Platone

PLATFORM FOR OPERATION OF DISTRIBUTION NETWORKS

Platone

PLATform for Operation of distribution NEtworks

D6.10 v1.0

Standardised grid models

Project name	Platone
Contractual delivery date:	28.02.2021
Actual delivery date:	28.02.2021
Main responsible:	Enea Bionda, RSE
Work package:	WP6 – Standardisation, Interoperability and Data Handling
Security:	CO
Nature:	R
Version:	V1.0
Total number of pages:	29

Abstract

This deliverable describes the methodology that was used to translate Italian, German and Greek demo grids from DIgSILENT data format to IEC Common Information Model, using a Model Driven Architecture approach.

Keyword list

Standard – Grid Model – Meta-Model – DigSilent – IEC CIM – Interoperability – Model Driven Architecture

Executive Summary

In the context of the Platone project, aiming to develop an architecture for testing and implementing a new data acquisition system based on a two-layer approach, an important issue regards standardization and legislation. The activity described in this report focuses on the development of standardized grid models, deploying the IEC Common Information Models (CIM) standards.

CIM represents concepts, relationships, constraints, rules and operations to specify data semantics for the electrical domain and has been adopted by the International Electrotechnical Commission as a common semantic model that describes the components of a power system at an electrical level and the relationship between each component. The use of a common model is useful to facilitate the exchange of power system network data between companies and to allow the exchange of data between applications within a company, increasing functionality for managing and optimizing the electrical system. This report describes a methodology to transform the three electrical grid demos from the DIgSILENT data format to the IEC CIM, using a Model Driven Architecture approach. The results of the activity are a partial DIgSILENT model, used to describe a portion of electrical grid, and the CIM instances of the considered networks. The transformation has been validated comparing the power flow results.

The results will be used to show the possibility to use IEC CIM standard in combination with the Platone DSO Technical Platform (DSOTP) to test existing or future tools of the platform.

Authors and Reviewers

Main responsible		
Partner	**Name**	**E-mail**
RSE		
	Enea Bionda	enea.bionda@rse-web.it
Author(s)/contributor(s)		
Partner	**Name**	
RSE		
	Gabriele Paludetto Francesca Soldan	
Areti		
	Maurizio Ceotto	
HEDNO		
	Eleni Daridou Stavroula Tzioka	
Reviewer(s)		
Partner	**Name**	
RWTH		
	Padraic McKeever	
Avacon		
	Benjamin Petters	
Approver(s)		
Partner	**Name**	
RWTH		
	Padraic McKeever	

Table of Contents

1 Introduction

The project "PLATform for Operation of distribution NEtworks – Platone" aims to develop an architecture for testing and implementing a data acquisition system based on a two-layer approach (an access layer for customers and a Distribution System Operator (DSO) observability layer) that will allow greater stakeholder involvement and will enable an efficient and smart network management. The tools used for this purpose will be based on platforms able to receive data from different sources, such as weather forecasting systems or distributed smart devices spread all over the urban area. These platforms, by talking to each other and exchanging data, will allow collecting and elaborating information useful for DSOs, transmission system operators (TSOs), customers and aggregators. In particular, the DSO will invest in a standard, open, non-discriminating, economic dispute settlement blockchain-based infrastructure, to give to both the customers and to the aggregator the possibility to more easily become flexibility market players. This solution will see the DSO evolve into a new form: a market enabler for end users and a smarter observer of the distribution network. By defining this innovative two-layer architecture, Platone removes technical barriers to the achievement of a carbon-free society by 2050 [1], creating the ecosystem for new market mechanisms for a rapid roll out among DSOs and for a large involvement of customers in the active management of grids and in the flexibility markets. The Platone platform will be tested in three European trials (in Greece, Germany and Italy) and within the Distributed Energy Management Initiative (DEMI) in Canada. The Platone consortium aims to go for a commercial exploitation of the results after the project is finished. Within the H2020 programme "A single, smart European electricity grid" Platone addresses the topic "Flexibility and retail market options for the distribution grid".

WP6 focuses on the topics of standardization and legislation. With regards to standardization, the goal is to support the Platone demos on this aspect by presenting and analysing the standardization ecosystem and highlight standards that are relevant to Platone. This is important because in this way the Platone project demos will have a clear reference on which standards can be used and which functionalities lack any standardization.

1.1 Task 6.3

The purpose of Task 6.3 is to develop standardized grid models, which will enable better interoperability of devices, cooperation among the involved parties and exchange of data that is of great importance to the rest of the activities of the project. In order to develop the grid models the CIM standard will be employed. The results will be used to show the possibility to use IEC CIM standard in combination with the Platone DSO Technical Platform (DSOTP) to test existing or future tools of the platform.

1.2 Objectives of the Work Reported in this Deliverable

The objective of the work reported in this deliverable is to share the developed methodology for the translation from a non CIM model description (like DIgSILENT data format) into an IEC CIM grid model description.

1.3 Outline of the Deliverable

Section 2 discusses the Model-Driven-Approach used to realise the mapping between the DIgSILENT model and IEC CIM model. In section 3 the results obtained for the Italian demo are presented. Section 4 concludes the report.

1.4 How to Read this Document

This document records the mapping procedure between two different grid models. The reader is not required to have any specific knowledge, as the main concepts about Information Models about Model-Driven-Transformations will be introduced. The reader is not expected to have a detailed knowledge of Platone or the specific standards but is expected to have a basic understanding of electrical engineering, ICT and the modern power system structure.

2 Electrical grid transformation: a model-driven approach

2.1 Introduction

The purpose of the here described activity is the mapping of electrical grids. In order to develop the grid models one or both of the CIM and SAREF standards could be employed. Being more interested in distribution grid aspects, like topology, geographical information and to be able to manage power flow calculation, the choice falls on CIM standard. In this deliverable we describe the transformation from DIgSILENT DGS data format to the IEC 61970 [2, 3].

IEC 61970 defines the part of Common Information Model (CIM) used by EMSs, which includes the core components such as wires, switches, transformers and the connectivity of these devices. The standard CIM was developed by the Electric Power Industry (EPRI) and has been adopted by the International Electrotechnical Commission (IEC) as a common semantic model that describes the components of a power system at an electrical level and the relationship between each component.

Besides IEC 61970, CIM consists of IEC 61968 [4], which extends the model to cover other aspects of power system software data exchange (such as asset tracking, work scheduling and customer billing), and of IEC 62325 [5], which covers the data exchanged between participants in electricity markets [6].

CIM is defined as an information model, which represents concepts, relationships, constraints, rules and operations to specify data semantics for the electrical domain.

The use of a common model has two primary aims [7]: to facilitate the exchange of power system network data between companies and to allow the exchange of data between applications within a company, increasing functionality for managing and optimizing the electrical system [6].

An example about the use of CIM and how it can help to integrate different systems is presented in Figure 1.

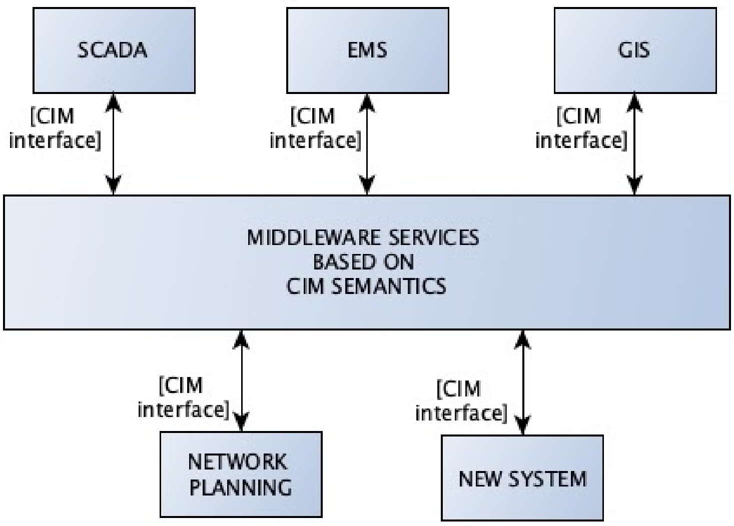


Figure 1 - Example about the use of CIM in order to integrate different systems

However, different standards and data formats are still available for utilities to describe and operate electrical grids, DIgSILENT providing one of them: the DGS (DIgSILENT) data format [8, 9]. DIgSILENT (PowerFactory) software is the power system analysis software application of German DIgSILENT GmbH company, which contains almost all the conventional functions for Power System Analysis. It covers the full range of functionalities from standard features to highly sophisticated and advanced applications including windpower, distributed generation, real-time simulation and performance monitoring for system testing and supervision [9]. DIgSILENT PowerFactory provides a standard interface, named DGS, for data exchange with other applications. The DGS import interface allows importing of complete network models as well as updating existing models. The DGS export interface provides the possibility to export network model data and calculation results. DGS uses a characteristic format, which defines the structure of the electrical grid data [8].

In the context of this specific activity, we had at disposal the DGS description of portions of electrical grids (Italian Rome ARETI grid, German Twistringen Avacon grid and Greek Mesogeia HEDNO grid); from this pre-existing data structure we had to extract the DGS model of the considered network. It is important to note that the developed model is not the complete DGS model, but the portion of DGS model used to describe the considered networks and it does not include all the possible classes and particular cases. Therefore, the aim of the present activity is to express in a standard format the three different demo grids, whereas the development of the complete DGS model goes beyond this work.

