Model-based Testbed Design for Electric Vehicles

Martin Paczona[1], Heinrich C. Mayr[2] and Guenter Prochart[3]

Abstract: Electric cars boom. This puts pressure on providing and improving tools and systems for electric car development. Electric vehicle testbeds (EVTs) are such systems: they serve for testing all high voltage vehicle components like batteries, inverters or complete engines and help to reduce the need of cost intensive road tests. EVT users like manufacturers of automobiles, aircrafts or train engines mostly have individual requirements. EVTs are therefore typically tailor-made solutions. Today’s approach to customized testbed (component) design starts with drawing the overall architecture using tools like MS Visio; based here-on, software developers, circuit plan designers, and engineers use their specific low-level design and development environments, obviously with no transformation or generation out of the initial drawing with causes all known challenges of such procedure. This paper presents a novel, innovative and scalable approach to EVT design based on an ontology grounded Domain Specific Modeling Language (DSML). It enables the user to describe the customer requirements in the familiar form. The resulting model can then be used to generate circuit diagrams and software configurations. Such approach not only may reduce development time and cost but may increase the quality of the resulting EVT.

Keywords: Modeling, Meta-Model, Ontology, Circuit plan, Testbed, MDA, DSML, Electric vehicle.

1 Introduction

We present here an innovative, model-based approach to the design of technical systems using Electric Vehicle Testbeds (EVTs) as an example. Most technical disciplines have their specific design methods and languages, for example circuit diagrams in the case of electrical systems. Rarely, these methods are derived from more general conceptualizations like metamodels or ontologies, and just as little are based on universal or standardized conceptual modeling languages like UML, Petrinets or similar. In the case of EVT development projects, which mostly target customized individual solutions, designers typically draw the overall architecture using tools like MS Visio. Based here on, software developers, circuit plan designers, and engineers use their specific low-level design and development environments, obviously with no transformation or generation out of the initial drawing – with all the notorious problems caused by such procedure like inconsistencies, loss of information, manual rework, late design feedback or poor reuse.

[1] AVL List GmbH, Electrification Products, Hans-List-Platz 1, Graz, 8020, martin.paczona@avl.com
[2] Alpen-Adria-Universität Klagenfurt, Application Engineering Research Group, 9020, Klagenfurt, heinrich.mayr@aau.at
[3] AVL List GmbH, Electrification Products, Hans-List-Platz 1, Graz, 8020, guenter.prochart@avl.com

By the example of EVT design, we will show that conceptual modeling may be useful for comparable engineering domains, in particular for mechatronic systems where engineers from different disciplines like electronic, mechatronic, software and management have to collaborate. Our approach is inspired by the state of the art in domain specific modeling methods (DSMMs) [Gh11, MM15, OM14]. In particular, we propose to provide, for a given engineering domain, a tailored domain specific conceptual modeling language (DSML), including tool support and mechanisms for model transformation to the lower domain levels. The work presented here is part of a dissertation project, which currently is implemented in close cooperation with the company AVL List GmbH. We adopt the process model for DSMM development discussed in [MM15] and follow Thalheim's [Th14] definitions regarding conceptual modeling, that presume the background of conceptual models to consist of "a base, a context and a community of practice".

The structure of the paper is as follows: In section 2 we outline the essential aspects of EVTs and their design. Section 3 presents a metamodel and the conceptual DSML we have developed based on the results of an intensive domain analysis including interviews with EVT designers and developers. In addition, we sketch the modeling process using that DSML and the creation of an appropriate modeling tool by deploying the metamodel framework ADOxx [FK13]. Section 4 shortly reviews related work. The paper ends with an outlook on future research in section 5.

2 Electric Vehicle Testbed Design

Electric vehicle testbeds serve for testing all high voltage vehicle components like batteries, inverters or complete engines and help to reduce the need of cost intensive road tests. Users of EVTs are manufacturers of automobiles, aircrafts, train engines and similar. They mostly have individual requirements so that EVTs are typically customized solutions. We concentrate here on the main EVT-components:

- **Unit under Test UUT**: In the case of batteries, the related EVT is called battery testbed (see Fig. 1). If the UUT is an E-Motor together with an inverter (DC to AC), the EVT is called electric powertrain component testbed (see Fig. 2).
- **Battery Management System BMS**: a low voltage component for controlling the battery functions.
- **Engine Control Unit ECU**: a low voltage component for controlling the inverter of the E-Motor.
- **Supply**: A direct current (DC) source that quickly can react on set point chances of voltage, current and power. In an EVT it is used for both, traction battery emulation (Fig. 2), and battery testing (Fig. 1).

