MODEL-DRIVEN SOFTWARE DESIGN FOR SMART GRID DATA ANALYTICS

Hamed VALIZADEH HAGHI
K. N. Toosi University – Iran
valizadeh@ieee.org

Masoud ALIAKBAR GOLKAR
K. N. Toosi University – Iran
golkar@kntu.ac.ir

Homayoun HAERI
Tavanir Company– Iran
haeri@tavanir.org.ir

ABSTRACT

Practical data analytics for the smart grid requires a software platform which enables the distribution companies to better correlate the projects, improve understating of system requirements and simplify system design by decomposing its complexity and large-scale data. This paper proposes the model-driven software design (MDSD) to managing and optimizing the smart grid data. It defines a language for visualizing, specifying, analyzing, and documenting the distributed object-oriented data. An MDSD experience is reported that employs three software environments interfaced with each other in order to create a chain of the desired methods for an energy efficiency program. The aim is to transform the data into actionable decisions accessible and understandable by distribution companies by taking software development into a higher level of abstraction. As a pilot project, a comprehensive data logging were performed for an MV/LV distribution system. Subsequently, the technical losses were studied using the developed software package. A Java-based GUI, MATLAB/Simulink, and an embedded C-based data management and calculation module are used in the proposed MDSD.

INTRODUCTION

Distribution companies seeking to attain the maximum benefits from advanced metering infrastructure (AMI) need a data analytics strategy. Planners can determine the analytical processes needed to turn unprocessed data into actionable information for improved decision-making and guide more efficient maintenance and operations. Collecting accurate, timely and relevant data is the core of any data analytics program, but the data needs to be put into an appropriate context to become useful information. Five fundamental analytical data transformations have immediate relevance to smart grid [1]:

1) Aggregation is a summary of data combined from several measurements.
2) Correlations identify statistical relationships between related data that are useful for building models.
3) Trending is the practice of collecting information and attempting to spot a pattern.
4) Exceptions are unexpected or abnormal conditions that is useful in data cleaning.
5) Forecasts are predictions of future events or values using historical data. Forecasts can also be built using correlation data.

Because of its multivariate nature, measuring the effectiveness of energy efficiency and smart grid programs is a complex undertaking (e.g. see [2], [3]). The smart meter data supply the computing power that enables the placement of sensors and process of cleaning, transforming, and modeling their data with the goal of highlighting useful information, suggesting conclusions, and supporting decision making in tools designed for energy efficiency calculations and spatial analysis. This is a part of comprehensive distribution system studies. Such studies may possibly cause high computational burden especially when considering enormous amounts of data from the AMI and other data logger meters online. Hence, practical system-wide simulation and calculation software need to be modified to take advantage of different types and sources of data recordings by combining abovementioned analytical data transformations while maintaining computational efficiency. Due to advances in hardware technology, however, smart grid often have more memory and computational power. Meanwhile, new software development projects should adopt three characteristics of this emerging scheme:

1) Higher levels of abstraction are needed as the demands of designing more complex and data-intensive smart grid applications requires the development of models that let software designers focus on higher-level design issues. This would manage the system complexity by managing levels of abstraction and levels of detail. Abstraction is recognized as a key concept in software development by which data and programs are defined with a representation similar in form to its meaning while hiding away the implementation details.
2) System managers should be able to test multiple views and concerns. Domain experts cannot understand all the technology stuff involved in software development. This problem is managed by selecting specific tool solutions to meet the unique needs of each domain expert. Also, the recorded data and real system response need to be properly visualized and represented.
3) Models should be designed not only to make the system comprehensible but also to integrate its functions and make the computations efficient and scalable within each function or domain.

Hence, data analytics and software for sensor-based power grid is desirable to be more domain-related as opposed to computing related. It is also about making software development in a certain domain more efficient. This is referred to as model-driven software design (MDSD) which is currently a highly regarded development paradigm in software engineering. Models are ideal means for abstraction and can enable developers to master the increasing complexity of software systems. MDSD is an approach where models are the central artefacts in software development and drive the code generation. This forms a component-based assembly-centric paradigm which is different from a third-generation language programmingcentric paradigm. Fig. 1 shows the components of the MDSD concept. According to this figure, the developer develops models based on certain metamodels. Then, using code generation templates, the model is transformed to executable code. Optionally, the generated code is merged with manually written code. One or more model-to-model transformation steps may precede code generation [4].

