries related to the simulated street reconstruction and derived from the competency questions. On the left, I show only the graphical result of the queries of the identification phase (1, 2, and 3). The result is a visualization of all pipes, appurtenances under the reconstructed street, and of the street itself. For the sake of brevity, I do not show here the tabular results linked to this graphical presentation. On the right, I show a tabular view of the results from query 6 (characterization of soil and groundwater) for one pipe. The query can be easily expanded to show more of the soil and groundwater characteristics as they become available.

The presentation of the results is not well suited for the report format, so the reader is invited to interact with the system, freely available at the online repository (Fossatti, 2020b).



id	class	function_of_line	status	network_name
1137	thermalpipe	returnLine	functional	District Heating Return
1373	oilgaschemicalspipe		outOfService	Gas Low Pressure

trenching_information - Feature Attributes	
id	1611
status	functional
network_name	Gas Low Pressure
location_quality	surveyed
location_accuracy	30 cm
exterior_diameter	75
exterior_diameter_unit	mm
depth_quality	surveyed
depth_level_measurement	-1,5
depth_level_measurement_unit	m
depth_reference_level	groundLevel
depth_accuracy	30cm
date_of_depth_measurement	1995-10-05
soil_type	silty sand
groundwater_level	-3
groundwater_level_unit	m
groundwater_reference_level	groundLevel
colour	yellow

FIGURE 4-2. **Left.** Visualization of the graphical part of queries 1, 2 and 3. **Right**: Tabular presentation of the results of query 6.

4.4.2. Do the effects satisfy the requirements?

***RO** The encoding must allow the representation and handling of the data that are necessary to answer the competency questions for which the ontology was designed by Ter Huurne (2019, pp. 28–29)*

19 out of the 34 original competency questions —plus the 2 new ones at the end of ***Table 4-2***— were **successfully operationalized** into formal queries, bringing the totals to 21 out of 36. Even though the operationalization (i.e. translation into queries) was deemed successful, the experts had comments on the outputs of 2 of them, and I noticed a problem with another one. I elaborate that below.

Competency question: 7*.4 What has been done during the last maintenance activity*? Clarification: the original O&M-model stores the maintenance information as a Uniform Resource Identifier (URI) pointing to a file in a computer detailing the activity. Because C&FM could not provide me with a maintenance report, and to make the explanation of the tests clearer and more didactic I replaced the URI with a description of the task executed and showed that as the outcome.

The experts pointed out the difficulty of representing the unique maintenance activities associated with the different utility networks using only one table column to describe the activity. This problem would probably not occur should a URI pointing to the location of the information be used (as was originally intended). I consider that it may be beneficial to expand the maintenance table to include (1) the URI, (2) a standardized and brief description of the activity, and (3) a separate explanation on the outcome of the activity.

Competency question*: (new) What are the surrounding soil and groundwater properties in the study area*? An expert commented that in the Netherlands there is a freely available databank containing soil surveys of the whole country (DINOloket, n.d.). Thus, he suggested exploring if it would be convenient to obtain missing data for the table *uom5_soil_and_groundwater* from that site.

Competency question*: 6.1 Which performance requirements are set for the utility network and its components*? C&FM currently lacks a scale to measure performance, so I arbitrarily set it as 1-very low to 5-very high. Moreover, I only explored engineering performance which leaves serviceability, safety, financial, and sustainability performance untested.

The last piece of feedback about **R0** was an expert's comment on the difficulty of knowing what kind of Operation & Maintenance data to store about **appurtenances** because of the great variety of nodal features present in networks. The O&M-model and the derived database have stronger representation capabilities of linear elements than of nodal ones, so it is more difficult to answer the competency questions related to appurtenances than to pipes and cables. The expert suggested contacting the Dutch major utility owners to discuss improvements in the representation of appurtenances.

R1 *The encoding must be useful for C&FM and other organizations that manage utility networks, i.e. the stakeholder must have a favourable opinion about its utility for asset management*

- A team of experts from an asset management firm had a favourable overall opinion on the system and its usefulness. Specifically:
  - An expert in the engineering design of utility networks commented favourably on the capability of the database to store and show the z-coordinate. He said that this is important for the design and execution of network expansions when network clashes must be accounted for and resolved. He suggested improving the presentation of the test results by using 3D visualizations.
  - An expert in the inspection of gas pipes deemed that the model would be able to provide him with the data necessary to plan inspection campaigns for leak detection. These are based on the locations, materials, and age of the pipes, information that can be stored in the database. Moreover, he viewed favourably the capability of recording previous maintenance activities in the database. With enough data, this would allow to predict where to inspect in future campaigns
- Several experts expressed favourable views on the chosen technologies. Specifically, they pointed out that QGIS, PostgreSQL/PostGIS are open, free, and state-of-the-art and that FME is practically the industry standard for Extract-Transform-Load (ETL) operations. One expert suggested looking into OpenLayers, a well-known JavaScript library to publish maps as web pages.
- An expert involved in the development of IMKL commented favourably on (a) the topological capabilities of the O&M-model and (b) on the usefulness of having a city model (CityGML) as a base for the utilities model. This allows representing different kinds of assets in a single model. He also mentioned that using the prototype database as a backend component to a GIS in the Netherlands would be possible because the mapping to IMKL (which is compulsory) using ETL tooling would not be complicated.
- However, a representative from C&FM noted that the variety of IT skills necessary to manage the system would be a barrier for the department at present. Moreover, the lack of official workflows and documentation for the Utility Network ADE poses a technical barrier even to experts. There are no best practices documents on data mapping, on how best to represent the network's topology, and on how to manage at the database level tasks such as a pipe replacement (i.e. a step-by-step guide on how to update the database in such a routine event). The present level of maturity of the system

is one of two reasons why the full potential of the system to support asset management cannot yet be exploited

- The other reason is the availability of data. Nearly all experts commented on the lack of data as a major barrier for the model to be useful for asset management. They pointed out the high costs involved in surveys, the time these campaigns take, and the uncertainty (as of now) on what to record. However, this thesis might contribute to a better understanding as far as this uncertainty is regarded. Asset management is about data-driven decisions, so if C&FM lacks data and does not plan to collect more in the short term, the use of a more complex database than their present one is unjustified. It would add higher entry barriers for users and lack the data to be of much use.

***R3** The encoding must allow testing the comprehensiveness of the O&M-model from which the data transfer format is derived*

The database encoding in combination with the GIS application allows storing, manipulating, analysing, and presenting all the concepts represented in the O&M-model, as shown in the test. So, in a superficial sense, this requirement can be considered satisfied. This is because the database allowed to test the O&M-model's comprehensiveness with a simulated but realistic test case. However, some of the experts said that it would be more interesting to test the format itself, as explained next.

There was great demand from experts from different backgrounds (both in data modelling and in asset management consultancy) for seeing a simulated data exchange using the data transfer format from the O&M-model. In other words, to see the use of the O&M-model as a solution to interoperability issues. This was, however, out of the scope of this study. They suggested involving in such tests the largest utility network owners in the Netherlands (e.g. Enexis, Stedin, and Liander) and their contractors. Some questions the experts had that could not be answered with the help of the test are:

(a) Does the use of the ontology allow contractors to operate on their own data environment, without the need to maintain several technical environments dedicated to communication with their clients?
(b) Is the utility operator able to share information more easily with one or more of its contractors without vendor lock-in, and sharing the same data with all of them?
(c) How much work is it needed to produce the export and import modules to enable the exchange of data?

Chapter 5

DISCUSSION, CONCLUSIONS, RECOMMENDATIONS AND FUTURE WORK

5.1. Discussion

Utility owners and managers normally use systems with spatial databases as backend for storage and management of the data. These databases tend to be unique for each organization, structured according to a schema corresponding to the organization's ***internal*** data model. The uniqueness of the organizations' databases means that not only syntactic encoding is necessary when systems exchange information, but also the semantic mapping between the concepts and terminology of the different models.

In this research I propose a different approach, namely, to use as the organization's database schema for asset management the ***open*** O&M Domain Ontology. If the O&M Domain Ontology were to become a standard for data exchange, using a database derived from it would simplify one of the mapping steps necessary to exchange the data. This is because the data from one of the systems would already conform to the application schema of the transfer format. Furthermore, the O&M-model was designed to cover multiple types of utilities in a single schema, which means that most types of networks could be supported. Thus, in principle, most utility owners could potentially migrate to the O&M-model and expand it modularly according to their needs. *This study contributes to the scientific literature by showing that an open conceptual model with a broad scope can be effectively used —in the form of a database— as a lifecycle asset management support tool by an organization that holds different types of networks.*

The tests on the database and the feedback from the experts showed that the database can be successfully used as an asset management support tool for routine engineering tasks. However, during the test case, this application was greatly hampered by the lack of data owned by C&FM. The solution is to acquire data over the years. Thus, the most urgent task at present is planning how to deal with data collection to enrich the existing datasets. *This study contributes to the scientific literature by providing a guide on relevant lifecycle data to collect during field surveys.* Moreover, there is the potential of a virtuous cycle of using the surveyed data to improve the classes in the O&M-model, and its related database and data transfer format schemas.

The competency questions that were **not formalized into queries** are shown in black in ***Table 4-2***. As the table shows, two whole use cases were not trialled, for the reasons explained next.

- *Use case 4 'Mapping the impact of a utility network and / or its individual components'* was left out of the tests for two reasons. First, there was no impact data in the original dataset, thus I would have had to create 'dummy' impact data. However, and unlike the other dummy data that I created, impact data is difficult to create because it requires studies specific to the context, type of network, surroundings, etc. These were out of the scope of the thesis. Second, I was unsure about how to upload data to the table 'uom5_impact'. Specifically, it was not certain to me whether it would be practical to upload data at the component level (very localized impacts) or the network level (widespread impacts). Experts pointed out that from an asset management perspective impacts are very important to know, so focus should be given to this topic.
- Similarly, *use case 5 'Mapping the risks involved within a network and its individual components'* and its associated competency questions were not tested. This would require to not only determine the impacts but also to do an assessment of the probability of occurrence of an event. It is not clear how to operationalize this with the database in its current form.

Both of these use cases are complex enough to be studied on their own and are too broad to be fitted in a research project dealing with other issues.

Finally, two experts on data modelling commented on the design decision to derive the O&M-model by re-writing the Utility Network ADE v0.9.2 instead of expanding it. This could cause interoperability issues between data structured using the O&M-model and data structured using the Utility Network ADE. Moreover, improvements in the Utility Network ADE would be harder to implement in the O&M-model, and by extension in the database encoding of the O&M-model. Thus, I discuss integration possibilities with the Utility Network ADE v0.9.4 in **Appendix F**.

5.2. Conclusions

The goal of the project was to design an encoding of the O&M Domain Ontology that can be used by asset managers and owners as a component of a GIS for the management of utility networks. Thus, I designed a spatial-relational database and validated that it is fit for this purpose.

Before transforming the original formal conceptual model to the database physical model, I simplified and enhanced the former to obtain a more compact database, in line with CityGML. I based the simplification on design decision from CityGML (and 3DCityDB), and the Energy ADE (and its database encoding). Thus, the new O&M-model (v.0.5.8) differs from the original (v.4.1). Moreover, based on the design choices by the CityGML team, I chose the relational paradigm as Implementation Schema, and free database software as a platform for implementation, namely PostgreSQL/PostGIS.

I did the transformations from levels of abstraction (conceptual to logical to physical) manually because the automatic transformation tool that I trialled returned a complicated database physical model. I obtained the criteria for the manual mapping from the documentation of 3DCityDB and the Energy ADE database. Next, I implemented (i.e. installed using the database scripts) a prototype of the spatial-relational database and connected it to a GIS (QGIS, also free). The scripts to create the database, as well as diagrams showing the structure of the tables are freely available online (Fossatti, 2020b).

I used a dataset containing two out of a possible eight utility networks of the campus of the University of Twente and a dataset containing the streets to create a test case for the database. The data transformation and upload process that I developed uses SAFE software FME, a proprietary ETL tool. The dataset was not rich in terms of attribute data, so I had to add 'dummy' data to make the tests meaningful.

I validated the database by performing a single-case mechanism experiment combined with expert opinion. The test was the simulation of a routine street reconstruction at the campus and it allowed me to show graphically the support that the database offers to asset managers. To create the stimuli for the test I translated approximately half of the competency questions of the O&M-model into SQL queries. The system returned the expected results and uncovered areas for improvement in the representation of appurtenances and maintenance activities. Further, in the Netherlands, the data for the class surrounding soil and groundwater could be obtained from DINOloket, an online repository containing soil surveys.

Of the other half of the competency questions, I purposefully left out of scope the ones related to impacts and risks (use cases 4 and 5 respectively, from ***Table 4-2***) because of their intrinsic complexity, which necessitates a separate study. The rest of the competency questions that I left out of the study simply had little relevance to the test case.

I also set out to test the comprehensiveness of the O&M-model taking advantage of the tests necessary to validate the database. The database technology I chose allows assessing practically how comprehensive the model is by showing how instances of the abstract classes from the O&M-model are stored in practice. However, experts commented on the need to simulate realistic data exchanges between utility owners and contractors for two reasons. First, such organizations hold realistic maintenance information, which could be beneficial for assessing the representation capabilities of the O&M-model through the database. Second, the main use case for the ontology is as a solution for interoperability. There is a need to show the use of the ontology in a standardized data exchange over the internet.

5.3. Recommendations

This study both shows that an open data model can be effectively used as an asset management support tool, and guides asset manager on what data they need to collect for rationalising their decision making

- The biggest challenge is to rationalize and standardize the collection of operation & maintenance data through field surveys, whenever such an opportunity presents itself at the campus (e.g. when a component is uncovered). The lack of data precludes using the database to do advanced analysis, such as predictive maintenance.
- An organization using the database in the Netherlands would still need to comply with the WIBON law. That can be done by creating a process to transform the data in the proposed system to the IMKL data model to submit it to the Kadaster.
- The 6 remaining utility networks at campus would need to be transformed and uploaded to the database. I estimate that 1 person needs about 2-3 months to first get acquainted with the project and then transform the water, sewer-gravity, sewer-under-pressure, medium-voltage, low-voltage, and data transport networks and upload them into the database. The sewer network appears to be the more complex one. This could be done during another master assignment that addresses another aspect of utility networks and management.
- Further improvements to the data model and database (classes and attributes) can be done by a student as a Master's project. The student could work embedded in C&FM for 5-6 months and try out the proposed system in practice. The student and C&FM can judge together based on the practical experience with its use what improvements the model needs. For example, an attempt to load real maintenance data will probably uncover room for changes in the model and the database.
- I estimate that a partial, trial implementation of the system as a replacement of the existing database for utility networks would require C&FM to either hire 1 person or outsource this to a GIS consultancy firm. The knowledge requirements for implementing the system are:

    - Experience with the 'enterprise' use of databases (versioning, backups, permissions, roles, etc.)
    - Knowledge of GIS (e.g. a Geomatics expert)

  It would be a full-time job during the first few months, but once the database has been implemented, the procedures for handling data created, and all utility networks loaded, specific interventions would suffice. The database only requires actions every time there are changes in the network, or new information from topographic research becomes available.

- Finally, full implementation of the system as a full-fledged GIS would require additional knowledge in programming, to enable:
    - Designing the input-output interfaces of the system (e.g. a Web Feature Service)
    - Web-mapping (using the maps powered by the database on the web)
    - Format definition (the way data is saved in the system)
    - Design of procedures for data management and data quality checks

  This may entail hiring an additional expert or outsourcing to a GIS consultancy firm. However, I cannot say this would be cost-effective for the campus, given the relatively small scale of their networks.

5.4. Future scientific work

Based on the discussions with experts and my own experience with the database and the O&M-model I propose three future lines of research.

5.4.1. Testing the use of the ontology to exchange utility data

The GML data transfer format schema of the O&M Domain Ontology can be easily derived using freely available tooling (ShapeChange Development Team, n.d.). This schema could potentially be used as the 'system-independent' transfer format to enable exchanges between two systems, as shown in **Figure 1-5**. The experts consulted as part of this study expressed great interest in seeing the use of the ontology for enabling standardized data exchanges.

To enable such transfer, the researcher should compose the processes to transform the data for the source and the target organizations. The processes shown in **Figure 1-5** are:

- Data transformation M_{I_A} (**M**apping **I**nternal Schema to **A**pplication Schema). Semantic mapping of the internal schema of the source organization to the external application schema (O&M-model).
- Data transformation E_{SD_SI} (**E**ncoding **S**ystem **D**ependent to **S**ystem **I**ndependent). Syntactic encoding of the system-dependent source data into system-independent transfer data.
- Data transfer from the source to the target system.
- Data transformation E_{SI_SD} (**E**ncoding **S**ystem **I**ndependent to **S**ystem **D**ependent). Syntactic encoding (decoding) of the system-independent transfer data into the system-dependent target data.
- Data transformation M_{A_I} (**M**apping **A**pplication Schema to **I**nternal Schema). Semantic mapping of the external application schema to the internal schema of the target system.

Academic level	MSc
Required background	Geo-information sciences (or IT) with some understanding of civil engineering.
Host organization	Utility owner (preferably Enexis, Liander, or Stedin) and one contractor.

5.4.2. Improving the ontology by using the database in practice

The database derived from the O&M-model in this study was not transferred to the department of C&FM of the University of Twente. It remains as a prototype with no real implementation. The use in practice of the database to record the maintenance activities and new data from surveys would enable to further test its usefulness as a support system and to improve the classes in the O&M-model.

The activities needed for such research would be:

- Installing the prototype database to be used in parallel with the current system used by the client.
- Mapping the utilities of the client to the database. In case of working with C&FM, mapping the remaining 6 utility networks to the database.
- Using the database to support maintenance in practice and to capture new data from surveys.
- Reflecting upon possible improvements to the database and the O&M-model.

Academic level MSc
Required background Civil engineering with some experience in geo-information sciences
Host organization Utility owner (e.g. C&FM, Enexis, Liander, or Stedin)

BIBLIOGRAPHY

3DCityDB. (n.d.). *3D City DB*. Retrieved January 17, 2020, from https://www.3dcitydb.org/3dcitydb/

3DCityDB Development Team. (n.d.). *3DCityDB GitHub Repository*. Retrieved January 22, 2020, from https://github.com/3dcitydb

3DCityDB Development Team. (2016). *3D City Database for CityGML version 3.3.0*. https://www.3dcitydb.org/3dcitydb/fileadmin/downloaddata/3DCityDB_Documentation_v3.3.pdf

Agugiaro, G. (n.d.-a). *3dcitydb_utility_network_ade GitHub Repository*. Retrieved January 22, 2020, from https://github.com/gioagu/3dcitydb_utility_network_ade

Agugiaro, G. (n.d.-b). *3DCityDB Metadata module - GitHub Repository*. Retrieved January 23, 2020, from https://github.com/gioagu/3dcitydb_metadata_module

Agugiaro, G., & Holcik, P. (2017a). *3D City Database extension for the CityGML Energy ADE 0.8* (Issue September). https://github.com/cstb/citygml-energy/tree/master/doc/guidelines

Agugiaro, G., & Holcik, P. (2017b). *3D City Database extension for the CityGML Energy ADE 0.8 PostgreSQL Version* (Issue September). https://github.com/gioagu/3dcitydb_energy_ade/blob/master/manual/3DCityDB_Energy_ADE_0.8_Documentation.pdf

Becker, T., Nagel, C., & Kolbe, T. (2012). Semantic 3D Modeling of Multi-Utility Networks in Cities for Analysis and 3D Visualization. *Progress and New Trends in 3D Geoinformation Sciences*, 41–62. https://doi.org/10.1007/978-3-642-29793-9-3

Becker, T., Nagel, C., & Kolbe, T. H. (2011). Integrated 3D Modeling of Multi-utility Networks and Their Interdependencies for Critical Infrastructure Analysis. *Advances in 3D Geo-Information Sciences*, 1–20. https://doi.org/10.1007/978-3-642-12670-3_1

Biljecki, F., Kumar, K., & Nagel, C. (2018). CityGML Application Domain Extension (ADE): overview of developments. *Open Geospatial Data, Software and Standards*, *3*(1), 1–17. https://doi.org/10.1186/s40965-018-0055-6

Boates, I., Agugiaro, G., & Nichersu, A. (2018). Network modelling and semantic 3d city models: Testing the maturity of the utility network ADE for citygml with a water network test case. *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, *4*(4), 13–20. https://doi.org/10.5194/isprs-annals-IV-4-13-2018

Brown, R. E. (2010). *Business Essentials for Utility Engineers* (1st ed.). CRC press.

