

Implementierung eines Werkzeuges zur Interpretation von Versorgungsnetzen aus dem Programmsystem STANET®

Melanie Riester, Andreas Geiger

(Melanie Riester, Institut für Automation und angewandte Informatik (IAI) Karlsruher Institut für Technologie (KIT), Herrmann-von-Helmholtz-Platz 1 76344 Eggenstein-Leopoldshafen, melanie.riester9@kit.edu)

(Andreas Geiger, Institut für Automation und angewandte Informatik (IAI) Karlsruher Institut für Technologie (KIT), Herrmann-von-Helmholtz-Platz 1 76344 Eggenstein-Leopoldshafen, andreas.geiger@kit.edu)

1 ABSTRACT

Die Modellierung von Ver- und Entsorgungsnetzen spielt im Hinblick auf vernetzte Energiesysteme eine wichtige Rolle und kann für verschiedene Netzmedien mit dem Programmsystem STANET® realisiert werden. Im Rahmen dieses Beitrages wird ein Analyse- und Transformationswerkzeug für STANET® vorgestellt. Das in Python entwickelte Werkzeug STANETransformer erlaubt die tabellarische Darstellung der Ausgangsdaten, die Analyse und Konsistenzprüfung der Daten, sowie den Export in mehrere Datenformate die bislang nicht von STANET® unterstützt werden. Es wird auf Möglichkeiten der topologischen und semantischen Modellierung von Netzwerken in den Datenmodellen CityGML mit der Erweiterung Utility Networks und dem von der internationalen Organisation buildingSMART entwickelten standardisierten Datenmodell Industry Foundation Classes (IFC) eingegangen. Basierend auf der Analyse der STANET® Ausgangsdaten werden Anforderungen an das Softwarewerkzeug abgeleitet und ein entsprechendes Konzept vorgestellt. Schließlich wird auf den aktuellen Stand der Realisierung des STANETransformers eingegangen, wobei insbesondere die Berücksichtigung der Netztopologie und der Sachdaten im Vordergrund steht.

Keywords: semantische Datenmodelle, Simulation, STANET, Versorgung, Netzwerke

2 EINLEITUNG

Das Karlsruher Institut für Technologie entwickelt unter dem Helmholtz-Forschungsinfrastrukturprojekt Living Lab Energy Campus (LLEC) mit dem Projekt Energy Lab 2.0 einen multimodalen und intelligenten Forschungscampus. Im Rahmen dieses Forschungsprojektes wird das Zusammenspiel der Komponenten zukünftiger vernetzter Energiesysteme erforscht, wobei die Modellierung von Ver- und Entsorgungsnetzen eine entscheidende Rolle spielt. Für die Simulation solcher Netze kommt unter anderem das Programmsystem STANET® zum Einsatz, das die stationäre und dynamische Berechnung verschiedener Netze mit den Medien Strom, Fernwärme, Gas und Flüssigkeiten ermöglicht.

Ein Teilaspekt des Forschungsprojektes ist die Nutzung standardisierter semantischer Datenmodelle. In der aktuellen Version von STANET® werden eine Reihe von Exportformaten gelistet, um Simulationsergebnisse, sowie die Netztopologie weiter verarbeiten zu können. Allerdings wird bei diesen Formaten keines der im Umfeld der Stadt- oder Gebäudemodellierung eingesetzt standardisierten Datenmodelle unterstützt. Aus diesem Grund wurde die Entwicklung eines entsprechenden Werkzeuges notwendig, welches die STANET® Datensätze einliest, die semantischen Daten in geeigneter Form darstellt und Exportfunktionalitäten in standardisierte Datenformate bietet.

3 GRUNDLAGEN

3.1 Das STANET®-Programmsystem

Das vom Ingenieurbüro Fischer-Uhrig entwickelte Programmsystem STANET® ist ein Netzberechnungs- und Analysesystem, welches stationäre und dynamische Berechnungen von Ver- und Entsorgungsnetzen ermöglicht. Mit STANET® können Simulationen von Strom-, Fernwärme-, Gas-, Dampf- und Flüssigkeitsnetzen für einen festen Zeitpunkt (stationär) oder einen vorgegebenen Zeitbereich (dynamisch) realisiert werden (Ingenieurbüro Fischer-Uhrig, 2018a). Das Programmsystem wird von Ver- und Versorgungsunternehmen, Forschungseinrichtungen sowie Ingenieur- und Planungsbüros im Bereich der Infrastrukturentwicklung verwendet (vgl. Otillinger et al., 2008; Nouvel et al., 2015; Stacic & Ebe, 2017; Ruf et al., 2018).

STANET® nutzt für die Netzberechnung neben manuell digitalisierten Daten vorwiegend Netzdaten aus Fremdsystemen wie beispielsweise Geoinformationssystemen (GIS) oder CAD-Systemen. Hierfür bietet das Programm entsprechende Importschnittstellen. Für den Datenexport werden unter anderem Formate für Geoinformationssysteme, CAD-Systeme und verschiedene Netzberechnungsprogramme unterstützt, nicht jedoch die in der Stadt- und Gebäudemodellierung eingesetzten Formate City Geography Markup Language (CityGML) und IFC. Auch ein Datenexport in die Formate OpenStreetMap (OSM) und KeyholeMarkupLanguage (KML) wird nicht unterstützt (Ingenieurbüro Fischer-Uhrig, 2018a, 2018b). In den folgenden Unterkapiteln werden die Formate CityGML UtilityNetwork und IFC in Bezug auf die Modellierung von Netzwerken und Netzkomponenten näher betrachtet.

3.2 CityGML UtilityNetworkADE

Die City Geography Markup Language (CityGML) ist ein XML-basiertes Datenmodell für die Speicherung und den Austausch von 3D Stadtmodellen und ist in der aktuellen Version CityGML 2.0 ein offizieller Standard des Open Geospatial Consortium (OGC) (Gröger et al., 2012). Das Datenmodell enthält vierzehn thematische Module für die wichtigsten Objektarten virtueller 3D Stadtmodelle. Für spezielle Anwendungsfälle bietet CityGML mit dem integrierten Application Domain Extension (ADE) Mechanismus eine Möglichkeit das Datenmodell durch zusätzliche Konzepte zu ergänzen (Gröger et al., 2012).