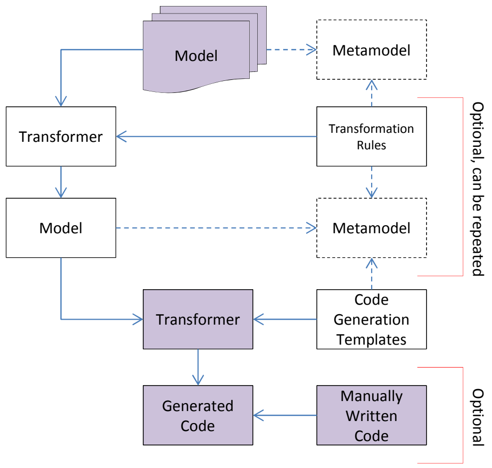

Fig. 1. A general concept of model-driven design methodology.

This paper introduces and outlines the idea of using MDSD as a key to smart grid data analytics. Furthermore, adopting and applying a MDSD methodology for studying effectiveness of an energy efficiency program within a smart grid is one of the original contributions of this proposal. Hence, a multivariate data modeling module and several metamodels that goes in parallel with network simulation tools is proposed in this paper.

MODEL-DRIVEN SOFTWARE DESIGN

Outline of the reported project

The domain-specific components as well as analytical data transformers are considered as effective parts of the MDSD. Fig. 2 illustrates the proposed methodology flow. The data modeling module is capable of representing different data trends and behaviours with good flexibility in software applications. In addition, each of the identified components provides some set of functionalities. The main purpose of the models is to give a straight definition for further components deployment. Moreover, our goal is not only to provide a description of interfaces but also to propose the most optimal way that those functionalities could be achieved. In order to implement the idea, a comprehensive data logging were performed for an MV/LV distribution system. There are 33 data loggers installed which recorded over 40 million samples of 32 quantities spanning a whole year. Fig. 3 shows two installed data logger meters. Subsequently, the technical losses were studied using the developed software package. A Java-based GUI (Fig. 4), MATLAB/Simulink (Fig. 5), and an embedded C-based data management and calculation module are used in the proposed MDSD.

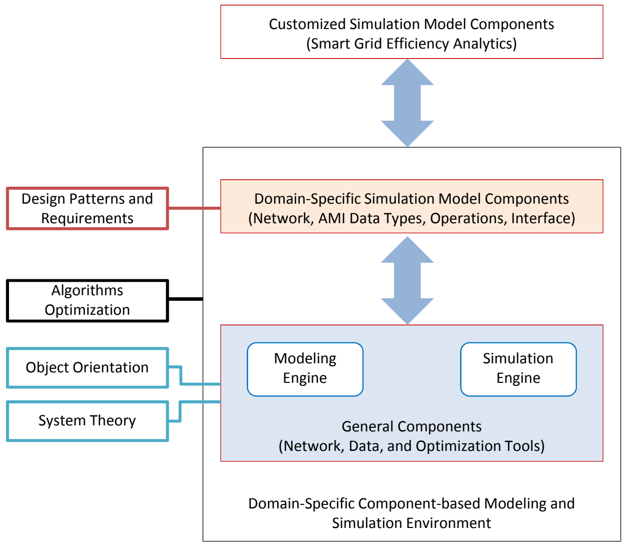

Fig. 2. Model-driven software design for a smart grid data analytics based on an implemented efficiency monitoring platform.

Fig. 3. Two of the data logger meters installed on-site.

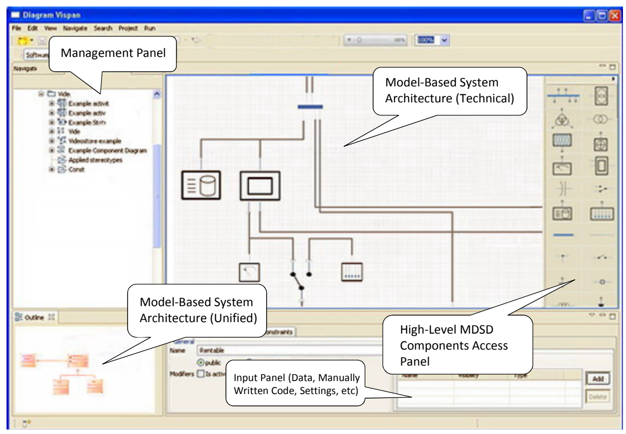

Fig. 4. Gateway view of a requirement set (domain specific) captured in the GUI concept that is linked to both embedded C++ and Simulink model elements.