The mapping of DGS Model to IEC CIM Model has been realized through a Model Driven Architecture approach, based on the languages of the source and target models. In this domain, the syntax of such a language is usually called a meta-model: a model is an instance of a meta-model when it conforms to the specifications of the meta-model [10]. Finally, the IEC CIM model of the considered portion of the electrical grid could be extracted as output.

In conclusion, the main results of this activity are the DGS meta-model, retrieved from a specific DGS grid description, and the CIM instances of the considered network. We specify that the CIM version used is IEC CIM 15 [11].

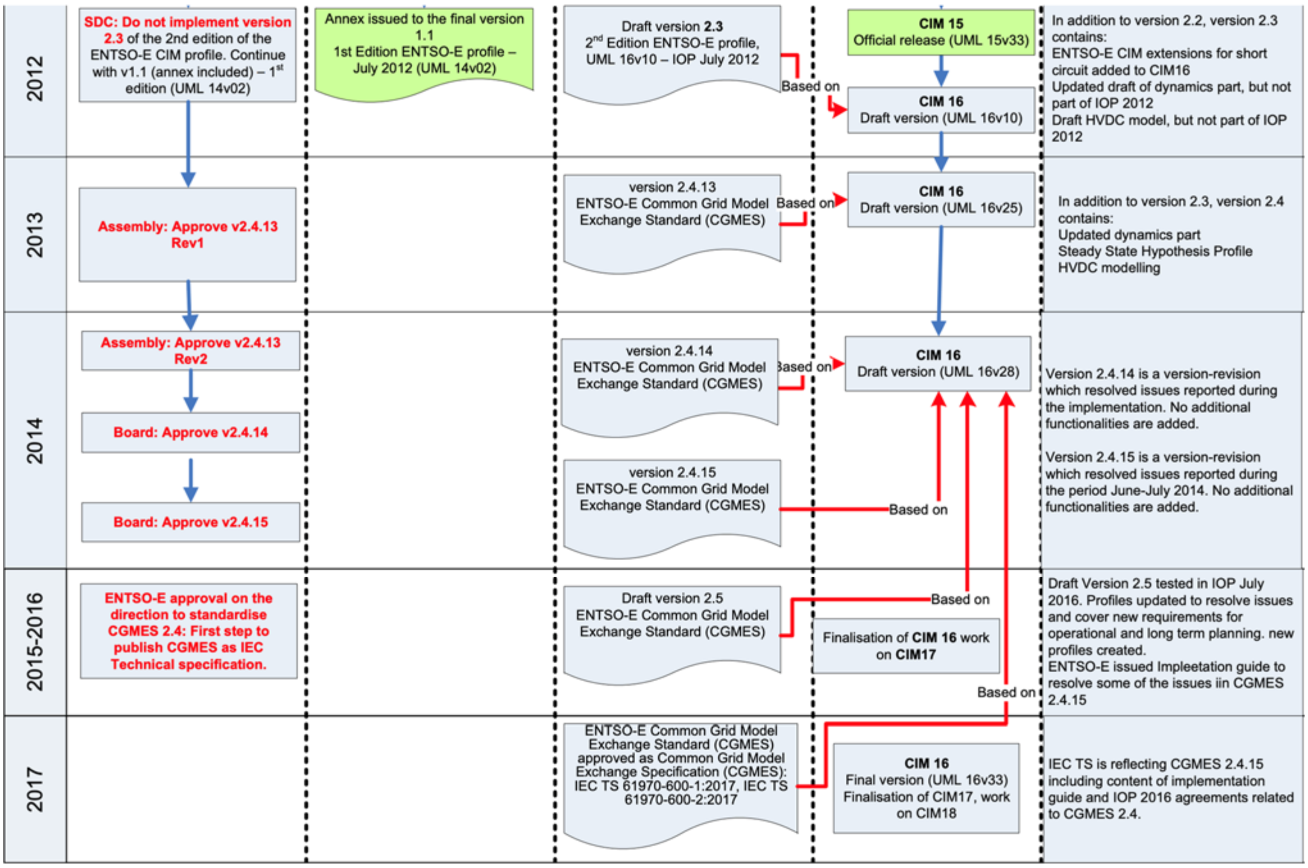


Figure 2 - ENTSO-E CGMES and IEC CIM history [11]

We opted for the IEC CIM 15 because it is the Official Release and probably the most used in real use cases.

Maybe the choice of a more recent version could help us for a better representation of some grid components as renewable energy sources (RES) elements. Once the mapping between DGS and IEC CIM models has been formally established, the same transformation can be easily applied to other DGS network portions; this can be considered as the main advantage of the proposed process, that can be able to transform a complete medium/low voltage network of a metropolitan city such as Rome. Problems for the translation arise (that we have experienced, too) when different utilities use customized versions of the source meta-model, in order to solve their specific issues. The necessity of the choice of a common standard arises in these cases but keeping the transformation between the source and target models as general as possible is a real challenge.

2.2 Model Driven Architecture (MDA) approach

2.2.1 Main MDA concepts and definitions

The Object Management Group (OMG) defines a Model Driven Architecture (MDA) as an approach to derive value from models and modelling, helping to deal with the complexity and interdependence of complex systems [12]. There are multiple ways to derive value; in our case the derivation is realised through automated transformation, which involves a "source model" (with some parameters for how that source model is to be interpreted), a "transformation" and a "target artifact" (in many cases another model). This derivation process also includes logical inference, defined in ontology languages such as OWL and common logic [12].

Before explaining the practical activity realised, some of the main MDA concepts and definitions are reported [12]:

- **System**: is a collection of parts and relationships among these parts, that may be organized to accomplish some purposes;
- **Model**: is a selective representation of some systems whose form and content are chosen based on a specific set of concerns. A model should include the set of information about a system that is within scope, the integrity rules that apply to the system and the meaning of terms used;
- **Modelling Language**: is constituted by the structure, terms, notations, syntax, semantics and integrity rules that are used to express the model. A model is said to conform to a modelling language, meaning that the model should satisfy the defined rules. The best way to express a modelling language is as a model, called a **meta-model**; a meta-model is a model that defines the structure of a modelling language and is also expressed using a modelling language [13].

A recurring problem of meta-modeling is how to set the initial meta-model. If a meta-model is a model of a modelling language, there must be a meta-model describing its modelling language, and so on, in higher levels and more abstract meta-models. The common solution to overcome this problem is to use a language that, at a particular level of this hierarchy, describes itself in its own language. In the field of modelling languages, the solution proposed by OMG is based on a four-layer architecture, as depicted in Figure 3. At the top of that hierarchy there is the meta-metamodeling layer (designed as M3) that is mainly responsible for providing a language to specify meta-models, instantiated from its own model. In the layer below (designed as M2), meta-models are defined by instantiation of the meta-metamodel (each element of the meta-model is an instance of an element defined in the meta-metamodel). In the layer below M2 (designed as M1), the models are defined according to the interest and needs of its users. Finally, the lowest level of hierarchy (the M0 layer) contains real instances of elements defined in the model that actually exist in the context of computational environment or the real world [13].

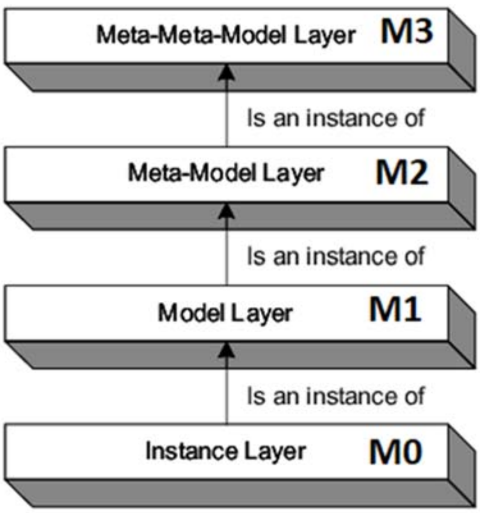


Figure 3 - The meta-model definition: relationships between meta-model and model

2.2.2 Activity of mapping between models

The block diagram of the complete activity reported in this deliverable is described in Figure 4; in this section the whole process will be described in detail.