- **(Power)Distribution Box PDU**: Connection point for the UUT including safety components (see Fig. 3).
- **Switch Box:** Connects the supply with one or more Distributions Boxes via power cables, or directly with the UUT, if the customized solution does not provide Distribution Boxes.
- **Test Automation System TAS**: Controls the interplay of the EVT components to perform the required test functionality according to a 'Vehicle Model' describing driving patterns, and a 'Battery Model' describing charging and discharging patterns and set points.

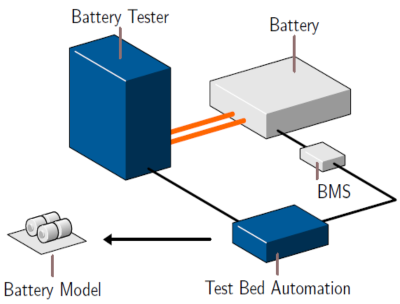

Fig. 1: Battery testbed

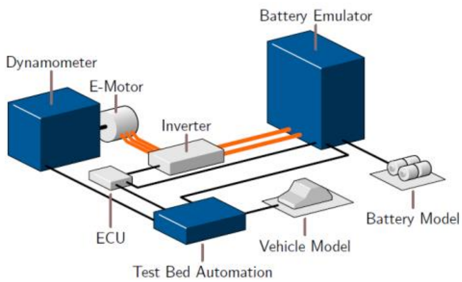

Fig. 2: Electric powertrain component testbed

As mentioned in the Introduction, developing a customized EVT solution starts with drawing an overall structural design using a standard drawing tool. Figure 3 shows a very simple example of such drawing, which even does not contain some of the before mentioned EVT components.

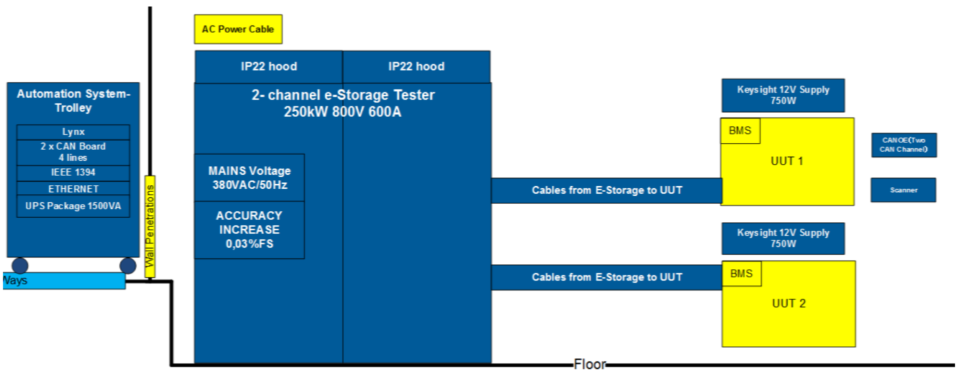

Fig. 3: TTD Example

The elements of such design may be conceptualized, abstracted and composed to a domain specific metamodel (according to level 2 of the OMG Meta Object Facility MOF [OM14]). A related and appropriately instrumented modeling language (a DSML) then serves to create models (on MOF Level 1) of EVT types as extensions of this metamodel. The MOF level 0 elements, being extensions (instances) of such models, are descriptions of concrete customized EVTs.

Clearly, practitioners would not be happy with having to deal with a pure modeling language. Rather they expect appropriate tool support for model creation, combination, reuse and analysis as well as guidance about how to proceed. In addition, circuit plan designers would expect the possibility of automatically generating circuit plans out of such models as this would help to reduce development time and costs, is flexible regarding agile modifications, and could contribute to increase the quality of the EVT development process. Consequently, this leads to the development of a comprehensive domain specific modeling method, i.e. a DSMM. [MM15] present a process model for DSMM development that proposes, on the top level, the following five stages: Preparation, Language, Modelling process, Modelling tool and Evaluation. Following this process model, the first activity in the stage Specification was to elicit the requirements and expectations of the engineers involved in EVT design and development. The results of this endeavor fall into four main categories:

(1) Understandability: The modeling language as well as the models should be intuitively understandable for all stakeholders involved in order act as a "bridge that overcomes the semantic gap" [Gh11] between the various user groups. The language, therefore, must cover all concepts needed for EVT design and only those; the visual notation should be similar to the familiar one (to be effective in the sense of [Mo09]). In addition, the engineers asked for the availability of stakeholder group specific views on models [Br17] increasing understandability for experts from different disciplines collaborating in an EVT project. In the traditional approach, such views would require extra drawing effort and, therefore, are rarely produced.

(2) Consistency: The modeling environment (tool) should provide means for ensuring syntactical correctness, for supporting compliance with semantic construction rules ("EVT Rules"), and for checking specific consistency constraints. Clearly, the traditional approach has to manage without such means and thus is error-prone with all related problems of late error detection.