Da in der aktuellen Version keine Ver- und Entsorgungsnetze modelliert werden können, wurde eine entsprechende Application Domain Extension in Becker et al. (2011, 2013) entwickelt und vorgestellt. Die UtilityNetworkADE ist ein Datenmodell für die Repräsentation von netzwerkartigen Infrastrukturen wie beispielsweise Gas-, Strom- und Fernwärmenetzen. Sie liefert Klassen für die verschiedenen Netzwerkkomponenten, Materialinformationen und Netzwerkeigenschaften (Becker et al., 2013). Das Konzept der UtilityNetworkADE ist die gleichzeitige topographische und topologische Repräsentation eines Netzwerkes mit allen Netzwerkkomponenten. Die topographische Beschreibung erfolgt durch die Klasse Network, welche auf eine beliebige Anzahl von Abstract_NetworkFeature Objekte verweist. Diese bilden die Basisklasse für alle abgeleiteten Netzwerkkomponenten. Die topologische Repräsentation des Netzes wird durch die Klasse NetworkGraph modelliert. Dieser Graph referenziert auf beliebig viele FeatureGraph Objekte, wobei ein FeatureGraph die topologische Repräsentation eines Objektes der Klasse Abstract_NetworkFeature darstellt (siehe Abbildung 1 NetworkCore). Ein FeatureGraph Objekt wird aus Kanten (Klasse InteriorFeatureLink) und Knoten (Klasse Node) zusammengesetzt, wobei jede Kante einen Start- und einen Endknoten aufweist. Verbindungen von FeatureGraph Objekten zu einem NetworkGraph können durch eine Kante der Klasse InterFeatureLink umgesetzt werden, welche auf Knoten der beiden betroffenen FeatureGraph Objekte verweist. Um die gegenseitigen Abhängigkeiten verschiedener Versorgungsnetze repräsentieren zu können, besteht die Möglichkeit die Außenknoten von zwei FeatureGraph Objekten unterschiedlicher Netze durch eine Kante der Klasse NetworkLink miteinander zu verbinden (Becker et al., 2011).

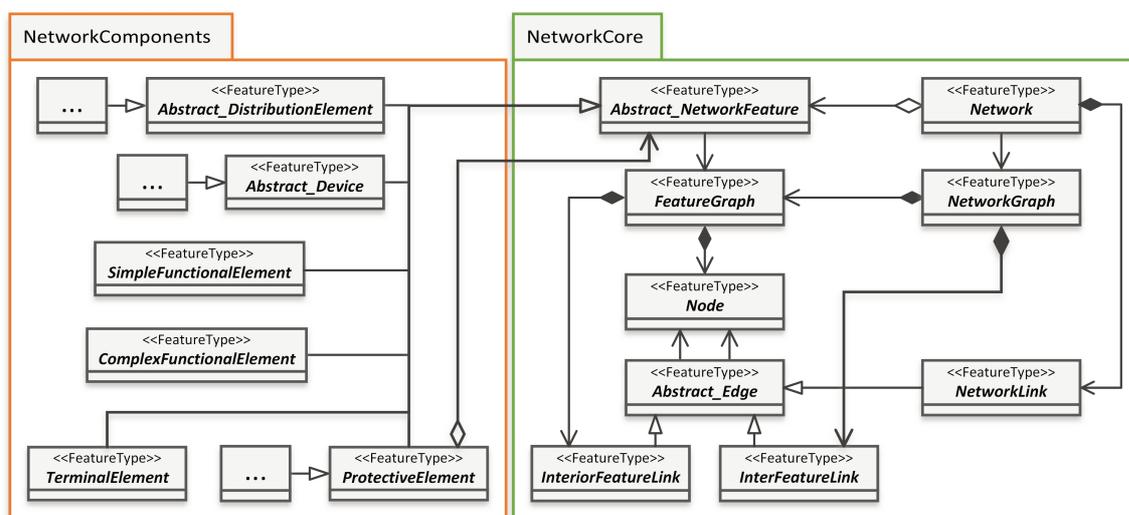


Abbildung 1: Vereinfachtes UML Diagramm der Netzwerkkomponenten (links) und des NetworkCore Moduls mit dem enthaltenen Topologiemodell (rechts) (nach Becker et al., 2013).

Die Klasse `Abstract_NetworkFeature` bildet die Basisklasse für alle Netzwerkkomponenten (Abbildung 1 NetworkComponents). Sie definiert generelle Netzwerkeigenschaften, worunter Eigenschaften für die geometrische Repräsentation der Komponenten in verschiedenen Detaillierungsgraden (LoD), sowie beispielsweise Angaben zu Status, Funktion und weiteren Qualitätsmerkmalen fallen. Zusätzlich ist es möglich, dem `Abstract_NetworkFeature` Materialdaten und Daten zum vom Netzwerk transportierten Medium zuzuweisen. Die Klasse `Abstract_NetworkFeature` verweist auf sechs abgeleitete Klassen, welche zusätzliche technische und geometrische Daten definieren (Becker et al., 2013).

Das vom Netzwerk transportierte Medium bzw. Material wird anhand seines physikalischen Aggregatzustandes oder seiner Übertragungsart in eine der fünf Datentypen `LiquidMedium`, `GasMedium`, `SolidMedium`, `Electrical` oder `Optical` eingeteilt. Je nach Einteilung stehen entsprechende Eigenschaften zur Beschreibung des Mediums zur Verfügung (Becker et al., 2013). Bei dem Material der Netzwerkkomponente (`FeatureMaterialType`) kann ein Materialwert aus einer vorgegebenen Liste ausgewählt werden. Für Verbundwerkstoffe besteht die Möglichkeit die Materialien in `ExteriorMaterial`, `InteriorMaterial` und `FillingMaterial` zu unterscheiden (CityGML Wiki, 2016).

3.3 IFC (Industry Foundation Classes)

Das Datenmodell Industry Foundation Classes (IFC) (ISO 16739:2013) wird von buildingSMART entwickelt und ist ein offener Standard für die Speicherung und den Austausch von Building Information Models (BIM), welche disziplinübergreifende Informationen über den gesamten Lebenszyklus eines Bauwerks bereitstellen (Borrmann et al., 2015). Die Industry Foundation Classes sind mit der Datenmodellierungssprache EXPRESS (ISO 10303-11:2004) beschrieben. Neben dem EXPRESS Schema existiert zusätzlich auch eine XML Schema Definition. IFC bietet ein umfangreiches Datenmodell, wovon im Weiteren nur auf den kleinen Teil zur Modellierung von Netzwerken und Netzkomponenten eingegangen wird.

Ein Netzwerk besteht aus Komponenten (`IfcDistributionElement`), welche durch Anschlüsse (`IfcDistributionPort`) miteinander verbunden werden können. Die Anschlüsse fungieren als Ein- bzw. Auslass für feste, flüssige und gasförmige Stoffe sowie für Elektrizität. Bei der Verbindung zweier Anschlüsse kann die Fließrichtung durch das Attribut `FlowDirection` (Enumeration) mit den Werten `Source`, `Sink` und `SinkAndSource` festgelegt werden. Die Verbindung wird durch die Relation `IfcRelConnectsPorts` umgesetzt, welche auf die zu verbindenden Anschlüsse verweist und optional eine physische Komponente (`RealizingElement`) als Verbindung zwischen den Anschlüssen ermöglicht (siehe Abbildung 2) (buildingSMART, 2015).