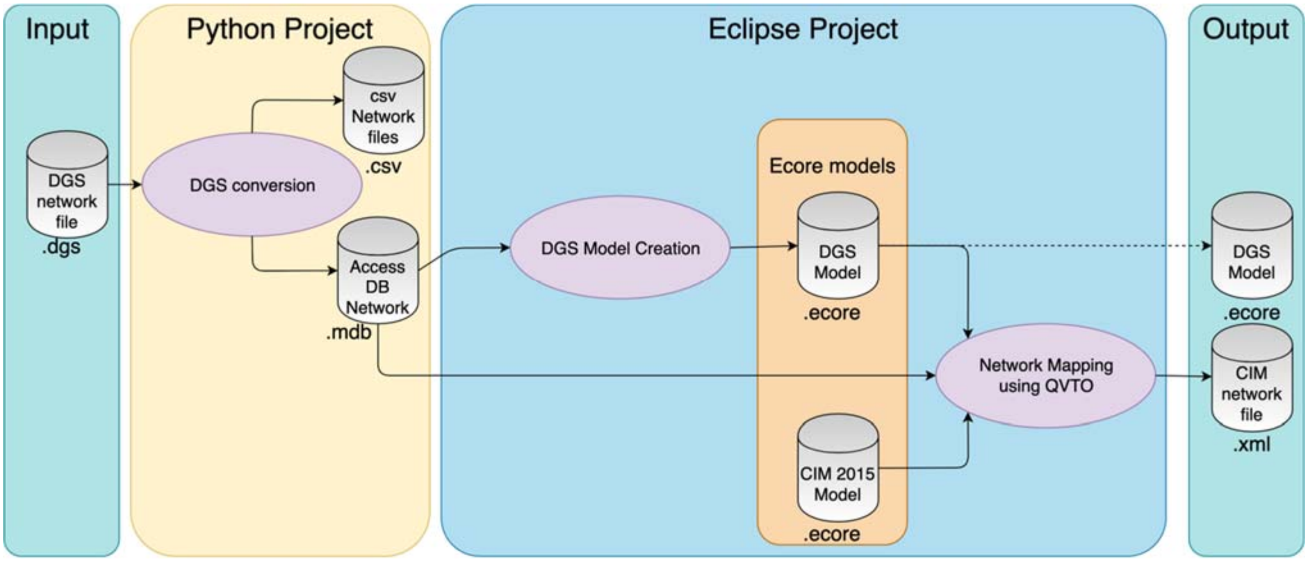


Figure 4 - Block diagram describing the process of creating the DGS meta-model and the mapping from the DGS to the IEC CIM meta-model. The operation has been done in the Eclipse Modelling Framework (EMF) [14], where the meta-model format is called Ecore. The resulting outputs are the DGS Ecore model and the grid instances in IEC CIM format

As introduced in section 2.1, the input of the transformation (our source model) was a serialization of a portion of an electrical grid in DGS data format; an example of input file is presented in Figure 5.

DGS RSE.dgs

```
********************************************************************************
*
* DIgSILENT (R) DGS Export V1.0.33
* Copyright (C) DIgSILENT GmbH 2020. All rights reserved.
*
* DIgSILENT (C) PowerFactory V14.0.519
*
* Date:     11/27/2020 15:37
* Project:  02_Cluste 134 (Study Case: Cluster_134)
*
********************************************************************************


$$General;ID(a:40);Descr(a:40);Val(a:40)
********************************************************************************
*  General Information
*
*  ID: Unique identifier for DGS file
*  Descr: Setting
*  Val: Value
********************************************************************************
  1;Version;5.0


$$ElmCoup;ID(a:40);loc_name(a:40);for_name(a:20);bus1(p);bus2(p);on_off(i);isclosed(i)
********************************************************************************
*  Breaker/Switch
*
*  ID: Unique identifier for DGS file
*  loc_name: Name
*  for_name: Foreign Key
*  bus1: Terminal i in StaCubic
*  bus2: Terminal j in StaCubic
*  on_off: Closed
*  isclosed: Actual State:open:closed
********************************************************************************
  ##EI-030606;I-01394A-2-030606;EI-030606;##NS01394A5;##NFI01394A20306062;0;0
  ##EI-014639;I-80740A-2-014639;EI-014639;##NS80740A5;##NFI80740A20146392;0;0
  ##EI-002194;I-00086A-5-002194;EI-002194;##NP00086A4;##NFI00086A50021942;1;1
  ##EI-080487;I-00086A-6-080487;EI-080487;##NP00086A1;##NFI00086A60804872;1;1
  ##EI-074086;I-31402C-19-074086;EI-074086;##NP31402C4;##NFI31402C190740862;1;1
  ##ES-000058;S-01111A-000058;ES-000058;##NF1S01111A0000581;##NF2S01111A0000581;1;1
  ##ES-000060;S-99171A-000060;ES-000060;##NF1S99171A0000601;##NF2S99171A0000601;1;1


$$ElmFeeder;ID(a:40);for_name(a:20);loc_name(a:40);chr_name(a:20);desc:0(a);desc:1(a);obj_id(p)
********************************************************************************
*  Feeder
*
*  ID: Unique identifier for DGS file
*  for_name: Foreign Key
*  loc_name: Name
*  chr_name: Characteristic Name
*  desc: Description
*  obj_id: Cubicle in StaCubic*
********************************************************************************
  ##SEL_20_09A-NP-000276;SEL_20_09A-NP-000276;SEL_20_09A-NP-000276;MARKETGS;SEL_20_09A-MARKETGS|297;0;##NP31402C4
  ##VLR_20_01A-NP-000250;VLR_20_01A-NP-000250;VLR_20_01A-NP-000250;CAMPUS;VLR_20_01A-CAMPUS|324;0;##NP00086A4
  ##VLR_20_05A-NP-000250;VLR_20_05A-NP-000250;VLR_20_05A-NP-000250;MEDAGLIA;VLR_20_05A-MEDAGLIA|297;0;##NP00086A1
```

Figure 5 - Example of electrical grid description in DGS data format

Generally, DGS defines the structure of the data and is therefore not bound to a specific file format or database schema; it supports the databases Oracle, MS-SQL and ODBC System DSN and can deal with data in the formats ASCII Text, XML, Microsoft Excel and Microsoft Access [8].

Therefore, different serializations are available; in the context of the present activity, we have focused on the mapping of DGS serialization, expressed in Figure 5.

The core principle of DGS is to describe grid elements using a relational database, where exactly one table is required for each class. The name of the table corresponds to the class name and the columns are the attributes of the specific class, one of which must be the "ID" column [8], which takes the role of key in the table. As usual in the case of relational databases, keys are attributes that help to identify a row in a table and allow to find the relationship between two tables.

Keys can be distinguished in primary keys and foreign keys: a primary key constraint is a column that uniquely identifies every row in the table of the relational database management system, while a foreign key is a column that creates a relationship between two tables.

Devices in DIgSILENT data model are connected by ElmTerm, StaCubic and StaSwitch [15]. Using a line between two buses as an example, DIgSILENT topology data model is shown in Figure 6, where [15]:

- ElmTerm stands for Bus/Node;
- StaCubic belongs to ElmTerm and its fold_id is consistent with the corresponding ElmTerm's ID;
- ElmLne's bus1 and bus2 are consistent with StaCubic's IDs in the corresponding bus, and StaCubic's obj_id is consistent with ElmLne's ID, meaning that the line is connected to the bus;
- StaSwitch belongs to StaCubic and StaSwitch's fold_id, consistent with the corresponding StaCubic's ID. Setting StaSwitch's on_off can change the device's operation status.

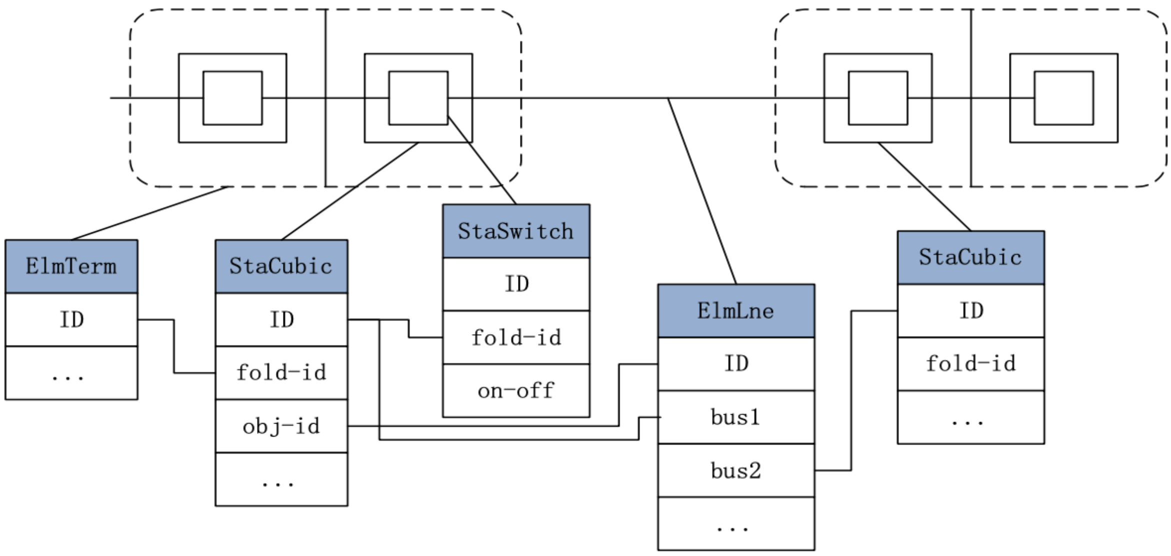


Figure 6 - Topology data models in DIgSILENT [15]

As a comparison, IEC CIM uses Terminals and Connectivity Nodes to define how components within a power system network join together: every piece of conducting equipment has one or more Terminals associated with it and these Terminals are connected through Connectivity Nodes [7]. Therefore, IEC CIM does not model interconnections by associating each component directly with the other components it connects to but indirectly via Terminals and Connectivity Nodes, as better depicted in Figure 7 for a simple circuit.