Brown, R. E., & Humphrey, B. G. (2005). Asset Management for Transmission and Distribution. *IEEE Power & Energy Magazine*, *may/june*, 39–45. https://doi.org/1540-7977/05/$20.00

den Duijn, X., Agugiaro, G., & Zlatanova, S. (2018). Modelling Below- and Above- Ground Utility Network Features with the CityGML Utility Network ADE: Experiences from Rotterdam. *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, *4*(4/W7), 43–50. https://doi.org/10.5194/isprs-annals-IV-4-W7-43-2018

DINOloket. (n.d.). *DINOloket Data and Information on the Dutch Subsurface*. Retrieved May 30, 2020, from https://www.dinoloket.nl/en/about-us

Enexis. (n.d.). *Open data*. Retrieved January 22, 2020, from https://www.enexis.nl/over-ons/wat-biedenwe/andere-diensten/open-data

Fossatti, F. (2020a). *Data model and database for operation & maintenance of utility networks*. https://www.youtube.com/watch?v=IpXXMTpEVGs

Fossatti, F. (2020b). *Utility-Network-ADE-OM-extension*. https://github.com/FedericoFossatti/Utility-Network-ADE-OM-extension

Geonovum. (2019). *Informatiemodel Kabels en Leidingen (IMKL)*. https://register.geostandaarden.nl/informatiemodel/imkl2015/

Gröger, G., & Plümer, L. (2012). CityGML - Interoperable semantic 3D city models. *ISPRS Journal of Photogrammetry and Remote Sensing*, *71*, 12–33. https://doi.org/10.1016/j.isprsjprs.2012.04.004

Huisman, O., & de By, R. A. (2009). *Principles of Geographic Information Systems* (4th Editio). The International Institute for Geo-Information Science and Earth Observation.

IBM. (n.d.). *What is a relational database?* Retrieved January 17, 2020, from https://www.ibm.com/analytics/relational-database

INSPIRE. (2013). *D2.7: Guidelines for the encoding of spatial data, Version 3.3rc3*.

Kutzner, T. (2016). *Geospatial Data Modelling and Model-driven Transformation of Geospatial Data based on UML Profiles* (Issue November).

Kutzner, T., Hijazi, I., & Kolbe, T. H. (2018). Semantic Modelling of 3D Multi-Utility Networks for Urban Analyses and Simulations: The CityGML Utility Network ADE. *International Journal of 3-D Information Modeling*, *7*(2), 1–34. https://doi.org/10.4018/ij3dim.2018040101

Lake, R., Burggraf, D. S., Trninić, M., & Rae, L. (2004). *Geography mark-up language: foundation for the geo-web*. John Wiley & Sons.

Ledoux, H., Arroyo Ohori, K., Kumar, K., Dukai, B., Labetski, A., & Vitalis, S. (2019). CityJSON: a compact and easy-to-use encoding of the CityGML data model. *Open Geospatial Data, Software and Standards*, *4*(1). https://doi.org/10.1186/s40965-019-0064-0

Nyerges, T. (2010). *GIS Database Primer*.

OGC. (n.d.-a). *CityGML*. Open Geospatial Consortium. Retrieved January 17, 2020, from https://www.opengeospatial.org/standards/citygml

OGC. (n.d.-b). *Geography Markup Language*. Retrieved May 14, 2020, from http://www.ogc.org/standards/gml

OGC. (2012). *OGC City Geography Markup Language (CityGML) Encoding Standard*. https://portal.opengeospatial.org/files/?artifact_id=47842

olde Scholtenhuis, L., Hartmann, T., & Dorée, A. G. (2016). 4D CAD Based Method for Supporting Coordination of Urban Subsurface Utility Projects. *Automation in Construction*, *62*, 66–77. https://doi.org/10.1016/j.autcon.2015.10.013

olde Scholtenhuis, L., Zlatanova, S., & den Duijn, X. (2017). 3D Approach for Representing Uncertainties of Underground Utility Data. *Civil Engineering*, *3*, 326–334. https://doi.org/doi:10.1061/9780784407943

OMG. (n.d.). *MDA® - THE ARCHITECTURE OF CHOICE FOR A CHANGING WORLD*. Retrieved May 14, 2020, from https://www.omg.org/mda/

OMG. (2005). *WHAT IS UML*. https://www.uml.org/what-is-uml.htm

PostGIS Development Team. (n.d.). *PostGIS*. Retrieved January 17, 2020, from https://postgis.net/

PostgreSQL Development Team. (n.d.). *About*. Retrieved January 17, 2020, from https://www.postgresql.org/about/

Publieke Dienstverlening Op de Kaart. (n.d.). *Over PDOK*. Retrieved February 18, 2020, from https://www.pdok.nl/over-pdok

QGIS. (n.d.). *QGIS*. Retrieved January 17, 2020, from https://www.qgis.org/en/site/

SAFE Software. (n.d.). *What is FME?* Retrieved May 28, 2020, from https://www.safe.com/fme/

ShapeChange Development Team. (n.d.). *ShapeChange*. Retrieved May 14, 2020, from https://shapechange.net/

Sohail, M., Cavill, S., & Cotton, P. (2005). Sustainable Operation and Maintenance of Urban Infrastructure: Myth or Reality? *JOURNAL OF URBAN PLANNING AND DEVELOPMENT*, *131*(March), 39–49. https://doi.org/10.1061/(ASCE)0733-9488(2005)131

Steenhuisen, B., Dicke, W., & de Bruijn, H. (2009). “Soft” public values in jeopardy: Reflecting on the institutionally fragmented situation in utility sectors. *International Journal of Public Administration*, *32*(6), 491–507. https://doi.org/10.1080/01900690902861753

ter Huurne, R. (2019). *Modeling utilities by developing a domain ontology* [University of Twente]. https://research.utwente.nl/en/publications/modelling-utilities-by-developing-a-domain-ontology

Truyen, F. (2006). The Fast Guide to Model Driven Architecture. *Practice*, 12. http://www.omg.org/mda/mda_files/Cephas_MDA_Fast_Guide.pdf

University of Twente. (2016). *Long-Term Housing Strategy* (Issue July). https://www.utwente.nl/en/ltsh/

University of Twente. (2019). *FACTS & FIGURES 2019*. https://www.utwente.nl/en/facts-and-figures/

van der Velde, J., Klatter, L., & Bakker, J. (2013). A holistic approach to asset management in the Netherlands. *Structure and Infrastructure Engineering*, *9*(4), 340–348. https://doi.org/10.1080/15732479.2012.657650

W3C. (2006). *Extensible Markup Language (XML) 1.1 (Second Edition)*. https://www.w3.org/TR/2006/REC-xml11-20060816/

WIBON. (2019). *Wet informatie-uitwisseling bovengrondse en ondergrondse netten en netwerken*.

Wieringa, R. (2010). *Design science methodology for Information Systems and Software Engineering*. Springer. https://doi.org/10.1145/1810295.1810446

Yao, Z., & Kolbe, T. H. (2017). Dynamically Extending Spatial Databases to support CityGML Application Domain Extensions using Graph Transformations. *Wissenschaftlich-Technische Jahrestagung Der DGPF in Würzburg – Publikationen Der DGPF*, 316–331.

Yao, Z., Nagel, C., Kunde, F., Hudra, G., Willkomm, P., Donaubauer, A., Adolphi, T., & Kolbe, T. H. (2018). 3DCityDB - a 3D geodatabase solution for the management, analysis, and visualization of semantic 3D city models based on CityGML. *Open Geospatial Data, Software and Standards*, *3*(1). https://doi.org/10.1186/s40965-018-0046-7

APPENDICES

Appendix A. Utility networks & data models

A.1. Introduction

Utility networks exist to convey the diverse commodities which are necessary for the functioning of society, such as drinking water, wastewater, stormwater, water for district heating, gas, oil, chemicals, electricity, and data. Individual networks are associated with the supply or disposal of a single commodity from a source to a supply area or sink. Moreover, utility networks consist of interconnected components that enable the flow of the commodities, and which depend on the type of network (e.g. pipes, pumps and manholes for sewer networks, transformers, and cables for electricity networks). The networks and their components can exist both above or below ground level, depending on factors such as the network type, location, safety aspects, costs, and the soil's characteristics. In an urban environment, they are located predominantly below ground level.

This section presents (1) the ways in which utility networks are owned and managed, (2) how cities are represented digitally in CityGML, (3) how utility networks are represented in an extension of CityGML called the Utility Network ADE, (4) a review of a the Operation & Maintenance Domain Ontology, (5) a comparison with IMKL, and (6) other studies with models for utility networks.

A.1. Ownership and management of utility networks

Ownership of a utility network can be national (by the state), regional (e.g. by a municipality) or private. However, ownership of a utility network does not necessarily imply that it is managed and/or operated by the same party. Moreover, the utility network owner does not necessarily have the right to trade and supply its associated commodity, a function that may be performed by a different organization. As a result –in most nations– there is a myriad of companies involved in the utilities/commodities sector. Each has its own goals and practices and holds unique information about the utilities (olde Scholtenhuis et al., 2016). This growing fragmentation of the utility sector has resulted in more layered decision-making (among more heterogeneous actors), and in a need for coordination between not only organizations responsible for different utility network types, but also organizations responsible for the different functions involved in their management (olde Scholtenhuis et al., 2016; Steenhuisen et al., 2009). Significantly, it has resulted in the development of several data models and formats used to digitally represent, store, manipulate and exchange the information required by the diverse stakeholders about the physical utility networks, including their geometry, position and also non-spatial attributes (Becker et al., 2011).

A.2. CityGML: Digital representation of cities

The diverse origins of the data models for the digital representation of real-world objects entail that they are domain-specific and are often not directly interoperable with each other. Therefore, the exchange of data between organizations (and even within them) often requires costly and time-consuming data conversion steps. Governments, industry, and academia have recognized this issue and created standards such as the Geographic Markup Language (a data format) and CityGML (a data model).

The Geography Markup Language (GML) is a vendor-neutral mark-up language used to describe and represent spatial objects digitally and to share them through the internet (Lake et al., 2004, pp. 3–4). It is a standard from the Open Geospatial Consortium (OGC). In GML spatial objects are called features and can be either concrete or abstract. However, GML does not describe specific features. These are defined in application schemas based on this specification and created by data modelers from different domains/communities (tourism, urban administration, or sewer management to name a few). GML only provides the modelling framework to define said models. The power of these semantic 3D models is that they allow users not only to visualize a digital representation of the world but also to perform complex analyses of their components and their interdependencies based on semantic and topological information.

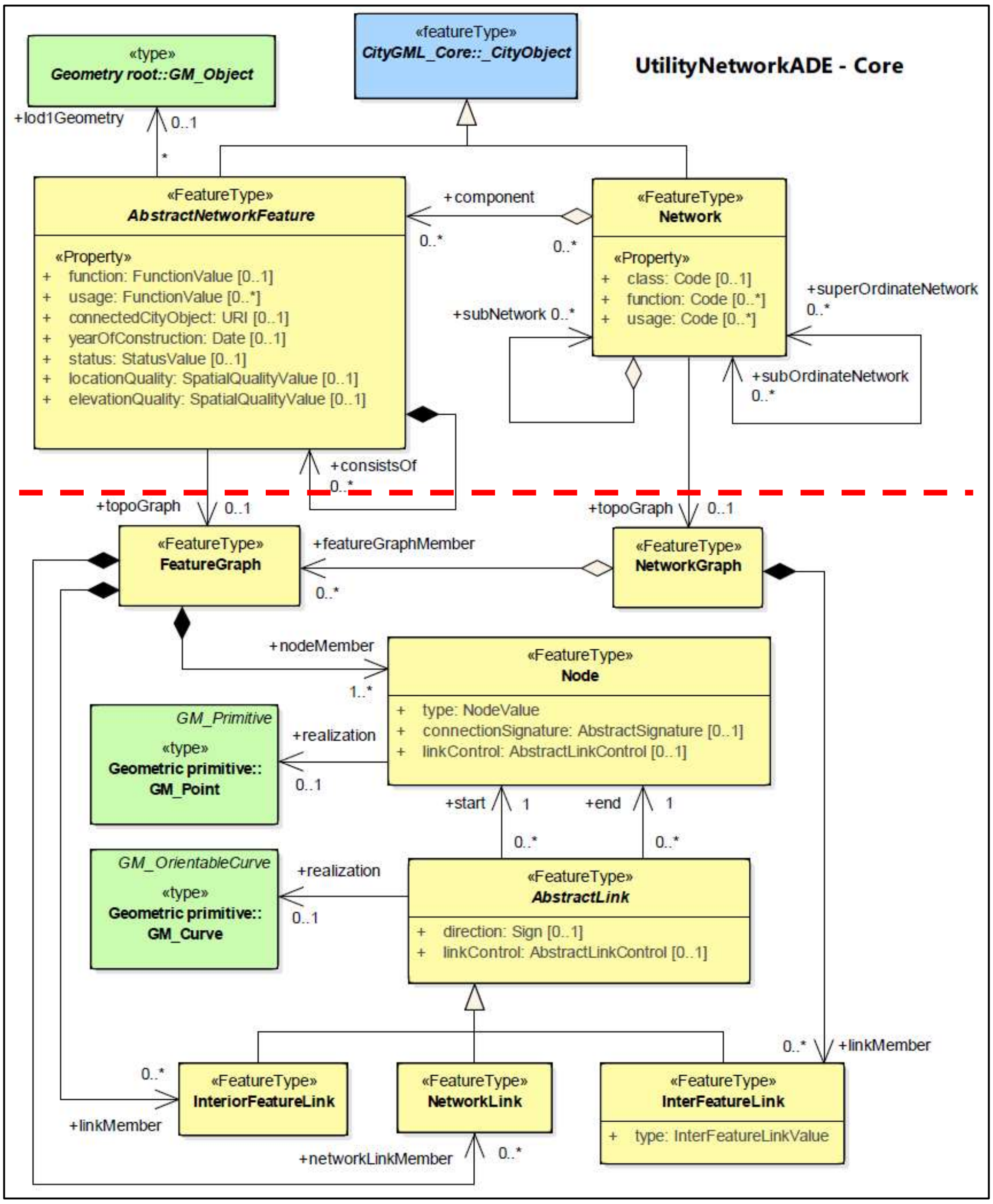
CityGML is one such application schema and a standard from the OGC. This standard is meant for representing and exchanging 3D city models for a broad range of applications. It allows representing a great variety of above-ground objects (multi-theme coverage), their appearance, geometry (in multiple levels of detail), topology, and other semantic information (Gröger & Plümer, 2012). Moreover, it includes a mechanism to extend the city model to other domains, by using Application Domain Extensions (ADEs). In its current version (CityGML 2.0), utility networks are not natively supported but they can be integrated using an ADE.

A.3. Utility Network ADE: Digital representation of utilities

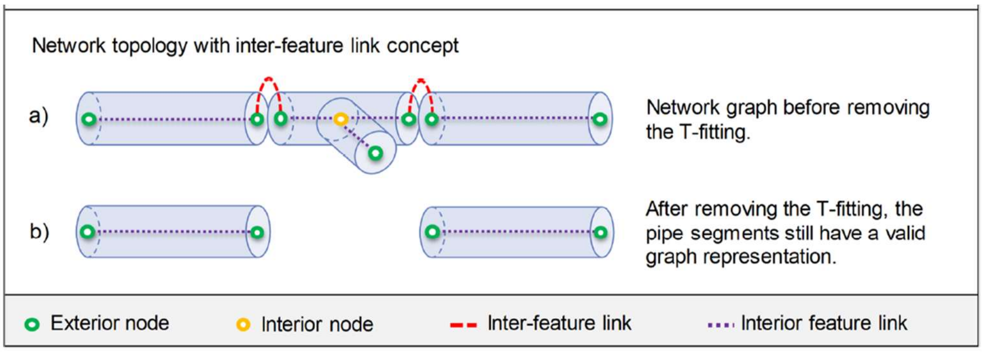
Utility Network ADE (Becker et al., 2011) allows integrating multiple utility networks into CityGML by extending this model with additional classes and attributes. It was originally developed for disaster management, analysis and simulating cascading effects between interdependent networks (both physically and implicitly) and has been expanded to support more uses (Kutzner et al., 2018). Moreover, it allows representing the spatial relation of utilities to above-ground city objects. For example, den Duijn, Agugiaro, & Zlatanova (2018) have shown the usefulness of modelling the connection between lamp posts and electricity lines. Moreover, the model allows the connection between individual buildings and the networks that supply them. Some of the most important capabilities of the model are (Kutzner et al., 2018):

- A dual representation of networks and network components (NetworkFeature), (1) by their 3D topography and (2) as a graph structure showing the functional, structural and topological aspects of individual components (**App. Figure A-1**). In the topological representation, inter-feature links provide connectivity between feature graphs to avoid them sharing end-nodes, which could cause topological issues upon the deletion of a feature (***App. Figure A-2***).
- Connection to city objects (***App. Figure A-3***).
- The modelling of hierarchies of network components –which allows the representation of a network feature as a composition of other features to an arbitrary depth–, and the modelling of hierarchies of networks, for example, high-, medium-, and low-pressure gas networks or subnetworks at any of those levels (***App. Figure A-4***).
- Representation of multiple utilities (different commodities) and their interdependencies in a single model.
- Finally, there Utility Network ADE (v 0.9.2) has a working and publicly available database schema (Agugiaro, n.d.-a; Boates et al., 2018). Storing semantic and spatial data in spatial-relational databases (such as PostgreSQL) –as opposed to file-based storage (such as GML)– is currently the preferred encoding in the field of Geomatics. In fact, CityGML –the parent data model of Utility Network ADE– also has a database encoding called 3DCityDB (3DCityDB Development Team, n.d.) for more efficient data storage and exchange than with GML files, which is not an OGC standard (Ledoux et al., 2019). The Utility Network ADE database encoding is built modularly on top of 3DCityDB.

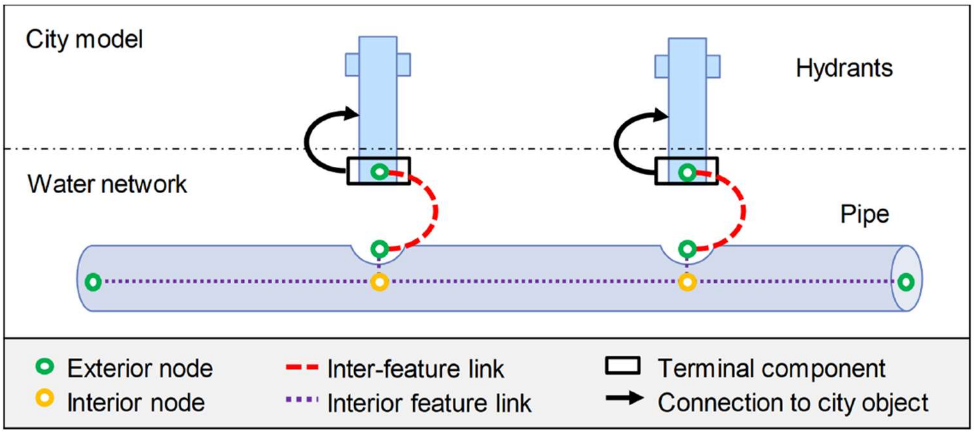
The specific reasons that motivated the creation of the Utility Network ADE were the lack of an open standard to store and exchange comprehensive information about multiple utility network types with a city scope (as opposed to individual buildings), including topographic and topological representation, and the various attributes that characterize a network and its components (Becker et al., 2011, 2012). For example, the INSPIRE utility network data model lacks detail (large geographic units involved), the Industry Foundation Classes (IFC) are very detailed but lack the possibility to use projected geometries (small geographic scale) and the ArcGIS network model is proprietary.



App. Figure A-1. Excerpt from the core module of Utility Network ADE 0.9.2. Above the dashed red line: topographic representation. Below the dashed red line: topologic representation. References: blue, CityGML classes; green, geometry classes; yellow UN ADE classes.



App. Figure A-2. Inter-feature links are the connections between network features. Image from Kutzner et al. (2018).



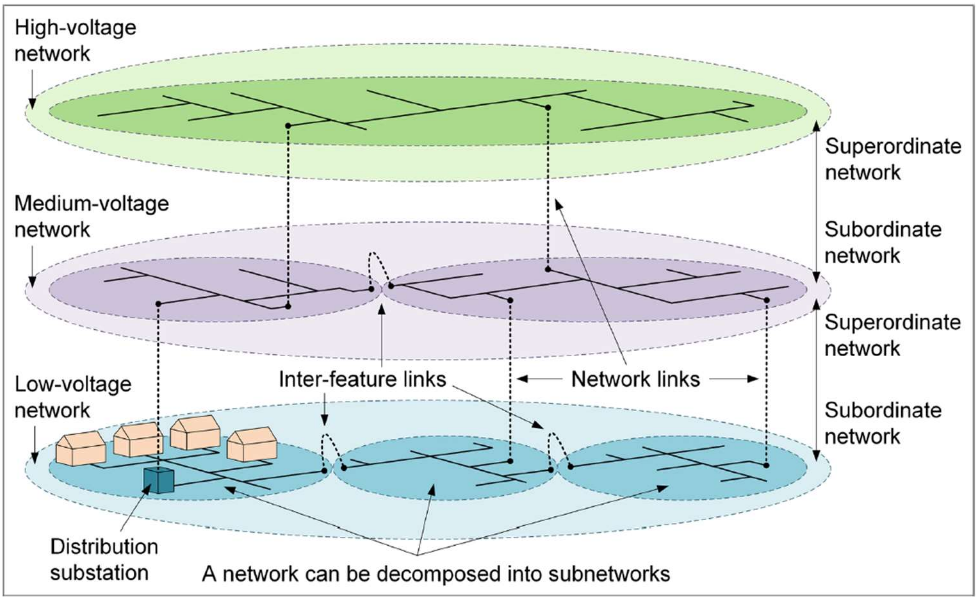
App. Figure A-3. Connection between CityGML objects and Utility Network ADE. Image from Kutzner et al. (2018).