Introducing MDSD for smart grid data analytics

It is time for appearance of MDSD approaches within software intensive smart grid industry . There are some progress in the last few years in terms of interoperability standards for software tools of smart grid. However, the majority of software systems today are implemented in general-purpose languages that address a broad domain. Classical programming languages for power grid software still survive alongside modern object oriented languages

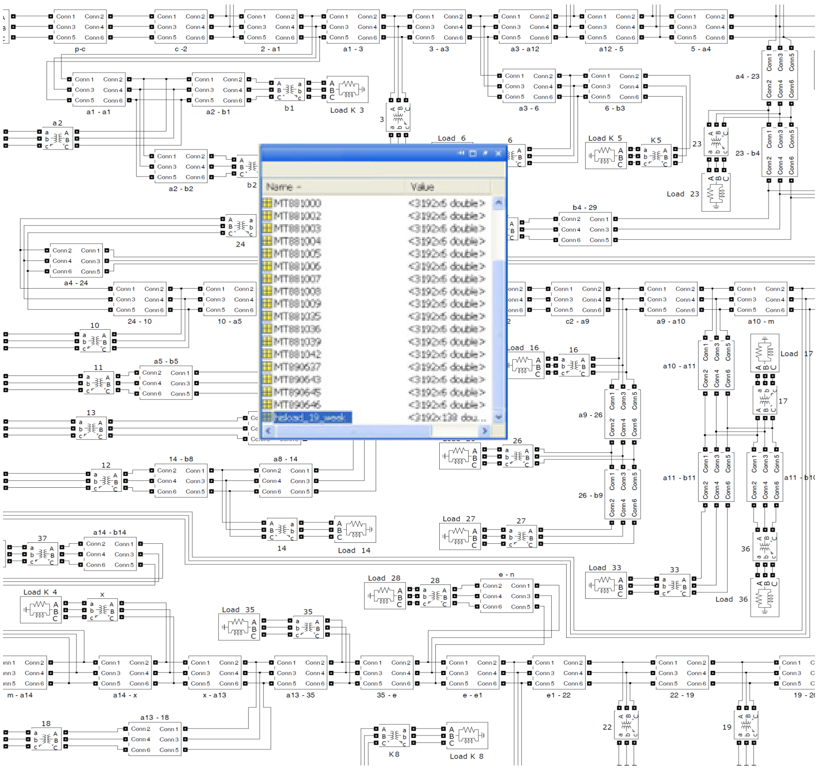

Fig. 5. A part of the network under study in MATLAB/Simulink along with a data analytics transformer window.

such as Java or C++. However, the characteristics of a good data analytics software, as mentioned in the Introduction, can hardly be represented by using classical or object-oriented code scripts. Additionally, MDSD languages such as the OMG's unified modeling language (UML) [5] provide a formal visual syntax to describe the structural aspect and part of the behavioural aspect of object oriented software systems [6]. On the other hand, all advanced applications of MDSD involve domain specific components (DSCs). The DSCs are formal specification components that incorporate domain concepts as first class language elements. Familiar DSCs could be mathematical computation or modeling and data management software. Here, the Simulink and a C-based coded module are the two DSCs of the presented project.

Starting to employ DSCs by distribution utilities requires full management commitment within the organization. A main advantage is that the utilities will be able to use standard mainstream software programs in some areas and maintain some of the previous tools, while introducing model-driven DSCs to represent a superior alternative for specific smart grid needs. One practical approach is to use domain analysis, and the initial design of appropriate DSCs, to clarify the question of scope before a rollout is considered [5]. The success of using a DSC can be measured by the compactness of the resulting specifications. If it takes a disproportionate amount of time to maintain specifications expressed in the DSC, it will not be possible to convince an application development team to use the new approach.

Metamodeling, as shown in Fig. 1, is an essential activity in any MDSD. Metamodels are used to specify the abstract syntax of a DSC, and they are also used in the context of MDSD to describe the structural elements of independent models, and model transformations. When defining domain specific models in software systems, the key concepts are visual elements, properties, and some structural rules that enables grouping or classification of model elements. This forms a major part of the MDSD, as shown in Fig. 4 for the deployed project, and is responsible for much of its usefulness and appeal.

In view of the fact that most distribution utilities are still within the early phases of the journey towards smart grid data analytics software through MDSD, it is worthwhile to map out a path that ensures successful future developments.