(3) Re-Usability: Currently, designs are "re-used" by simple copy&paste of diagram parts, again with all related problems. Consequently, the engineers ask for a flexible multi-level [Fr13] modeling environment supporting all kind of model re-use and integration.

(4) Cost-Effectiveness: This relates to all previous requirements, but in addition should be strengthened by the availability of generators for producing lower-level artifacts out of models like circuit plans as well as configuration data for system components. Clearly, in the traditional approach, such means are not available. At least, most of the common circuit plan tools like EPLAN or WSCAD allow for creating circuit plans by selecting and combining "prefabricated" building blocks [SB07, Wa17, EP18a] but this obviously still is a manual process.

3 EVT-MM: A Modeling Method for EVT Design

This section is structured according to the five DSMM development stages introduced in the previous section.

3.1 Analysis of domain-specific documents

Developing a domain specific modeling method requires a deep analysis of the domain to understand and figure out the important aspects to be conceptualized and included in the metamodel. For that purpose, we analyzed a large number of existing EVT design documents, like the TTD for a battery testbed as shown in figure 3. Such TTD contains information about the main system components, cable lengths, software packages to be installed, information about services. Documents that are more detailed contain information about specific customer requirements regarding controller parameters, customized interface or system limits. Analyzing the product structure of common testbed applications may reveal additional information, as far as EVT standard products with customizable options are already available.
Our analysis came up with a series of aspects to be considered when developing an EVT DSMM: connections, placement of parts, dynamic specifications, timings, the current load of all HV components, the maximum electric strength of components, the cable length influencing the current dynamics, the temperature range and power overload capacity as well as the possible operation modes of the UUTs. In addition, general constraints like Ohm’s law, electromagnetic induction, electric capacity and Electric Energy are to be taken into account as well as the relevant Standards: IEC 61439, IEC 13849, IEC 81346, and IEC 60146. As an example think of the length and diameter of a power cable as they these direct impact on the current dynamics that can be achieved by the supply.

3.2 Meta-Model and DSML

Based on these findings we iteratively worked out the necessary conceptualizations and combined these to the metamodel as depicted in Fig. 4 and Fig. 5 , the latter refining the concept of “Hardware Component” including the concepts needed for circuit design. The italic-labeled concepts are abstract in the sense that they have no instantiation on the level M1. The semantic foundation of these concepts is done by means of an EVT ontology, which we cannot present here due to lack of space. It will be published elsewhere.

Models, i.e., MOF level M1 extensions (instances) of this metamodel, represent EVT types. M0 level instances of such models then are concrete customized EVT designs corresponding to the traditional TTDs. Table 1 shows some examples of the modeling concepts on the three MOF-layers. The representation concepts for the EVT DSML follow the style of the TTDs. An example is given in figure 8. This conceptual model has been created using a prototype of the modeling tool we are currently developing based on the ADOxx metamodeling framework (see section 3.4).

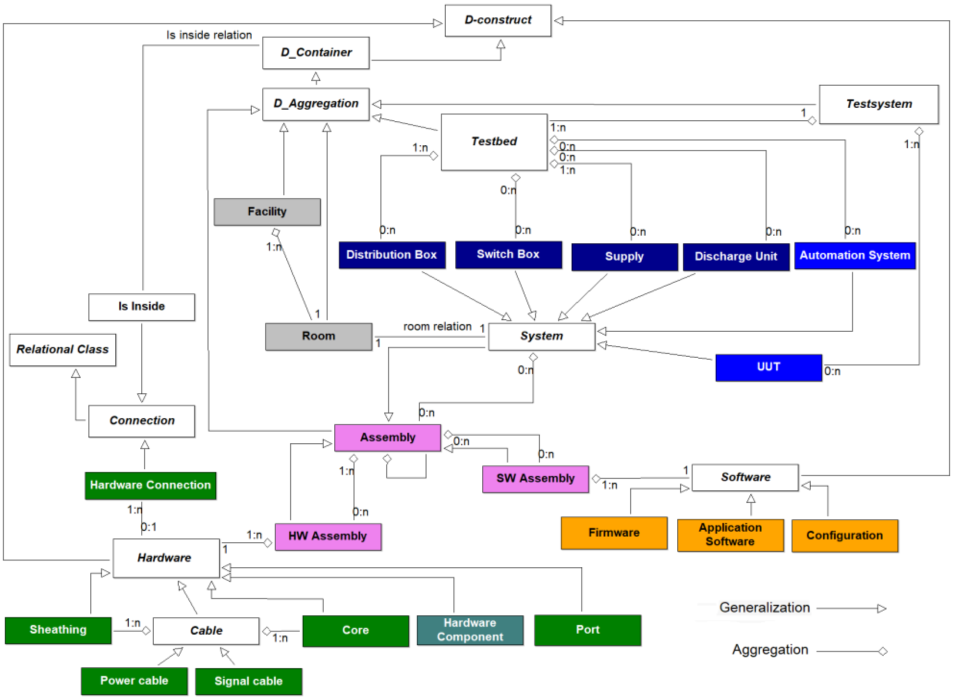

Fig. 4: EVT Metamodel (M2)