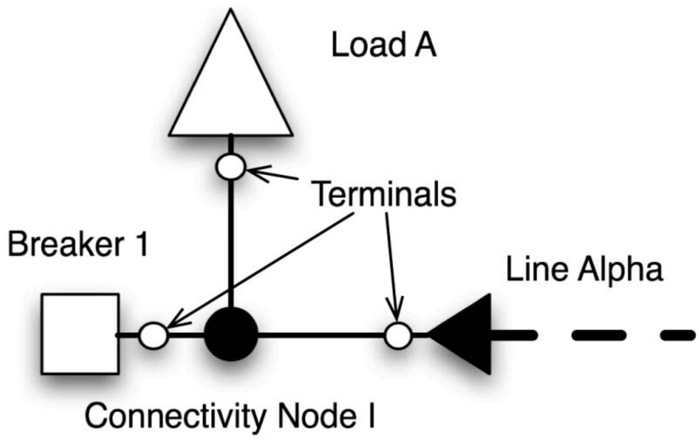


Figure 7 - Connectivity Example circuit with Connectivity Node and Terminals [7]

Considering in more detail the DGS class StaCubic, it has the following attributes [8]:

- ID: unique identifier for DGS file;
- loc_name: name;
- fold_id: the cubicle's belonging area in the grid;
- obj_bus: Bus Index (side of connection);
- obj_id: ID of the element connected to the cubicle;
- chr_name: Characteristic Name.

The table in Figure 8 represents the DGS description of an element of class StaCubic from the ARETI electrical grid in Microsoft Access file format; it is evident that the formal DGS standard is not completely reflected, but some model customizations are introduced: the attribute obj_id is called cBusBar, the attribute chr_name is called for_name, while fold_id is not used.

StaCubic

ID	loc_name	obj_id	for_name	cBusBar
##NF1S01111A0000581	Cub_1	##ES-000058	NF1S01111A0000581	##NF1-S-000058

Figure 8 - Instance of the class StaCubic in DGS standard (Microsoft Access file format) for the ARETI electrical grid

As a comparison, in Figure 9, we present an instance of class StaCubicle from the Twistringen Avacon grid; in this case the only not used attribute is chr_name, there are two more attributes (data_src and OP) and the ID column is named FID.

StaCubic

FID	loc_name	fold_id	obj_id	obj_bus	dat_src	OP
842	CUB1	112	324	0	GIS	C

Figure 9 - Instance of the class StaCubic in DGS standard (Microsoft Access file format) for the Avacon electrical grid

All these utility customizations make it difficult to develop a general mapping from the DGS standard to IEC CIM standard. However, it is a general practice for utilities to use models in a customized way, in order to solve specific issues and grid peculiarities. During the implementation phase the formal standard undergoes changes and becomes a dialect, which uses classes and attributes in a specific way, often to compensate for the standard lack of clarity and completeness in order to deal with specific situations.

Cubicle elements in dgs are mapped into instances of the class Terminal in the IEC CIM model, with the following main attributes:

- name;
- connected: true implies the terminal is connected to the related topological node and false implies it is not;
- ConnectivityNode: ID of the element connected to the Terminal;
- sequenceNumber: orientation of the terminal connections for a multiple terminal conducting equipment (the sequence numbering starts with 1 and additional terminals should follow in increasing order).

In Figure 10 an example of Terminal instance is shown, using the characteristic CIM RDF/XML serialization. The extra attributes are useful to define a topological description of the grid.

```xml
<cim:Terminal rdf:ID="_92b1cbe0-5849-11e4-a400-4437e60e960b">
  <cim:IdentifiedObject.name>Terminal  1 Q12C3 Busbar Q12C</cim:IdentifiedObject.name>
  <cim:Terminal.connected>true</cim:Terminal.connected>
  <cim:Terminal.sequenceNumber>1</cim:Terminal.sequenceNumber>
  <cim:IdentifiedObject.DiagramObjects rdf:resource="#_92b1cbe1-5849-11e4-a400-4437e60e960b" />
  <cim:Terminal.ConductingEquipment rdf:resource="#_2f3c5ee2-5849-11e4-a400-4437e60e960b" />
  <cim:Terminal.ConnectivityNode rdf:resource="#_6f02d130-5849-11e4-a400-4437e60e960b" />
</cim:Terminal>
```

Figure 10 - Example of Terminal data description in CIM standard

The elements defined in Figure 8 and Figure 10 are instances respectively of the classes StaCubic (belonging to the model DGS) and Terminal (belonging to the model CIM).

Having at disposal an example of defined instances of DGS classes, the aim was to retrieve the useful portion of DGS meta-model; this operation has been done in the Eclipse Modelling Framework (EMF) [14], where the meta-model format is called Ecore.

Therefore, the first result of the presented activity is a part of the DGS model in Ecore format, as presented in Figure 11.

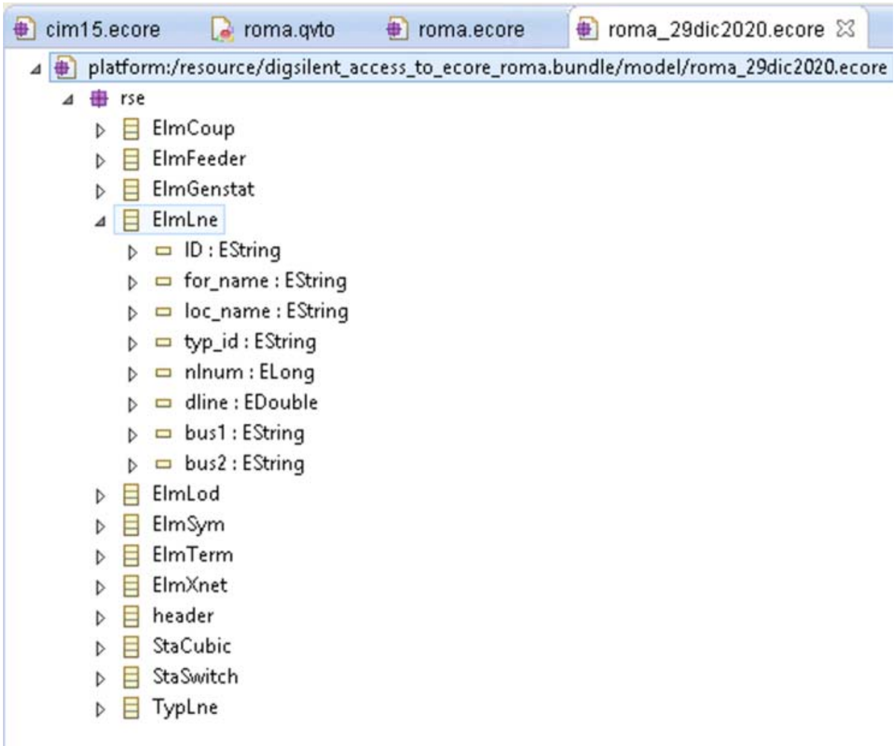


Figure 11 - DGS meta-model in Ecore format

The IEC CIM Ecore meta-model (Figure 12) had already been generated in the context of previous activities conducted by RSE.

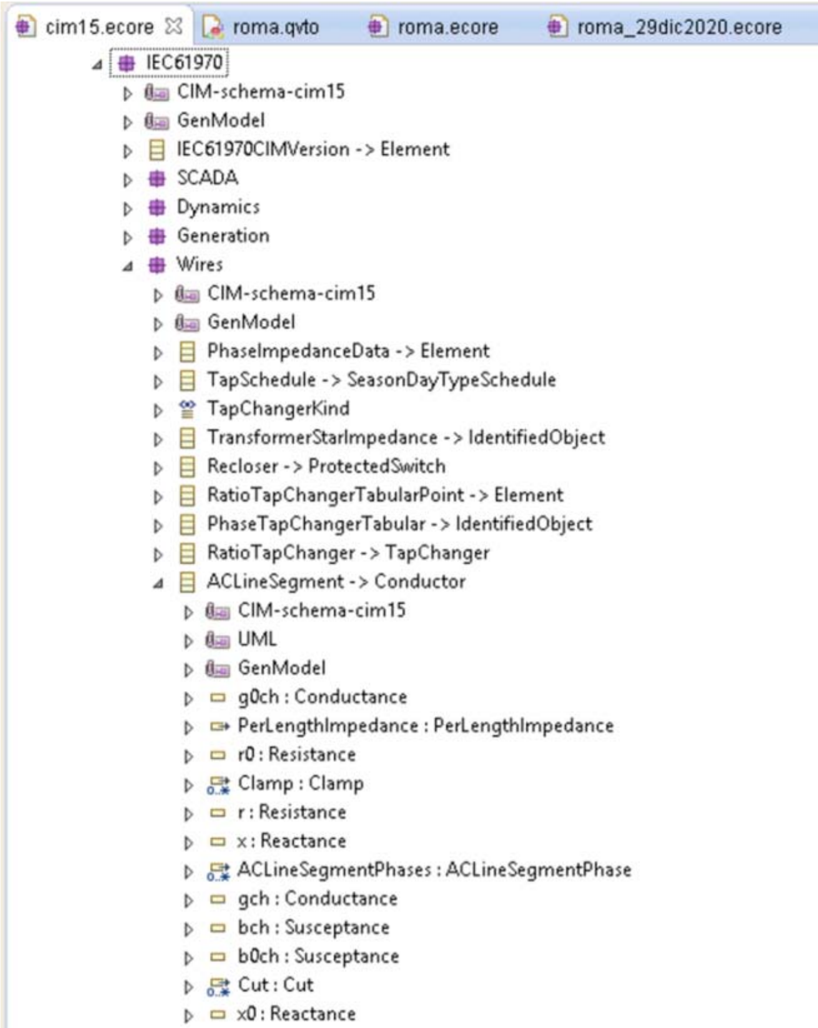


Figure 12 – IEC CIM meta-model in Ecore format

Once the Ecore meta-models have been extracted it was possible to implement the mapping between their classes.