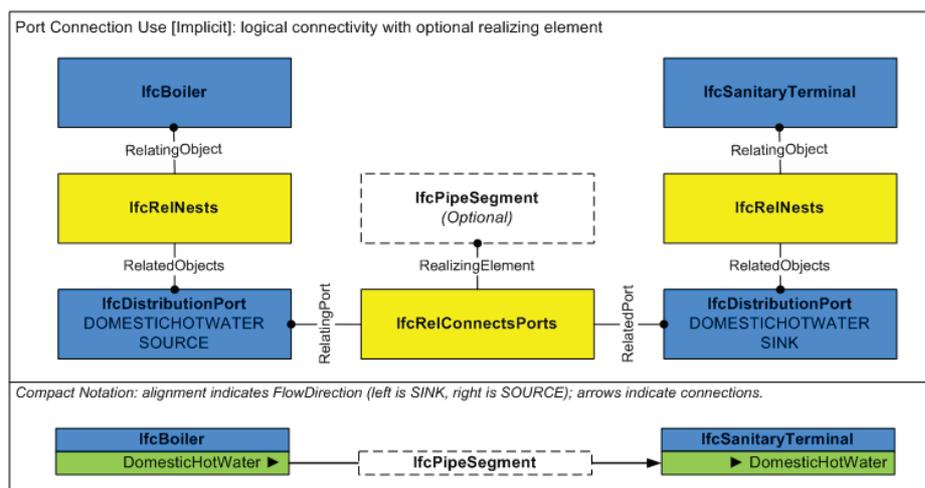


Abbildung 2: Beispiel der Verbindung zweier Anschlüsse (Ports) mittels der Relation `IfcRelConnectsPorts` unter Verwendung eines `RealizingElement` (hier `IfcPipeSegment`) (buildingSMART, 2015).

Alle Komponenten des Netzwerkes werden von der Klasse `IfcDistributionElement` abgeleitet. Diese Klasse unterteilt sich in die zwei abgeleiteten Klassen `IfcDistributionControlElement` und `IfcDistributionFlowElement` mit ihren jeweiligen Unterklassen. Kontrollkomponenten können Parameter (z.B. Temperatur, Druck) anderer Komponenten überwachen. Hierfür wird eine Verbindung (`IfcRelFlowControlElement`) zwischen den jeweiligen Komponenten hergestellt. Die Subkomponenten der

Klasse `IfcDistributionElement` können als Attribute vordefinierte Typen, vordefinierte Eigenschaften und Basismengen aufweisen. Die geometrische Repräsentation der Komponenten erfolgt optional durch Volumen-, Oberflächen- und Liniengeometrien. Allen Komponenten können durch die Beziehung `IfcRelAssociatesMaterial` Materialeigenschaften zugewiesen werden. Um Komponenten eines Netzwerkes zusammenzufassen, können diese zu einem `IfcDistributionSystem` aggregiert werden. Hierdurch besteht auch die Möglichkeit eine Verbindung zu weiteren räumlichen Elementen (`IfcSpatialElement`) herzustellen (buildingSMART, 2015).

Das vom Netzwerk transportierte Medium wird bereits durch die Angabe des `IfcDistributionElement` annähernd festgelegt. Beispielsweise impliziert die Klasse `IfcPipeSegment`, dass das Medium fest, flüssig oder gasförmig ist, aber in keinen Fall elektrische Daten überträgt. Auch die Anschlüsse (`IfcDistributionPort`) können semantische Informationen zum transportierten Medium beinhalten. Dies geschieht zum Beispiel durch Angabe eines Anschlusstyps (z.B. PIPE), Systemtyps (z.B. OIL) sowie weiterer vordefinierter Eigenschaften (z.B. Velocity) (buildingSMART, 2015).

4 ANALYSE

Das Kapitel Analyse umfasst die Analyse der Ausgangsdaten sowie die Analyse der Anforderungen an das Softwarewerkzeug STANETtransformer. Bei der Analyse der Ausgangsdaten wird die vorliegende Datenstruktur der Simulationsdaten genauer betrachtet. Diese Daten sollen durch das Werkzeug interpretiert werden und liegen in einem STANET® spezifischen Format vor. Die Anforderungen an den STANETtransformer beschreiben, welche Funktionalitäten von dem Werkzeug unterstützt werden sollen und setzen somit die Rahmenbedingungen für die Entwicklung.

4.1 Analyse der Ausgangsdaten

Die Ausgangsdaten liegen in einem STANET® spezifischen CSV Format vor. Das CSV (Comma Separated Value) Format (Shafranovich, 2005) ist ein Textformat, bei welchem Datensätze durch ein Trennzeichen wie zum Beispiel durch ein Semikolon oder ein Komma voneinander getrennt werden. Die Ausgangsdaten teilen sich in drei Abschnitte, welche im Folgenden anhand einer vereinfachten CSV-Datei weiter erläutert werden.

Jede Ausgangsdatei weist einen Kopfteil mit wichtigen Dateiinformatoren für den Import und Export auf. Die Zeilen des Kopfteiles setzen sich aus einer Satz-Kennung (3-stellige Kennung) und dem dazugehörigen Inhalt zusammen (siehe Abbildung 3 – Kopfteil). Die Satz-Kennung dient als Abkürzung für einen Begriff. Beispielsweise steht die Satz-Kennung „DAK“ für Dateikennung und „VER“ für Version. Im Kopfteil der Datei stehen wichtige Informationen für die Interpretation der Daten, wie beispielsweise das Format (FMT), der Texttrenner (TTR) und der Dezimaltrenner (DTR) (Ingenieurbüro Fischer-Uhrig, 2018b).

Nach dem Kopfteil folgen die Felddefinitionen für die einzelnen Datentabellen der Netzelemente, welche eine genaue Definition aller Felder der jeweiligen Datentabelle angeben. Pro Zeile wird ein Feld beschrieben, wobei zu Beginn drei Satz-Kennungen stehen. Diese kennzeichnen die Zeile mit „FEL“ als Felddefinition und geben an, welches Netzelement beschrieben wird. Nach den Satz-Kennungen folgt die eigentliche Felddefinition des Datensatzes, in welcher der Kurzname, der Datentyp, die Gesamtfeldlänge, die Nachkommastellen, die Bedeutung und die Einheit eines jeden Feldes beschrieben werden (siehe Abbildung 3 – Felddefinition) (Ingenieurbüro Fischer-Uhrig, 2018b).

Schließlich folgen die Daten der Netzelemente. Diese werden mit Kommentaren zu den in der Felddefinition definierten Beschreibungen, Einheiten und Kurznamen der Felder eingeleitet (siehe Abbildung 3 – Netzelemente). Jede darauffolgende Zeile beschreibt ein Netzelement und wird mit der Satz-Kennung des Netzelementes am Anfang der Zeile eingeleitet z.B. „LEI“ für Leitungsdaten. Wie in Abbildung 3 zu sehen, können die Daten beispielsweise durch den Texttrenner Semikolon voneinander getrennt und somit den Feldern zugeordnet werden.

In STANET® können Netzelemente in vier verschiedene Gruppen eingeteilt werden. Zu den Gruppen gehören „Knoten“, „Elemente an Knoten“, „Zweipolelemente“ und „Elemente auf Leitungen“ (siehe Abbildung 4). Das Netzelement „Knoten“ wird durch seinen Knotennamen identifiziert und enthält als zwingende Attributwerte Koordinatenangaben in einem von STANET® unterstützten Koordinatensystem.