M2	M1	M0
Supply	2 channel supply	2 channel supply 800V/600A
Supply	1 channel supply	1 channel supply water cooled
Power cable	Power cable	Power cable 10m, q=150mm^2
Discharge unit	Discharge unit 4x	Discharge unit 4x 1200V
UUT	Battery	Battery Dura Ultra 50kWh
Assembly	Door switch	Door switch DS03
Configuration	Controller parameter	Controller parameter T_{90}=0.5ms

Tab. 1: Example of EVT language concepts on MOF: M2, M1 and M0

3.3 Notation

For each element of a MOF hierarchy one (or possibly several) appropriate representation language(s) have to be provided [Ma18]. Usually, for the metamodel itself, a UML class diagram like notation is chosen as we did in section 3.1. For the making the metamodel concepts available on M1 we designed a modeling language EVT-ML by following the principles and guidelines of [Re13, Gu13 and Mo09]. The notations can be divided into the categories *Container Elements* (symbol is an unfilled square), *Software Elements* (yellow rounded rectangle elements), *System Elements* (blue rectangle, based on real system), *Cables* (line) and *Hardware Components* (based on IEC 60417). For reasons of intuitive understandability, aggregations drawn overlapping, for example, System Elements within a Room symbol (see fig. 8) that again might be drawn within a Facility symbol.

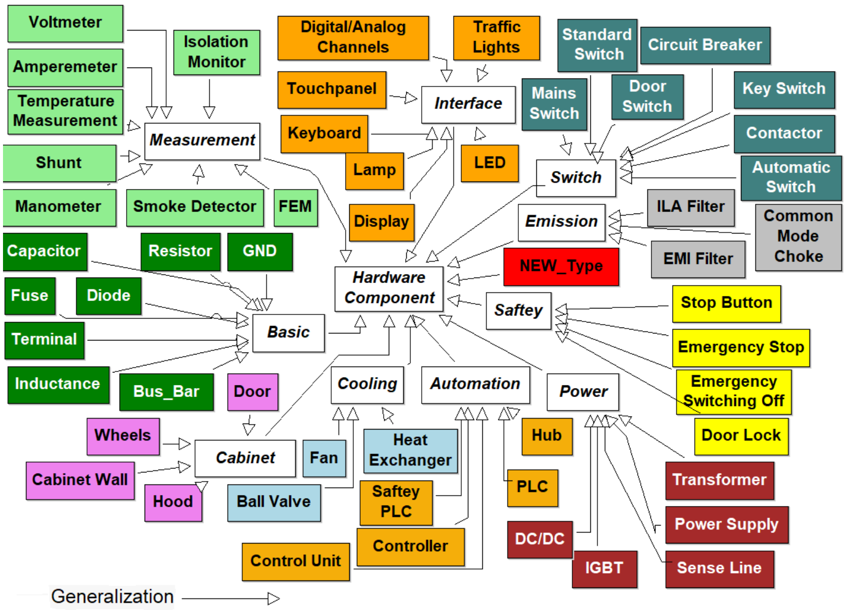

Fig. 5: Hardware Component Metamodel refinement (M2)

Figures 6 and 7 give an overview over the notational elements for testbed and circuit diagram representation. The latter correspond largely to the existing IEC 60417 notations, which are well established in the electrical engineering domain. By using already existing notations, the barrier to learn the new language is reduced [Ka09].

3.4 Modeling process

A modeling process definition (describing a modeling process model) provides guidance to the modeler: which model view to start with, how to use the modeling concepts, how to structure models hierarchically using aggregation and generalization relationships, and how to establish and deploy naming conventions [MM15]. Such guidelines should be oriented at the daily needs of EVT designers in order to be considered by them.
The EVT modeling process is part of the overall EVT development process, which starts with initiating a new project and eliciting the customer requirements. For this purpose, a first rough testbed model is established (possibly building on an appropriate pattern if available in the model base) which then is refined iteratively. It should contain the facility, the rooms, the main parts (systems), the connections (signal cables or power cables) between the systems and the automation system. In addition to that, the properties of the modeling elements must be specified (to allow transformation and model checking techniques). The circuit plan is created using the circuit modeling toolbox (see below) including cables, ports and hardware component elements. A circuit plan example create by use of our modeling prototype is shown in Fig. 9. Each circuit plan belongs to a system (specified by a relation INTERREF).

