Cross-factor analysis of software modeling practices versus practitioner demographics in the embedded software industry

Deniz Akdur
ASELSAN Inc., Ankara, Turkey
Department of Information Systems,
Informatics Institute
Middle East Technical University (METU)
Ankara, Turkey
denizakdur@aselsan.com.tr

Vahid Garousi
SnT center, University of
Luxembourg, Luxembourg
Department of Computer Engineering
Hacettepe University, Ankara,
Turkey
garousi@svv.lu

Onur Demirörs
Department of Information Systems,
Informatics Institute
Middle East Technical University
(METU)
Ankara, Turkey
demirors@metu.edu.tr

Abstract— **Software-intensive embedded systems have evolved to be essential tools of our lives. To cope with growing complexities in embedded software industry, software modeling and model-driven engineering (MDE) have become popular for design, development and testing of these systems. However, the usage of models in embedded software industry and the relevant practices usually vary since challenges, requirements and purposes differ among systems as well as among sectors. To understand the state-of-the-practice of software modeling in embedded systems industry, we designed and conducted a world-wide survey, which accepted 627 responses from 27 different countries. The goal of this paper is to present our work built on this survey to better understand and characterize the cross-factor analysis of software modeling versus practitioner demographics, e.g., degree of using software modeling versus target sectors of the products.**

Keywords-Embedded software; modeling; model-based; model-driven; model-driven engineering; cross-factor analysis

I. INTRODUCTION

Design, development and testing of software for embedded systems are not trivial due to their multiple constraints across different dimensions of performance and quality. Moreover, the increasing amount of components, which require seamless integration of many hardware and software systems, makes the embedded systems development more challenging.

Software modeling is a widely used approach in the embedded software industry and a tool to manage the complexity of these systems. Modeling helps engineers to work at higher levels of abstraction and facilitates communication. However, the usage of models in embedded software and the relevant practices usually vary since requirements, challenges ~~differ~~ among sectors (e.g., aviation, appliances and automotive [1]). At one extreme, some practitioners use software modeling at a very informal level, where diagrams are sketched on a paper or a white board in order to help communicate ideas with colleagues. In such cases, the emphasis is on selective communication rather than comprehensive specifications, and these diagrams are usually either discarded or quickly become inaccurate since they are not kept updated along with the

source code [2]. At the other extreme, modeling turns into programming with automated generation of code from models.

Our practitioner survey showed the state-of-the-practice in software modeling practices in the embedded software industry by providing a view on the latest approaches, languages and tools used by practitioners. The goal of this paper is to present our work built on this survey to better understand and characterize the cross-factor analysis of software modeling versus practitioner demographics, e.g., degree of using software modeling versus target sector of the products. With the help of this analysis, both embedded software professionals and also researchers benefited from our results and this will encourage more academia-industry collaborations.

The remainder of this paper is structured as follows. Section 2 discusses terminology used in this study and a brief overview of our recent survey. Section 3 presents the cross-factor analysis. Section 4 reviews the potential validity threats. Finally, Section 5 concludes this study.

II. BACKGROUND

We present an overview of the concepts of Model-Based Engineering (MBE) versus Model-Driven Engineering (MDE) and Model-Driven Development (MDD). A brief overview of our survey is reviewed next.

A. Terminology: MBE versus MDE and MDD

In the literature, there are various terminologies in the context of software modeling. In our work, we followed the terminology offered by Brambilla et al. [3] for describing and differentiating between "model-based" and "model-driven" ~~MDD~~ treats models as the ~~key artifacts of the~~ process. Usually, in MDD, the implementation is automatically generated from the models. In addition to development, MDE encompasses all the other tasks of the software engineering (SE) process such as testing and maintenance, and thus, MDE is considered a superset of MDD. On the other hand, MBE is a process, in which software models still play an important role although they are not necessarily the key artifacts of the development. For example, designers specify the models (e.g., on paper or by

using modeling tools), but then these models are directly handed out to the programmers to manually write the code (i.e., there is no automated code generation). In that sense, MBE refers to a 'lighter' version of MDE [4]. Therefore, all model-driven processes are model-based but not the other way round. Figure 1 (adopted from [4]) depicts these concepts.
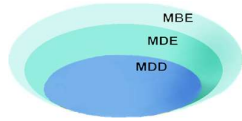