1) Utilities need to set up a small pilot project similar to the case proposed in this paper. Loss estimation would make a good case since it is a part of comprehensive distribution system studies and involves several primary challenges both on the computationally efficiency and technical data gathering.
2) Template languages should be used for developing DSCs. Open source and widely accepted languages would make a good choice in this regard. For example, the C++ and MATLAB are used in the pilot project outlined in this paper.
3) A large part of MDSD is about knowledge management, and building domain-specific assets. Further business DSCs can be added later if required. For instance, an operation planning module, such as a program for

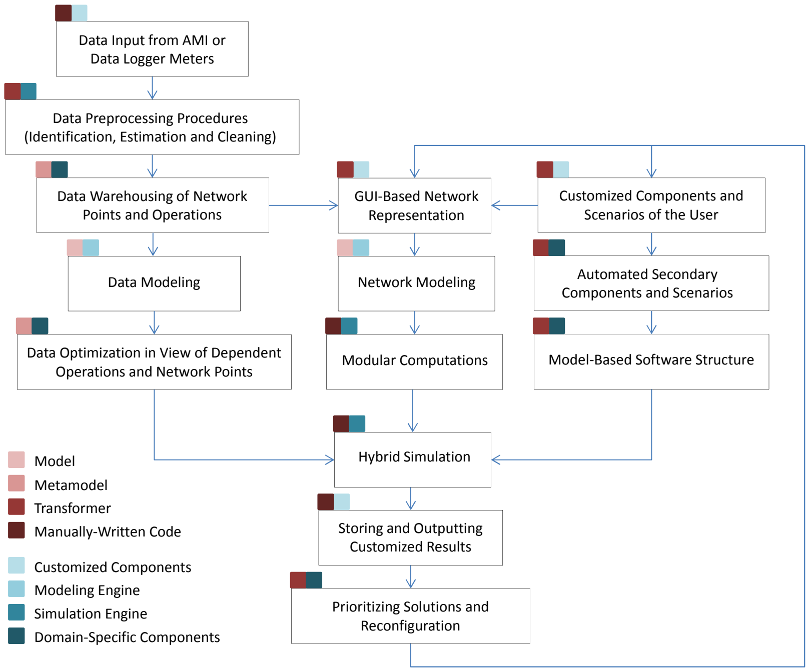

Fig. 6. Software design workflow for loss estimation and optimal reconfiguration from three perspectives displayed by arrows and two colour maps.

optimal reconfiguration of distribution feeders based on minimal losses, could be added to the outlines MDSD.

4) Domain-specific framework development can be a very substantial part of MDSD. For example, the reference implementation and tool kit of the outlined pilot project should be kept up to date for the purpose of being able to test new changes and upgrades in the distribution network.

PRACTICAL CASE STUDY: TECHNICAL LOSS ESTIMATION

Architects of MDSD-based smart grid software are required to have a high level of expertise in the field of distribution systems. On the other hand, The MDSD approach is underpinned by a variety of technical standards, some of which are yet to be specified or are yet to be implemented in a standard manner. Therefore, pilot projects will be customized based upon the chosen projects of the utilities and needs.

A practical system-wide simulation and loss calculation software were chosen to examine a prototype of the MDSD-based smart grid software. Main components of the developed software package is shown in Fig. 6. This figure uses the definitions of the MDSD presented in Figs. 1 and 2. Using transformers, metamodels and domain-specific components (or model-based high-level abstractions) discriminates this software from object-oriented software platforms which solely use high-level programming languages. Based on customizable components and user interfaces, utility managers would be able to examine different scenarios and designs within an integrated software environment with no need to re-write or re-design the software due to changing network policies and designs.

CONCLUSIONS

This paper introduces model-driven methodology into the area of smart grid software development and data analytics. The outlined concepts has been tested within a pilot project. This project is carried out on a 33-data bus distribution network. The main goal of the developed software is to perform a full distribution system analysis based on the data gathered from data logger meters. Testing the performance of the MDSD-based program has however been placed on the loss estimation. The main idea is at fully providing a tool for exploring and comparing algorithms for optimal smart grid data management. The future work includes identification of design patterns by comparing various concrete solutions and model transformations.

REFERENCES

[1] Siemens eMeter and Greentech Media, 2012, "Understanding the potential of smart grid data analytics".

[2] H. Valizadeh Haghi, M. A. Golkar, M. Tavakoli Bina, 2011, "Energy loss forecasting in active distribution networks", *Proc. 21st CIRED*, Frankfurt, Germany, 1-4

[3] M. A. Golkar, H. Valizadeh Haghi, "Using a multivariate design of experiment (DOE) method for congestion study in distribution systems under impacts of plug-in electric vehicles", *Proc. 21st CIRED*, Frankfurt, Germany, 1-4

[4] M. Völter, et al., 2006, *Model-Driven Software Development: Technology, Engineering, Management*, Wiley, 446.

[5] J. Bettin, 2006, "Transitioning to model driven software development: preparing for the paradigm shift", *SoftMetaWare Ltd.*, 1-16.

[6] www.uml.org, Accessed Jan. 2013.