Automated Model Transformation for Cyber-Physical Power System Models

Gustav Holm

***Abstract*—Standardized information and mathematical models, which model the characteristics of the power generation and power transmission systems, are requirements for future development and maintenance of different applications to operate the electrical grid. Available databases such as Nordpool provides large amounts of data for power supply and demand [1]. The typical misconception with open availability of data is that existing power system software tools can interact and process this data. Difficulties occur mainly because of two reasons. The first on is the amount of data produced. When the topology of the electrical grid changes e.g. when a switch opens or closes, the flow of electrical power changes. This event produce changes in generation, transmission and distribution of the energy and different data sets are produced. The second problem is the representation of information [2]. There are a limited number of software tools that can analyze this data, but each software tool requires a specific data format structure to run. Dealing with these difficulties requires an effective way to transform the provided data representation into new data structures that can be used in different execution platforms. This project aims to create a generic Model-to-Text (M2T) transformation capable of transforming standardized power system information models into input files executable by the Power System Analysis Tool (PSAT). During this project, a working M2T transformation was never achieved. However, missing functionality in some programs connected to sub processes resulted in unexpected problems. This led to a new task of updating the information model interpreter PyCIM. This task is partially completed and can load basic power system information models.**

I. Background

IN December 2013 the European Network of Transmission System Operators for Electricity (ENTSO-E), published the Common Grid Model Exchange Standard (CGMES) [3]. CGMES is a subset of the International Electrotechnical Commission (IEC) Common Information Model (CIM) standard [4]. It specifies how equipment properties, steady-state and dynamic information of a power transmission system should be implemented in available software tools, for the efficient exchange of information. The CIM/CGMES semantic model is an open standard that utilize Object Oriented Programming principles which organize all the available data into attributes, objects and relations among objects [5]. The adaptation of the CIM/CGMES resulted in the representation of information models with a specific structure compatible with existing power system analysis tools and is easy to map to other formats.

A. *Information models*

Information contained in the CIM/CGMES is sufficient to describe all parts of the power system, but the information need to be transformed to new representations for use in different power system analysis methods. Model Driven Software engineering (MDSE) techniques, such as Model-to-Text (M2T) transformations and Model-to-Model (M2M) transformations are currently the bridge that connects model information to mathematical models for analysis and simulation within the available software tools. These two techniques have the same general intention of transforming the model representation, and produce data structures for different applications. M2M transformations and M2T transformations both use models as inputs, but produce different types of outputs [6]. M2M transformations produce new models as output and M2T transformations produce source code or text files [6]. Together these two techniques provide the necessary tools to transform information models into data structures required by execution platforms [7].

B. *Goal*

This project was focused on M2T transformations, with the objective to create a generic program in the python programming language. The idea was that the produced program should be able to transform power system information models to text files executable by the Power System Analysis Toolbox (PSAT) plugin tool to Matlab [8]. Since the information models are standardized, this should in theory eliminate the time consuming process of manually mapping data from power system information models to new input files.

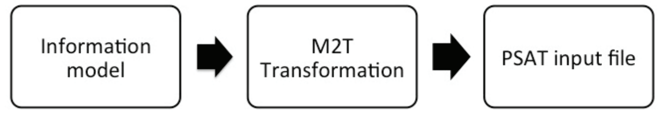

Fig. 1. Showing the M2T transformations targeted in this project.

C. *Outcome*

Changes in electrical grid topology produce large quantities of data that needs to be processed. The properties of the power transmission system determine how the electrical power varies throughout the system. This in combination with other factors, such as constant changes in electrical power production and consumption results in a complex power transmission system. Software tools for power system analysis exist, but in order to take full advantage of these tools and predict future power grid behavior, effective methods that automatically transform generated data to input files are required. The expected outcome of this project would provide the opportunity to execute power system analysis and simulation more frequent with fresh data, size independent of the targeted power system. This would improve the development and maintenance conditions of power transmission system applications by making analysis and simulation more accessible.

II. Theory

This project connects information and knowledge between several different fields. In this section, basic terminology and concepts will be explained to provide a deeper understanding of this project and the method used.

A. Government agencies and standardization

Government agencies such as Svenska kraftnät control the development and maintenance of national power transmission systems [9]. To secure societies' needs and supply of electricity, many countries have joined forces. The European organization in charge, ENTSE-E actively works to create transparency and provide means to improve collaboration between involved nations [10]. Mainly, this is achieved through implementation of standards. The standards relevant to this project are called CIM/CGMES, and specify management of power transmission data [3].