Figure 1. *Venn Diagram - The relationship among MBE, MDE and MDD*

B. A Brief Overview of Our Recent Survey

Although there have been a few prior surveys related to modeling in the embedded software industry (e.g., [5-7]), they have either focused on only one aspect of modeling, (e.g., the use of Unified Modeling Language (UML)) or the use of formal models, or modeling in regional contexts (e.g., in Brazil or in Greece). There are also surveys, whose participants were involved with model-based techniques on a single sub-domain of embedded systems (i.e., the automotive industry [5]). However, our survey takes a larger and more holistic scope on the subject and with a higher scale (data from worldwide).

The research approach we used in our survey is the Goal, Question, Metric (GQM) [8]. Stated using it template, the goal of our survey was to understand the state-of-the-practice of modeling in the embedded software domain by identifying to what degree, why and how modeling is conducted.

In designing the survey, we made sure that the questions are relevant to the industry and also capture the most useful information as relevant to the goal of the survey. In order to develop a survey that would adequately cover the latest trends on modeling, we reviewed the similar past surveys, benefitted from our professional experiences in industrial projects, and prepared a draft set of questions. We conducted a round of peer reviews and according to their results, some questions were improved. Since performing a pilot study before distribution is an important step, the survey was piloted by eight colleagues from different industries, software engineering roles, experiences and nations (four Turkish, two English, one French and one Taiwanese embedded software professionals). Given their feedback and the time they needed to fill out the survey, the questionnaire was refined and the final version of the questionnaire consisted of 27 questions, in the form of multiple-choice, single-choice and Likert-scale answers. Where applicable, free-text areas for additional input were provided to respondents as "Other" [9]. To design and execute the survey, we used the Google Forms tool. The ethics approval for the survey was issued by the Human Subjects Ethics Committee of Middle East Technical University (METU) in March 2015. The survey was then executed in the period of April-May 2015.

The identified target audience is anyone working in embedded software industry, with a variety of different software engineering roles from requirements engineer to business analyst and from software developer/programmer to

tester. In terms of descriptive statistics of the dataset gathered via the survey, 50% and 11% of respondents had a Master's and PhD's degree respectively; whereas 39% of respondents had Bachelor's degrees. Most of the participants had "Software Developer/Programmer" role; "Software Designer", "Software Architect" and "Software Tester" roles were the other frequent responses. The majority of respondents had 10+ years (52%), followed by 6-10 years (40%) of work experience. As for the modeling experience, the majority were in 6-10 years range (46%), followed by 10+ years of modeling experience (40%). More detailed demographics of the participants and further details can be found in [10].

After analysis of responses, we made several interesting observations:

(1) Software engineers use modeling approaches in varying degrees, which usually depends on their experience and project needs. 11% of respondents have not been using any software modeling (neither formal nor informal); whereas the remaining 89% are somehow (partially or fully) using software modeling.

(2) Although there is a wide spectrum in terms of the latest software modeling practices, the C programming language and Eclipse-based tools seem to be the most popular choices. The majority of respondents use UML and the second most frequently reported response is "Sketch/No formal modeling".

(3) The most used diagram types are sequence diagrams, state-machine diagrams, and class diagrams.

(4) The majority of respondents use modeling in the "systems/software design" phase, followed by "implementation" and "requirements/systems analysis" phases of software development life cycle (SDLC).

(5) 59% of all participants do not use any model-driven techniques; these participants are either using model-based approaches or using sketches/no formal modeling. The remaining 30% are model-driven users at some degree.

(6) Top four motivations for adopting MDE are "Cost Savings", "Shorter Development Time", "Reusability" and "Quality Improvements".