The language used for transforming models is QVT (Query, View, Transformation), defined by the OMG [10, 16]. The name of the language reflects its three-part structure: the first part is named Query because queries can be applied to the source model, an instance of the source meta-model; the View is a description how the target model should look like; the Transformation is the part where the results of the queries are projected on the view, creating the target model [10]. QVT consists of three different domain-specific languages: Relations, Core and Operational Mappings. In the context of this activity, we have used QVT operational (QVTo) mapping language. QVTo allows to define operational transformations, meaning unidirectional transformations expressed imperatively [16].

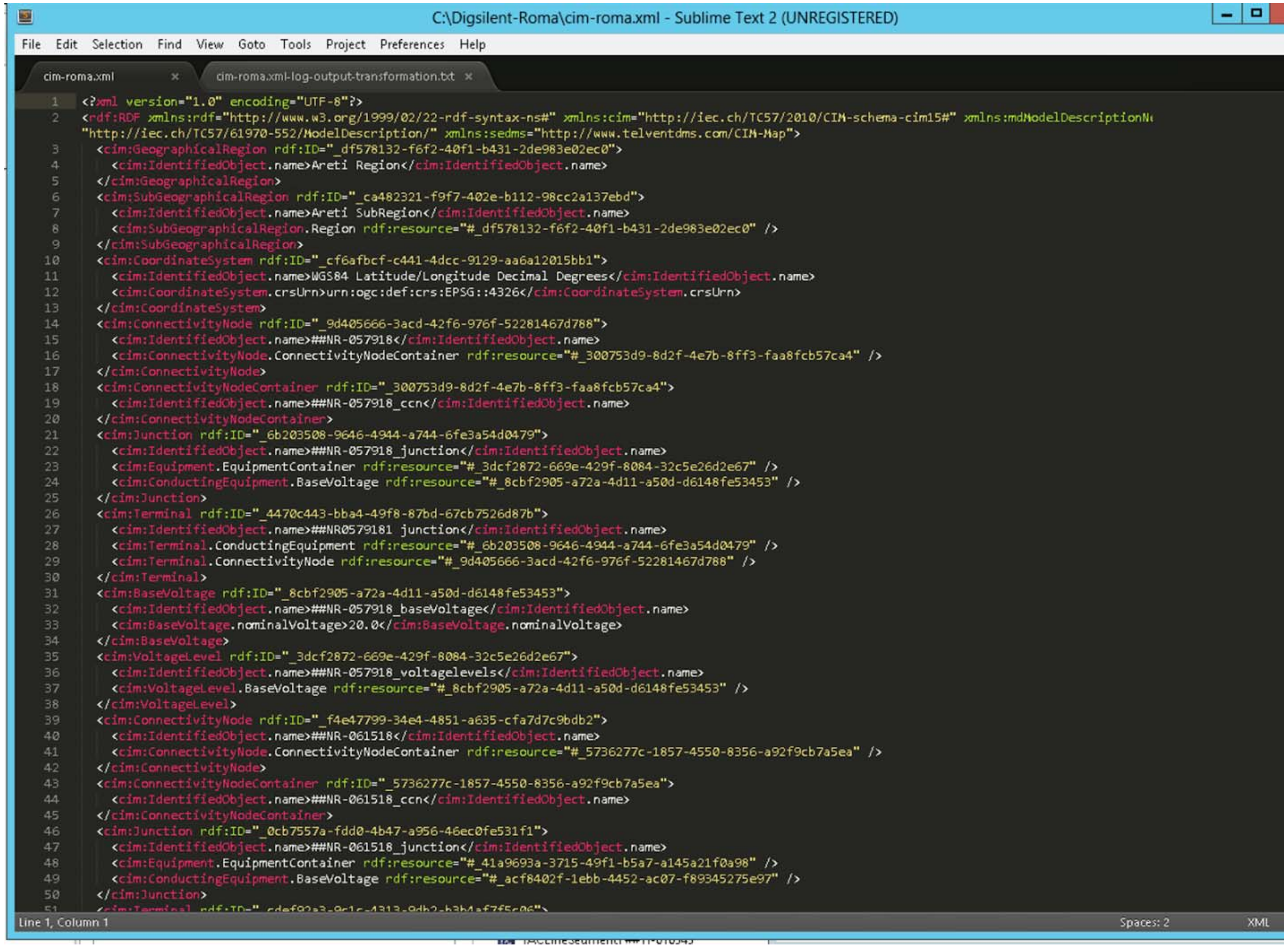
In Figure 13 we show an example of QVTo transformation, regarding the mapping of a DGS StaCubic into a CIM Terminal: the DGS field “ID” is mapped into the CIM field “name”, while the CIM connectivity node information is extracted from the DGS “cBusBar” field.

```
mapping StaCubic::toTerminal(inout cNodes : Dict(String, Core::ConnectivityNode), inout terminals : Dict(String, Core::Terminal)):Core::Terminal
when (self.ID.toString().length() <> 0 and terminals-> get(self.ID)->isEmpty() = true)
{
            log("Transform StaCubic " + self.ID);
            result.name := self.ID;

            result.ConnectivityNode := cNodes->get(self.cBusBar);


            terminals->put(self.ID , result);

}
```

Figure 13 - Example of QVTo mapping of a DGS StaCubic into a CIM Terminal

Finally, the second resulting output is the CIM RDF/XML description of the input grid (Figure 14).

In this case, in order to validate the implemented transformation a visual check of the network topology and physical parameters could be enough, as the grids are not very large and complex. However, a formal method to validate a transformation operation, applicable also with complete and larger networks, is to run a power flow from both the input and output network descriptions, in this case the DGS and CIM description of the same network. In Figure 15 an example of power flow results for the CIM transformed network is presented.




```xml
<?xml version="1.0" encoding="UTF-8"?>
<rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#" xmlns:cim="http://iec.ch/TC57/2010/CIM-schema-cim15#" xmlns:mdModelDescriptionN
"http://iec.ch/TC57/61970-552/ModelDescription/" xmlns:sedms="http://www.telventdms.com/CIM-Map">
  <cim:GeographicalRegion rdf:ID="_df578132-f6f2-40f1-b431-2de983e02ec0">
    <cim:IdentifiedObject.name>Areti Region</cim:IdentifiedObject.name>
  </cim:GeographicalRegion>
  <cim:SubGeographicalRegion rdf:ID="_ca482321-f9f7-402e-b112-98cc2a137ebd">
    <cim:IdentifiedObject.name>Areti SubRegion</cim:IdentifiedObject.name>
    <cim:SubGeographicalRegion.Region rdf:resource="#_df578132-f6f2-40f1-b431-2de983e02ec0" />
  </cim:SubGeographicalRegion>
  <cim:CoordinateSystem rdf:ID="_cf6afbcf-c441-4dcc-9129-aa6a12015bb1">
    <cim:IdentifiedObject.name>WGS84 Latitude/Longitude Decimal Degrees</cim:IdentifiedObject.name>
    <cim:CoordinateSystem.crsUrn>urn:ogc:def:crs:EPSG::4326</cim:CoordinateSystem.crsUrn>
  </cim:CoordinateSystem>
  <cim:ConnectivityNode rdf:ID="_9d405666-3acd-42f6-976f-52281467d788">
    <cim:IdentifiedObject.name>##NR-057918</cim:IdentifiedObject.name>
    <cim:ConnectivityNode.ConnectivityNodeContainer rdf:resource="#_300753d9-8d2f-4e7b-8ff3-faa8fcb57ca4" />
  </cim:ConnectivityNode>
  <cim:ConnectivityNodeContainer rdf:ID="_300753d9-8d2f-4e7b-8ff3-faa8fcb57ca4">
    <cim:IdentifiedObject.name>##NR-057918_ccn</cim:IdentifiedObject.name>
  </cim:ConnectivityNodeContainer>
  <cim:Junction rdf:ID="_6b203508-9646-4944-a744-6fe3a54d0479">
    <cim:IdentifiedObject.name>##NR-057918_junction</cim:IdentifiedObject.name>
    <cim:Equipment.EquipmentContainer rdf:resource="#_3dcf2872-669e-429f-8084-32c5e26d2e67" />
    <cim:ConductingEquipment.BaseVoltage rdf:resource="#_8cbf2905-a72a-4d11-a50d-d6148fe53453" />
  </cim:Junction>
  <cim:Terminal rdf:ID="_4470c443-bba4-49f8-87bd-67cb7526d87b">
    <cim:IdentifiedObject.name>##NR0579181 junction</cim:IdentifiedObject.name>
    <cim:Terminal.ConductingEquipment rdf:resource="#_6b203508-9646-4944-a744-6fe3a54d0479" />
    <cim:Terminal.ConnectivityNode rdf:resource="#_9d405666-3acd-42f6-976f-52281467d788" />
  </cim:Terminal>
  <cim:BaseVoltage rdf:ID="_8cbf2905-a72a-4d11-a50d-d6148fe53453">
    <cim:IdentifiedObject.name>##NR-057918_baseVoltage</cim:IdentifiedObject.name>
    <cim:BaseVoltage.nominalVoltage>20.0</cim:BaseVoltage.nominalVoltage>
  </cim:BaseVoltage>
  <cim:VoltageLevel rdf:ID="_3dcf2872-669e-429f-8084-32c5e26d2e67">
    <cim:IdentifiedObject.name>##NR-057918_voltagelevels</cim:IdentifiedObject.name>
    <cim:VoltageLevel.BaseVoltage rdf:resource="#_8cbf2905-a72a-4d11-a50d-d6148fe53453" />
  </cim:VoltageLevel>
  <cim:ConnectivityNode rdf:ID="_f4e47799-34e4-4851-a635-cfa7d7c9bdb2">
    <cim:IdentifiedObject.name>##NR-061518</cim:IdentifiedObject.name>
    <cim:ConnectivityNode.ConnectivityNodeContainer rdf:resource="#_5736277c-1857-4550-8356-a92f9cb7a5ea" />
  </cim:ConnectivityNode>
  <cim:ConnectivityNodeContainer rdf:ID="_5736277c-1857-4550-8356-a92f9cb7a5ea">
    <cim:IdentifiedObject.name>##NR-061518_ccn</cim:IdentifiedObject.name>
  </cim:ConnectivityNodeContainer>
  <cim:Junction rdf:ID="_0cb7557a-fdd0-4b47-a956-46ec0fe531f1">
    <cim:IdentifiedObject.name>##NR-061518_junction</cim:IdentifiedObject.name>
    <cim:Equipment.EquipmentContainer rdf:resource="#_41a9693a-3715-49f1-b5a7-a145a21f0a98" />
    <cim:ConductingEquipment.BaseVoltage rdf:resource="#_acf8402f-1ebb-4452-ac07-f89345275e97" />
  </cim:Junction>
```