B. MDSE

Model driven software engineering was a central part of this project. Research in this area revolved around different types of models, the various ways to create, transform and use the information represented within. When considering an information model, the typical person visualizes a diagram with arrows connection different processes. This is a valid and useful way to represent information [11], but it does not utilize the full strength of MDSE techniques. A sufficient information model is a subset of data classes providing a complete description of a system e.g. concepts, relationships constraints and rules [12]. This enables the user to transform the model representation, but also allows for data extraction and modification of specific parts in the information model without the risk of affecting other parts of the power system model.

C. Model transformations

M2M and M2T transformations are two adapted concepts in MDSE. The main purpose of these transformations is to automate construction and modification of information models where possible. Model transformations are programs created to process one or more information models and produce new information models or representations suitable for the targeted purpose.

D. Power system analysis

To understand the data and the structure of the information models, an increased understanding of power system analysis was required. The focus was directed towards terminology, how to measure and calculate data from the electrical grid. Analyzing power grid behavior requires methods for determining the electrical parameters e.g. current, voltage, phase, active- and reactive power. The use of electrical buses [13] together with the characteristics of the power transmission system provides a way to determine how electrical parameters vary throughout the power grid. There are three types of buses implemented in a power transmission system [14]. Together they allow operators to determine the required parameters.

	P = Active power	Q = Reactive power	V = Voltage	δ = Phase angle
P-Q Bus	measured	measured	calculated	calculated
P-V Bus	measured	calculated	measured	calculated
Slack bus	calculated	calculated	measured	measured

Fig. 2. The three different types of buses implemented in a power transmission system and their functionality.

Measuring parameters from the electrical grid requires a connection between the three high current conductors [15]. This can be done at any location throughout the power transmission system, but the most effective way is to use the existing buses. Nearly all buses are uninsulated to provide cooling for the power grid conductors. The buses are used to connect conductors to electrical equipment or a different conductor, and are therefor strategically placed throughout the system. For power system analysis this has resulted in that, the number of buses is used to describe the size of the targeted power transmission system e.g. the Single machine infinite bus (SMIB), IEEE9-bus and Nordic-44bus, which were examined in this project.

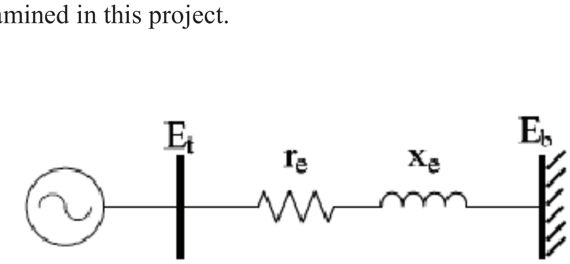

Fig. 3. Showing the Single Machine Infinite Bus (SMIB), which is a small power transmission system.

E. Programming techniques

One of the main usages of programming is data and data structure processing. The python programming language provide sufficient data handling capabilities [16], and for this reason python is one of the most used programming languages in the world [17]. But programming languages can be challenging, often requiring updates and additional plug in tools to function properly. This is what makes general purpose programming languages useful [6], but it also means that more programming knowledge is required when additional functionality is needed.

F. XML data structure

The information represented within the information models is organized after the XML data structure, according to the CIM and CGMES [3]. What this means, is that the attributes are directly linked to their corresponding value. This provides a data structure that allows a user to interpret the information, but also a format that a computer or programming language can process. The XML data representation also provides classification of different types of data, and this is what makes it suitable in power transmission system applications where different types of data occur.

```
<cim: RotatingMachineDynamics.damping>0.6700000166893005<cim:
RotatingMachineDynamics.damping>
<cim:RotatingMachineDynamics.inertia>3.3299999237060547<cim:
RotatingMachineDynamics.inertia>
```

Fig. 4. Showing example of code structure utilized in the XML data format. In this case, a damping value of 0.67 and an inertia value of 3.33.

G. PyCIM interpreter

PyCIM is an open source python program for data interpretation designed to read the XML data structure [18], and organize attributes into objects containing data from several sources e.g. the voltage measured from all buses throughout a power transmission system. Creating a generic program capable of transforming information models requires looping between different types of data originating from different sources. The organization functionality within PyCIM provides the necessary tool that allows access to specific values, and therefore makes M2T transformations possible.