III. CROSS FACTOR ANALYSIS

One of the opportunities the survey data provided as a further study was to analyze relations among software modeling practices and practitioner demographics. To investigate the potential factors that could correlate with other factors, we created a two-dimensional matrix where the individual survey questions are listed in both rows and columns. For this study, due to space constraints, we report the cross-factor correlation analysis for four combinations only (marked with 'x' in Table I) after using the expert opinion of the authors and in consultation with industry contacts. The other remaining combinations have been planned to be investigated in future works. As seen from Table I, all 'x' marked combinations are related with Q7 in the survey [10], which is about the target sector(s) of the products developed by the company.

TABLE I. CROSS-FACTORS CORRELATIONS CHOSEN AS THE MOST IMPORTANT ONES TO BE INVESTIGATED

			Demographic aspects			
			Q6	Q7	Q8	Q9
			Profile of the companies participants work for			
		Description	Type(s) of application developed	Target sector(s) of the product(s) developed	Number of employees working in software engineering roles	Size of typical software development team
Model-driven engineering practices	Q10	Degree of Using Software Modeling in SDLC		X		
	Q13	Mediums to Create Sketch or Model				
	Q14	Modeling Languages		X		
	Q15	Programming Languages				
	Q16	Modeling Environments/Tools				
	Q17	Diagram Types		X		
	Q18	SDLC phases, where software modeling is used				
	Q19	Degree of Using Model-Driven Techniques		X		

Seven possible choices were pre-provided in the questionnaire, which were designed in discussions with embedded industry partners [11]. Note that since this was a multiple-response question, multiple sectors could be recorded, e.g., a person could work in both consumer electronics and IT & Telecommunications. As it can be seen from Figure 2, there is a good mix of respondents from various embedded software industry sectors. For this study, due to space constraint, we excluded "Government", which is the least chosen sector.
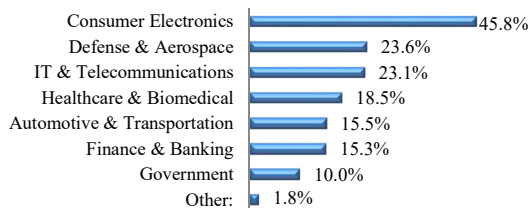

Figure 2. Target sectors of products

We asked how often the participants use software modeling in the SDLC by including both formal and informal usage (i.e., models or sketches) using a 5-point Likert-scale. After analyzing the results of the data based on target sector(s) of the product developed by the companies, we have Figure 3 results.
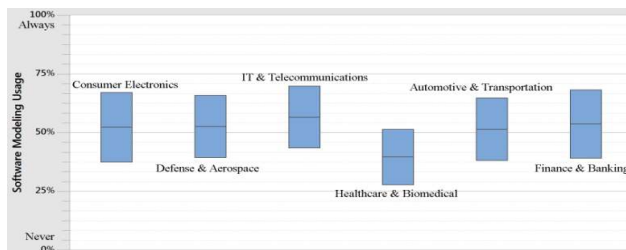

Figure 3. *Software modeling usage based on sectors*

Accordingly, "Healthcare & Biomedical" sector is using software modeling the least, whose overall software modeling ratio is at "Sometimes (<50%)" level. The other sectors are at "Often (>=50%)" level usage category.

Notice that any informal usage of modeling is seen as "modeling usage" in the survey and Q14 tried to understand the modeling language that participant use, if any. Since this was again a multiple-response question, multiple items could be recorded (i.e., participants might use both UML and DSL). The majority of participants (77%) use UML (not surprisingly), but it is interesting that the second most frequently selected response is "Sketch/No formal modeling language". Another interesting result is that some respondents chose both UML and "Sketch/No formal modeling language", which show that these participants use modeling both formally and informally as in [1] depending on their purposes (See details in [10]).

When we analyzed the choices of modeling languages versus sectors, we see Table II results. (Notice that rows are ordered according to the most selected responses in the survey. i.e., the most selected response in the survey is at 1st row, the least selected one is at last row).