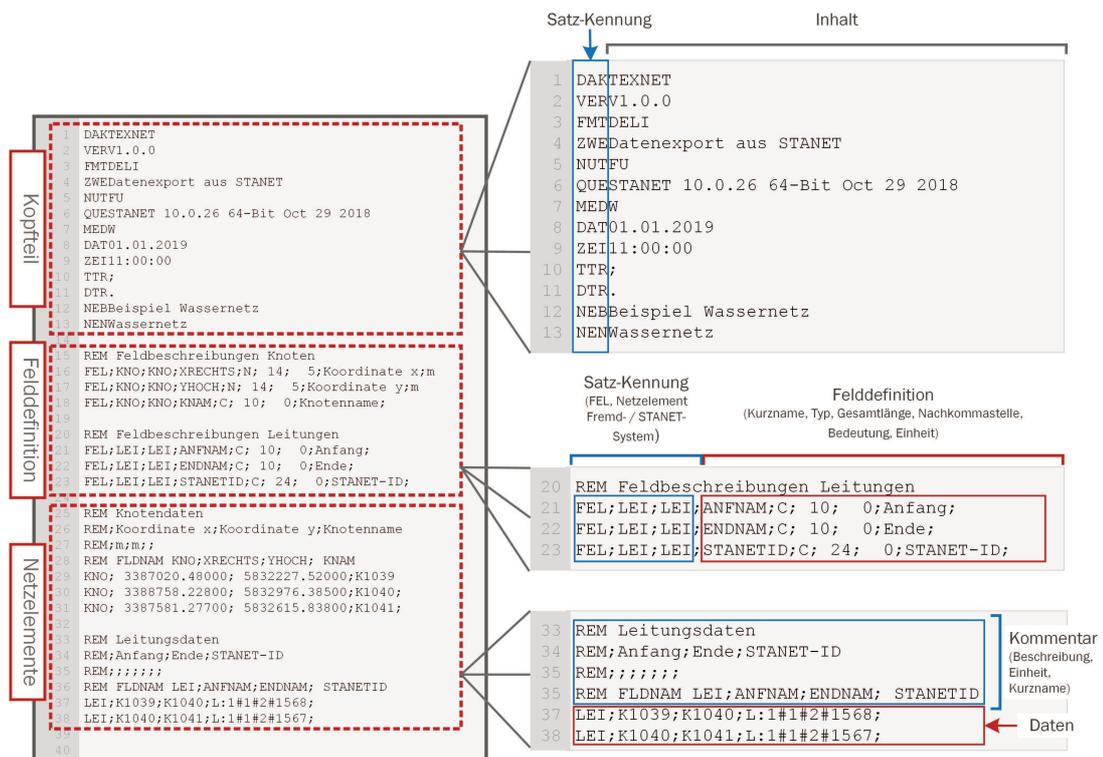


Abbildung 3: Auszug aus einer CSV-Datei mit Einteilung in Dateiabscnitte (links) und Erläuterungen zur Interpretation der Abschnitte (rechts) (nach Ingenieurbüro Fischer-Uhrig, 2018b).

Für die anderen Netzelemente gilt die STANET-ID als eindeutiger Schlüssel. Die STANET-ID wird vom Programmsystem STANET® für neue Netzelemente automatisch erzeugt und dient zur eindeutigen Identifikation dieser (Ingenieurbüro Fischer-Uhrig, 2018b). „Elemente an Knoten“ wie beispielsweise Abnehmer, Behälter und Generatoren verweisen mit dem Feld Knotenname als Fremdschlüssel auf den zugehörigen Knoten. Als „Zweipolelemente“ gelten alle Netzelemente, welche durch einen Anfangs- und Endknoten (Fremdschlüssel auf Knotennamen) definiert werden und aus topologischer Sicht einer Kante entsprechen. Dies trifft auf eine Reihe von Netzelementen, wie beispielsweise Leitungen, Pumpen und Ventile zu. Die letzte Gruppe von Netzelementen sind „Elemente auf Leitungen“. Diese werden analog zu Zweipolelementen durch Anfangs- und Endknoten definiert und dadurch einer Leitung zugewiesen. Topologisch gesehen entsprechen diese Elemente jedoch einem Knoten, weshalb sie zusätzlich Koordinaten aufweisen. Beispiele für „Elemente auf Leitungen“ sind Schieber und Hydranten.

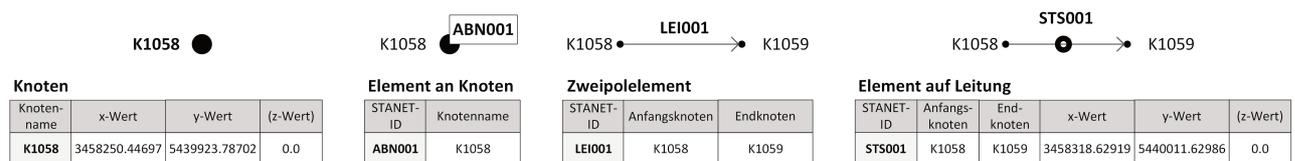


Abbildung 4: Schematische Darstellung der STANET® spezifischen Einteilung der Netzelemente in die Gruppen Knoten, Element an Knoten, Zweipolelement und Element auf Leitung mit zugehörigen zwingenden Attributwerten zur eindeutigen Identifikation.

4.2 Anforderungen an den STANETTransformer

Generelle Anforderungen an den STANETTransformer sind der Datenimport der STANET® CSV Daten sowie der Export in verschiedene semantische Datenformate. Durch die Analyse der Ausgangsdaten werden weitere Funktionalitäten notwendig, wodurch die Anforderungen konkretisiert werden. Die folgenden Anforderungen legen die Rahmenbedingungen für die Entwicklung des Werkzeuges fest.

A. Vollständige Interpretation des STANET® CSV Formates

Das STANET spezifische CSV Format (siehe Kapitel 4.1) soll vollständig interpretiert werden. Dazu müssen alle in den Ausgangsdaten vorhandenen Informationen eingelesen und in einer geeigneten Datenstruktur gespeichert werden. Der Einleseprozess der Daten soll möglichst dynamisch gestaltet sein und robust auf Fehler reagieren.

B. Umsetzung einer graphischen Benutzeroberfläche mit tabellarischer Darstellung der Ausgangsdaten

Der STANETtransformer soll eine graphische Benutzeroberfläche aufweisen. Die Oberfläche soll benutzerfreundlich aufgebaut sein und die eingelesenen Ausgangsdaten in geeigneter Form tabellarisch darstellen.

C. Analyse und Abfragen

Der STANETtransformer soll Möglichkeiten bieten alle eingelesenen Informationen abzufragen. Dazu gehört die Anzeige der Dateinformationen des Kopfteils der Datei. Falls geographische Informationen vorhanden sind, sollen diese sowie die Bounding Box (Eckpunkte des minimal umgebenden Rechtecks um alle Koordinaten) abgefragt werden können. Es soll eine Statistik zum Einleseprozess erstellt werden, aus welcher die eingelesenen, sowie falls vorhanden, die übersprungenen Netzelemente hervorgehen. Bei Einlesefehlern sollen diese in einer aussagekräftigen Meldung angezeigt werden. Das Werkzeug soll außerdem Möglichkeiten zur Netzprüfung der eingelesenen Daten bieten.