This data model is however not without disadvantages. For example:

1. There is no stable release of the official Utility Network ADE yet, and no organizations using it. Moreover, there is no documentation or best practices guide to illustrate how to map real-world objects to classes and attributes in the model or to show the preferred way of modelling topology.
2. There are high entry barriers for new users.
    a. There are few publicly available datasets compliant with any of the two preferred encodings of the Utility Network ADE data model (either GML files or a spatial relational database). This is an issue for newcomers who could learn how to apply the model to their own data by studying existing implementations.
    b. Furthermore, in order to take advantage of the data model, its users first need to transform their spatial data into any of the two preferred encodings. To date, there is no fully automatic process that performs the data transformation. Moreover, there is no universal workflow to transform data from one encoding to the other (GML file to a spatial-relational database and vice versa). The transformation of the data requires previous knowledge of several technologies, to name the most relevant:
        i. eXtensible Markup Language (XML, including Xlinks and schemas), Geographic Markup Language (GML) and Universal Markup Language (UML)
        ii. CityGML
        iii. A data transformation tool such as Safe Software's FME
        iv. Databases (such as Oracle or PostgreSQL and their spatial extensions, and 3DCityDB which is the database implementation of CityGML), Structured Query Language (SQL), Entity-Relationship Models (ER models), etc.

3. The great modelling flexibility that is one of the strengths of CityGML also means that it is rather complex because of, for example, the size of the data model, its deep hierarchies (it allows for arbitrarily deep nesting of objects), and the multiple ways of storing geometry it provides. This results in complex queries between multiple tables when using the database encoding of the data model.
4. There is not much software support for the model. The few existing Utility-Network-ADE-complaint GML files can be viewed directly only with the freeware FZK viewer from the Karlsruher Institut für Technologie and with FME Data Inspector. For visualizing the information in GIS applications (for example QGIS and ArcGIS), they must be transformed and connected with a spatial relational database beforehand.

Finally, even though this model provides a solid base for the representation of multi-utility networks for diverse purposes, it is not geared towards the domain of O&M and lacks the extensive information required for planning or executing O&M. To fill this gap ter Huurne (2019) developed a domain ontology partly based on Utility Network ADE which is specifically aimed at O&M. This extension was possible because ADEs –which are extensions of CityGML– are also extensible themselves, thus providing the flexibility to overcome perceived limitations in a given data model (Biljecki et al., 2018). An explanation of the O&M extension is provided next.



App. Figure A-4. Network hierarchies. Image from Kutzner et al. (2018).

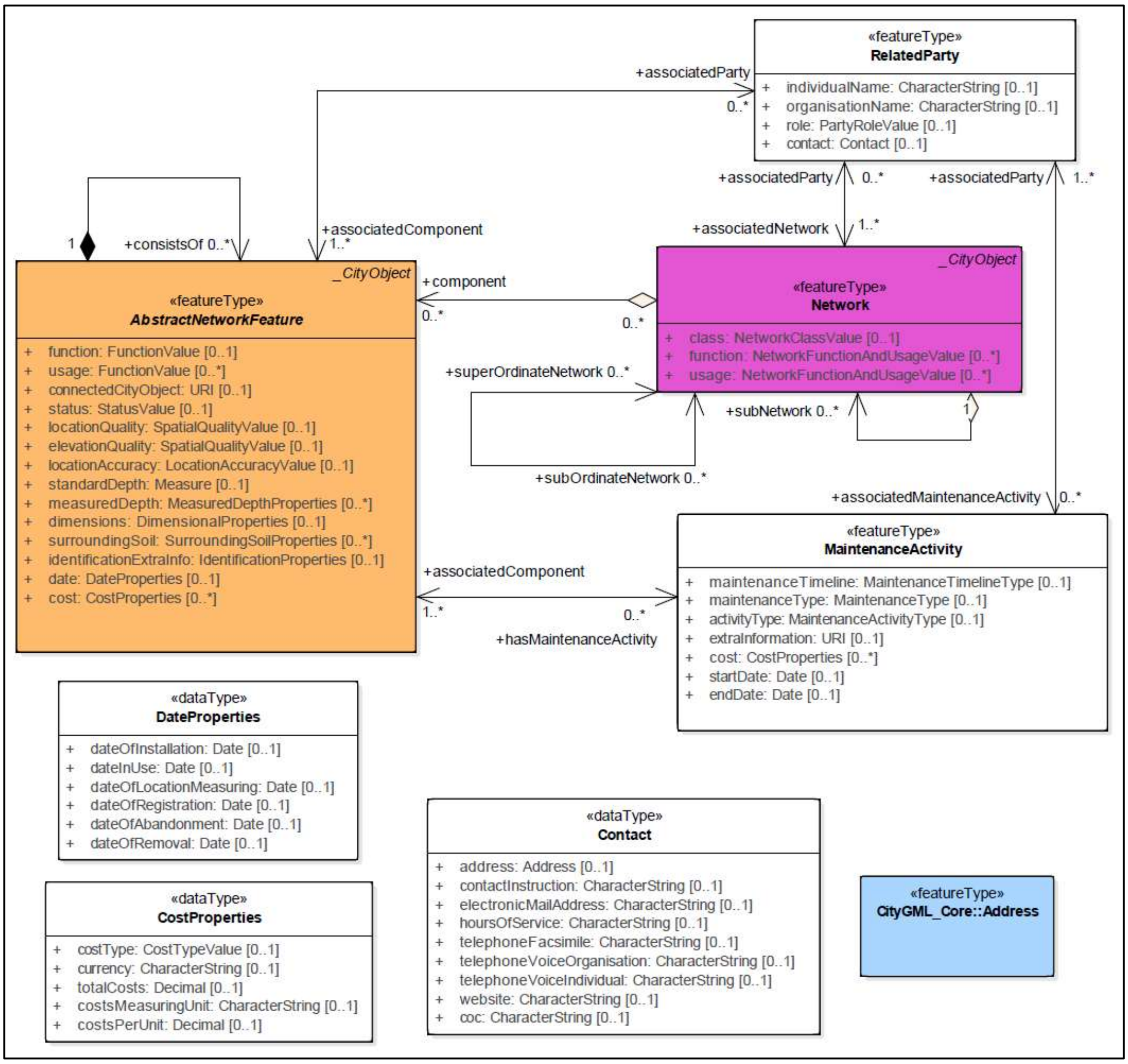
A.4. The Operation & Maintenance data model

The Utility Network ADE O&M extension was originally developed to compensate for the shortcomings and/or disadvantages of using the existing data modelling standards for representing utility networks (ter Huurne, 2019, p. 21). It was designed in collaboration with the client organization (which makes the model targeted) and taking into consideration the information they need to capture for O&M tasks (the model is therefore focused). However, it does not lose any of the functionalities of the base models (CityGML and the Utility Network ADE 0.9.2) described in Section A.2 and Section A.3, which means that the O&M model does not reduce their scope. Furthermore, it offers improved support for infrastructure asset management compared to Utility Network ADE 0.9.2, due to the inclusion of new classes and attributes that were not present in Utility Network ADE 0.9.2 and many of which are still absent from Utility Network ADE 0.9.4 (current version). An example of such additions is provided in App. Figure A-5, showing the new 'Operation & Maintenance' module of the data model, in which the classes *RelatedParty*, *Contact, MaintenanceActivity*, *Cost*, *Date,* and were added.

- The class 'Related party' is related to the classes 'Network', 'Network Feature' and 'Maintenance Activity', and therefore, parties involved with different tasks can be independently assigned to whole networks, certain components, and even maintenance tasks. The information that can be stored with this new class is the general information about the company involved, the contact person in it, as well as a reference to the class 'Contact' where the contact information like P.O. box, e-mail address and phone number is kept.
- The class 'Maintenance Activity' adds the capability to store -in a standardized way- the maintenance records of components (or 'Network Feature(s)' in the terminology of Utility Network ADE) such as pipes, pumps, manholes, etc. It includes attributes such as (a) timeline and date, to indicate when the activity was executed, (b) maintenance type, to indicate the general approach to a specific maintenance task (corrective, predictive or preventive), and (c) activity type, to indicate precisely the task that was carried out (inspection, survey, rehabilitation, etc.), among other attributes. Moreover, it has an attribute called 'extra information' designed to store a Uniform Resource Identifier pointing to a document or repository with additional documents concerning the maintenance.
- The class 'Date' adds the ability to store significantly more milestones in the lifecycle of a component than was possible in Utility Network ADE 0.9.2, which only has one date attribute (year of construction).

Other classes that were used to expand Utility Network ADE 0.9.2 are:

- The characterization of the site conditions is possible via the classes 'Surrounding soil properties' and 'Groundwater properties. The soil properties considered are its type (e.g. silty sand), permeability, reactivity, moisture content, density, and strength. The groundwater is characterized by its depth and an indication whether it is currently present or not. With these two classes, the costs and safety measures for trenching can be better estimated.
- The class 'Performance' allows the representation of the current environmental, safety and engineering performance of components, including their minimum and current scores. Collecting performance information is essential for conducting preventive maintenance as opposed to corrective maintenance, executed after the failure of a component.
- 'Measured depth properties' is modelled as a dataType stereotype, an id-less class. This is because it is considered a property of the components ('Network Feature(s)') that has no meaning without them. With this class, components can be assigned depth information related to survey campaigns, as well as the location of the measurement, the date and the reference level used. Since liners components can have more than one depth associated to them because of the irregularity of the terrain, as well as the slope of the component, an unlimited number of 'measured depth properties' can be assigned to a single component.



App. Figure A-5. Excerpt from UN O&M data model. References: blue, CityGML classes; purple, UN ADE 0.9.2 unedited classes; orange: UN ADE 0.9.2 modified classes; white, new classes in the O&M data model.

Moreover, it is not only in the addition of new classes that the O&M model expands UN_ADE but also in the expansion of the available code lists associated with attributes of classes (e.g. materials and commodities). The extended scope of the model was determined by the functional requirements of experts in the domain of O&M (ter Huurne, 2019, pp. 7, 12–13). These functional requirements were expressed as groups of competency questions associated with use cases. ***App. Table A-1*** provides a high-level comparison of Utility Network ADE 0.9.2 with Utility Network ADE 0.9.4 and the O&M extension, showing the differences in structuring and in scope.

Module		UN ADE 0.9.2	UN ADE 0.9.4	UN O&M	Major differences with 0.9.2	
#	Name	Supported?			0.9.4	O&M
1	Core	✔	✔	✔	Adds 'related party' attribute	Adds 'depth' feature
2	Feature material	✔	✔	✖	Adds layering of cross section materials	Moved to 'component properties' module
3	Functional characteristics	✔	✔	✔	-	-
4	Network properties	✔	✔	✔	Deleted 'commodity classifier' feature. Added 'phase change' feature.	-
5	Network components	✔	✔	✔	Improved taxonomy of functional components	Improved taxonomy of functional components. Improved taxonomy of distribution components. Note: Geometry root is missing.
6	Hollow space	✔	✖	✔	Deleted	-
7	Geometry of network components	✖	✔	✖	Improves significantly the geometric representation of features	-
8	Component properties	✖	✖	✔	-	Adds 'related party', 'abstract dimensional properties', 'surrounding soil properties', and 'abstract extra information' features. Feature material relocated module here.
9	Maintenance and operations properties	✖	✖	✔	-	Adds 'maintenance properties', 'cost properties', and 'date properties' features.
10	Performance properties	✖	✖	✔	-	Adds 'performance properties' and 'abstract impact properties' features.

App. Table A-1. Comparison of the structure of the data models. Utility Network ADE 0.9.2 compared with Utility Network 0.9.4 and the O&M extension.

In **App. Table A-2**, the Utility Network ADE 0.9.2, 0.9.4 and the O&M extension are compared by analysing their ability to answer the competency questions that were used in the design of the O&M extension (ter Huurne, 2019, pp. 28–29). The comparison is not in-depth and is carried out by taking note of the classes and attribute present in each model. Since the development of the O&M extension took some input from changes between UN ADE 0.9.2 and 0.9.3, and also the development of UN ADE 0.9.4 took input from the O&M extension there are points of overlap in the improvements (such as the class RelatedParty). The resulting comparison shows a promising coverage of the domain by the O&M data model.

A.5. Why not IMKL2105?

In the Netherlands, utility owners are obliged by law (WIBON, 2019) to exchange their utility network with the Cadastre (Dienst voor het kadaster en de openbare registers) using an electronic information system that includes a standard data model called ‘Information Model Cables and Pipes’, abbreviated IMKL in Dutch (Geonovum, 2019). IMKL is not a data format, it uses GML to exchange utility data. The purpose of the law that gave origin to the IMKL data model is: (1) to prevent excavation damage to utilities, (2) to promote the construction of electronic communication networks, and (3) to promote the co-use of physical infrastructure and coordinated construction. A review of the classes, attributes and relations in IMKL (**App. Table A-3**) shows that it is not suited for storing comprehensive information about utility networks such as needed for operation & maintenance, and it is not its purpose.

#	Use case / competency question	Supported? 0.9.2	0.9.4	O&M	
1	Assessment of the functioning and usage of a utility network and / or its individual components.				
1.1	What kind of commodity does the utility network distribute?	✓	✓	✓	
1.2	What is the function of the utility network and / or its components?	✓	✓	✓	
1.3	What is the actual usage of the utility network and / or its components?	✓	✓	✓	
1.4	What is the state of operation of a utility network and / or its components?	✓	✓	✓	
2	Identification of a utility network and / or its individual components.				
2.1	What is the location of a utility network and / or its components in the (1) x-, (2) y-, and / or (3) z-coordinate?	✓	✓	✓	
2.2	What is the spatial accuracy of the utility network and / or its components?	✗	✗	✓	
2.3	What are the interior and exterior dimensions of the utility component?	–	–	✓	*1
2.4	What is the shape of the utility component?	✗	✗	✓	
2.5	What is the (main) colour of the utility component?	✗	✗	✓	
2.6	What exterior, interior and filling material does the utility component exist of?	✓	✓	✓	*2
2.7	What is the unique ID of the utility network and / or its components?	✓	✓	✓	*3
3	Assessment of dependencies and redundancies between (1) utility networks, (2) a utility network				
3.1	How are utility networks related to each other?	✓	✓	✓	
3.2	How are utility networks and their components related to each other?	✓	✓	✓	
3.3	How are utility components within a single network related to each other?	✓	✓	✓	
3.4	How are utility components within distinctive networks related each other?	✓	✓	✓	
4	Mapping the impact of a utility network and / or its individual components.				
4.1	What is the social impact of the utility network and / or its components?	✗	✗	✓	
4.2	What is the environmental impact of the utility network and / or its components?	✗	✗	✓	
4.3	What is the economic impact of the utility network and / or its components?	✗	✗	✓	
5	Mapping the risks involved within a network and its individual components.				
5.1	What is the risk to damages due to excavation?	–	–	–	*4
5.2	What is the risk to damages due to vegetation?	–	–	–	
5.3	What is the risk to damages due to settlement of the subsurface?	–	–	–	
5.4	What is the risk to function loss due to deterioration or any other reason?	–	–	–	
6	Assessment of the performance of a utility network or its individual components.				
6.1	Which performance requirements are set for the utility network and its components?	✗	✗	✓	
6.2	What is the nominal flow of a commodity through a distribution line?	?	?	?	
6.3	What is the operational flow of a commodity through a distribution line?	✓	✓	✓	
6.4	What is the maximum flow of a commodity through a distribution line?	✓	✓	✓	
6.5	What is the age of the utility component?	✓	✓	✓	
6.6	For how long has the utility component been in use?	✗	✗	✓	
6.7	What is the expected lifespan of the utility component?	✗	✗	–	*5
7	Planning of operations and maintenance related activities.				
7.1	Who is the owner of the utility network and / or its components?	✗	✓	✓	
7.2	Who is the operator of the utility network and / or its components?	✗	✓	✓	
7.3	When was the last maintenance activity performed?	✗	✗	✓	
7.4	What has been done during the last maintenance activity?	✗	✗	✓	
7.5	How many utility components are enclosed within a particular protective component?	✓	✓	✓	

*1 Improved support in O&M

*2 The best system is used in 0.9.4 (layers)

*3 All the models inherit ID from CityGML core

*4 Risk is not explicitly quantified. Impact can be used but still failure probability is missing. The addition of surrounding soil in O&M is useful for this tasks.

*5 Provides current performance parameters that indirectly allow to calculate expected lifespan

App. Table A-2. A priori comparison of the three models in their ability to answer the competency questions

Legend:
↑ Fully supported
⇨ Partially supported
↓ Not supported

Property / capability	UN ADE 0.9.2	UN O&M	IMKL 1.2	
	Supported?			
Open (extensible)	↑	↑	↑	
Multiple utility networks	↑	↑	↑	
Network hierarchies	↑	↑	↓	
Topography	↑	↑	⇨	*1
Topology	↑	↑	⇨	*2
Connection to city objects (CityGML)	↑	↑	↓	
Maintenance activities	↓	↑	↓	
Performance properties	↓	↑	↓	
Impact	↓	↑	↓	
Depth	↓	↑	↑	
Related party	↓	↑	↑	
Component labels / annotations	↓	↑	↑	
Surrounding soil properties	↓	↑	↓	
Indication of precautionary measures	↓	↓	↑	

*1 UN ADE can store topographies (because CityGML can). IMKL must store them separately.
*2 UNADE has a more advanced topological module.

App. Table A-3. Comparison of Utility Network ADE 0.9.2, the O&M extension (adopted in this study), and IMKL2015 v1.2 (the Dutch standard)

A.6. Related work. Studies based on the Utility Network ADE

Three studies which share similar departure points to this study have been carried out in the past:

- Olde Scholtenhuis, Zlatanova, & den Duijn, (2017). The focus was on extending Utility Network ADE to allow the integration of multiple datasets about the same utility, and to provide a visualization mechanism for the combined spatial information.
- Den Duijn et al. (2018). The focus was on the suitability of Utility Network ADE for the integrated management of above- and below- ground assets in the city of Rotterdam.
- Boates et al. (2018). The focus was on presenting a workflow for populating a spatial database compliant with Utility Network ADE and on testing the maturity of Utility Network ADE for performing analyses (such as routing).

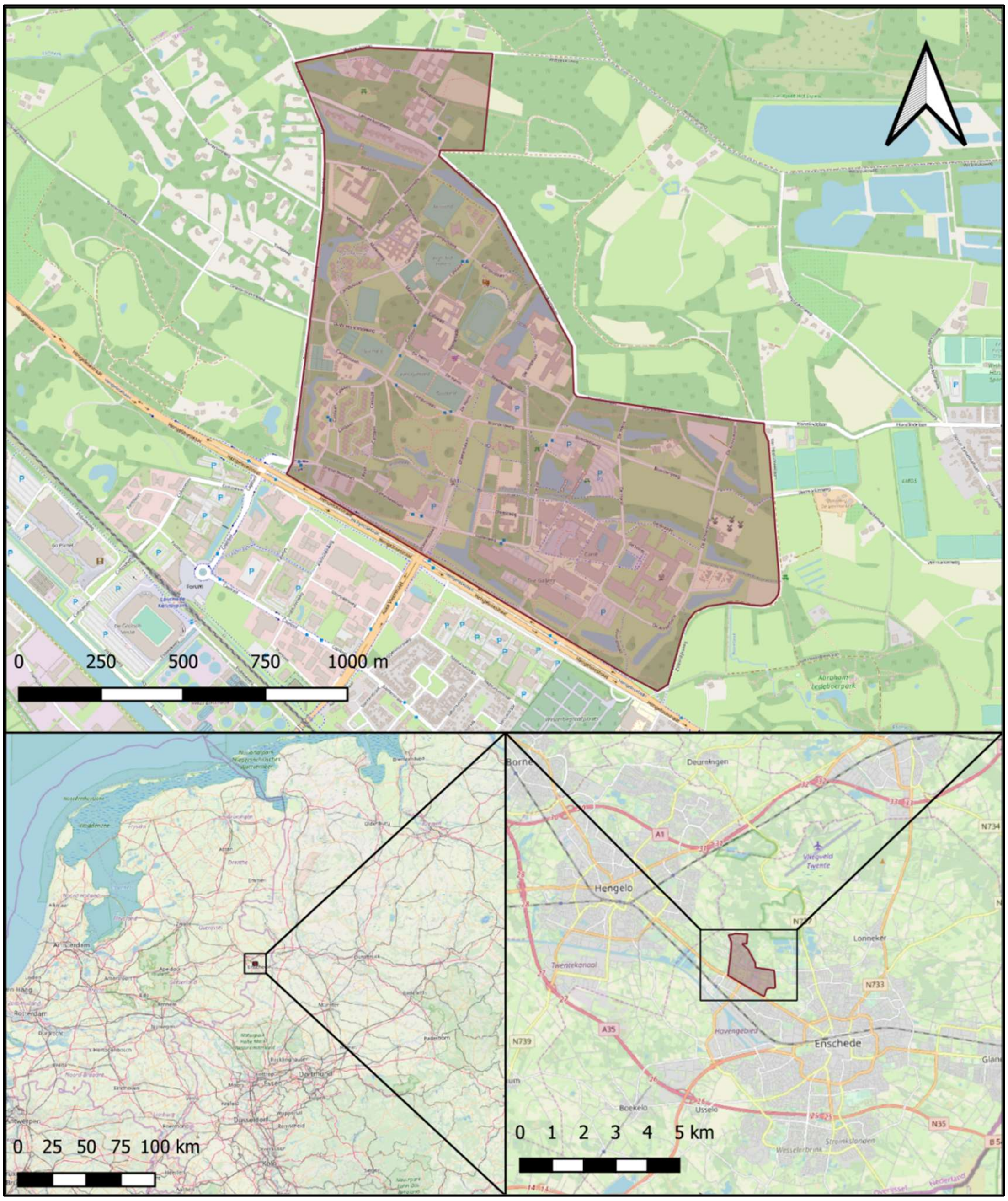
The following differences with the mentioned studies have been identified:

1. Unlike the first study, the focus of the present one is on semantic rather than geometric aspects.
2. Fundamentally, the chosen data model in the last two studies was the unmodified Utility Network ADE 0.9.2 and I intend to use its O&M extension. The focus of the new model is on O&M, but it keeps the base functionality of the base model (such as multi-utility representation). Moreover, the choice of a different data model means that the transformation workspaces (shapefile to GML) developed for Utility Network ADE 0.9.2 are not directly reusable. The same goes for the spatial-relational database configuration.
3. The previous studies did not add ‘dummy’ information to attributes to test the models.