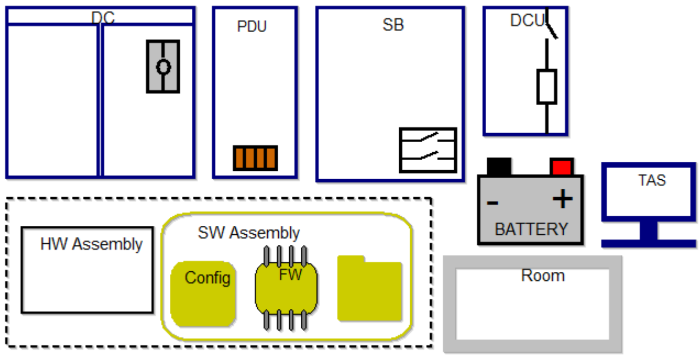

Fig. 6 Notations (Testbed View)

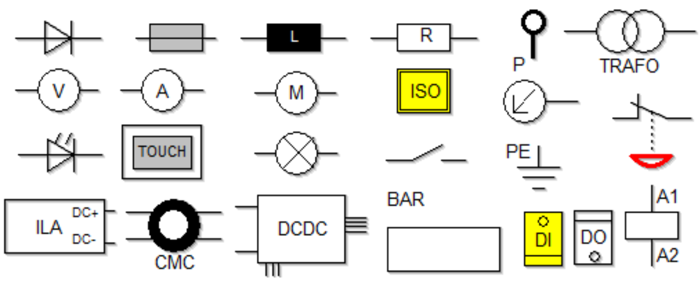

Fig. 7 Excerpt of the Notations (Circuit Diagram View)

3.5 Modeling tool

The most efficient way for tool implementation is to use a so-called metamodeling platform like ADOxx [FK13], Eclipse EMF or MetaEdit+ [5,7]. We decided to deploy ADOxx based on the fact that ADOxx has been successfully used and tested since long in a large number of research and industrial projects, in particular also by our department. Moreover, we used it successfully for creating a tool for supporting a DSML for circuit plan design. The Testbed Example Model shown in figure 8 has been created using the current stage of our EVT modeling tool under construction. The colors used correspond to the metamodel concepts from which the particular model element is instantiated. Even this first prototype allows for some basic consistency checks and model analysis (inbuilt in ADOxx) and avoids constructing syntactically incorrect models (correct buy construction [Ke14]). This allows for fast feedback during the design phase.

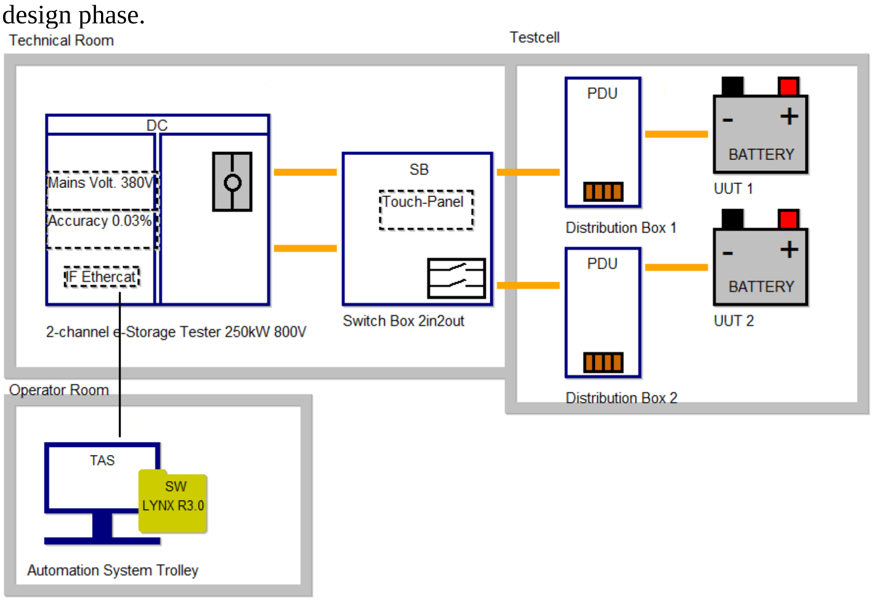

Fig. 8 Simple Testbed Example Model (M1)

Because the EVT modeling tool is to be used by managers and by circuit plan designers who operate at a different level of detail, the EVT modeling tool provides different views on the model. The "Testbed View" which is shown in Fig. 8 is used for modelling the overall testbed and the "Circuit Plan" view is used for modeling the circuit plan of the EVTs (Fig. 9). To reduce complexity the modeling tool provides composition technology. To increase the productivity, template models can be loaded (from a database) which then are further modified.