Figure 14 - Description of the electrical grid in format CIM RDF/XML, after the QVTo transformation

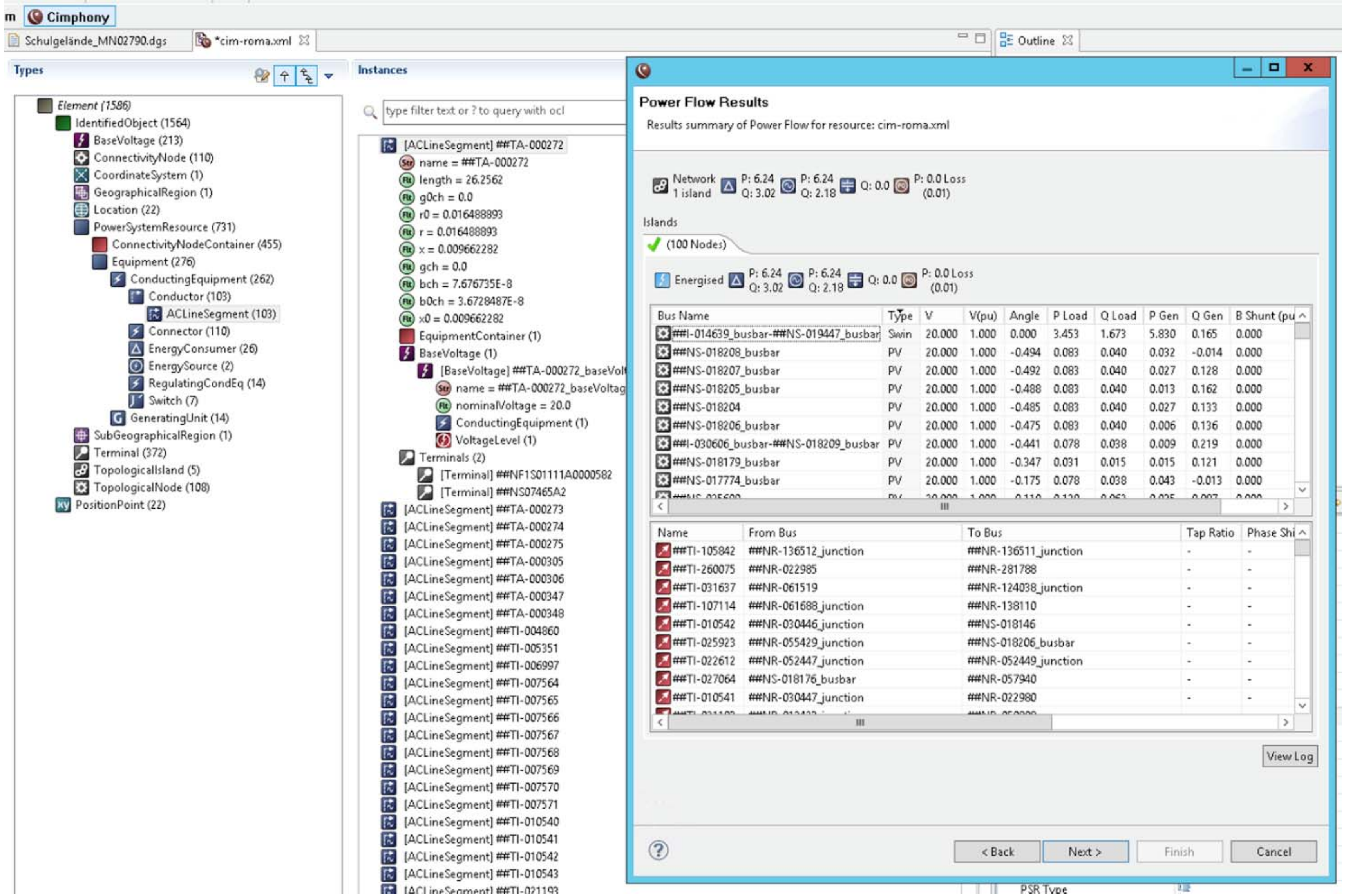


Figure 15 - Example of power flow results for the CIM transformed network

3 Demo results

Section 2 reports the Model-Driven methodology to transform grid network descriptions from a model to another model.

In the following sections the three demo grids will be presented more in detail in section 3.1 for the Italian demo, section 3.2 for the Greek demo and section 3.3 for the German demo. An overview of the demonstration sites is presented in Figure 16.

	ITALY	GREECE	GERMANY
Operator	Areti	HEDNO	Avacon
Area	**Rome** Large metropolitan area	**Mesogeia** (Attica region) Mix of rural, urban and sub-urban areas	**Twistringen** (Diepholz) Rural area (agricultural region)
Load	Households Small / medium industries Public buildings EV charging station	Households Small / medium / large industries	Households Agricultural buildings Storage heaters Batteries
Variable generation	Rooftop PVs	Rooftop and ground PVs	Rooftop PVs

Figure 16 - Overview of demonstration sites

3.1 ARETI Grid (Rome - Italy)

3.1.1 Grid description

ARETI grid is located in the large metropolitan area of Rome (Figure 17), where the characteristic loads are households, small/medium industries, public buildings and EV charging stations. The variable electricity is mainly generated by rooftop PVs.

The portion of the network provided by ARETI refers to two primary substation busbars. The first has one outgoing feeder, the second one has two feeders. The feeders have 29 secondary substations, 26 energy consumers, 103 lines and 14 generation units (divided in synchronous machines and PV units). From a topological point of view the network is represented as composed by 2 topological islands (one for each primary busbar) with 24 and 81 nodes respectively.

Figure 17 - ARETI grid location within the Italian territory

3.1.2 Transform validation

Once obtained as output the CIM RDF/XML description of the grids, the mapping has been validated comparing the power flow results from the DGS and CIM description of the same network. For the ARETI grid transformation, the maximum percentage difference between the power flow results is 0,01%, supporting the mapping correctness.

3.2 HEDNO Grid (Mesogeia - Greece)

3.2.1 Grid description

The HEDNO grid is located in the area of Mesogeia at the south-eastern part of Attica, near Athens (Figure 18). Mesogeia is an ideal demonstration area because of its location, close to the capital, but including both parts of the mainland and interconnected islands and providing a mix of rural, urban and suburban areas. The consumer mix includes households, small, medium and large industries [17]. Finally, Mesogeia has a good RES penetration of various types, mainly rooftop and ground PVs [17].

Figure 18 - HEDNO grid location within Greece

The available network for this demo describes a portion of a medium voltage grid. In particular two feeders have been provided. Both feeders have similar features i.e., one external grid source element, MV loads and MV generation distributed on the grid nodes.

The input data consist of two files, one for each feeder. Those files are in pfd proprietary format and not in dgs, so a further step is needed for the conversion into a dgs file format. The metadata structure of the feeders is the same, therefore, the same mapping pipeline has been used to transform the two networks. Regarding the components description, it is observed that in this demo RES generation units are described using the DIgSILENT element ElmLod i.e., they are represented as loads with negative power. This choice is different from the Italian demo where the PV generation is described using the DIgSILENT element ElmGenstat. As a consequence, the generation units are mapped into CIM as energy consumers and not as generation units. A customization of the transformation would be needed in order to map a load with negative power into a CIM generation unit, as it has been done for ElmGenstat elements. However, for a validation with static power flow, the two mappings are equivalent. Regarding the GPS coordinates, although the coordinate fields are present in the metadata, there is no information about the substation position as in Italian and German cases. In Figure 19 and Figure 20, a representation of the CIM mapped networks is shown.