Appendix B. Study area & chosen utility networks

B.1. The University of Twente & utility networks

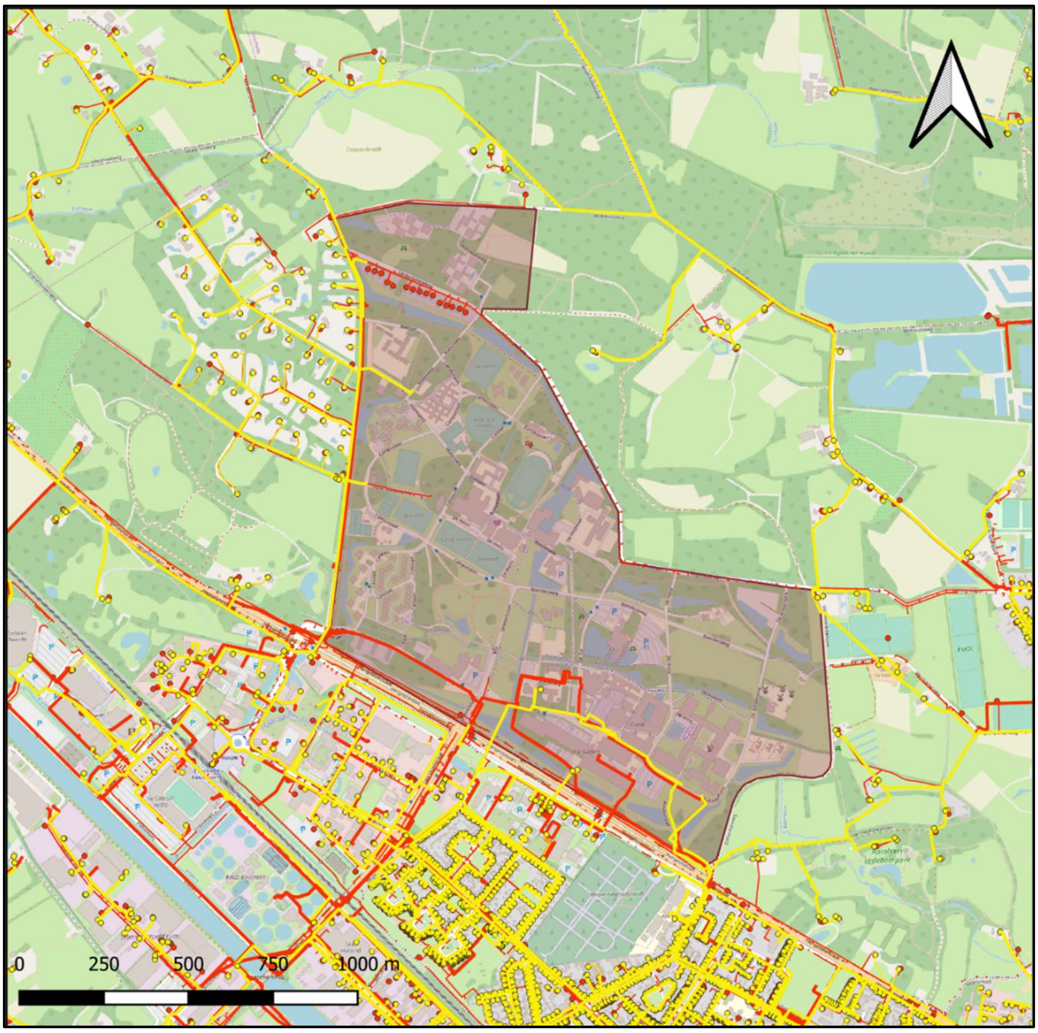
The University of Twente lays in the East of the Netherlands, in the province of Overijssel, between the cities of Enschede and Hengelo and less than 10 km from the border with Germany (App. Figure B-1).



App. Figure B-1.Location of the campus of the University of Twente in the Netherlands (burgundy polygon)

Over 11.000 students are enrolled in its five different faculties and 3.150 staff work there (University of Twente, 2019). Most of the University's buildings are in its 146-hectare, park-like campus, with the main exceptions being the faculty of Geo-Information Science and Earth Observation, the ITC hotel, and the Pakkerij building located in Enschede, and the Therm building located in Enschede Kennispark. As of 2016, the total floor area of the buildings within the campus was 186.000 m^2 (University of Twente, 2016). Apart from the many work and study buildings of the faculties and research centres, the campus has 2.575 apartments for students, 60 apartments for staff, 2 hotels, multiple canteens, a supermarket, a sports centre, a doctor's and dentist office, and a day-care among other buildings.

Several underground utility networks run through the campus to provide the commodities that sustain it and its buildings (e.g. electricity, gas, water and wastewater, and telecommunications). The University is in a relatively uncommon situation as it owns most of the utility networks within the campus, except for the distribution pipes from service providers that deliver the commodities to determined points where they are transferred to the university-owned infrastructure (**App. Figure B-2**), as well as the district heating network. The University is not only the owner but also operates the utility networks through the service department Maintenance & Real Estate (M&RE) of the Campus & Facility Management (C&FM) department.

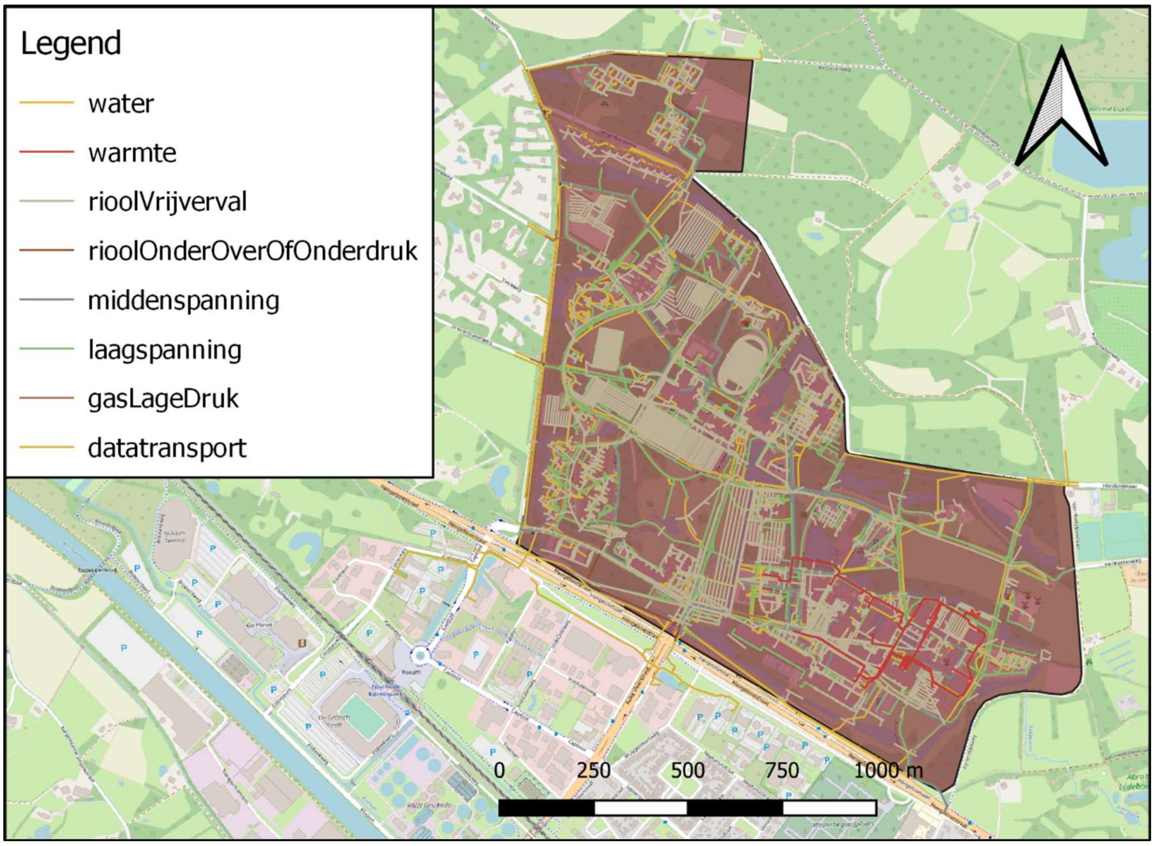


App. Figure B-2.Example of a service provider and its distribution infrastructure reaching the campus. In red: electricity lines and appurtenances, in yellow: gas lines and appurtenances. Source: Enexis open data (Enexis, n.d.).

B.2. Current system

Since 2018, C&FM has embarked in a campaign to consolidate all the University's real estate information into a centralized digital repository: a PostgreSQL relational database -spatially enabled with PostGIS- as backend and the free application Quantum GIS as a frontend. An ongoing survey of the campus buildings, landscape features, utilities, and street furniture as well as topography has resulted in an ever-more-complete centralized database with these data. In the case of pipes and cables, the database contains all the utility networks owned by the university (plus the district heating network which is owned by a third party), totalling approximately 300 km of lines (**App. Figure B-3** & **App. Table B-1**).

Before the implementation of this database, the utility network data was stored as separate AutoCAD drawings (.dwg files) for each network. Because of the inherent limitations of AutoCAD drawings, component information was sparse and heterogeneously recorded as labels placed in the vicinity of the referred element. The existence of the new database greatly facilitates IMKL compliance when compared to the AutoCAD drawings as it requires much less processing for the transformation of the data. Furthermore, it allows to record field information consistently. However, populating the database presents new challenges: as scarce asset information is available, acquiring information about the assets requires the inspection of the network over the next years -which is costly- and the estimation of the properties of the components that cannot be determined unambiguously. Moreover, knowing what information to record from each type of component inspected is not trivial. Finally, the database is not very comprehensive as it only allows to store 31 attributes in total for both linear elements and appurtenance and many of them are currently empty. In this context, the O&M data model based on Utility Network ADE was first developed (ter Huurne, 2019).

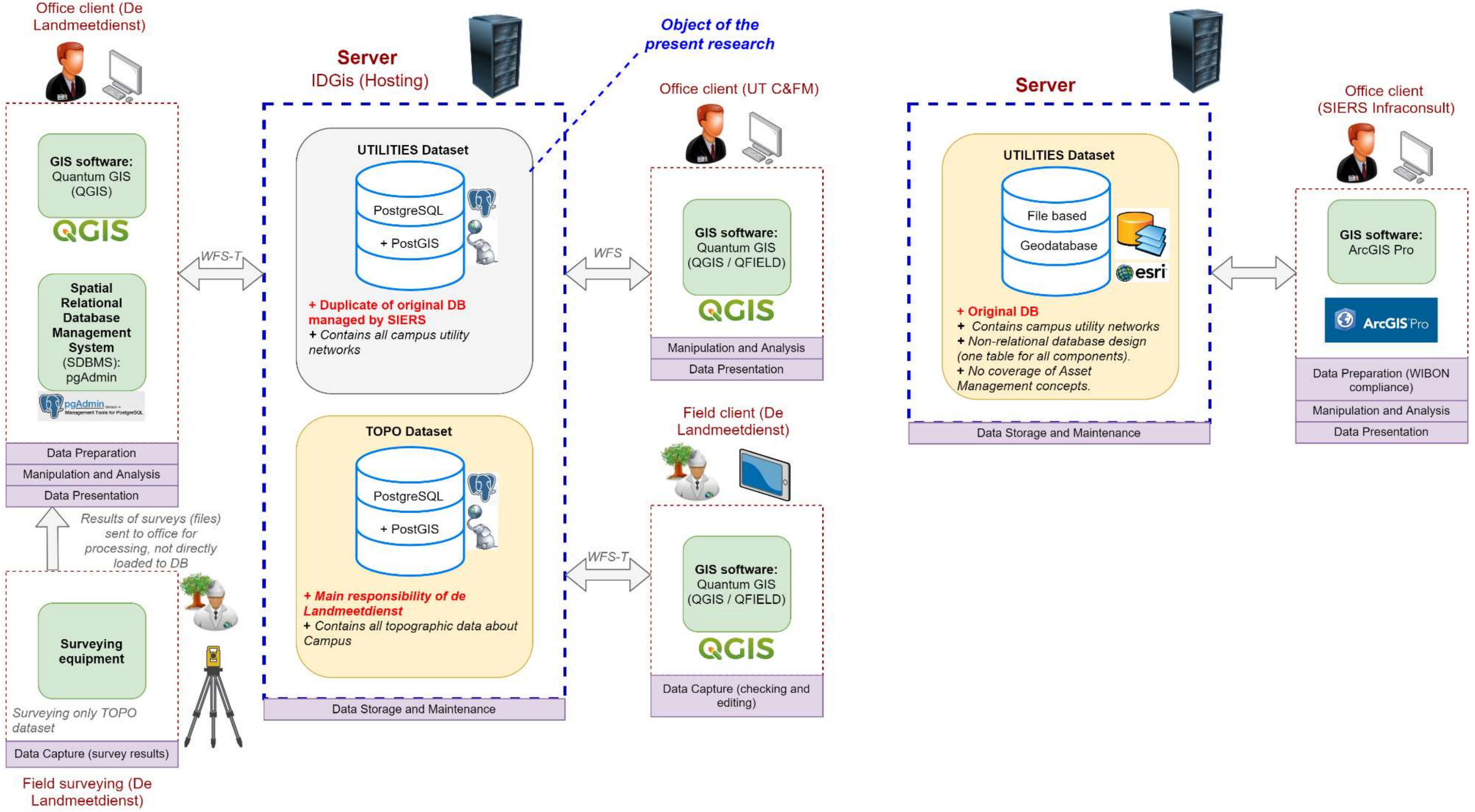


App. Figure B-3. All registered utility networks owned by the University of Twente

#	Network [-]	Identification [-]	Length [km]
1	Electricity (medium voltage)	middenspanning	14.17
2	Electricity (low voltage)	laagspanning	58.07
3	Gas (low pressure)	gasLageDruk	16.73
4	Freshwater	water	59.46
5	Combined sewer (gravity)	rioolVrijverval	77.01
6	Combined sewer (pressure)	rioolOnderOverOfOnderdruk	0.44
7	Heating	warmte	5.46
8	Telecommunications	datatransport	72.32
Total			**303.65**

App. Table B-1. Inventory of utility networks at the University of Twente campus

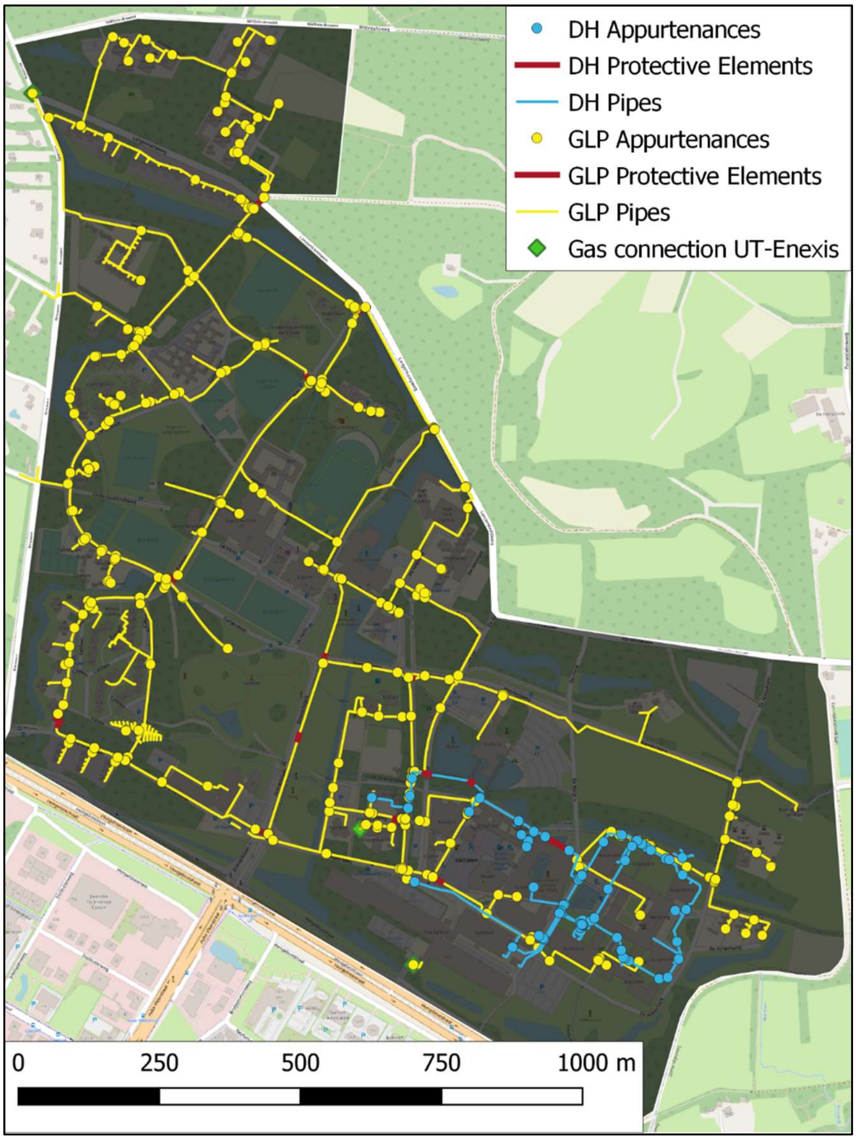
App. Figure B-4 shows the full setup of the GIS in place at campus. The organizations responsible for maintaining the databases are De Landmeetdienst for the TOPO database containing all topographic features of the campus, and SIERS Infraconsult, maintaining the UTILITIES database.



App. Figure B-4. GIS at campus

B.3. Chosen utility networks

In total, two different utility networks were chosen for this thesis to showcase the power of the model to support multiple utilities in one database. The chosen networks –shown in **App. Figure B-5**– were the 16,7 km-long gas low pressure network (GLP) and the 5,5k km-long district heating network (DH). Furthermore, the district heating network is composed of two subnetworks (supply and return), bringing the actual total number of networks used to three. Both network types (DH and GLP) are made up of the same three components: pipes, protective elements, and appurtenances of various sorts. The GLP network is fully owned by the University and is shaped as a closed circuit to have redundancy in case of a component going out of service. The DH on the other hand is owned by Ennaturlijk, a Dutch company that builds and operates district heating networks in several locations in the Netherlands.



App. Figure B-5. Gas Low Pressure (GLP) and District Heating Supply and Return (DH) networks chosen for this study.

Appendix C. Data gathering and pre-processing

This appendix contains additional explanations on the dataset obtained and on its pre-processing.

C.1. Data gathering

As explained in Appendix B, the department of Maintenance & Real Estate of Campus & Facility Management has a PostGIS database containing all their utility lines and appurtenances, among many other spatial features (such as trees, lamp posts, bushes, etc.). Through the surveying company that maintains the database, they have facilitated access to all the digitized campus data via a Web Feature Service (WFS). Thereafter, the relevant data (only lines and appurtenances) was downloaded as shapefiles and loaded into a GIS application.

Furthermore, the digital elevation model (DEM) of the campus was obtained from the online portal PDOK.nl, which provides access to geographical datasets in the Netherlands (Publieke Dienstverlening Op de Kaart, n.d.). The dataset downloaded a part of Actueel Hoogtebestand Nederland 2, the digital elevation map of the Netherlands. Four raster images containing the campus were downloaded.

C.2. Pre-processing methodology

The pre-processing was done in steps:

1. As explained before, two shapefiles were loaded into QGIS: UT_lines and UT_appurtenances. Each contained distinct information pertaining all the different utility networks located in the campus. Ut_lines had all the pipes, protective elements and cables (linear features) and UT_appurtenances had all the appurtenances (point features) in the campus.
2. Both shapefiles were then colour-coded using the attribute ‘thema’, which has the information on the utility network type, to visualize each network with a different colour. Eight different themes were identified: Electricity low voltage, electricity medium voltage, gas low pressure, freshwater, gravity sewers, pressure sewers, telecommunications, and district heating.
3. The length of each network was determined and simultaneously they were broadly visually inspected for assessing the amount of work needed to create topologically correct networks in terms of connectivity, and to check for overlapping elements and digitization mistakes.
4. Gas low pressure and district heating were chosen due to the perceived simplicity of pre-processing them when compared to the rest of the networks.
5. Pre-processing of district heating:
    a. Separating the supply from the return lines and appurtenances by attribute filtering
    b. Identifying both (a) line features unassigned to the correct circuit (supply or return) and (b) protection elements (casings) by filtering.
    c. Correcting digitization mistakes in line and point features. Assigning protection elements (lines) to a different layer and deleting repetitions.

   d. Point features did not have attribute information about what circuit they belong to. Spatial filtering to match them to either the supply or return circuit. Generating a separate layer for each circuit.
6. Pre-processing of gas low pressure:
   a. Looking for 'disused' line and point features wrongly marked as 'functional'
   b. Correcting digitization mistakes in line and point features. Assigning protection elements (lines) to a different layer and deleting repetitions.
7. Pre-processing of the digital elevation model:
   a. Merging the four raster images into a single image using QGIS. Cropping raster only to campus extents and discarding unnecessary information.
   b. The raster images contained points with no data at locations with surface water. The missing data was obtained using an FME workbench to avoid having vacuums in the DEM. The TIN-based DEM was stored as a CityGML complaint GML file.