TABLE II. CHOICES OF MODELING LANGUAGES VERSUS SECTORS

	Consumer Electronics	Defense & Aerospace	IT & Telecommunications	Healthcare & Biomedical	Automotive & Transportation	Finance & Banking
UML	84%	77%	69%	67%	71%	79%
Sketch/No formal modeling language	76%	55%	63%	57%	59%	76%
DSL	33%	36%	32%	34%	47%	40%
UML extensions (profiles)	13%	28%	17%	13%	21%	15%
SysML	10%	25%	19%	15%	12%	15%
MATLAB	7%	23%	14%	5%	15%	8%
BPML	9%	7%	7%	8%	9%	7%
SoaML	7%	6%	10%	9%	7%	9%
AUTOSAR	0%	1%	0%	0%	15%	0%
EMF	2%	1%	1%	2%	0%	2%
EAST-ADL	0%	0%	0%	0%	3%	0%
Markov Chain Markup Language	2%	0%	0%	0%	0%	0%
AADL	0%	4%	0%	0%	0%	0%
Modelica	0%	1%	1%	0%	0%	0%

The dominant modeling language is UML in all sectors; however, there are some interesting results based on the sectors. The usage of "Sketch/No formal modeling language" is very similar to UML usage in "Finance & Banking". DSL is mostly used in "Automotive & Transportation", where AUTOSAR (AUTomotive Open System ARchitecture) usage is ~15% although it was not in the pre-given answer set. Besides, specific modeling language for target sectors (i.e. AADL (Architecture Analysis & Design Language) for "Defense & Aerospace", EAST-ADL for "Automotive &

Transportation" and Markov Chain Markup Language for "Consumer Electronics") are also interesting results.

As another question, participants were asked about the diagram types used while modeling via the 5-point Likert scale. According to the most used diagram types are sequence diagrams, state-machine diagrams, and class diagrams [10]. When we analyze this result based on target sector(s) of the

product developed by the companies, we see Figure 4. The 'x' axis in Figure 4 corresponds to the diagram usage *(from "Never (0%) to Always (100%))*, whereas the 'y' axis corresponds the diagram types (The most used is "Sequence Diagram", the least used is "EPC" [10]). Moreover, the green boxes in Figure 4 indicate the most used; whereas the red ones indicate the least used diagram types in each sector category.
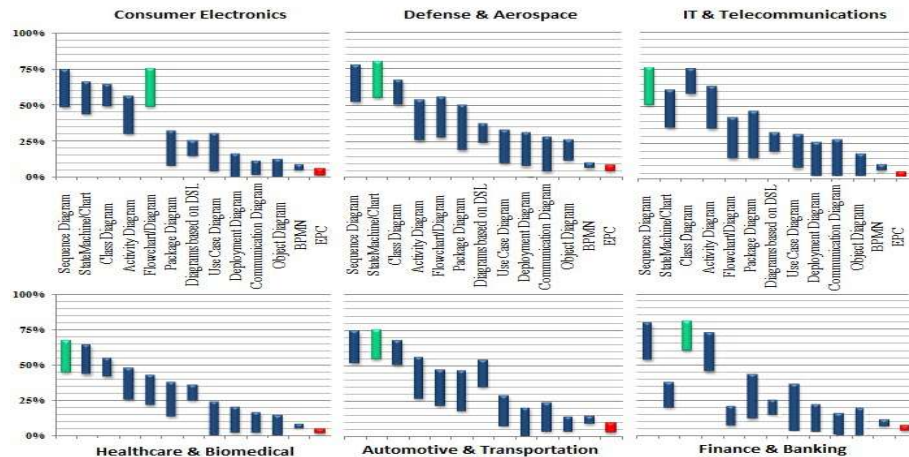

Figure 4. *Diagram types usage versus sectors*

The most used diagram type according to the survey result *(i.e., Sequence Diagram, which is left-most side)* is also the most used diagram for only two sectors *(i.e., "IT & Telecommunications" and "Healthcare & Biomedical)*; the other sectors have different most frequently used diagram types (e.g., for "Consumer Electronics" is "Flowchart/Diagram" or for "Defense & Aerospace" is "State Machine/Chart"). This means that each sector has different level of diagram usage, but in common, they mostly use "Sequence Diagram" although their most frequently used diagramming technique might be different. Moreover, if we consider the similarity of activity diagrams and flowcharts, we can probably conclude that it is one of the most frequently used techniques.