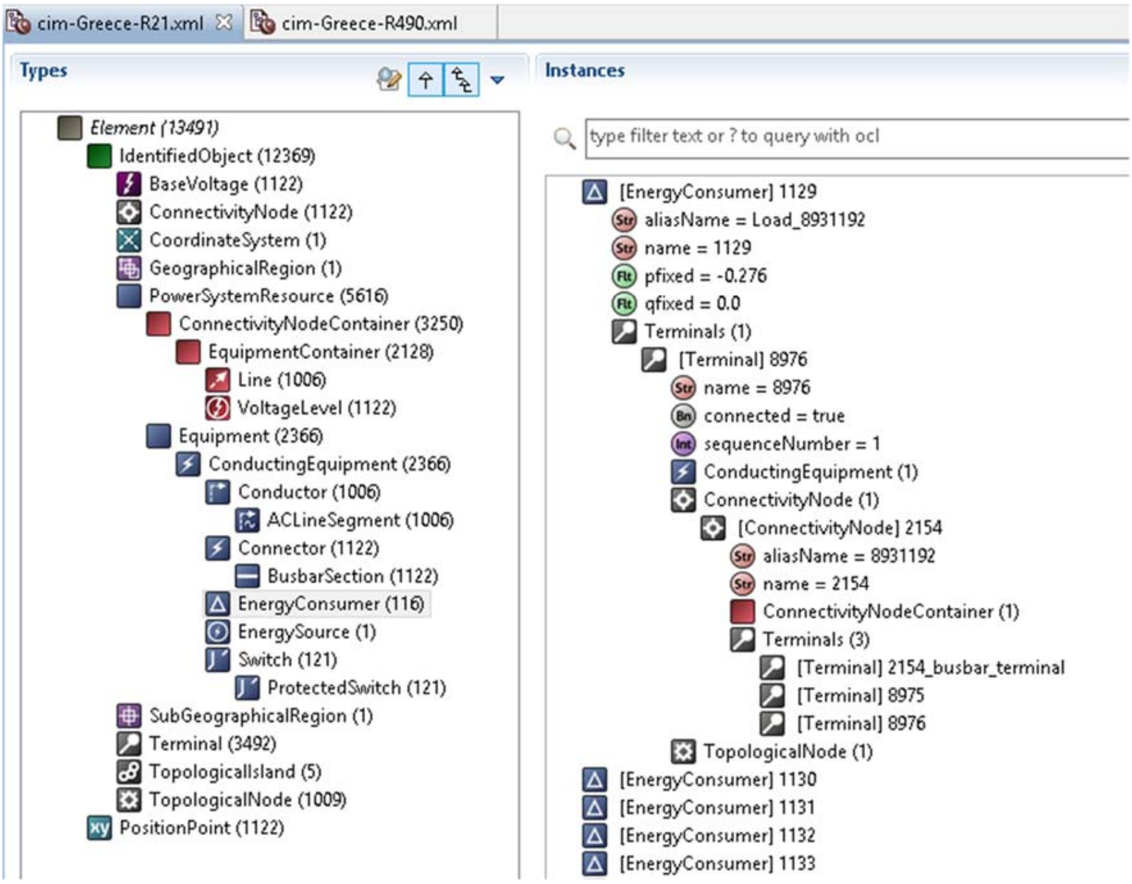


Figure 19 - CIM representation of the first feeder in the Greek demo network

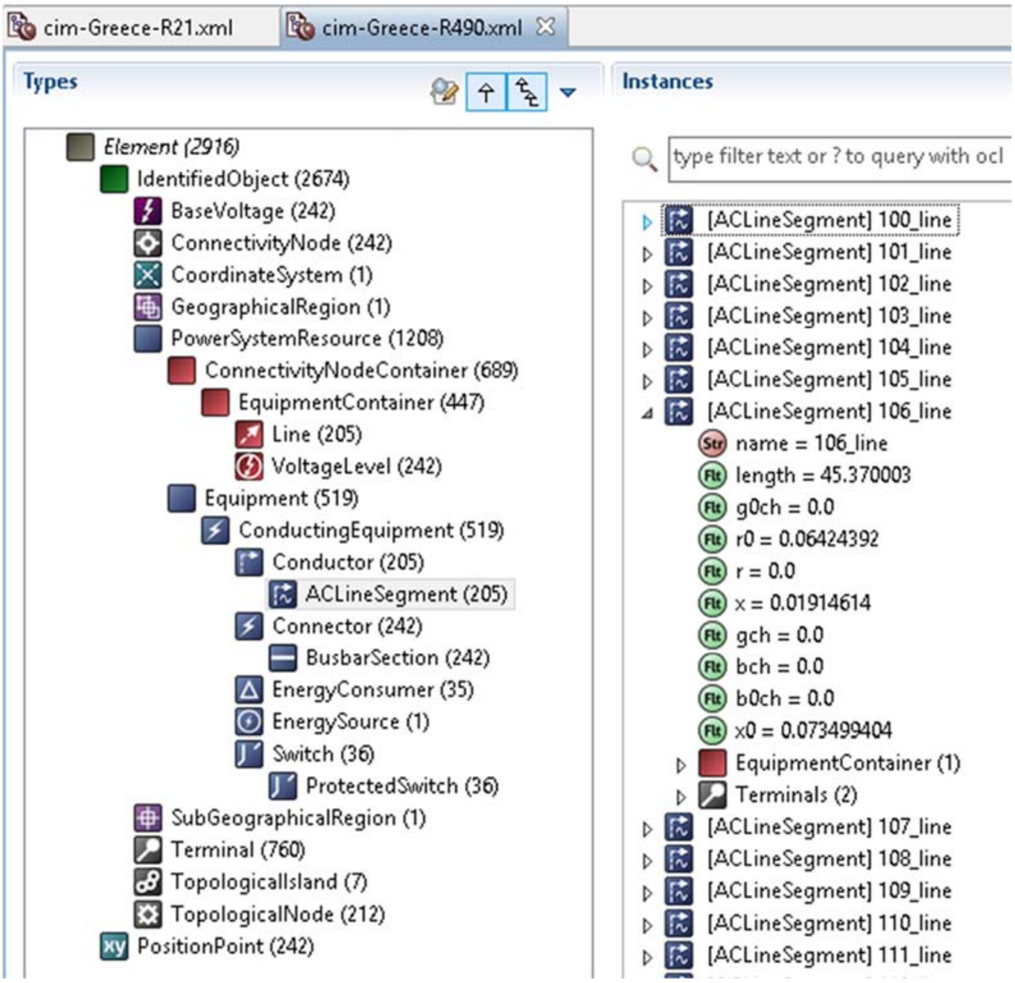


Figure 20 - CIM representation of the second feeder in the Greek demo network

3.3 AVACON (Twistringen - Germany)

3.3.1 Grid description

The Avacon grid, depicted in Figure 21, consists of a low voltage network in a rural area with a high penetration of DER, mainly rooftop PVs.

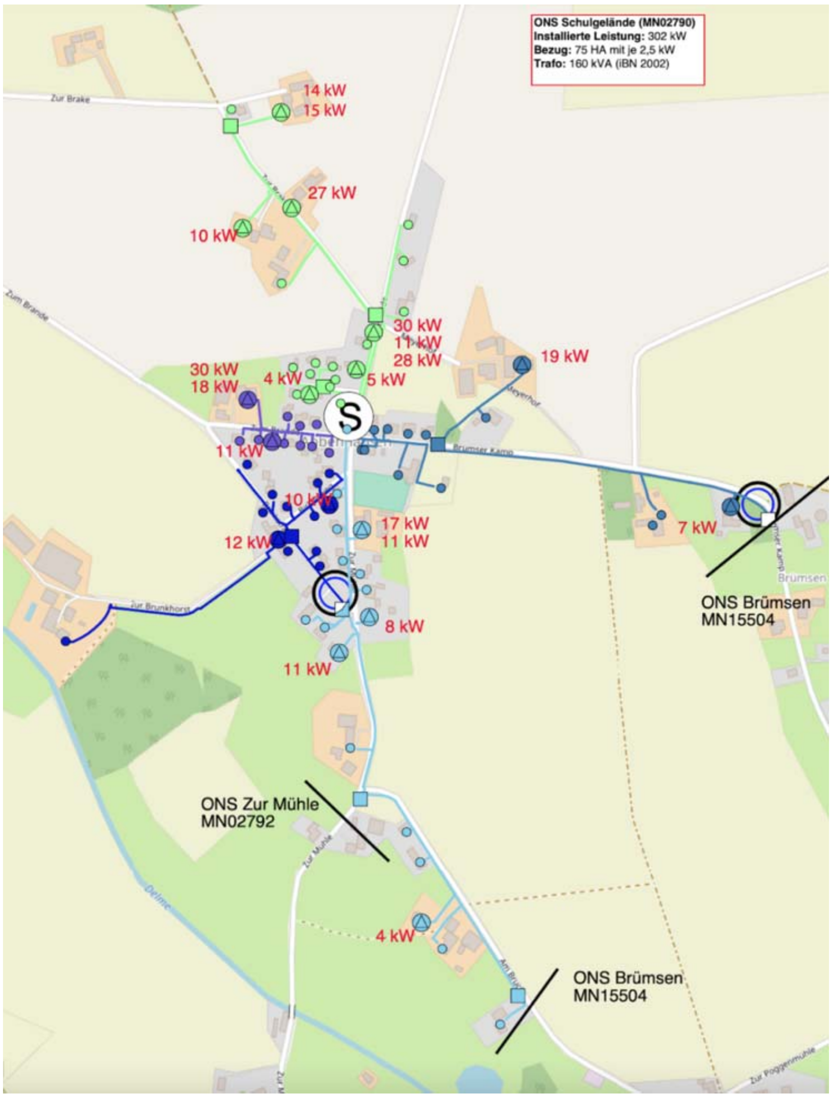


Figure 21 - Twistringen Avacon grid

The grid is located in an agricultural region, in North-West Germany (Figure 22). The typical loads are households, agricultural buildings and storage heaters [17].