Input: WFS with campus data.
Output: Nine shapefiles with the network data and one DEM stored as GML file.
1. Shapefiles
   a. Gas low pressure: pipes, protective elements and appurtenances.
   b. District heating supply: pipes, protective elements and appurtenances.
   c. District heating return: pipes, protective elements and appurtenances.
2. GML file: TIN-based DEM of the campus

Tooling: QGIS, FME

C.3. Pre-processing results

The two shapefiles obtained from the university -UT_lines and UT_points- were loaded into QGIS. The dataset UT_lines was decomposed into six different shapefiles using attribute filtering: gas low pressure pipes, gas low pressure protective elements, district heating supply pipes, district heating supply protective elements, district heating return pipes, and district heating return protective elements. The dataset UT_points was decomposed into three shapefiles: gas low pressure appurtenances, district heating supply appurtenances and district heating return appurtenances. The total number of features in each layer is presented in **App. Table C-1**.

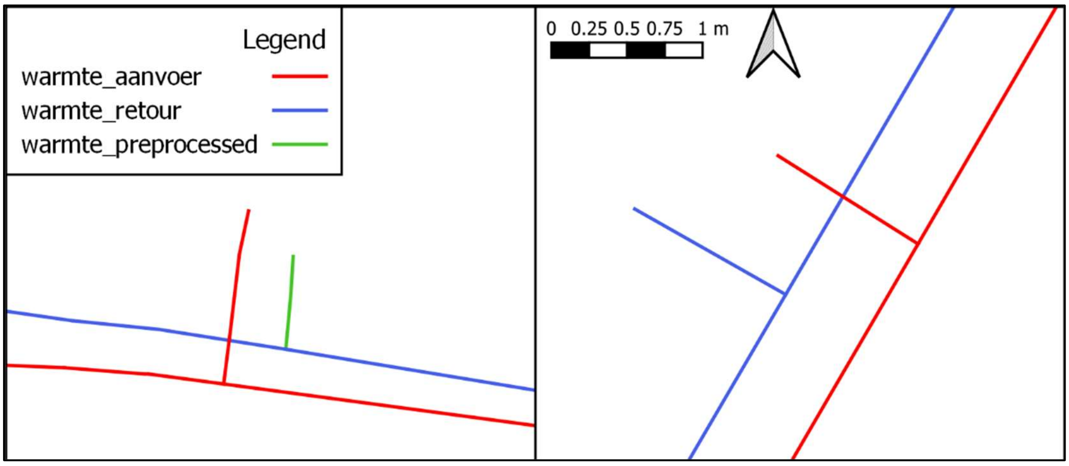
#	Shapefile	Component	# of features
1	Gas low pressure	Pipes	508
2	Gas low pressure	Protective elements	22
3	Gas low pressure	Appurtenances	413
4	District heating flow	Pipes	101
5	District heating flow	Protective elements	6
6	District heating flow	Appurtenances	83
7	District heating return	Pipes	83
8	District heating return	Protective elements	6
9	District heating return	Appurtenances	72
Total			**1,294**

App. Table C-1. Base shapefiles and their number of features

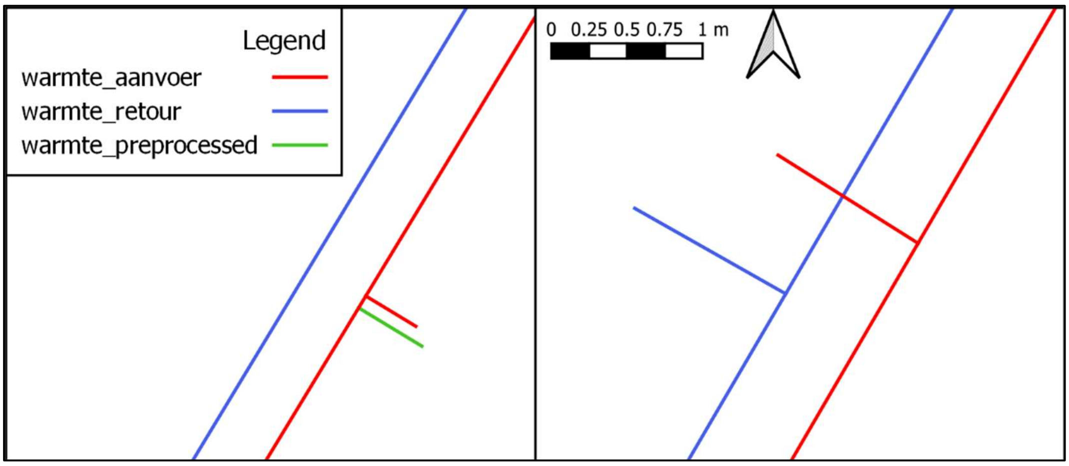
Next, all the shapefiles were pre-processed to correct all digitization issues, as detailed in the following sections.

C.3.1. Pre-processing of district heating networks

First, the district heating pipes were checked for missing attributes describing to which network the pipe belongs to. A filter was placed identifying pipes with the theme 'warmte_aanvoer' (supply), 'warmte_retour' (return) and unclassified. ***App. Figure C-1*** & ***App. Figure C-2*** show two pipes (marked in green) missing this information. Moreover, in ***App. Figure C-2*** the unclassified pipe is also incorrectly drawn, as it should reach the district heating return circuit.



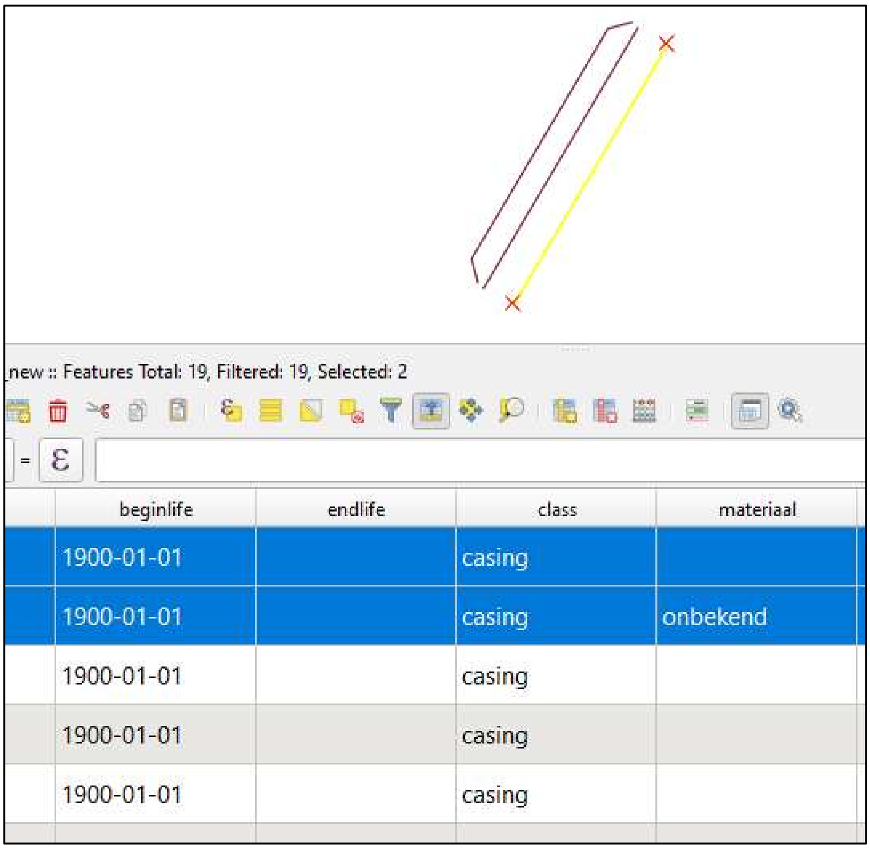
App. Figure C-1. District heating. Line digitisation issues. Left: The layer warmte_preprocessed (green) contains a return pipe that was not assigned to the correct subnetwork (return) via an attribute. Right: Correct digitisation, for reference.



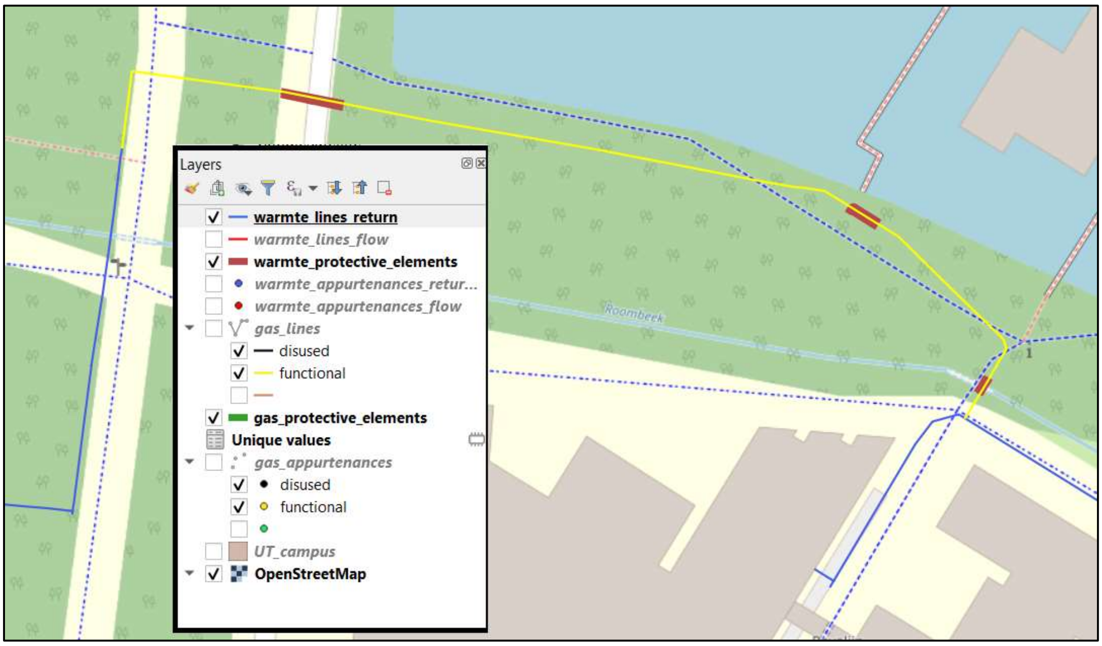
App. Figure C-2. District heating. Line digitisation issues. Left: The layer warmte_preprocessed (green) contains a return pipe that was neither assigned to the correct subnetwork (return) via an attribute nor drawn correctly. Right: Correct digitisation, for reference.

Second, the protective elements were checked for repetition of the information. ***App. Figure C-3*** shows (in yellow) a selection that should yield one protective element, but instead yields two with the same characteristics. One of the elements was deleted.

Third, in Utility Network ADE pipes can be inside only one protective element at a time. ***App. Figure C-4*** shows (in yellow) a pipe which was drawn as a single element in the campus database. In order to model it in the O&M data model it needs to be broken into subsegments, with one subsegment having at most one protective element.

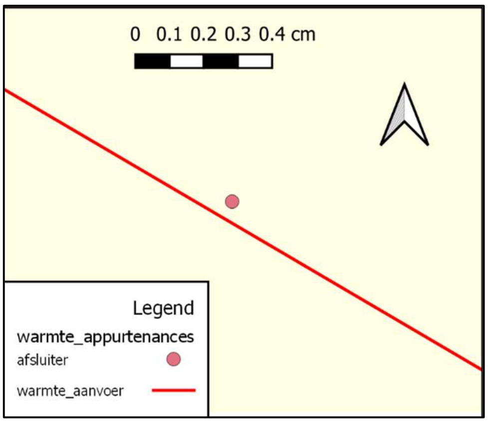


App. Figure C-3. District heating. Line digitisation issues. The image shows the protective elements (casings) of the district heating network. Two identical casings are occupying the same space, so one is deleted.



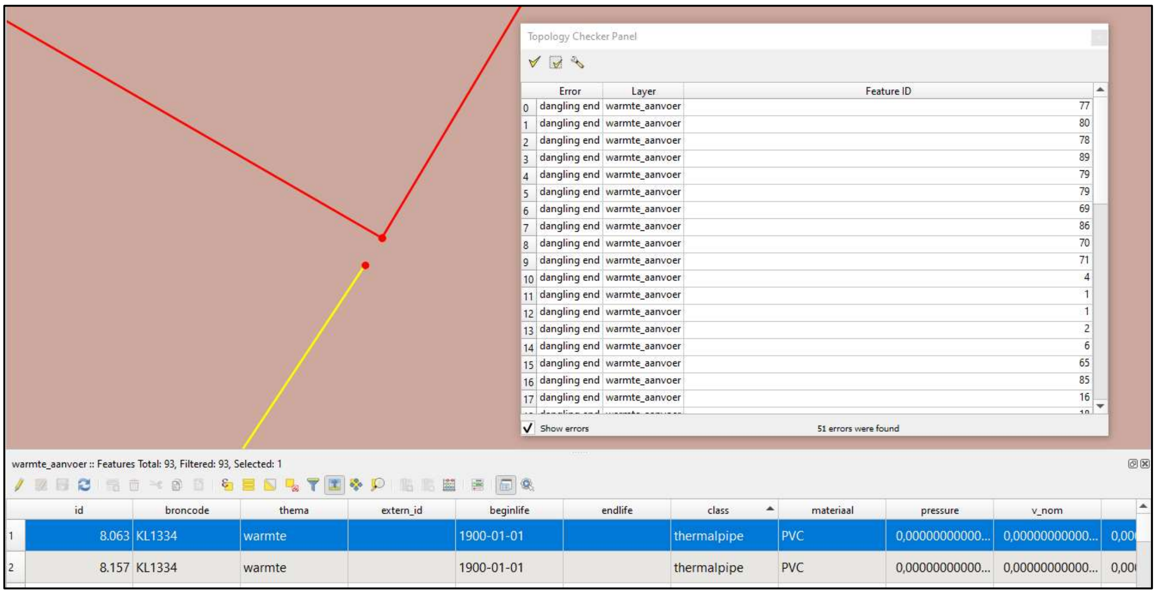
App. Figure C-4. District heating. In yellow, one district heating pipe segment selected. In brown, three protective elements that intersect it. Only one protective element is allowed per feature in the data model. The pipe segment needs to be broken into segments.

Fourth, the network was checked for the precision of the ‘snapping’ between points and lines. ***App. Figure C-5*** shows a point that was not snapped properly to the line it is on.



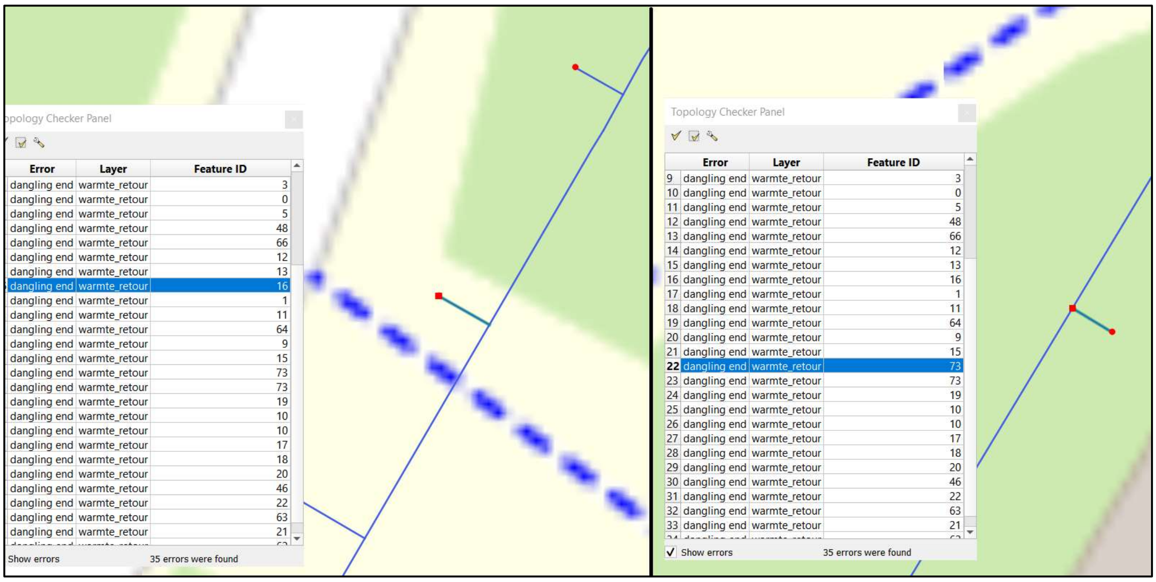
App. Figure C-5. District heating. Point digitisation issues. The image shows a valve (afsluiter) that was not drawn precisely on its corresponding line feature. 19 appurtenances have this issue.

Fifth, the model was checked for ‘undershoots’: lines that do not reach their intended destination. ***App. Figure C-6*** shows in yellow a case of a line and its end node falling short of the intersection with the rest of the network.

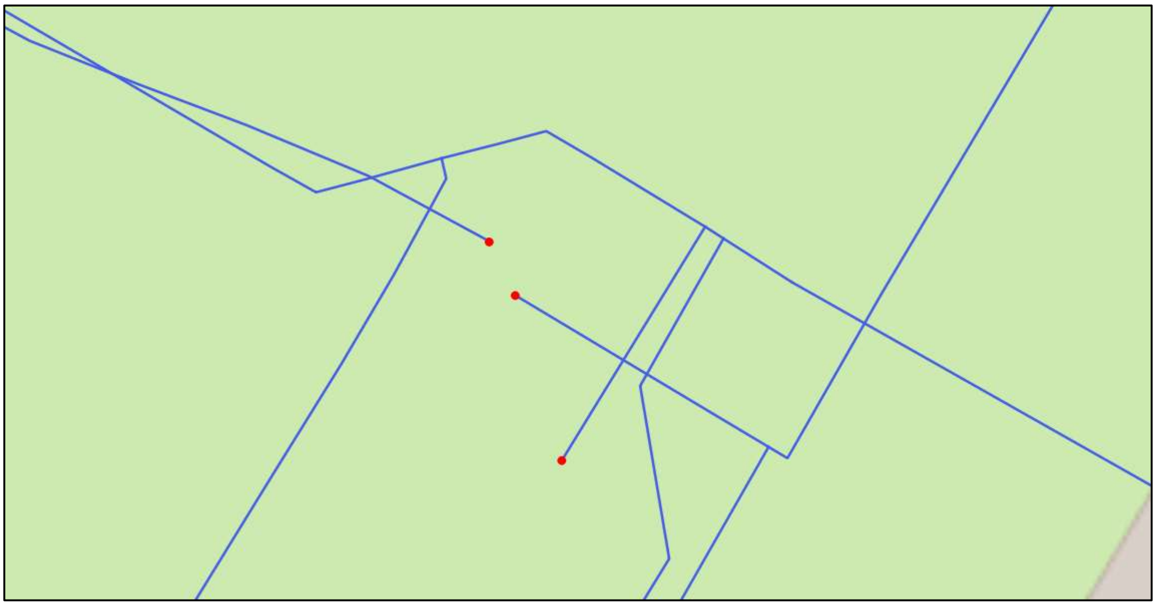


App. Figure C-6. District heating. Line digitisation issues. Incorrect topology.

Sixth, the model was checked for lines that intersect other lines outside of nodes (i.e. along the segment of the intersected line) as shown in ***App. Figure C-7***. All lines not originally connected end-to-end had to be broken down into segments to achieve this. Ideally, this step could have been avoided and done automatically in FME, but the situation shown in ***App. Figure C-8*** prevented this. Lines crossing each other at different depths cannot be differentiated by FME, so the segmentation had to be carried out manually.

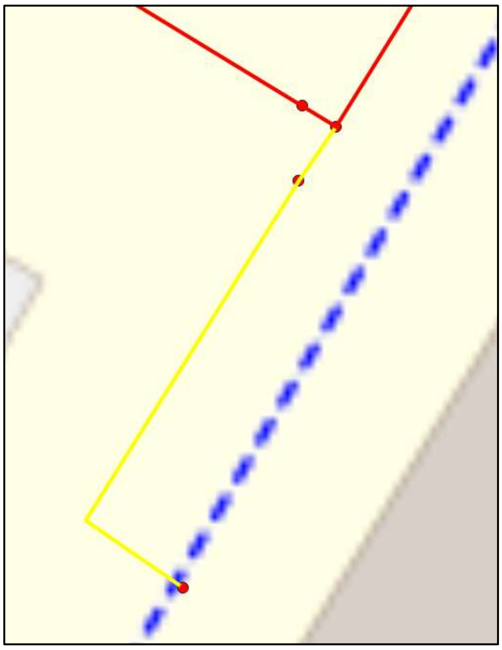


App. Figure C-7. District heating. Line digitisation issues.). Left: lines not connected end-to-end (not desired for recreating the topology of this network). Right: lines connected end-to-end in raw data (desired configuration for this specific network



App. Figure C-8. District heating. Line digitisation issues. Non-coplanar lines running over each other prevent the use of an automatic topology builder that generates the network's topology by intersecting segments and end nodes.