D. Datenexport in standardisierte semantische Datenmodelle

Der Datenexport des Netzes soll in mehrere standardisierte Datenmodelle ermöglicht werden. Insbesondere soll der Datenexport in die Formate CityGML Utility Network ADE und IFC möglich sein. Der Fokus liegt auf dem Export des Netzes mit seiner Topologie und der semantischen Informationen.

5 KONZEPT UND REALISIERUNG DES STANETTRANSFORMERS

Das Softwarewerkzeug wird in der Programmiersprache Python Version 3.7 (Python, 2018) entwickelt. Für die graphische Benutzeroberfläche wird die plattformübergreifende GUI Bibliothek wxPython Phoenix (Version 4.0.3) (wxPython, 2018) genutzt. Die Bibliothek ermöglicht das Erstellen von nativen graphischen Benutzeroberflächen mit gewohntem Look-and-Feel. Ergänzend wird das Rapid Application Development (RAD) Werkzeug wxFormBuilder (Version 3.8.1) (wxFormBuilder, 2018) eingesetzt. Dieses Werkzeug bietet die Möglichkeit maschinengenerierten Python-Code für Standardoberflächenelemente zu generieren.

5.1 Konzept

Eine schematische Darstellung des entwickelten Werkzeuges ist in Abbildung 5 zu sehen. Der STANETtransformer bietet die Möglichkeit zum Import des STANET® spezifischen CSV Formates (siehe Kapitel 4.1). Dabei wird die Datei zeilenweise interpretiert und in eine geeignete Datenstruktur überführt. Aus dieser Datenstruktur kann die topologische Struktur des Netzes abgeleitet und auf die Sachdaten der Komponenten zugegriffen werden. Im folgenden Abschnitt wird das Datenhaltungskonzept näher erläutert.

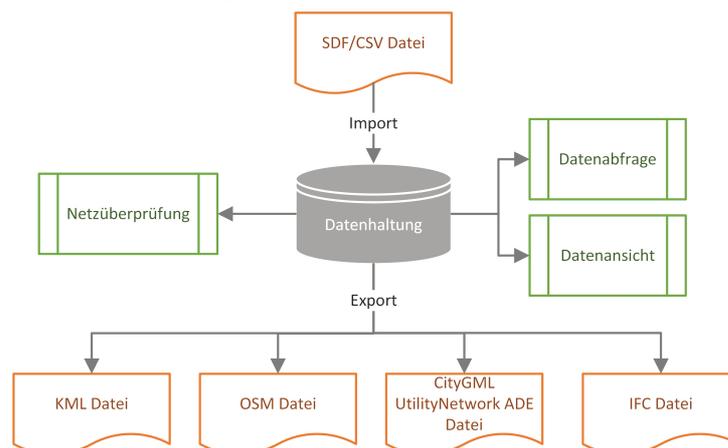


Abbildung 5: Schematische Darstellung des Softwarewerkzeuges

Das Konzept der Speicherung der topologischen Struktur ist in Abbildung 6 veranschaulicht. Die Beziehungen der Komponenten zueinander werden durch zwei assoziative Datenfelder (in Python Dictionary (Python, 2019)) realisiert. Diese weisen eine unsortierte Menge von Schlüsseln auf, welche auf die dazugehörigen Daten verweisen. Hier liegen die Daten in Form einer Liste vor. Das erste Dictionary (Abbildung 6 links oben) enthält als Schlüssel Koordinaten-IDs, welche jede Koordinate eindeutig identifizieren. Diese Schlüssel verweisen auf eine Liste mit Einträgen zu den Koordinaten (hier 2D Koordinaten) und dem eindeutigen Schlüssel der Netzkomponente, welche durch diese Koordinaten

beschreiben wird. Dies können nur Elemente vom Typ „Knoten“ sowie „Elemente auf Leitungen“ (siehe Kapitel 4.1) sein, da nur diese Komponenten Koordinaten zur Identifikation aufweisen. Die nachfolgenden Listeneinträge verweisen mit einem eindeutigen Schlüssel auf benachbarte Komponenten. Nachbarn können „Zweipolelemente“, „Elemente auf Leitungen“ und „Elemente an Knoten“ sein. Ein Beispiel hierfür ist für den Knoten „K1058“ (Koordinaten-ID 1) gegeben. Wie aus Abbildung 6 (oben links) und dem Schaubild Abbildung 6 (rechts) hervorgeht, besitzt dieser Knoten als benachbarte Komponenten ein Zweipolelement (Leitung – LEI001), ein Element am Knoten (Abnehmer – ABN001) und ein Element auf der Leitung (Streckenschieber – STS001). Da der Knoten „K1058“ nur ein Zweipolelement als Nachbar aufweist, kann er als Außenknoten des Netzes identifiziert werden.

Das zweite Dictionary (Abbildung 6 links unten) enthält als Schlüssel die Komponenten-IDs, auf welche im ersten Dictionary in der Liste verwiesen wird. Die Komponenten-IDs verweisen auf eine Liste, welche als ersten Eintrag einen Datenschlüssel enthält. Dieser Schlüssel erlaubt zusammen mit der Komponenten-ID den Zugriff auf die Sachdaten der jeweiligen Netzkomponente. An zweiter, dritter und vierter Stelle der Liste wird eine Referenz auf Koordinaten-IDs aus dem ersten Dictionary gegeben. Beispielsweise verweisen „Knoten“ oder „Elemente an Knoten“ auf eine Koordinaten-ID, durch welche diese beschrieben werden können. „Zweipolelemente“ weisen Referenzen auf die Koordinaten-ID des Anfangs- und Endknotens auf, wobei durch die Reihenfolge der Listeneinträge die Richtung der Netzkomponente resultiert. Schließlich enthält die Liste für ein Element auf einer Leitung drei Referenzen auf Koordinaten-IDs. Die ersten beiden Referenzen fungieren analog zu den Zweipolelementen als Referenz auf den Anfangs- und Endknoten der Leitung. Der dritte Verweis liefert anhand der referenzierten Koordinaten die Position des Elementes auf der Leitung. Somit kann anhand der Länge der verknüpften Liste einer Komponenten-ID die Einteilung der dazugehörigen Komponente in die verschiedenen Elementgruppen erfolgen. Zudem können Rückschlüsse getroffen werden, ob es sich um knotenartige oder kantenartige Komponenten handelt.

Der Zugriff auf die Sachdaten der Netzkomponenten erfolgt ebenfalls durch Dictionaries. Der Datenschlüssel, welcher angibt zu welcher Datentabelle ein Netzelement gehört (z.B. „KNO“ für Knotendaten), dient als Schlüssel für den Zugriff auf die Datentabelle. Auf diese Weise kann effektiv auf die Sachdaten der Netzkomponenten zugegriffen werden.