At the beginning of 19th question, we gave the terminology, which clearly explained the difference between model-based and model-driven techniques (See [9]) so that participants could consistently answer Q19, which investigates how often the participants use MDE. Results are given in [10]. When we analyze this result based on target sector(s) of the product, we observe the results in Figure 5.
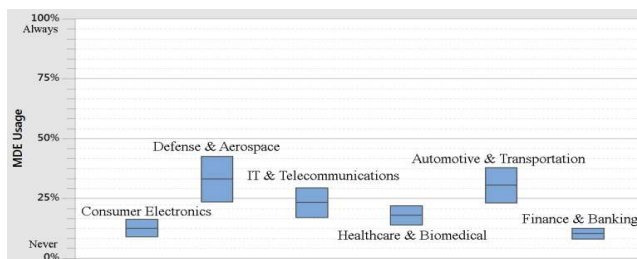

Figure 5. *MDE usage ratio based on sectors*

Accordingly, all sectors under 50% usage, which is at "Sometimes (<50%)" level. According to results, "Finance & Banking", whose "Never" usage percentage is at the top, is the least model-driven user sector. One of the interesting results here although "Consumer Electronics" might be probably considered as one of the sectors where innovation and time to market drives the business, MDE usage ratio is between 9%-17%. MDE is a technique established to support these values at most; but it might be important to analyze what and where is the problem in this sector although its software modeling usage ratio is high. This means that the participants in this sector use model-based or sketch/no formal modeling approaches, but what are the specific consumer electronics challenges or bad experiences on MDE, which resulted such a situation? It might be a good future work to investigate it. On the other hand, "Defense & Aerospace" sector is the one, which uses MDE at most, whose MDE usage ratio is between 24%-43% as shown in Figure 5. Perhaps, the project length and necessary investigation on MDE (its corresponding costs, i.e., tool, training, etc.) might be suitable for this sector.

IV. THREATS TO VALIDITY IN OUR SURVEY

In this section, we discuss the possible validity concerns in our survey since this cross-factor results also depend on these threats, which we minimized or mitigated.

Construct validities are concerned with the extent to which the objects of study truly represents theory behind the study [12]. We collected data from different sources (e.g, different countries, industrial sectors, etc.) in order to avoid mono-operation bias. It is also common for people to deflect their answers when they feel being evaluated and based on what

they think is the intended result of a study. To mitigate these, we informed participants prior to the survey that our motive in this study was to take a snapshot of the industry and that we will not collect any identifying information so that participants will remain anonymous.

We tried to reduce the threat related with definitions of MDE vs. MBE by making sure the participants distinguished these terminologies. In order to prevent any misunderstanding and potential threat in this terminology, we conducted a pilot study in which several practitioners filled the survey and we met with them to assess their common understanding of these terminologies. However, the definition provided by Brambilla et al. [3] unfortunately still leave room for subjectivity. Thus, this issue stays as a potential threat, e.g., a given practitioner might in fact use MBE, even though s/he stated to use MDE.

Internal validity reflects whether all causal relations are studied or if unknown factors affect the results [12]. Instrumentation was improved by using a pilot study. This reduces the likelihood for learning effects and, therefore, maturation effects.

External validity is concerned with the extent to which the results of this study can be generalized [12]. In order to decrease the effect of possible dominant participant number in a specific sector due to authors' work experiences' network (i.e., defense & aerospace, consumer electronics, academia), the survey has been distributed to embedded software professionals via various social network sites in all around the world. Therefore, we have done our best to reach the subjects with a variety of different backgrounds for the embedded software industry. Our sample size is quite high compared to previous surveys. While we did our best to achieve an even geographical distribution, the samples were mostly based from Europe (66%), followed by Asia (17%) and then the Americas (14%) (See [10]). Nevertheless, note that we used non-probabilistic sampling design and external validity is limited.