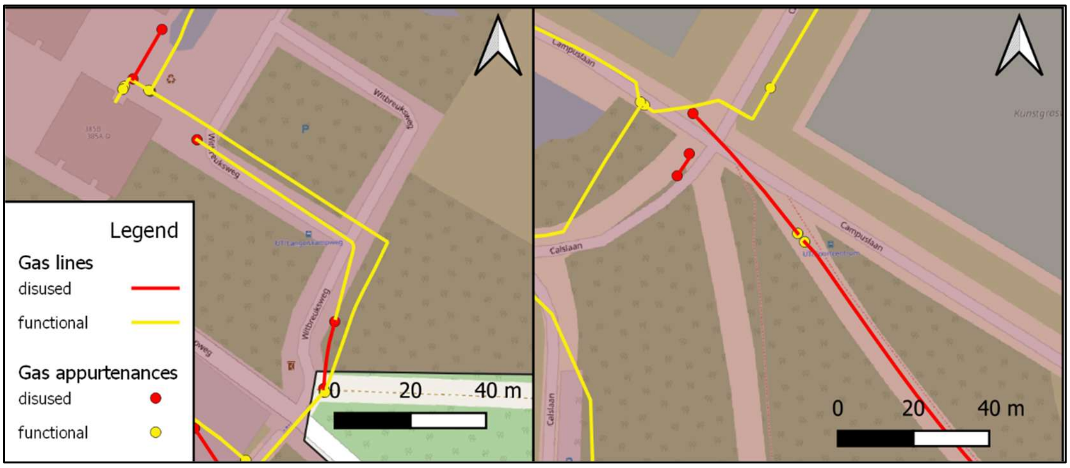
Finally, situations like the one shown in ***App. Figure C-9*** where an appurtenance is not located at the end of a line, but instead it is located along its length were not processed in QGIS, because they can be handled automatically by FME.



App. Figure C-9. District heating. Line digitisation issues. In yellow: selected pipe segment. Note that an appurtenance intersects the pipe along its length and not at either end.

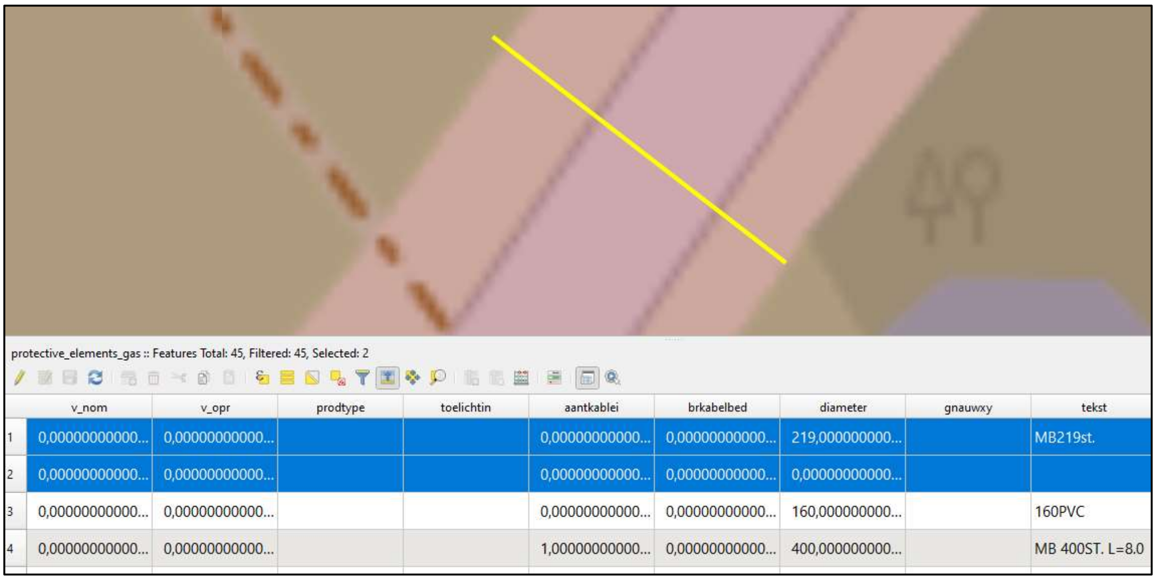
C.3.2. Pre-processing of gas low pressure network

First, the points and lines (appurtenances and pipes) in the network were scanned for incorrectly assigned 'status' attributes. ***App. Figure C-10*** shows on the left a pipe marked as 'functional' but that is located on a disconnected sector of the network and surrounded by 'disused' features and on the right two appurtenances incorrectly marked as functional.



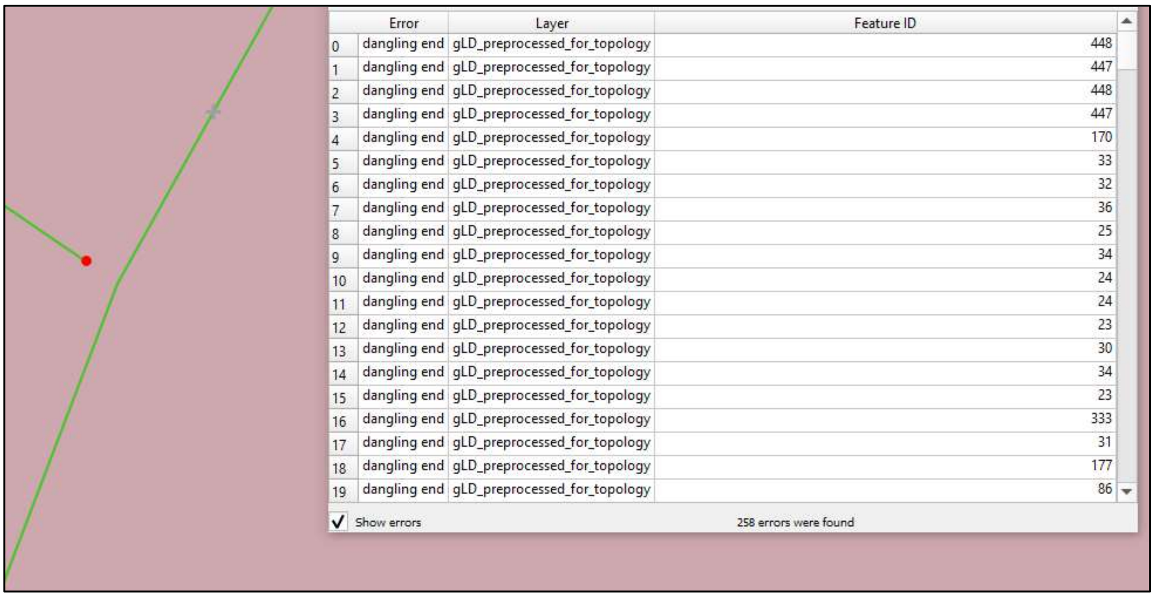
App. Figure C-10. Gas. Point and line digitisation issues Left: wrongly marked 'disused' line feature. Right: wrongly marked 'disused' point features.

Second, the protective elements were checked for repetition of the information. ***App. Figure C-11*** shows (in yellow) a selection that should yield one protective element, but instead yields two with the same characteristics. One of the elements was deleted.

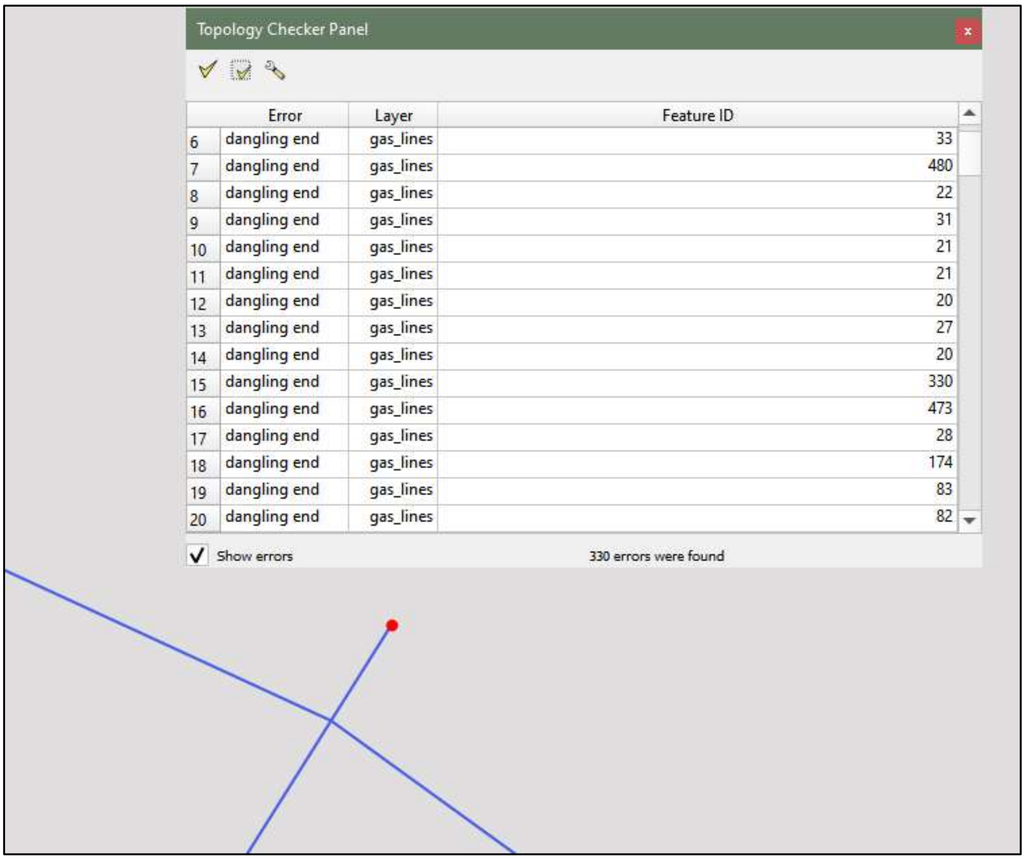


App. Figure C-11. Gas. Line digitisation issues. Repeated protective element.

Finally, the network topology checker of QGIS was used to detect and correct all undershoots (***App. Figure C-12***) and overshoots (***App. Figure C-13***).



App. Figure C-12. Gas. Line digitisation issues. Incorrect topology: undershoot.



	Error	Layer	Feature ID
6	dangling end	gas_lines	33
7	dangling end	gas_lines	480
8	dangling end	gas_lines	22
9	dangling end	gas_lines	31
10	dangling end	gas_lines	21
11	dangling end	gas_lines	21
12	dangling end	gas_lines	20
13	dangling end	gas_lines	27
14	dangling end	gas_lines	20
15	dangling end	gas_lines	330
16	dangling end	gas_lines	473
17	dangling end	gas_lines	28
18	dangling end	gas_lines	174
19	dangling end	gas_lines	83
20	dangling end	gas_lines	82

App. Figure C-13. Gas. Line digitisation issues. Incorrect topology: overshoot.

Appendix D. Data transformation process

This appendix contains additional explanations on the data transformation process from the original dataset to the database.

D.1. Introduction

The transformation consists essentially of two parts:

(a) Manipulating the information from the pre-processed shapefiles and the DEM to digitally reconstruct the three utility networks studied in this project, departing from the simple data structure of the original shapefiles and using as a new data structure the database encoding of CityGML and of the Operation & Maintenance data model. The transformation of the data from one structure to the other enables to leverage the capabilities of CityGML, UtilityNetwork ADE and the O&M extension.

(b) Simultaneously, uploading the transformed data into the PostgreSQL database housing the relational structure derived for the Operation & Maintenance data model.

Both the transformation (mapping) and insertion of the data from the shapefiles and the DEM into the database were handled in a single FME workbench, consisting of around 300 transformers, 9 shapefile readers, 1 DEM reader, and which writes to 16 of the tables in the database (1 out of 59 CityGML tables and 15 out of 36 tables from the O&M extension). Significant tasks executed in the workbench are briefly introduced next:

1. Assigning numbering (id). The workbench reads the (empty) database and fetches the next value for each sequence associated with a table. For example, it reads the current value of the sequence "cityobject_seq" and assigns the next one to the id column of each city object imported. This step is crucial as the id columns are the primary keys of the tables so they must be unique (in each table, not globally).
2. Determining topology. The lack of Utility Network ADE documentation implied having to determine which topological representation was best suited for the data and for this research. ***App. Figure D-1 A)*** shows the typical geometric configuration of the information received: appurtenances exists both at pipe ends and pipe segments and continuous pipes are intersected by other pipes ***App. Figure D-1 B)*** shows the most complete representation of the network supported by Utility Network ADE, using internal nodes and length-less interior feature links connected to them from the exterior nodes. Finally, ***App. Figure D-1 C)*** shows the simplified topological representation supported by Utility Network ADE if the pipes are partitioned at every intersection with a network feature. In order to achieve this representation, the following steps were followed:
    a. Partitioning pipes into independent segments at intersections with other pipes and at intersections with nodes. Generating one node at the beginning of every loose end and at every intersection.
    b. Matching existing appurtenances to the generated nodes. Unmatched nodes are classified as auxiliary appurtenances, with no information available from the original dataset.
    c. Multiplying the nodes corresponding to appurtenances (either the matched or the unmatched ones) to generate external nodes of pipes.

  d. Generating interior feature links between exterior nodes of one pipe.
  e. Generating inter feature links between exterior nodes of pipes and nodes of appurtenances.
3. Assigning topographic data to the network components. Since the available location data for the lines and appurtenances is 2D, the DEM of the campus is used to assign to the pipes and appurtenances an elevation. Each utility is draped to the DEM and offset to the underground.
4. Creating networks, commodities and materials based on the datasets.
5. Extracting relevant attributes from each shapefile, renaming them according to the semantics of the O&M data model and assigning them to the corresponding table in the database.

Input: Nine shapefiles and one DEM stored as GML file.

1. Shapefiles
    a. Gas low pressure: pipes, protective elements and appurtenances.
    b. District heating supply: pipes, protective elements and appurtenances.
    c. District heating return: pipes, protective elements and appurtenances.
2. GML file: DEM of the campus

Output: Database loaded with the three networks.

Tooling: FME, PostgreSQL

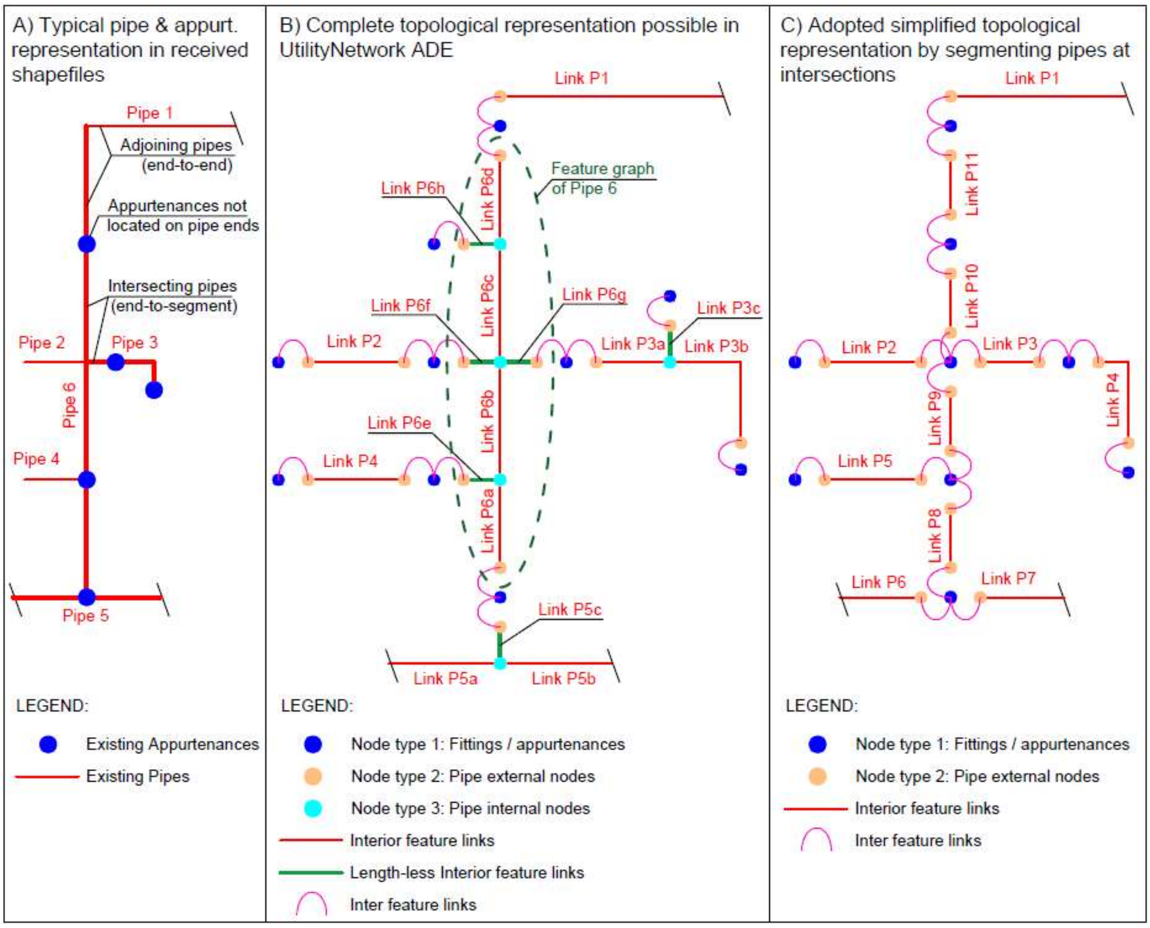
D.2. Transformation examples

To close this section, several examples are provided to illustrate the functioning of the FME workbench by (a) showing portions of the workbench and explaining the purpose of the transformers in them, and (b) showing examples of the data mapping from the original shapefiles to the database.

D.2.1. Example 1: Creating the commodities that flow in the networks.

In order to have a comprehensive representation of the networks that exploits the potential of the O&M data model, it was decided to represent and store in the database the natural gas and the district heating water that flow in the two networks under study.

(a) Transformers used. An FME ***creator*** transformer was used to create a blank feature and send it via two parallel processes: one assigning it the characteristics of natural gas, and the other the characteristics of district heating water. Simultaneously, a custom transformer called ***3DCityDB_Sequence_Getter*** was used to read the database, read the name and current count in all of the sequences, and finally create FME ***counters*** to aid in the numbering (assigning IDs) of the features passing through the processes. After assigning the IDs, the commodities were mapped according to the O&M data model. A peculiarity of the mapping of the commodities is that –in contrast with the mapping of pipes, appurtenances and protective elements– no information was available in the shapefiles about them, so the data mapping was in reality simply writing the necessary information using an FME transformer called ***attribute manager*** . Finally, the information was written to the database using an FME ***feature writer*** linked to the database table '*uom5_commodity*'. This portion of the workbench is shown and explained with comments in ***App. Figure D-2***. Auxiliary transformers called ***attribute creators*** were used to add traceability to the different features, for example by registering what kind of network is related to the feature.



App. Figure D-1. **A)** Pipe and appurtenance representation in the input shapefiles. **B)** Complex possible topological representation in UtilityNetwork ADE. **C)** Simplified topology, adopted in this work, obtained by segmenting pipes to reduce the number of links and nodes in the network and facilitate the generation of the topology.

(b) Data mapping. ***App. Figure D-3*** shows the ***attribute manager*** transformer used to map the attributes of the district heating commodity, district heating water. The id is assigned automatically; the objectclass_id is assigned manually based on the numbering in the objectclass table; the GMLid is created using a UUID generator to produce a unique identifier; the name '*Liquid Commodity*' is assigned manually and is taken from the corresponding UML diagram class; the type '*districtHeatingWater'* is assigned manually and taken from the lookup table called '*uom5_lu_EN_commodity*'. Since this table is an enumeration, the type needs to be entered precisely or it will be rejected by the database.

This example serves to illustrate the general approach of (a) adding traceability, (b) assigning numbering (IDs), and (c) mapping the received attributes to the corresponding columns of the O&M database. This workflow was followed 17 times in total in the FME workbench but with increasing complexity and adding additional processes, because dealing with the creation of commodities was the simplest part of the transformation.

D.2.2. Example 2: Populating a linking table (many-to-many)

Three linking tables were populated using the FME workbench: (1) network to network feature, (2) network feature to material, and (3) network graph to network feature graph. Since the process is very similar for the three tables, only one is shown here: network graph to network feature graph. This table links the topological representation of a feature, a feature graph made up of links and nodes, with the topological representation of a network.