Figure 22 - Avacon grid location within the German territory

The input data consist of one file containing not only DIgSILENT elements already represented in Greek and Italian networks but also new DIgSILENT elements describing new grid components. The data model and the naming convention are in good agreement with the DIgSILENT standard i.e., no customization has been introduced. However, the different structure of metadata respect to Italian and Greek demos has led us to develop a further pipeline with a slightly different mapping into IEC CIM.

Regarding the new elements found in this demo, there are:

- new kind of switches represented by the DIgSILENT element RelFuse;
- low-voltage loads expressed using the new element ElmLodlv;
- an explicit reference to the substations using the element ElmTrfstat.

The explicit description of the secondary substations allows to retrieve easily the geographical information through the GPS coordinates fields. In Figure 23, a representation of the CIM mapped network is shown.

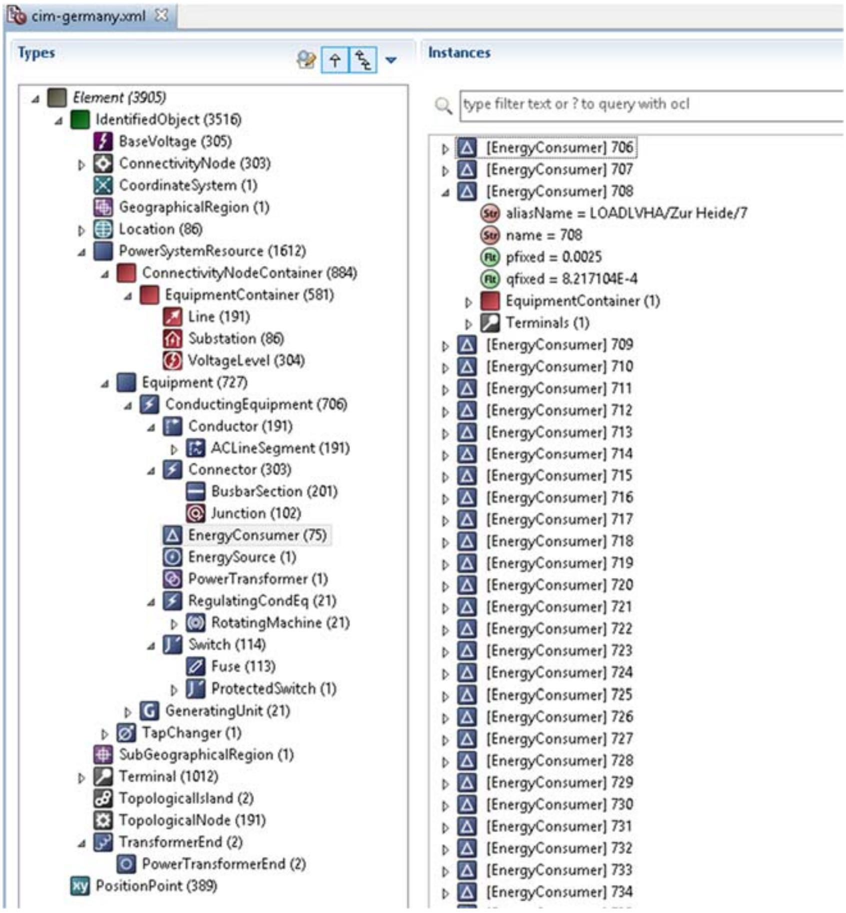


Figure 23 - CIM representation of the German demo network

4 Conclusion

In this report a methodology to map electrical grids from the DIgSILENT data format to IEC CIM 15 standard has been described, here briefly summarized. The input of the transformation is a portion of an electrical grid in DGS data format. The outputs are two artifacts: the DGS meta-model and the IEC CIM transformed grids. First, using a Model Driven Architecture approach the DGS meta-model has been retrieved. Once both DGS and IEC CIM meta-models have been extracted, it is possible to implement the mapping between their classes, using the QVTo mapping language. Finally, the CIM RDF/XML description of the input grid is obtained. To validate the implemented transformation, the power flow results from the DGS and CIM network descriptions can be compared.

This methodology has been applied to the Italian, Greek and German electrical grid demos. The transformation has been kept as general as possible taking into account the specific DGS model customizations of the different utilities.

The results will be used to show the possibility to use IEC CIM standard in combination with the Platone DSO Technical Platform (DSOTP) to test existing or future tools of the platform.

5 List of Figures

6 List of References

[1] European Commission, «2050 long-term strategy,» [Online]. Available: https://ec.europa.eu/clima/policies/strategies/2050_en.

[2] M. Uslar; M. Specht; S. Rohjans; J. Trefke; J.M. González, «The Common Information Model CIM: IEC 61968/61970 and 62325 - A practical introduction to the CIM,» *Springer Science & Business Media,* 2012.

[3] International Electrotechnical Commission, «IEC 61970 Energy Management System application program interface (EMS-aPI) - Part 301: Common Information Model (CIM) Base,» n. Edition 1.0, 2013.

[4] International Electrotechnical Commission, «IEC 61968 Application integration at electric utilities - System interfaces for distribution management - Part 11: Common Information Model (CIM)».

[5] International Electrotechnical Commission, «IEC 62325 Framework for energy market communications - Part 301: Common Information».

[6] EPRI, The Common Information Model for Distribution: An Introduction to the CIM for Integrating Distribution Applications and Systems, Palo Alto, CA, 2008.

[7] A. W. McMorran, An Introduction to IEC 61970-301 & 61968-11: The Common Information Model, Glasgow, UK: University of Strathclyde, 2007.

[8] DIgSILENT GmbH, DGS Interface, Germany, 2014.

[9] DIgSILENT, «PowerFactory - DIgSILENT,» [Online]. Available: https://www.digsilent.de/en/powerfactory.html.

[10] P.J. Barendrecht, Modeling transformations using QVT Operational Mapping, 2010: Eindhoven University of Technology, Eindhoven.

[11] ENTSOE, «CGMES Roadmap,» 05 12 2018. [Online]. Available: https://eepublicdownloads.entsoe.eu/clean-documents/CIM_documents/CGMES_Roadmap_2018-SOC.pdf.

[12] Object Management Group, «MDA Guide revision 2.0,» 14 June 2014. [Online]. Available: https://www.omg.org/cgi-bin/doc?ormsc/14-06-01.

[13] A.R. da Silva, «Model-driven engineering: A survey supported by the unified conceptual model,» *Computer Languages, Systems & Structures,* vol. 43, pp. 139-155, 2015.

[14] The Eclipse Foundation, «Eclipse Modeling Project,» [Online]. Available: https://www.eclipse.org/modeling/emf/.

[15] J. Lin; D. Zhang; J. Huang; Y. Huang; D. Liu, «Research and Development of Grid Model Online in E Imported to DIgSILENT,» *International Conference on Advances in Mechanical Engineering and Industrial Informatics,* 2015.

[16] Object Management Group, «Meta Object Facility (MOF) 2.0, Query/View/Transformation Specification, version 1.3,» 2016. [Online]. Available: https://www.omg.org/spec/QVT/1.3/PDF.

[17] Platone EU Horizon 2020 Project, «D5.2 Detailed Use Case Descriptions,» 2020.

[18] FEIN Aachen, «IEC61970 16v29a - IEC61968 12v08 Documentation,» [Online]. Available: https://cim.fein-aachen.org/libcimpp/doc/IEC61970_16v29a_IEC61968_12v08/index.html.

7 List of Abbreviations

Abbreviation	Term
CIM	Common Information Model
DEMI	Distributed Energy Management Initiative
DGS	DIgSILENT
DSO	Distribution System Operator
EMS	Eclipse Modeling Framework
EPRI	Electric Power Industry
GIS	Geographic Information System
IEC	International Electrotechnical Commission
MDA	Model Driven Architecture
OMG	Object Management Group
OWL	Web Ontology Language
Platone	PLATform for Operation of distribution NEtworks
QVT	Query, View, Transformation
QVTo	Query, View, Transformation operational
RDF	Resource Description Framework
SCADA	Supervisory Control And Data Acquisition
TSO	Transmission System Operator
XML	eXtensible Markup Language