3.6 Generation of artefacts

For reasons of cost-efficiency we aimed at generating artefacts directly out of the model. In particular, software configuration files (performance, hardware, network) and circuit plans (for EPLAN P8) are generated. EPLAN P8 circuit plans are widely used in the domain of electrical engineering to document the electrical structure of a system. Therefore, our *AdoScript Circuit Plan transformation* module generates out of a circuit model an input file for the *EPLAN Interface Script*, which uses the EPLAN API to create the circuit plan pages and the components on the pages [EP18b] in an 1:1 correspondence between model and circuit plan. In addition to the circuit plan pages the bill of material (BOM) is generated which contains the material that is needed for each system.

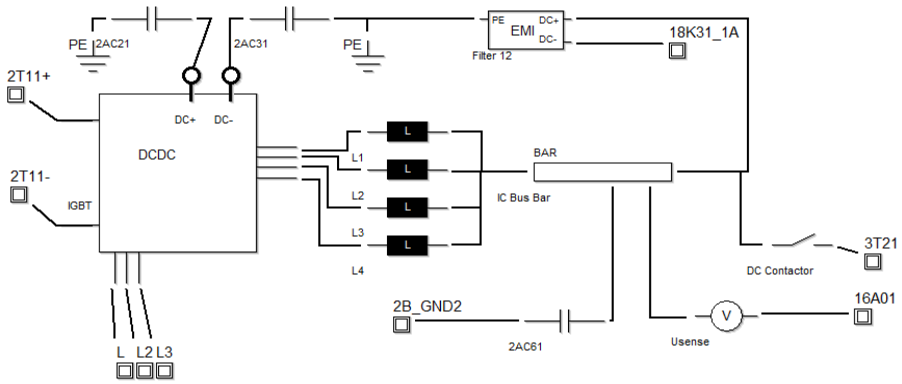

Fig. 9 Circuit Plan Example (M1)

3.7 Evaluation

Apparently, what we are presenting here is research in progress. However, several evaluations have been done already. At the beginning, an online survey in the EVT manufacturer’s development team was performed to figure out the requirements for the DSML and the Modeling tool. The survey included questions, related to the current EVT development process, used tools, important components, customer specific changes, and the application area. To evaluate the understandability of the modelling notation also an online survey has been performed. There the task was to select the correct name for the shown notation element. Based on the results the notation (for the system elements) was reworked. The notation is now based on the real-systems including graphic and text to be understandable intuitively. The modeling prototype has already been rolled out in the development department in order to collect first test results. The developers used it to model the testbed for project documentation, communication across stakeholder and troubleshooting. The main result was, that the models are clearly understandable to the involved persons (project manager, software developer, circuit diagram designer, team leader, support engineer and product manager). However, they proposed some improvements, e.g., a comment function, and an issue representation for troubleshooting. Moreover, they claimed, that it should be possible for reasons of complexity reduction, to hide elements that are drawn inside others. To evaluate the completeness of the DSML, we modeled twenty already existing EVTs (based on the TTD) using our modeling tool. As a result, all aspects of these EVT's were covered with the exception that there were no concepts for modeling components like a "Climatic Chamber" and similar. We are currently analyzing if it would be advisable to introduce a generic system element for covering such kind of components.

4 Related work

Since the 90s, domain specific modeling, languages and tools as well as ontological grounding are broadly investigated. Metamodelingplatforms like Eclipse EMF, ADOxx and MetaEdit+ provide powerful means for creating modeling and analysis tools for domain specific modeling languages. Tab. 2 shows some existing domain specific methods. [MM15] focus on Model Engineering, i.e. propose a process model for developing domain specific modelling methods. The researchproject IMoMeSA [HRZ15] came up with a development method for mechatronic systems. It enables the development of machines and systems starting from the concept of a virtual prototype. The method integrates already existing tools and enables generation of MCAD, ECAD and IEC 61131-3 code through model transformations. [CSV10] introduces a method to connect domain-specific tools through a higher-level SysML data model. [Ho17] introduces a method to integrate models in the plant engineering domain using the Anlagenreferenzstruktur. Sporer [Sp16] presents an approach to domain specific modeling of embedded automotive mechatronic systems that allows for linking requirements and specifications to the created models. However, so far no references on conceptual modelling method specifically for the domain of EVT design could be found. Clearly, SysML and UML based approaches are basically conceptual, but they are not domain specific in the sense propagated within this paper.