III. METHOD

Before the work on creating the M2T transformations could begin, there were three key elements that had to be identified. Understand the given problem, why it occurs and how to actually solve it. For these reasons, this project was divided into three phases.

A. Research phase

The first and most extensive phase of this project was the research phase. Here, the theory and connection between the relevant fields were studied. To provide an adequate knowledge, the four major areas were first studied on an individual level. When this was completed it was time to investigate how they interact with each other and applies to the CIM/CGMES standardization and model transformations. Acquiring this information consisted mostly of reading. Several books, websites, program manuals and documentations were reviewed to create wide perspective on the targeted goal.

B. Implementation phase

When a sufficient knowledge was acquired from the research phase, the implementation phase could begin and the task to discretize how M2T transformations are created. The Python programming language, and the PyCIM information model interpreter where chosen as the preferred tools for creating the M2T transformations. The PyCIM documentation and the PSAT manual were reviewed to determine the specifications and organization of the targeted input files.

C. Coding phase

The coding consisted in two parts. First, the raw data from energy production and consumption together with power transmission system characteristics had to be combined into the standardized information model format. This should be done with existing Python programs. When the information models were acquired, the PyCIM information model interpreter processed the information models and created a data structure required to create M2T transformations. The necessary information was then to be identified from the data structures and printed to a specific text file structure, representing the PSAT input files.

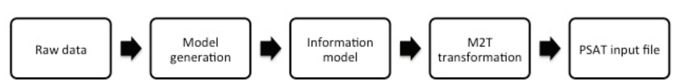

Fig. 5. The entire process going from raw data to input files.

In this phase, difficulties began to arise. Early, it was discovered that generating the power system information models was problematic. Software tools created to generate information models from raw data required commercial licenses that were quite expensive. The first step of producing the information models was not a part of the main goal to create M2T transformations. This resulted in the IEEE9-bus being chosen as the targeted power transmission system because a complete information model was already available. The IEEE9-bus is a bigger power transmission system than the Single machine infinite bus (SMIB), shown in figure 3, requiring more code, but it has a key feature necessary to create a generic M2T transformation. For the M2T transformation to be applicable to a power transmission system of varying size, the changes in the information model representation when dealing with an increased number of buses has to be considered. When the information model was acquired, it was time to load the data into the python programming language. This was quickly done with the PyCIM interpreter, only requiring a few lines of code. However, for the time unknown reasons, all the data contained in the information model would not upload into Python. Solving this problem became a big issue and required careful review of the PyCIM documentation. It was discovered that the current version 15 of PyCIM did not support the updated information models. This resulted in a new task to update the PyCIM library and add the missing dynamic functionality contained in the updated information models.

D. Updating PyCIM

The new task to update the PyCIM library consisted in several steps. First, a new module for the version 16 information model library has to be created. Since the basic information model structure was the same for version 15 and version 16, a copy of the version 15 library called CIM15 was made and named CIM16. CIM16 consists of approximately 800 files, each representing a Python program containing one class designed to interpret a specific object or connection in the power system information model. The second step was to open all the files contained in the CIM16 one by one and update the version from 15 to 16 and the online referencing to the current CIM/CGMES standard.

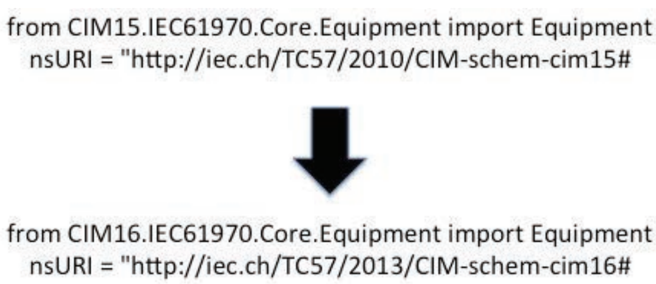

Fig. 6. Showing the two lines of code that has to be exchanged in every program. On top the CIM15 lines and underneath the substituted CIM16 lines.

The third and final step was to create the missing classes responsible for interpretation of the new functionality contained in the updated information models. This functionality was related to dynamic behavior of power transmission systems. A dynamic folder was created to hold the new programs and keep the existing structure intact. Inside this folder, functionality for Synchronous machine dynamics was created. Synchronous machine dynamics consisted of two classes, RotatingMachineDynamics and SynchronousMachineTimeConstantReactance. A copy of an existing class in the model interpreter was made and used as a baseline in the creation of the new classes. The class name, corresponding attributes and references were then exchanged to match the new functionality according to the CIM model report [4].