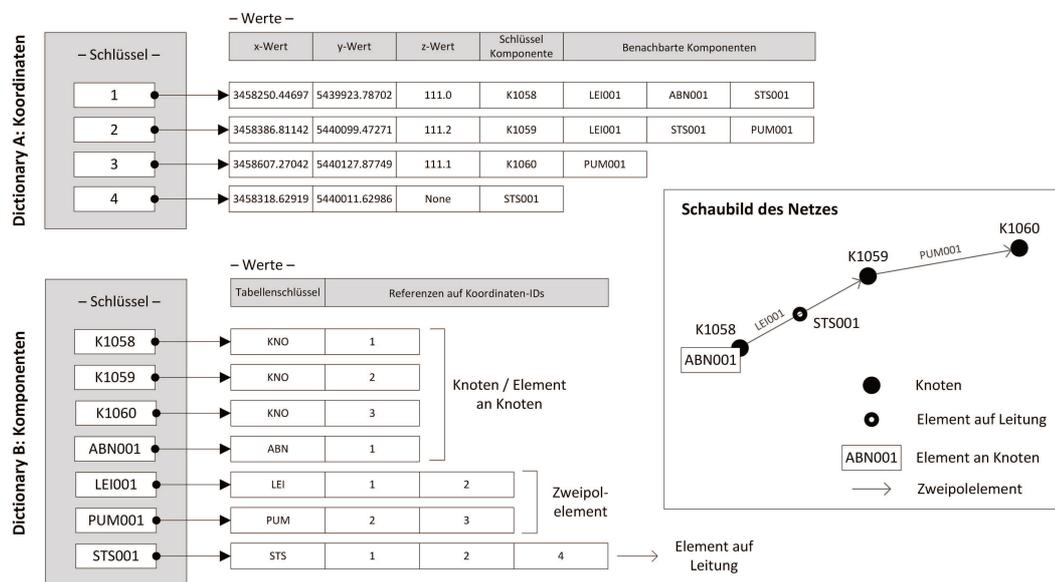


Abbildung 6: Darstellung des Konzepts zur Speicherung der topologischen Netzstruktur anhand eines vereinfachten Beispielnetzes: Aufbau der Datenstruktur anhand zweier Dictionaries (links) und Schaubild des resultierenden Beispielnetzes (rechts).

Der STANETransformer bietet verschiedene Funktionalitäten für die Darstellung sowie Analyse und Prüfung der in der Datenhaltung gespeicherten Netzwerksdaten (siehe Abbildung 5). Die Sachdaten zu den Netzelementen werden durch Tabellen dargestellt, aus welchen auch die Beschreibung der Felder sowie die Einheit der Sachdaten hervorgehen. Allgemeine Daten zum Netzwerk, wie beispielsweise Daten aus dem Kopfteil der Ausgangsdatei oder eventuell vorhandene geographische Daten können in Dialogen abgefragt werden. Zusätzlich zu diesem Darstellungs- und Abfragekonzept erlaubt die Datenhaltung eine Netzprüfung. Hierbei werden knotenartige und kantenartige Netzkomponenten auf wichtige Eigenschaften überprüft.

Beispielsweise wird geprüft, ob alle Knoten Koordinaten aufweisen, diese in der richtigen Dimension vorliegen und keine Knotenduplikate (aufeinanderliegende Knoten) im Netzwerk auftreten. Des Weiteren wird geprüft, ob isolierte Knoten vorkommen, welche keine Verbindung zu einer Kante aufweisen. Auch kantenartige Komponenten können auf Duplikate geprüft werden. Zudem besteht die Möglichkeit zur Prüfung der Zweipolelemente, indem die Anfangs- und Endknoten auf die oben genannten Eigenschaften geprüft werden.

Eine zentrale Funktionalität des STANETTransformers ist der Export von Netzen, semantischen Daten, sowie der Netztopologie. Der Export erfolgt in die Datenformate KeyholeMarkupLanguage (KML) (Open Geospatial Consortium, 2015), OpenStreetMap (OSM) (OSM, 2019), CityGML UtilityNetwork und Industry Foundation Classes (IFC) (siehe Abbildung 5). Die Exportformate KML und OSM dienen hauptsächlich der graphischen Visualisierung des Netzes und bieten nur eingeschränkte Möglichkeiten zur topologischen Modellierung und Speicherung von semantischer Information, wobei OSM deutlich mehr Möglichkeiten als KML bereitstellt. Bei dem Export in die Formate CityGML UtilityNetwork und IFC liegt der Fokus auf der topologischen und semantischen Datenmodellierung. Bei diesen Formaten ist im Gegensatz zu KML und OSM eine deutlich umfangreichere Modellierung der Topologie und Semantik möglich (siehe Kapitel 3.2 und 3.3).

5.2 Realisierung

Die technische Realisierung des STANETTransformers erfolgt unter anderem durch die Module „Import“, „Export“, „Datenhaltung“ und „Datenprüfung“. Die Standardoberflächenelemente werden mithilfe des Werkzeuges wxFormBuilder erstellt. Ihnen werden die eingelesenen Sachdaten zur Visualisierung der Daten zugewiesen.

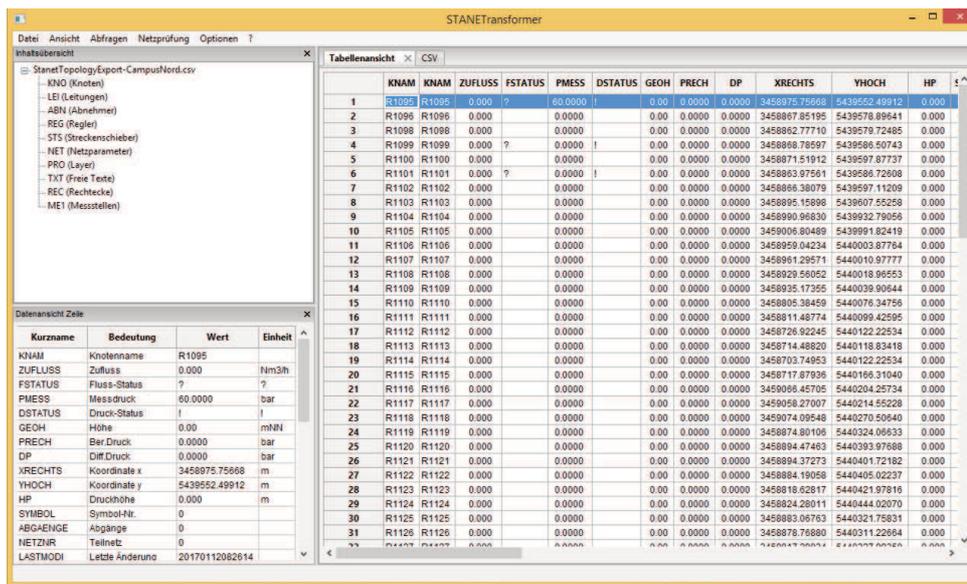


Abbildung 7: Benutzeroberfläche des Softwarewerkzeuges (unter Windows 8.1) mit eingelesenen anonymisierten Netzdaten und Auswahl der Datentabelle Knotendaten.