Name	Domain
EPLAN Electric [EP18a]	Mechatronic design
MATLAB [MA18]	Mathematic calculations
FUP [Ha15]	PLC programming
VHDL [IE00]	Electronic design automation
ComVantage [Bu16]	Production Management Systems
EMS-DSM [Sp16]	Mechatronic system development

Tab. 2: Modeling methods used in mechatronics

5 Outlook on next steps to do

This paper presents research and work in progress. However, as the idea of deploying a systematic conceptual DSML approach to EVT design is innovative, we decided to present here our first considerations and results. The approach proposed may not only reduce the development time for EVT but also improve the quality of developed customer specific EVTs, in particular when the lower-level design documents are generated by deploying model transformations according to the MDA principles. In addition using a DSMM-tool instead of drawing tools like MS Visio may help to overcome the well known problems caused by purely document based approaches. There is still much open research and work to do: First, more detailed end-user experiments will help evaluating the completeness and appropriateness of the proposed metamodel and DSML. Second, we need turning the prototype into a product including means for handling domain specific consistency constraints as well as continuous adaptations in the EVT domain. Moreover, to provide substantial improvements of effectivity and efficiency in EVT design, we will have to finalize and integrate generators into the EVT tool that produce, for example, circuit plans and configuration data out of a concrete EVT level M0 representations. This will reduce today's need of manual work substantially and thus contribute to the avoidance of related transcription errors.
During creating the metamodel (Fig. 4) it was found that the concepts of the metamodel are not only specific to the electric domain (especially EVT), but also have similarities with the hydraulic domain (e.g. filling plan). Such analogy was already detected in the 19th century by Oliver Lodge (drain-pipe theory) [Na02]. Examples for this are the Supply-pump, cable-pipe, capacitor-tank and resistor-constricted pipe connection. As a consequence we are currently thinking of a generalization of the presented approach.

Bibliography

[Br17] Bruneliere, H. et al.: A featurebased survey of model view approaches. Journal Software & Systems Modeling No. 10270, 2017.

[Bu16] Buchmann, R. A.: Modeling Product-Service Systems for the Internet of Things: The ComVantage Method. In (Karagiannis, D; Mayr, H; Mylopoulos, J. eds): Domain-Specific Conceptual Modeling, pp. 417–437, 2016.

[CSV10] Chami, M.; Seemullerm H.; Voos, H.: A SysML-based integration framework for the engineering of mechatronic systems. In: Proc. IEEE/ASME Int. Conf. on Mechatronic and Embedded Systems and Applications MESA 2010, QingDao, pp. 245–250, 2010.

[Di12] Dimitrieski, V. et al.: A Comparison of Ecore and GOPPRR through an Information System Meta Modeling Approach. In (Störrle et al. eds): Joint Proceedings of co-located Events at the 8th European Conference on Modelling Foundations and Applications (ECMFA 2012), Technical University of Denmark, Copenhagen, Denmark, pp. 217-228, 2012.

[EP18a] EPLAN Electric P8: Power for electrical planning and engineering, https://www.eplanusa.com/us/solutions/electrical-engineering/eplan-electric-p8/, accessed: 18/11/2018.

[EP18b] Eplan Software and Service GmbH. API User Guide, http://www.eplan.help/help/platformapi/2.7/en-us/help/API_UserGuide.html, accessed: 18/11/2018.

[FK13] Fill, H.; Karagiannis, D.: On the Conceptualisation of Modelling Methods Using the ADOxx Meta Modelling Platform. Journal Enterprise Modelling and Information Systems Architectures Issue 1/2013, pp. 4–25, 2013.

[FKT14] Frakes, W. B.; Kulczyski, G.; Tilley J.: A Comparison of Methods for Automatic Term Extraction for Domain Analysis. In: (Schaefer I., Stamelos I. eds.) Proc. Software Reuse for Dynamic Systems in the Cloud and Beyond. ICSR 2015. Lecture Notes in Computer Science Vol. 8919, Springer, Cham, pp. 269-281, 2014.

[Fr13] Frank, U.: Domain-Specific Modeling Languages: Requirements Analysis and Design Guidelines. In (Reinhartz-Berger, I. et al. eds): Domain Engineering, Springer Berlin Heidelberg, pp. 133–157, 2013.

[Gh11] Ghosh, D.: 2011. DSL for the Uninitiated, Communications of the ACM Vol. 54 Issue 7/2011, pp. 44–50, 2011.

[Gu13] Guizzardi, G.: Ontology-Based Evaluation and Design of Visual Conceptual Modeling Lanugages. In (Reinhartz-Berger, I. et al. eds): Domain Engineering, Springer Berlin Heidelberg, pp. 317-347, 2013.

[Ha15] Hanssen, D. H.: Function Block Diagram (FBD). In (Hanssen, D. H. ed): Programmable Logic Controllers: A Practical Approach to IEC 61131-3 using CODESYS, 2015 John Wiley & Sons, pp. 262–277, 2015.