IV. Result

A working M2T transformation program was not achieved in this project. Missing dynamic functionality in the PyCIM information model interpreter resulted in M2T transformations being difficult to achieve and created a new task to update the PyCIM interpreter library, to accept the latest information model standard. A partially completed update of the PyCIM information model interpreter has been created with the implementation of the new classes from version 16 of the CIM and CGMES. This update is still in progress and the CIM16 library will be added to the open source network GitHub for future work. The update is sufficient to load the three targeted power transmission systems, but further development is needed in order to process additional systems. Hopefully, this report also provides a simplified description of model transformation processes.

V. Discussion

This project has been an educational process. Research in power system analysis and MDSE has resulted in deeper understanding of the challenges electrical engineers face. However, the scope of this project for someone with basic prerequisites is extensive. The unexpected problems that occurred resulted in additional troubleshooting and research. This lead to additional time being spent on solving problems unrelated to M2T transformations, and the consequences were time constraints in the coding phase.

A. PyCIM update

PyCIM is designed to automate data handling and provide a new data structure from the generated power system information models. Because of this, the PyCIM update can be considered a model transformation and is therefor relevant to the context of this project. The completion of this update will not produce the desired M2T transformation, but it is a crucial step in the process that connects data to execution platforms and will simplify future work.

B. Impact

The concept of M2T transformations addresses one part in the process of simulating power transmission system behavior. Every step in this process that can be automated would reduce the overall time needed to map raw data to input files and therefore minimize errors and increase analysis and simulation capabilities. The steady data flow due to varying conditions in the power transmission system requires a fast process that can be redone in the event of electrical grid alterations. Automated process provides this feature and in combination with the current standardization, these tools result in transparency and the means to further develop power transmission systems applications.

C. Documentation

Absence of, or difficult documentation in some areas connected to model transformations produced extra work to sort out how M2T transformations were created. Of course this is expected during a development process, and MDSE is a relatively new approach to software development under improvement [6]. Most of the available documentation in MDSE focuses on implementation advantages and theoretical applications, making the research a demanding process. To further develop the use and understanding of MDSE, additional documentation describing how to create the information models and the process of producing model transformations are required. This would simplify the learning process, and make MDSE techniques more available.

D. Adapting to MDSE techniques

Application of MDSE can be used as an alternative to traditional programming, with pros and cons [6]. One of the big advantages of MDSE is the simplicity and the ability to reuse models, providing a quick and effective way to produce programs [19]. Adapting MDSE techniques requires time and the development relies heavily on individual contribution towards applications [20]. The general disadvantage is that the current stage calls for complementary coding [6] resulting in additional work.

E. Benefits of standardization

The adaptation and implementation of information modeling standards, CIM and CGMES in this case, is beneficial for many reasons. They allow for easier information processing, which is the key to process automation, represented by the model transformations in this project. Creating new software tools for power system analysis that can read the standardized information models should be investigated. Alternatively, modifying the CGMES standard or the existing software tools for analysis to process one another would produce the same result. This would require lots of work, but if this were achieved, the entire process connected to this project would be redundant, and further increase the capabilities of software and power transmission developers.

F. Version interference

CIM, CGMES, software tools used to process model information and execution platforms for simulation and analysis are under constant improvement and modification [21]. This can create interference when developers release updated versions of software e.g. adding or removing functionality. This is a common problem, experienced in this project. Working with raw data and data structures, requires different tools and programs to interpret information. Different tools connect to different stages of the development process and if one of these tools receives and update, there is a risk of halting the entire process, creating new problems that can be hard to identify and produce additional workload. When creating software and programs, there is a significant risk of ending up in a position solving unrelated tasks, instead of the main objective.

VI. Future Work

To continue the development of M2T transformations and progress further with this project, there are some factors that need to be considered. First of all the PyCIM interpreter needs to be completed and regularly updated to the current versions of available standards. The regular updates also apply to all involved software tools and programs, since the results and method demonstrated interference in several stages of this project. An increased knowledge in MDSE and programming would be beneficial, to reduce the time spent on research and understanding. The project description and work plan focused on the specific stage to create model transformations. Therefore this project relied on the functionality of previously completed stages. To avoid problems and difficulties associated with this project, it might be beneficial to review all stages from the beginning, before attempting to execute model transformations.