Die aktuelle Benutzeroberfläche ist in Abbildung 7 dargestellt. Am oberen Fensterrand befindet sich eine Menüleiste mit welcher auf Grundfunktionen wie „Öffnen“ und „Schließen“, sowie auf weitere Funktionalitäten der Datenabfrage, Netzprüfung und dem Export zugegriffen werden kann. Generell teilt sich die Oberfläche in drei Fensterbereiche auf. Links oben befindet sich der Fensterbereich „Inhaltsübersicht“, in welchem nach dem Öffnen einer Datei alle eingelesenen Datentabellen der Netzelemente aufgelistet sind. Bei Auswahl einer Datentabelle werden die dazugehörigen Daten der Netzelemente im zentralen Fenster „Tabellenansicht“ dargestellt. Dieser Fensterbereich ermöglicht die Navigation durch die Netzelemente in einer tabellarischen Struktur, wobei jede Zeile einem Netzelement entspricht. Die Felder geben die Sachdaten des Netzelementes wieder. An der ersten Stelle der Datentabelle steht der eindeutige Schlüssel der Netzkomponenten (die STANET-ID oder der Knotenname), damit diese identifiziert werden kann. Eine genauere Beschreibung der Sachdaten einer Komponente erfolgt im dritten Fensterbereich der Oberfläche (links unten). Hier werden alle Sachdaten der im zentralen Fensterbereich

markierten Komponente mit zusätzlichen Informationen zu der Bedeutung und der Einheit der Felder abgebildet.

Vom Menü aus erfolgt der Zugriff auf den Datenexport in die verschiedenen Datenformate. Aktuell ist der Datenexport in die Formate KML und OSM realisiert. Der OGC Standard KML ermöglicht die graphische Visualisierung der kantenartigen und knotenartigen Netzkomponenten auf einem virtuellen Globus z.B. Google Earth (Google Earth, 2019). Geometrieobjekte werden in KML durch *Placemarks* repräsentiert, welche die Attribute *name* und *description* aufweisen können (Google Developers, 2018a). Weitere spezifische semantische Daten werden nicht unterstützt. Um die Sachdaten der Netzkomponenten dennoch darstellen zu können, werden diese mithilfe des Tags *ExtendedData* (Google Developers, 2018b) einem jeden Geometrieobjekt als Schlüssel-Wert-Paare hinzugefügt. Auf diese Weise können die Sachdaten in einem sogenannten *Balloon* (Pop-up) auf dem virtuellen Globus angezeigt werden (siehe Abbildung 8 links). Da die Koordinatenangaben den Geometrieobjekten direkt zugewiesen sind, bestehen keine topologischen Beziehungen zwischen Kanten und Knoten.

Das Format OSM bietet mehr Möglichkeiten zur topologischen und semantischen Modellierung. OSM unterscheidet zwischen den grundlegenden Objekttypen Knoten (*Node*), Kante (*Way*) und *Relation*. Jedem Objekttyp können beliebig viele Attribute (*Tags*) zugeordnet werden, welche aus einem Schlüssel-Wert-Paar bestehen. Die OpenStreetMap Gemeinschaft empfiehlt bestimmte Schlüssel-Wert-Paare (*Map Features*) (OpenStreetMap Wiki, 2018) für die Beschreibung von Objekten. Für die STANET® Netzwerksdaten wird das Attribut „man_made = pipeline“ in Kombination mit dem Attribut „substance“ genutzt, um semantische Informationen zur Art und dem transportierten Material des Netzwerkes zu speichern. Zusätzlich dazu werden alle Sachdaten der Netzkomponenten als individuelle Attribute gespeichert. Kanten werden in OSM durch Referenzen auf bestehende Knoten definiert, wobei jede Kante anhand der Reihenfolge der angegebenen Knoten eine Richtung aufweist (siehe Abbildung 8 rechts). Somit besteht eine Knoten-Kanten-Beziehung zwischen den Objekten (OpenStreetMap Wiki, 2017).

Der Datenexport in das Datenmodell IFC ist derzeit in Entwicklung und erlaubt wie aus Kapitel 3.3 hervorgeht eine deutlich umfangreichere Beschreibung der Topologie und Semantik. Der nächste Schritt stellt der Export in das Datenmodell CityGML UtilityNetwork ADE dar.

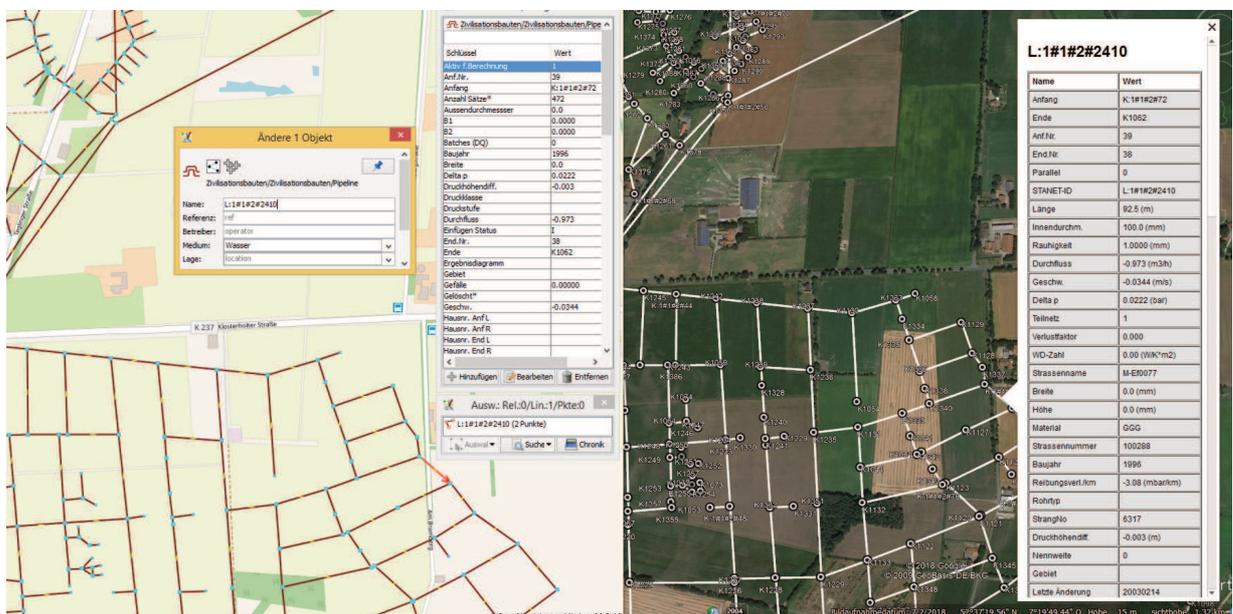


Abbildung 8: Ausschnitt eines exportierten anonymisierten Wassernetzes in den Formaten KML (links) und OSM (rechts) mit Darstellung der Sachdaten einer Leitung (Google Earth, 2018; JOSM, 2018).