(a) Transformers used. This portion of the workbench starts at the two ***feature writers*** responsible for writing to '*uom5_network_graph*' and '*uom5_network_feature_graph*', which are located immediately after an attribute manager transformer used to map the data. The data stored in a linking table is simply the ID of two related features. Therefore, a simplification of the incoming features is performed by using a ***geometry remover*** and an ***attribute keeper***, to keep only the essential information, which is traceability and id. Next, an ***attribute renamer*** (***App. Figure D-4***) is used to rename the incoming IDs from the two streams following the nomenclature of the O&M domain ontology: the ID of network graphs is renamed to '*network_graph_id*' and the ID of network feature graphs is renamed to '*feature_graph_id*'. Next, both the network graphs and the network feature graphs go through an ***attribute filter*** each, that outputs the features classified by the network they belong to (District Heating, District Heating Supply, District Heating Return, or Oil Gas Chemicals). Four ***feature mergers*** follow, one for each network. Their task is to add the 'network_graph_id' attribute to each incoming network feature graph, according to the network they belong to. The final step is using a ***feature writer*** linked to the database, which writes to the table '*uom5_feature_graph_to_network_graph*'. The whole process is shown in ***App. Figure D-5***.

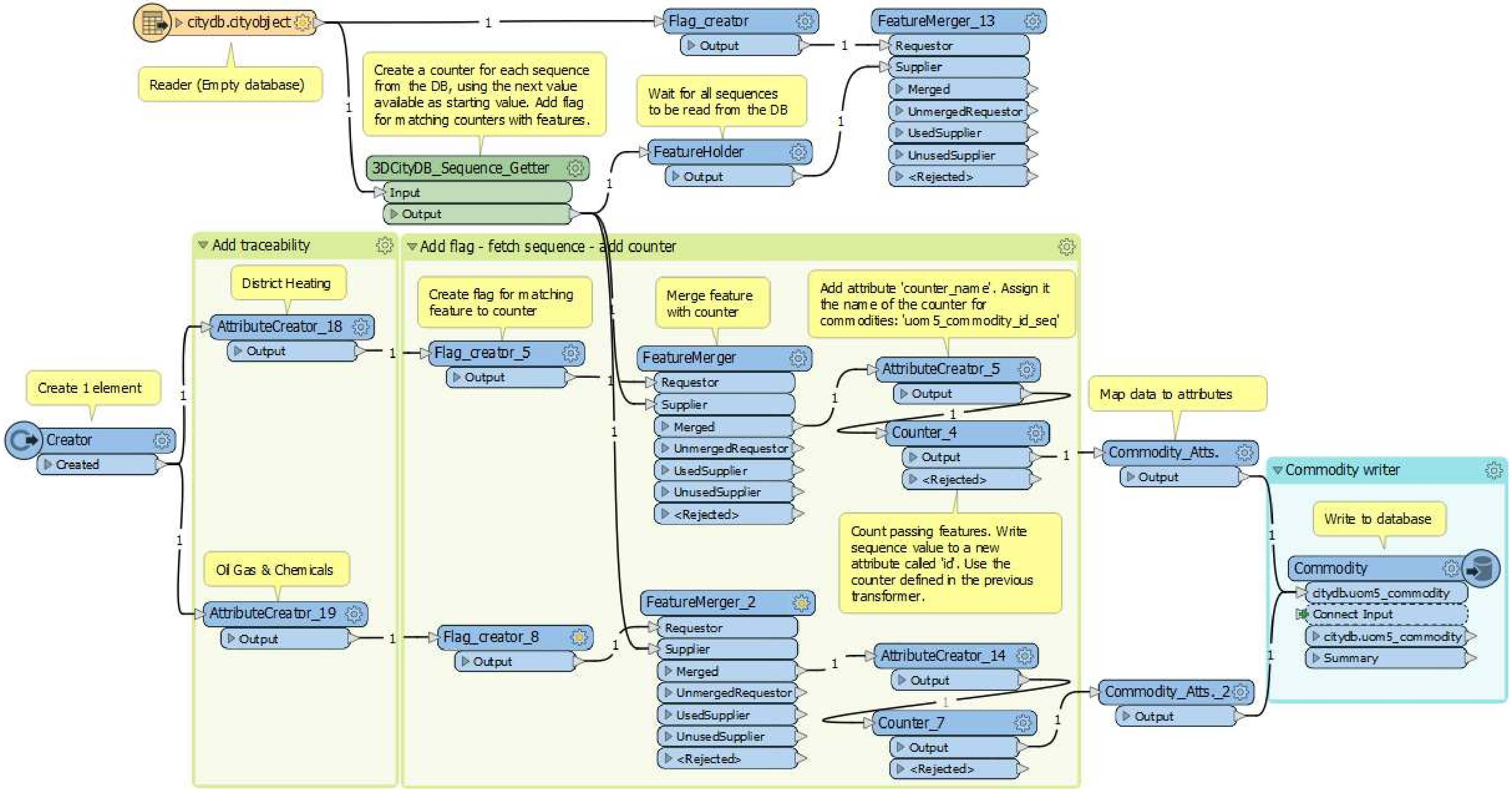
(b) Data mapping. Data mapping is very simple in the case of linking tables because they only have two attributes, which are the IDs of the objects they link together. Thus, no attribute manager is used for this process, as the ***attribute renamer*** presented in point (a) takes care of the data mapping ().

D.2.3. Example 3: Building the networks topology

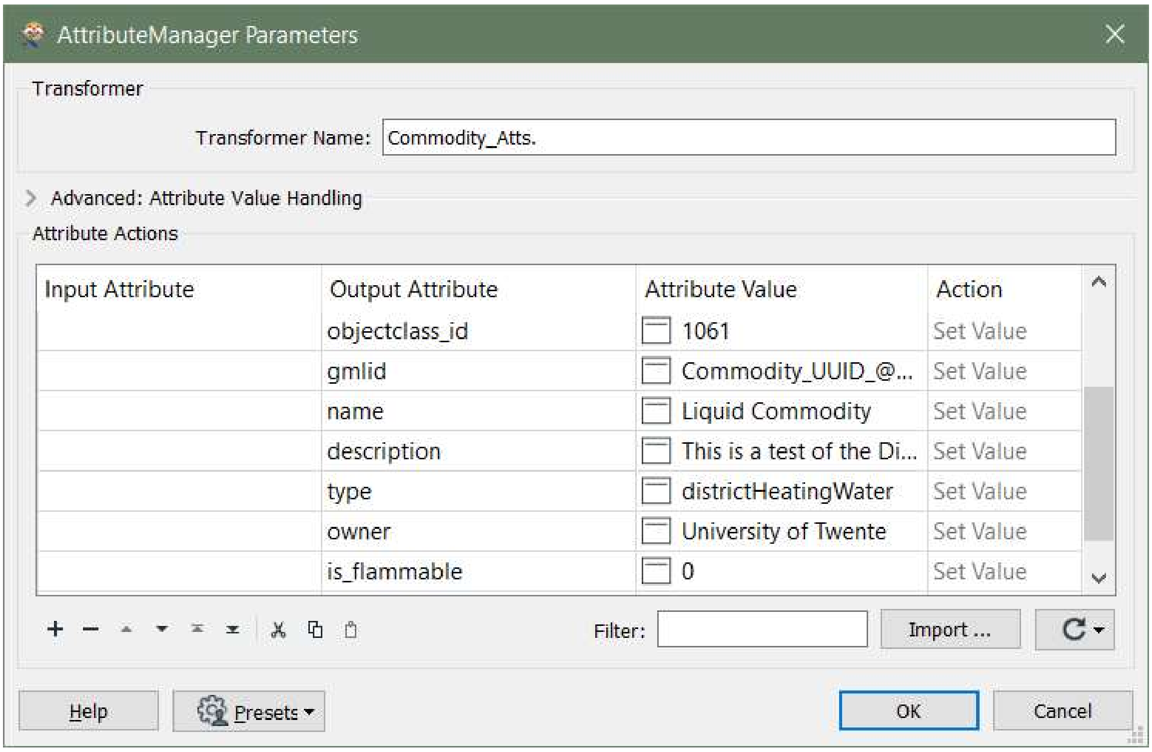
There are five portions in the FME workbench that perform the processes necessary to structure the input data according to the simplest topological representation possible in Utility Network ADE.

1. Partitioning of the pipes at (a) intersections with appurtenances, and (b) intersections with other pipes not occurring at an end of both pipes, and then generating the topology of the newly partitioned elements.
2. Matching the existing appurtenances present in the shapefile(s) to the nodes of the previously generated topology. Unmatched nodes are classified as 'auxiliary' appurtenances necessary to build the topology of the networks according to ***App. Figure D-1*** **C)**.
3. Creating the exterior nodes of pipes and indicating to which node representation of an appurtenance each one is connected.
4. Generating the interior feature link located between the two exterior nodes of a pipe, which represents topologically a single pipe.
5. Finally, the inter feature link located between the exterior node of a pipe and the node representation of an appurtenance is generated.

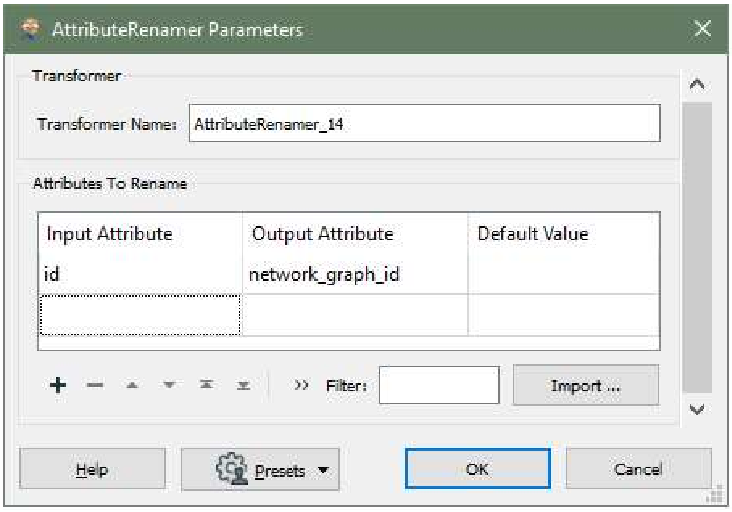
Even though the topological processes in each portion are quite different from each other, for the sake of conciseness only the first topological process is described here in more detail and shown in ***App. Figure D-6***. This process starts directly from the six ***shapefile readers*** of pipes and appurtenances of district heating supply, district heating return and gas low pressure. Three streams of features are created, one for each network. Only the gas pipes are routed through a ***line-on-line overlayer*** FME transformer, because, as explained in **Appendix C**, the district heating networks were pre-processed manually to achieve the same results that this transformer produces. The line-on-line overlayer partitions the gas pipes into segments at every intersection with another gas pipe that does not occur at an end. Next, all three streams are routed through ***point-on-line overlayers***, to break continuous pipes at their intersections with appurtenances. These two first processes would not be necessary if the topology of the networks was built according to the other possible topological representation of Utility Network ADE (***App. Figure D-1* B**)). In fact, they help to simplify the topology of the network and allow to build it according to ADE ***App. Figure D-1* C**). If the intersection with other lines and appurtenances was allowed to happen outside of end nodes of pipes (in other words, anywhere along the length of a segment), the feature graphs would have needed to use interior nodes and more interior feature links. This would have resulted in a much more complex process downstream for generating links and nodes. By partitioning the network like this, the *FeatureGraph* of a pipe is simply composed of one *InteriorFeatureLink* and two exterior *Node*(s). Each of the exterior *Node*(s) is in turn always connected to one *Node* corresponding to an appurtenance. Finally, each stream is routed through an FME ***topology builder***, which automatically determines the connectivity of the 'edges' and 'nodes' of the network. Furthermore, each of the generated nodes receives a list of all the edges it is connected to. This information is necessary for the topological processes 3, 4 and 5 previously mentioned. No data mapping is performed by these processes.



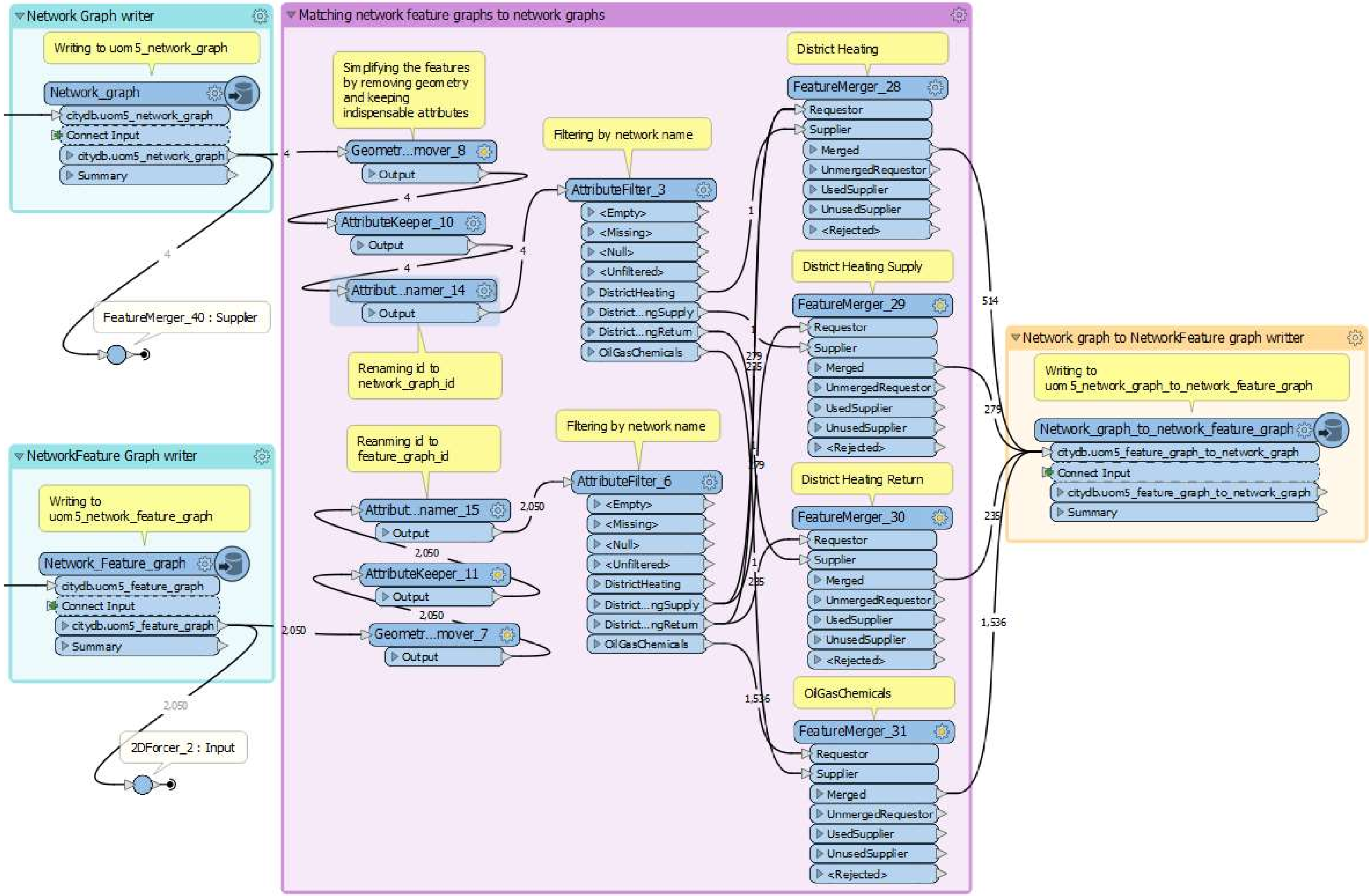
App. Figure D-2. Transformers used to create the commodities, number them, map the necessary information to the correct database columns, and writing the database table.



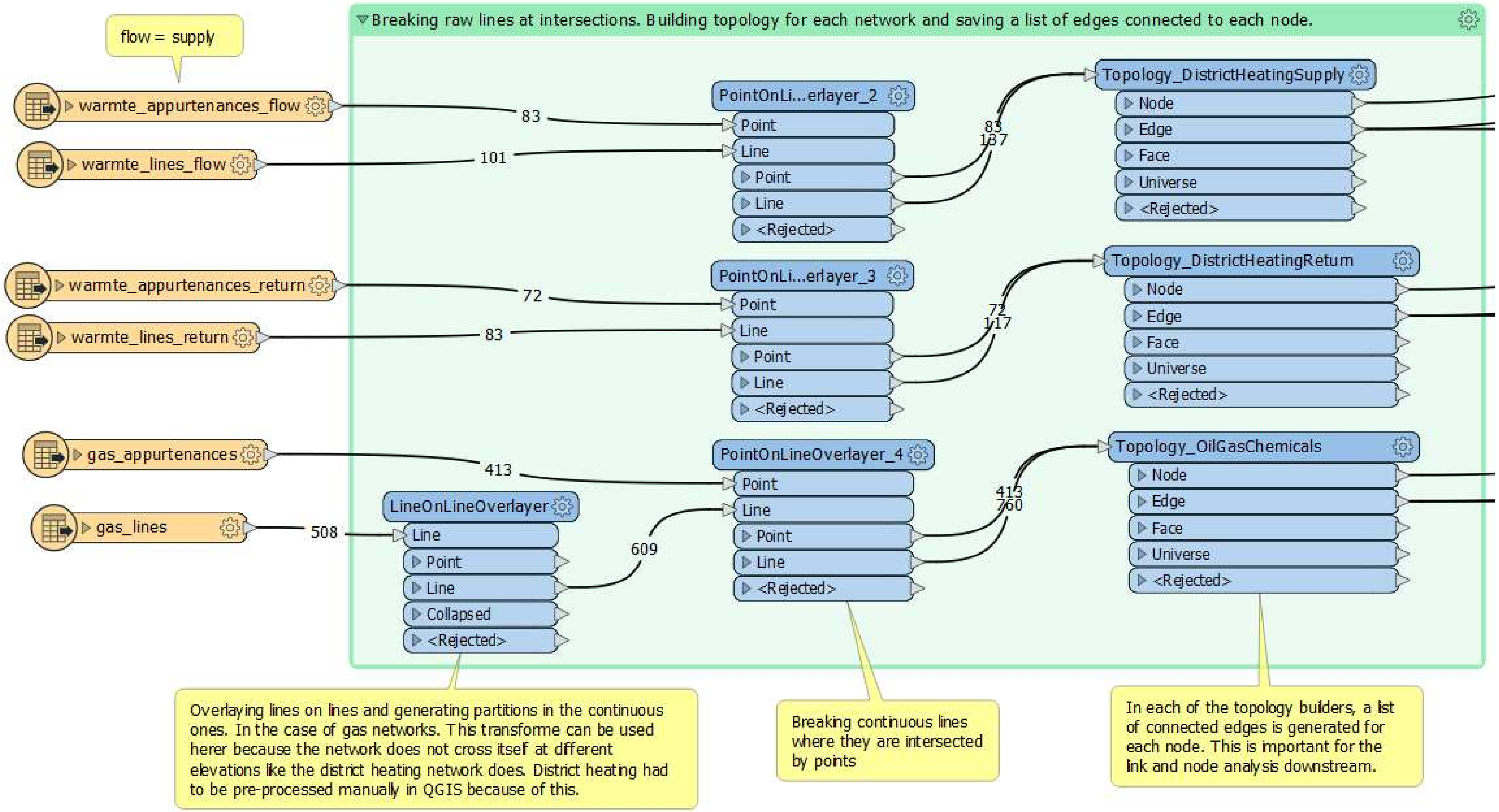
App. Figure D-3. Mapping of the 'district heating water' commodity to the attributes of the table '*uom5_commodity*' of the database encoding of the O&M domain ontology using an FME attribute manager. This dialogue box configures the '*Commodity_Atts*.' transformer from *App. Figure D-2*.



App. Figure D-4. Mapping of the 'network graph(s)' to the attributes of the table '*uom5_feature_graph_to_network_graph*' of the database encoding of the O&M domain ontology using an FME attribute renamer. This dialogue box configures the '*AttribureRenamer_14*' transformer from *App. Figure D-5*.



App. Figure D-5. Transformers used to create the linking table between feature graph(s) and network graphs(s) of the network topology.



App. Figure D-6. Transformers used to do the first of the five topological analyses to get the topological representation of the networks.