VII. Summary

The implementation of new hardware and software technologies in the operation of the electrical grid have increased the information available for different types of power system analysis, such as monitoring the network or development of models for time-domain simulations. There exist a limited number of software tools to analyze and simulate the power system. However, these software tools process the available information with non-standard data formats. The goal to create a generic M2T transformation was not achieved, but this project managed to produce different results. Flaws in the PyCIM interpreter library were discovered and an update has been partially completed. This updated library will be uploaded to the open source network GitHub for future work.

VIII. Acknowledgements

Finally, I would like to extend my gratitude to my supervisor Francisco José Gomez Lopez, and his team leader, Prof. Luigi Vanfretti. Their research on KTH SmarTS-LAB focuses on applicable solutions enabling cyber-physical power systems [22], to further improve the capacity of modern power transmission systems. They provided most of the material needed to conduct this project and offered support throughout the entire process. Without their help in the discretization and execution of model transformation processes this project would not have been feasible.

IX. References

[1] (2017, May). We are expanding. Nordpool [Online]. Available: http://www.nordpoolspot.com.
[2] (2017, April). What is representation of information? DCC [Online]. Available: http://www.dcc.ac.uk/node/9558.
[3] (2017, April). Standards. ENTSOE [Online]. Available: https://www.entsoe.eu/major-projects/common-information-model-cim/cim-for-grid-models-exchange/standards/Pages/default.aspx.
[4] (2017, May). Core IEC standards. IEC [Online]. Available: http://www.iec.ch/smartgrid/standards/.
[5] (2017, April). Common information model (CIM) for grid model exchange. ENTSOE [Online]. Available: https://www.entsoe.eu/major-projects/common-information-model-cim/cim-for-grid-models-exchange/Pages/default.aspx.
[6] M. Brambilla, J. Cabot, and M. Wimmer, Model-Driven Software Engineering in Practice, UK, London: Morgan & Claypool, 2012, pp. 1-182.
[7] (2017, April). What goes into functional specification. Bridging the gap [Online]. Available: http://www.bridging-the-gap.com/functional-specification/.
[8] F. Milano, Documentation for PSAT version 2.1.6, Canada, Waterloo: University of Waterloo, 2010, pp. 1-460.
[9] (2017, April). Drift av stamnätet. Svenska Kraftnät [Online]. Available: http://www.svk.se/drift-av-stamnatet/.
[10] (2017, April). Electricity market trancparency. ENTSOE [Online]. Available: https://www.entsoe.eu/data/entso-e-transparency-platform/Pages/default.aspx.
[11] (2017, April). 14 ways to present information visually. Velocitypartners [Online]. Available: https://velocitypartners.com/blog/infographics-14-ways-to-present-information-visually/.
[12] (2017, April). On the difference between information models and data models. IETF [Online]. Available: https://tools.ietf.org/html/rfc3444.
[13] (2017, April). Electrical bus-bar and its types. Circuitglobe [Online]. Available: http://circuitglobe.com/electrical-bus-bar-and-its-types.html.
[14] (2017, April). Slack bus. Wikipedia [Online]. Available: https://en.wikipedia.org/wiki/Slack_bus.
[15] F. Milano, Power system modelling and scripting, Germany, Berlin: Springer, 2010, pp. 1-556.
[16] H. P. Langtangen, Python scripting for computational science, Germany, Berlin: Springer, 2008, pp. 1-738.
[17] (2017, April). Index for April 2017. Tiobe [Online]. Available: https://www.tiobe.com/tiobe-index/.
[18] R. Lincoln. (2017, April). PyCIM 15.13.4. Pypi [Online]. Available: https://pypi.python.org/pypi/PyCIM/15.13.4.
[19] (2017, April). Benefits of modeling. Modeling languages [Online]. Available: http://modeling-languages.com/benefits-modeling-or-how-convince-your-project-manager/.
[20] (2017, April). Motivations to contribute to open-source software. Open social [Online]. Available: https://www.getopensocial.com/blog/motivation-contribute-open-source.
[21] (2017, April). Software evolution. Wikipedia [Online]. Available: https://en.wikipedia.org/wiki/Software_evolution.
[22] (2017, April). Smart transmission system - laboratory (SmarTS-Lab). KTH [Online]. Available: https://www.kth.se/en/ees/omskolan/organisation/avdelningar/epe/research/smart-transmission-systems-laboratory-smarts-lab-1.627203.