Conclusion validity of a study deals with whether correct conclusions are reached through rigorous and repeatable treatment [12]. This study was designed by one author, who has both researcher and practitioner hat and two other researchers from two different institutions; therefore the risk for "fishing" on the results is reduced.

V. CONCLUSION

With the help of this study, the state-of-the-practice of software modeling was better understood. It is interesting to see how embedded software professionals within different embedded target sectors have different software modeling usage and practices. Some modeling languages or diagrams are specific to some sectors or their usage ratio is different depending on their needs and challenges. We believe that both embedded software professionals and also researchers benefited from our results and this will hopefully encourage more academia-industry collaborations in this area.

As a future work, we aim at conducting semi-structured in-depth interviewing to capture some detailed, rich contextual description. We believe that after getting quantitative results, these interviews can be used for both descriptive, exploratory and then for an improvement purpose. Then, based on the findings of this study and our previous Action Research (AR) project [13] as well as others incorporating different classifications about software modeling, we plan to construct a conceptual model of development and usage for software modeling. This conceptual model will attempt to clarify the meaning of various, usually ambiguous terms, and ensure that problems with different interpretations of the concepts cannot occur in software modeling context.

REFERENCES

[1] M. Petre, "UML in practice," in *35th International Conference on Software Engineering (ICSE)*, 2013, pp. 722-731.

[2] W. J. Dzidek, E. Arisholm, and L. C. Briand, "A Realistic Empirical Evaluation of the Costs and Benefits of UML in Software Maintenance," *IEEE Transactions on Software Engineering,* vol. 34, pp. 407-432, 2008.

[3] M. Brambilla, J. Cabot, and M. Wimmer, "Model-driven software engineering in practice," *Synthesis Lectures on Software Engineering,* vol. 1, 2012.

[4] J. Cabot. (2009). *Relationship between MDA,MDD and MDE*. Available: http://modeling-languages.com/relationship-between-mdamdd-and-mde/

[5] M. Broy, S. Kirstan, H. Krcmar, and B. Schätz, "What is the benefit of a model-based design of embedded software systems in the car industry?," in *Emerging Technologies for the Evolution and Maintenance of Software Models*, ed, 2011, pp. 343-369.

[6] L. T. W. Agner, I. W. Soares, P. C. Stadzisz, and J. M. Simão, "A Brazilian survey on UML and model-driven practices for embedded software development," *Journal of Systems and Software,* vol. 86, pp. 997-1005, 2013.

[7] G. Liebel, N. Marko, M. Tichy, A. Leitner, and J. Hansson, "Assessing the State-of-Practice of Model-Based Engineering in the Embedded Systems Domain," in *Model-Driven Engineering Languages and Systems*. vol. 8767, ed: Springer International Publishing, 2014, pp. 166-182.

[8] V. C. Basili, G.; Rombach, D.H., "The Goal Question Metric Approach," in *Encyclopedia of Software Engineering*, ed: Wiley, 1994.

[9] D. Akdur, V. Garousi, and O. Demirörs, "MDE in embedded SW industry-Survey Form (Questions)," *https://dx.doi.org/10.6084/m9.figshare.4262978,* 2015, Last accessed: Nov. 27, 2016.

[10] D. Akdur, V. Garousi, and O. Demirörs, "MDE in embedded software industry, Technical Report," *METU II-TR-2015-55, https://dx.doi.org/10.6084/m9.figshare.4262990,* 2015, Last accessed: Nov. 27, 2016.

[11] D. Akdur, V. Garousi, and O. Demirörs. (2015). *Survey on Software Modeling in Embedded Systems Engineering*. Available: https://drive.google.com/file/d/0BzPI4c-GGTgoeGxwcWtfaVhNZjg/view

[12] C. Wohlin, P. Runeson, M. Höst, M. C. Ohlsson, B. Regnell, and A. Wesslén, *Experimentation in Software Engineering*: Springer Berlin Heidelberg, 2012.

[13] D. Akdur and V. Garousi, "Model-Driven Engineering in Support of Development, Test and Maintenance of Communication Middleware: An Industrial Case-Study," in *International Conference on Model-Driven Engineering and Software Development (MODELSWARD)*, 2015.