[He10] Hepp, M.: Volkswagen Vehicles Ontology, https://www.volkswagen.co.uk/vocabularies/vvo/ns, accessed: 18/11/2018, 2010.

[Ho17] Hooshmand, Y. et al.: An approach for holistic model-based engineering of industrial plants. In (Maier A. eds.): Proc. 21st Int. Conference on Engineering Design (ICED17) Vol. 3/2017, Vancouver, 2017.

[HRZ15] Hackenberg, G.; Richter, C.; Zäh, M.: IMoMeSA Abschlussbericht. Technical Report., https://mediatum.ub.tum.de/attfile/1253882/incoming/2015-May/570232.pdf, accessed: 18/11/2018, 2015.

[IE00] IEEE 2000. IEEE Standard VHDL Language Reference Manual.

[Je16] Jeusfeld, M. A.: SemCheck: Checking Constraints for Multi-perspective Modeling Languages. In (Karagiannis, D.; Mayr, H.; Mylopoulos, J. eds): Domain-Specific Conceptual Modeling, Springer, Cham, pp. 31–53, 2016.

[Ka09] Karsai, G. et al.: Design Guidelines for Domain Specific Languages. In: Proc. of the 9th OOPTSLA Workshop on Domain-Specific Modeling, Helsinki School of Economics, Orlando, USA, 2009.

[Ka14] Karbe, T. et al.: State of the art for automotive ontology D308.010, crystal, 2014.

[Ke14] Kezadri-Hamiaz, M. et al.: Correct-by-construction model composition: Application to the Invasive Software Composition method. In (Buhnova, B.; Happe, L.; Kofron, J.

eds.): Proc. FESCA 2014, Grenoble, France, pp. 108-122, 2014.

[KKW07] Ahmed-Kristensen, S.; Kim, S.; Wallace, K. M.: A Methodology for Creating Ontologies for Engineering Design (modified version DETC2005-84729). Journal of Computing and Information Science in Engineering 2007, pp. 132-140, 2007.

[KMP16] Kiritsis, D.; Milicic, A.; Perdikakis, A.: User Story Mapping-Based Method for Domain Semantic Modeling. In (Karagiannis, D.; Mayr, H.; Mylopoulos, J. eds.): Domain-Specific Conceptual Modeling, Springer, Cham, 439–454, 2016.

[Ma18] Mayr, H.C. et al.: A Model Centered Perspective on Software-Intensive Systems. In (Fellmann, M.; Sandkuhl, K. eds.): Proc. 9th Int. Workshop on Enterprise Modeling and Information Systems Architectures. Vol. 2097, CEUR-WS.org, Rostock, Germany, pp 58-64, 2018.

[MA18] MathWorks, Math. Graphics. Programming, https://ch.mathworks.com/products/matlab.html, accessed: 18/11/2018.

[MM15] Michael, J.; Mayr, H. C.: Creating a Domain Specific Modelling Method for Ambient Assistance. In: Proc. International Conference on Advances in ICT for Emerging Regions (ICTer 2015), IEEE, Colombo, Sri Lanka, 2015.

[Mo09] Moody, D. L.: The "Physics" of Notations: Toward a Scientific Basis for Constructing Visual Notations in Software Engineering. IEEE Transactions on Software Engineering. Vol. 35 Issue 6, pp. 756–779, 2009.

[Na02] Nahin, P. J.: Oliver Heaviside: The Life, Work, and Times of an Electrical Genius of the Victorian Age, illustrated reprint, John Hopkins University Press, 2002.

[OM14] OMG 2014. MDA Guide rev. 2.0., https://www.omg.org/cgi-bin/doc?ormsc/14-06-01, accessed: 18/11/2018.

[Re13] Reinhartz-Berger, I. et al.: Domain Engineering: Product Lines, Languages, and Conceptual Model, Springer Heidelberg New York Dordrecht London, 2013.

[SB07] Schaefer, D; Baguley P.: ECAD functionality suitable for webbased optimised automated product variant generation. International Journal of Internet Manufacturing and Services. Vol. 1 No. 1, 2007.

[Sp16] Sporer, H.: Mechatronic System Development: an Automotive Industry Approach for Small Teams, Doctoral Thesis, TUGraz DIGITAL Library, 2016.

[Th14] Thalheim, B.: The conceptual model ≡ An adequate and faithful artifact enhanced by concepts. Frontiers in Artificial Intelligence and Applications Vol. 260, pp. 241–254, 2014.

[Tu06] Tudorache, T.: Employing Ontologies for an Improved Development Process in Collaborative Engineering, Doctoral Thesis, 10.14279/depositonce-1477, 2006.

[Wa17] Walker, T.: Elektrokonstruktion im Sondermaschinenbau mit WSCAD hochgradig standardisiert, Hüthing Verlag, 2017.