6 ZUSAMMENFASSUNG UND AUSBLICK

Die Modellierung von Ver- und Versorgungsnetzen spielt im Hinblick auf vernetzte Energiesysteme eine wichtige Rolle und kann für verschiedene Netzmedien mit dem Programmsystem STANET® realisiert werden. In diesem Beitrag wird ein Softwarewerkzeug zur Analyse und Transformation der STANET® Datensätze in verschiedene semantische Datenmodelle vorgestellt. Hierbei wird bereits ein Großteil der

Anforderungen an das Werkzeug erfüllt (siehe Kapitel 4.2). So können Netze beliebiger Medien mit unterschiedlichen Netzkomponenten interpretiert und in einer Datenstruktur gespeichert werden, über welche auf topologische und semantische Informationen zugegriffen werden kann. Es wurde eine graphische Benutzeroberfläche mit tabellarischer Darstellung der Ausgangsdaten erstellt, sowie Datenabfragen und eine Netzüberprüfung implementiert. Schließlich wurde der Datenexport in die Datenmodelle KML und OSM realisiert, wobei die topologische und semantische Datenmodellierung im Rahmen der Möglichkeiten der beiden Datenmodelle umgesetzt wurde.

Im weiteren Verlauf des Projekts wird die Entwicklung des Exports in das IFC Datenmodell abgeschlossen sowie der Export in das Datenmodell CityGML UtilityNetwork realisiert. Bei beiden Formaten liegt der Fokus auf der topologischen und semantischen Datenmodellierung.

7 LITERATURVERZEICHNIS

- BECKER, T., NAGEL, C., KOLBE, T. H., 2011. Integrated 3D Modeling of Multi-utility Networks and Their Interdependencies for Critical Infrastructure Analysis. In: Advances in 3D Geo-Information Sciences. Springer Berlin Heidelberg.
- BECKER, T., NAGEL, C., KOLBE, T.H., 2013. Semantic 3D Modeling of Multi-Utility Networks in Cities for Analysis and 3D Visualization. In: Progress and New Trends in 3D Geoinformation Sciences. Springer Berlin Heidelberg.
- BORRMANN, A., KÖNIG, M., KOCH, C., BEETZ, J., 2015. Building Information Modeling. Springer Fachmedien Wiesbaden, Wiesbaden.
- CITYGML WIKI, 2016. UtilityNetworkADE Version 0.2.0. UML Diagram for FeatureMaterial Model. Verfügbar unter: http://www.citygmlwiki.org/images/c/c0/FeatureMaterial_-_UtilityNetworkADE.pdf (12.01.19).
- GOOGLE DEVELOPERS, 2018a. KML Reference Placemark. Verfügbar unter: <https://developers.google.com/kml/documentation/kmlreference> (14.01.19).
- GOOGLE DEVELOPERS, 2018b. KML Reference Adding Custom Data. Verfügbar unter: <https://developers.google.com/kml/documentation/extendeddata> (14.01.19).
- GOOGLE EARTH, 2019. Outreach. Verfügbar unter: <https://www.google.com/earth/outreach/learn/> (08.01.19).
- GOOGLE EARTH V 7.3.2.5495, 02.07.2018. 52°37'19.56"N, 7°19'49.44"O, Höhe 1.32 km. GeoBasis-DE BKG. Verfügbar unter: <https://www.google.com/earth/index.html> (14.01.19).
- GRÖGER, G., KOLBE, T.H., NAGEL, C., HÄFELE, K.-H., 2012. OGC City Geography Markup Language (CityGML) Encoding Standard 344.
- BUILDINGSMART, 2015. IFC4 – Addendum 2 (Final Standard). Verfügbar unter: <http://www.buildingsmart-tech.org/ifc/review/IFC4Add2/ifc4-add2-rv/html/> (06.01.19).
- INGENIEURBÜRO FISCHER-UHRIG, 2018a. STANET Netzberechnung. Verfügbar unter: <http://stafu.de/de/home.html> (19.12.18).
- INGENIEURBÜRO FISCHER-UHRIG, 2018b. STANET Handbuch (Handbuch No. 10.0.24). Berlin.
- ISO 10303-11:2004, 2004. Industrial automation systems and integration - Product data representation and exchange - Part 11: Description methods: The EXPRESS language reference manual.
- ISO 16739:2013, 2013. Industry Foundation Classes (IFC) for data sharing in the construction and facility management industries.
- JOSM V 14620, 31.12.2018. 52.62926°N, 7.3330521°O. OpenStreetMap Carto (Standard) Zoomstufe 16. Verfügbar unter: <https://josm.openstreetmap.de/> (14.01.19).
- NOUVEL, R., BRASSEL, K-H, BRUSE, M., DUMINIL, E., COORS, V., EICKER, U., ROBINSON, D., 2015. SIMSTADT, a new workflow-driven urban energy simulation platform for CityGML city models. Präsentiert auf der CISBAT 2015, Lausanne, Switzerland.
- OPEN GEOSPATIAL CONSORTIUM, 2015. OGC KML 2.3 Encoding Standard 266.
- OPENSTREETMAP WIKI, 2018. Map Features. Verfügbar unter: https://wiki.openstreetmap.org/wiki/DE:Map_Features (08.01.19).
- OPENSTREETMAP WIKI, 2017. Elements. Verfügbar unter: <https://wiki.openstreetmap.org/wiki/Elements> (14.01.19).
- OSM, 2019. OpenStreetMap web page. Verfügbar unter: <https://www.openstreetmap.org/about> (05.01.19).
- OTILLINGER, F., FISCHER-UHRIG, F., AHRENS, J., 2008. Netzerneuerung mit Köpfchen spart Geld und schafft Vorteile. 3R Int. 47, 10.
- PYTHON V 3.7.0, 2018. Python 3.7.0. Verfügbar unter: Python.org. URL <https://www.python.org/> (07.01.19).
- PYTHON, 2019. Data Structures - Python 3.7.2 Documentation: Dictionaries. Verfügbar unter: <https://docs.python.org/3/tutorial/datastructures.html#dictionaries> (13.01.19).
- WXFORMBUILDER V 3.8.1, 08.08.2018. RAD tool for wxWidgets GUI design. Verfügbar unter: <https://github.com/wxFormBuilder/wxFormBuilder> (06.09.18).
- RUF, J., ZIMMERLIN, M., SAUTER, P.S., KOPPEL, W., SURIYAH, M.R., KLUWE, M., HOHMANN, S., LEIBFRIED, T., KOLB, T., 2018. Simulation Framework for Multi-Carrier Energy Systems with Power-to-Gas and Combined Heat and Power. In: 2018 53rd International Universities Power Engineering Conference (UPEC). Präsentiert auf der 2018 53rd International Universities Power Engineering Conference (UPEC), IEEE, Glasgow, United Kingdom.
- SHAFRANOVICH, Y., 2005. RFC 4180: Common Format and MIME Type for Comma-Separated Values (CSV) Files 2007.
- STAKIC, D.E., EBE, F., 2017. Sektorkopplung als Baustein in der Energiewende – Ergebnisse einer bidirektionalen Simulation von Energienetzen.
- WXPYTHON V 4.0.3, 2018. WxPython - GUI Toolkit Python. Verfügbar unter: <https://wxpython.org/index.html> (07.01.19).