Appendix E. SQL scripts for tests

E.1. Identification phase

E.1.1. Identification of streets

SELECT id as street_id, lod1_multi_surface_id, geom

FROM citydb.land_use_auxiliary

WHERE land_use_auxiliary.id = 1605 OR land_use_auxiliary.id = 495;

E.1.2. Identification of pipes

SELECT DISTINCT vp.id AS pipe_id, vp.class, vp.description, vp.network_name, vp.commodity, vp.status, vp.function, vp.usage, vp.location_quality, vp.location_accuracy, vp.exterior_diameter, vp.exterior_diameter_unit, vp.shape, vp.material_type, vp.colour,vp.date_of_installation, vp. date_of_removal,vp.geom

FROM citydb_view.uom5_view_pipes AS vp,

(SELECT geom

FROM citydb.land_use_auxiliary

WHERE land_use_auxiliary.id = 1605 OR land_use_auxiliary.id = 495) AS area

WHERE st_DWithin(vp.geom, area.geom,2);

E.1.3. Identification of appurtenances

SELECT DISTINCT vapp.id as appurtenance_id, vapp.class, vapp.description, vapp.network_name, vapp.commodity, vapp.status, vapp.function, vapp.usage, vapp.is_accesible,vapp.location_quality, vapp.location_accuracy, vapp.geom

FROM citydb_view.uom5_view_appurtenances AS vapp,

(SELECT geom

FROM citydb.land_use_auxiliary

WHERE land_use_auxiliary.id = 1605 OR land_use_auxiliary.id = 495) AS area

WHERE st_DWithin(vapp.geom, area.geom,2);

E.2. Decision phase

E.2.1. Pipe maintenance records

SELECT id as maint_id, pipe_id, network_name, maintenance_timeline, maintenance_type, maint_activity_type, maint_extra_information, maint_start_date, maint_end_date, maint_related_party_id, maint_related_party_role, maint_related_party_organisation_name, geom

```
FROM citydb_view.uom5_view_pipe_maintenance_history
WHERE pipe_id IN (
        SELECT vp.id
        FROM citydb_view.uom5_view_pipes AS vp,
                (SELECT geom
                FROM citydb.land_use_auxiliary
                WHERE land_use_auxiliary.id = 1605 OR land_use_auxiliary.id = 495) AS area
        WHERE st_DWithin(vp.geom, area.geom,2))
ORDER BY maint_start_date, pipe_id;
```

E.2.2. Pipe performance records

```
SELECT id as performance_id, pipe_id , status, network_name, date_of_installation, CAST (date_part('year',performance_date_measurement)-date_part('year',date_of_installation) AS integer) AS age_at_perf_measurment_years, class , performance_type, performance_requirement, performance_score, performance_is_sufficient, performance_date_measurement, performance_extra_information, geom
FROM citydb_view.uom5_view_pipe_performance
WHERE pipe_id IN (
        SELECT vp.id
        FROM citydb_view.uom5_view_pipes AS vp,
                (SELECT geom
                FROM citydb.land_use_auxiliary
                WHERE land_use_auxiliary.id = 1605 OR land_use_auxiliary.id = 495) AS area
        WHERE st_DWithin(vp.geom, area.geom, 2))
ORDER BY performance_date_measurement, pipe_id;
```

E.3. Execution phase

E.3.1. Site characterization

```
SELECT id AS pipe_id, status, network_name, location_quality, location_accuracy, exterior_diameter, exterior_diameter_unit, depth_quality, depth_level_measurement, depth_level_measurement_unit, depth_reference_level, depth_accuracy, date_of_depth_measurement, soil_type, groundwater_level,groundwater_level_unit, groundwater_reference_level, colour, geom
FROM citydb_view.uom5_view_pipes
WHERE id IN (
        SELECT vp.id
```

```
    FROM citydb_view.uom5_view_pipes AS vp,
        (SELECT geom
        FROM citydb.land_use_auxiliary
        WHERE land_use_auxiliary.id = 1605 OR land_use_auxiliary.id = 495) AS area
    WHERE st_DWithin(vp.geom, area.geom, 2));
```

E.3.2. Valves gas low pressure

```
WITH valves_pipe_1 AS
    (SELECT DISTINCT ON (pgr_KSP.path_id)
      appt.appurtenance_id AS appurtenance_id,
      app.class,
      app.geom AS geom
    FROM
      pgr_KSP('
                SELECT id, start_node_id::int4 AS source, end_node_id::int4 AS target, 1 AS cost
                FROM citydb.uom5_link',
                (SELECT start_node_id FROM citydb_view.uom5_view_pipe_topology WHERE id=4475),
                (SELECT node_id FROM citydb_view.uom5_view_appurtenance_topology WHERE appurtenance_id=3257),
                50,
                directed := FALSE)
    LEFT JOIN citydb_view.uom5_view_appurtenance_topology AS appt ON appt.node_id = pgr_KSP.node
    LEFT JOIN citydb_view.uom5_view_appurtenances AS app  ON app.id = appt.appurtenance_id
    WHERE app.class = 'valve'
    ORDER BY pgr_KSP.path_id, pgr_KSP.seq)
SELECT DISTINCT valves_pipe_1.appurtenance_id, valves_pipe_1.class, valves_pipe_1.geom
FROM valves_pipe_1
```

E.3.3. Valves district heating supply

```
WITH valves_pipe_1 AS
    (SELECT DISTINCT ON (pgr_KSP.path_id)
```

```
  appt.appurtenance_id AS appurtenance_id,
  app.class,
  app.geom AS geom
 FROM
  pgr_KSP('
        SELECT id, start_node_id::int4 AS source, end_node_id::int4 AS target, 1 AS
cost
        FROM citydb.uom5_link',
        (SELECT start_node_id FROM citydb_view.uom5_view_pipe_topology WHERE
id=3988),
        (SELECT node_id FROM citydb_view.uom5_view_appurtenance_topology
WHERE appurtenance_id=3006),
        50,
        directed := FALSE)
 LEFT JOIN citydb_view.uom5_view_appurtenance_topology AS appt ON appt.node_id =
pgr_KSP.node
 LEFT JOIN citydb_view.uom5_view_appurtenances AS app  ON app.id = appt.appurtenance_id
 WHERE app.class = 'valve'
 ORDER BY pgr_KSP.path_id, pgr_KSP.seq),
 valves_pipe_2 AS
 (SELECT DISTINCT ON (pgr_KSP.path_id)
  appt.appurtenance_id AS appurtenance_id,
  app.class,
  app.geom AS geom
 FROM
  pgr_KSP('
        SELECT id, start_node_id::int4 AS source, end_node_id::int4 AS target, 1 AS
cost
        FROM citydb.uom5_link',
        (SELECT start_node_id FROM citydb_view.uom5_view_pipe_topology WHERE
id=3988),
        (SELECT node_id FROM citydb_view.uom5_view_appurtenance_topology
WHERE appurtenance_id=2936),
```

```
                100,
                directed := FALSE)
        LEFT JOIN citydb_view.uom5_view_appurtenance_topology AS appt ON appt.node_id = pgr_KSP.node
        LEFT JOIN citydb_view.uom5_view_appurtenances AS app  ON app.id = appt.appurtenance_id
        WHERE app.class = 'valve'
        ORDER BY pgr_KSP.path_id, pgr_KSP.seq)
SELECT DISTINCT valves_pipe_1.appurtenance_id, valves_pipe_1.class, valves_pipe_1.geom
FROM valves_pipe_1
UNION
SELECT DISTINCT valves_pipe_2.appurtenance_id, valves_pipe_2.class, valves_pipe_2.geom
FROM valves_pipe_2;
```

E.3.4. District heating return

```
WITH valves_pipe_1 AS
        (SELECT DISTINCT ON (pgr_KSP.path_id)
          appt.appurtenance_id AS appurtenance_id,
          app.class,
          app.geom AS geom
        FROM
          pgr_KSP('
                SELECT id, start_node_id::int4 AS source, end_node_id::int4 AS target, 1 AS cost
                FROM citydb.uom5_link',
                (SELECT start_node_id FROM citydb_view.uom5_view_pipe_topology WHERE id=3852),
                (SELECT node_id FROM citydb_view.uom5_view_appurtenance_topology WHERE appurtenance_id=2781),
                50,
                directed := FALSE)
        LEFT JOIN citydb_view.uom5_view_appurtenance_topology AS appt ON appt.node_id = pgr_KSP.node
        LEFT JOIN citydb_view.uom5_view_appurtenances AS app  ON app.id = appt.appurtenance_id
```

```
WHERE app.class = 'valve'
ORDER BY pgr_KSP.path_id, pgr_KSP.seq),
valves_pipe_2 AS
(SELECT DISTINCT ON (pgr_KSP.path_id)
  appt.appurtenance_id AS appurtenance_id,
  app.class,
  app.geom AS geom
FROM
  pgr_KSP('
        SELECT id, start_node_id::int4 AS source, end_node_id::int4 AS target, 1 AS cost
        FROM citydb.uom5_link',
        (SELECT start_node_id FROM citydb_view.uom5_view_pipe_topology WHERE id=3852),
        (SELECT node_id FROM citydb_view.uom5_view_appurtenance_topology WHERE appurtenance_id=2851),
        100,
        directed := FALSE)
LEFT JOIN citydb_view.uom5_view_appurtenance_topology AS appt ON appt.node_id = pgr_KSP.node
LEFT JOIN citydb_view.uom5_view_appurtenances AS app  ON app.id = appt.appurtenance_id
WHERE app.class = 'valve'
ORDER BY pgr_KSP.path_id, pgr_KSP.seq),
    valves_pipe_3 AS
(SELECT DISTINCT ON (pgr_KSP.path_id)
  appt.appurtenance_id AS appurtenance_id,
  app.class,
  app.geom AS geom
FROM
  pgr_KSP('
        SELECT id, start_node_id::int4 AS source, end_node_id::int4 AS target, 1 AS cost
```

```
                    FROM citydb.uom5_link',
                    (SELECT start_node_id FROM citydb_view.uom5_view_pipe_topology WHERE id=3852),
                    (SELECT node_id FROM citydb_view.uom5_view_appurtenance_topology WHERE appurtenance_id=2815),
                    100,
                    directed := FALSE)
        LEFT JOIN citydb_view.uom5_view_appurtenance_topology AS appt ON appt.node_id = pgr_KSP.node
        LEFT JOIN citydb_view.uom5_view_appurtenances AS app  ON app.id = appt.appurtenance_id
        WHERE app.class = 'valve'
        ORDER BY pgr_KSP.path_id, pgr_KSP.seq)
SELECT DISTINCT valves_pipe_1.appurtenance_id, valves_pipe_1.class, valves_pipe_1.geom
FROM valves_pipe_1
UNION
SELECT DISTINCT valves_pipe_2.appurtenance_id, valves_pipe_2.class, valves_pipe_2.geom
FROM valves_pipe_2
UNION
SELECT DISTINCT valves_pipe_3.appurtenance_id, valves_pipe_3.class, valves_pipe_3.geom
FROM valves_pipe_3;
```

Appendix F. Integration possibilities with the Utility Network ADE v0.9.4

A final problem treated in this thesis is related to the compatibility of the O&M data model with Utility Network ADE. The alterations in Utility Network ADE that gave origin to the O&M data model have rendered the existing database implementation of Utility Network ADE (Agugiaro, n.d.-a) not directly reusable or expandable to accommodate the new classes. It must be rewritten, as well as the functions (stored procedures in database terminology) necessary to manage the database in terms of insertion and deletion of information. Furthermore, increasing the number of available data models goes against standardization efforts (which require users working with the same model), and having a more modular approach in the expansion of the base model would entail an easier exchange of data between the two, and easier updating of the O&M expansion every time Utility Network ADE gets updated.

F.1. Methodology

The methodology to analyse the architecture of the O&M extension and its relation to the base Utility Network ADE consisted of two parts. First, the available documentation was scanned for criteria pertaining the design of CityGML and ADEs (including their database encoding):

1. **CityGML 2.0 documentation**
    a. An ADE must be defined in its own XML schema definition file with its own namespace. It must be available (or accessible via internet) to everyone parsing CityGML instance documents (OGC, 2012, pp. 16, 151).
    b. An ADE may be defined for one or several of the original CityGML modules. No separate extension mechanism exists for ADEs, because they can be expanded modularly like CityGML (OGC, 2012, p. 16).
    c. The general approach chosen by CityGML for ADEs is (OGC, 2012, p. 151): Incorporating domain specific information into a CityGML instance document. Extending the model with new feature types and attributes to represent domain specific data, not originally present in CityGML. This way of extending the original model allows to use several ADEs simultaneously.
    d. There are two possible extension mechanisms of CityGML (Biljecki et al., 2018; OGC, 2012, p. 151):
        i. Subclasses. New feature types must be derived from existing CityGML feature types and inherit all their attributes and relations. For example, the Utility Network ADE 0.9.2 derives features directly from the following CityGML classes: _CityObject, CityObjectGroup, _AbstractBuilding, _Feature and _GML.
        ii. Hooks. Expanding existing CityGML feature types by adding additional properties to them in the ADE namespace.

2. **The Utility Network ADE documentation** (since there is no official documentation, the UML diagrams were used as reference)
    a. Action: Comparing the structure (classes & relations) of the O&M module with 'Electrical Network Package', a module newly added to the Utility Network ADE version 0.9.4
3. **Database documentation**:
    a. Consulting the 3DCityDB 3.3.0 documentation (3DCityDB Development Team, 2016). There are no references in this document on how to deal with Application Domain Extensions (ADEs).
    b. Consulting the Energy ADE 0.8 documentation. Relevant system and design decisions (Agugiaro & Holcik, 2017a, p. 9):
        i. Extending the 3DCityDB non-concurrently (multiple ADEs at the same time)
        ii. Use original 3DCityDB relations and stored procedures without modifications

Second, the O&M data model was checked against these criteria in search for discrepancies and changes to achieve a modular design are listed. The results are presented next.

F.2. Results

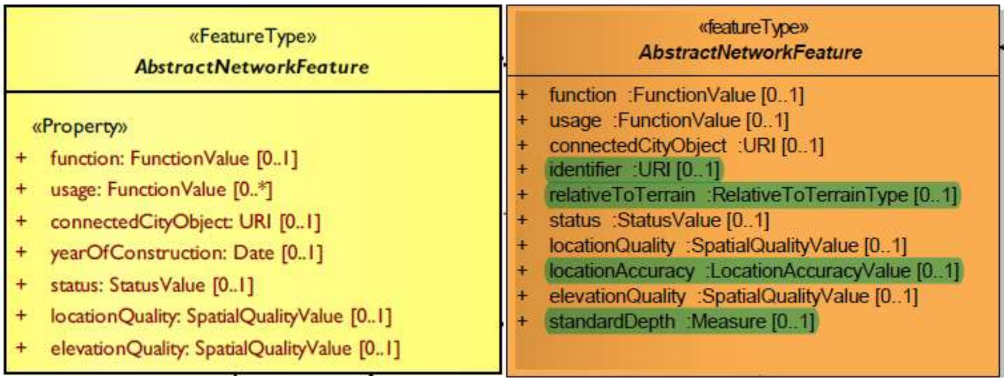
The first step in the methodology was the comparison of the UML diagram of the O&M extension with the documentation of CityGML, the Utility Network ADE, and the available databases. The findings are presented in this section.

Even though the O&M data model is very similar in its core to the Utility Network ADE 0.9.2 and it adds many new classes, it also alters existing classes in the base model by adding or deleting attributes and renaming elements. This is not one of the two suggested mechanisms for extending CityGML, as it does not expand the base ADE data model exclusively by adding subclasses within the new ADE's namespace or adding application specific properties to existing classes in the new ADE's namespace (hook mechanism). Even though it is not CityGML itself that is being altered, it could be argued that the same design principles should apply to its extensions, such as the Utility Network ADE.

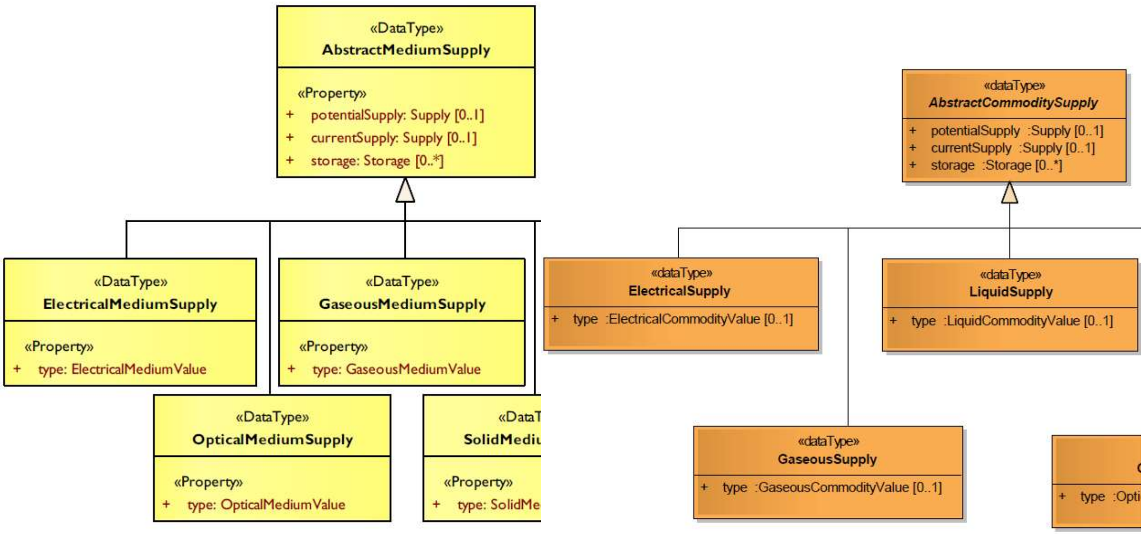
In short, the O&M extension does not extend the base model in a modular way, but instead by partially rewriting it (***App. Figure F-1*** & ***App. Figure F-2***). The benefit of this approach is that it improves the model's fit with its intended domain and allows to create a tailored database encoding, optimized for the model. The consequences however are twofold:

1. As explained in section 3.2, many changes were made during the development of this thesis to the base scripts meant for deriving the Utility Network ADE 0.9.2 database, in order to obtain new scripts apt for creating the O&M database, which is not ideal. It should be noted that this is partly because the O&M data model is in fact based on a hybrid between versions 0.9.2 and 0.9.3 of the Utility Network ADE, and not exclusively on version 0.9.2 as was the base database. So far there has not been a stable Utility Network ADE 1.0 release, and there are significant changes between versions: the database encoding of the Utility Network ADE 0.9.2 would need substantial work to be usable for the Utility Network ADE 0.9.4. Notwithstanding that, it would have been preferable to reuse as much as possible of the existing database and build an extension to it with only the new concepts added by the O&M data model and that was not possible due to the changes in the base model.
2. It creates slight semantic discrepancies between itself and the Utility Network ADE 0.9.2 data model. This means that semantic transformations would need to be performed to map data back and forth. Therefore, datasets that match one of the data models would need to be translated into the other one for reuse in another domain.

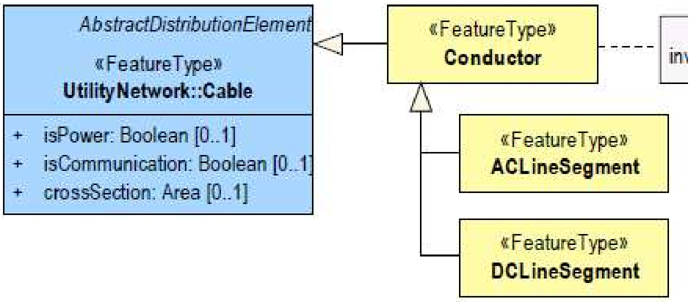
As mentioned in section 3.4, no official documentation exists for Utility Network ADE. However, the ‘Electrical Network Package’ -newly added to version 0.9.4- can serve as an alternative source of information to determine the best practices for expanding Utility Network ADE. In ***App. Figure F-3***, it can be seen how the new domain specific information from the electricity networks (Conductor, ACLineSegment and DCLineSegment) is added without altering the base class from the Utility Network ADE 0.9.4 (Cable). Unfortunately, this source of information is quite incomplete as it does not show how to proceed in case of redundant information being created by extending the model with newer classes. An example of such a situation is the newly added ‘DateProperties’ class (***App. Figure F-4***). The original class ‘AbstractNetworkFeature’ (***App. Figure A-1***, left) has an attribute called ‘yearOfConstruction’, which overlaps with the meaning of the attribute ‘dateOfInstallation’ of the ‘DateProperties’ class of the O&M extension. According to the modularity principles of CityGML, the original attribute should not be altered when extending the model.



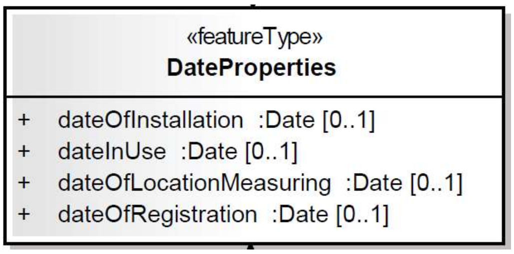
App. Figure F-1. Comparison of AbstractNetworkFeature in the base Utility Network ADE 0.9.2 (left) and O&M extension (right). Four new attributes added to the original class (highlighted) and one attribute deleted (yearOfConstruction).



App. Figure F-2. Comparison of Commodity / Medium Supply in the base Utility Network ADE 0.9.2 (left) and O&M extension (right). All classes were renamed in the O&M extension.



App. Figure F-3. Excerpt from 'Electrical Network Package'. In blue: unaltered class from the Utility Network ADE 0.9.4. In yellow: additional classes added by the package to add concepts from the domain of electricity networks.



App. Figure F-4 O&M extension 'DateProperties' class.

In conclusion, the benefits of a data model with a more modular architecture would be an easier exchange of data with users of the base Utility Network ADE, and better (even full) reusability of future database encodings that will likely be developed as the base model evolves. The downside of this approach would likely be:

- Regarding the data model: Less intuitive and clear structure because of information that should be grouped in one class being dispersed among several classes.
- Regarding its database encoding: The addition of more tables to express relations between the new and the base classes.

F.3. Possible changes in the data model for future integration to Utility Network ADE

Should closer integration with Utility Network ADE be desired in the future, several changes in the data model of the O&M extension are necessary:

- Working with the base data model by:
  - Respecting the names of the original classes
  - Avoiding alterations to the attributes of the original classes
  - Keeping relations between classes unaltered
- Expanding it by:
  - Using subclasses of either Utility Network ADE or CityGML to add new concepts specific to the domain
  - Avoiding repetition of existing attributes in new classes