International Journal of Research in Engineering and Science (IJRES)
ISSN (Online): 2320-9364, ISSN (Print): 2320-9356
www.ijres.org Volume 5 Issue 2 ǁ Feb. 2017 ǁ PP. 96-101

Prototyping Embedded Systems for Model-Driven Approach

THARABAI C. V.
Lecturer in Electronics
Women's Polytechnic College Ernakulam, Kochi, Kerala.

Abstract:
The complexity of developing an embedded system (ES) is increasing. This increase has several cumulative sources: some are directly related to ES constraints (reliability, compute intensive, resource constraints), while others are related to the industrial context in which they are developed (fast prototyping, early validation, parallelization of developments). Although several Model-Driven Engineering (MDE) processes for ES development have been proposed, the majority of them are not fully formalised. This has several disadvantages that prevent them from being used in prototyping where iterations must be brief and focused. In situations where quick results are expected with little effort, incomplete formalised processes are often avoided. Model-Based Development (MBD), when combined with Model-Based Validation (MBV), can help identify problems early and thus reduce rework costs. The use of tests based on designed models not only allows for the early detection of defects, but also for continuous quality assurance. Testing can begin as early as the first development iteration.
Keywords:
Embedded Systems, Model Driven Engineering, Model-Driven Approach, Model Driven Development, Model Driven Testing

I. Introduction:

As the demand for compact devices grows, the sizes of processors and microchips continue to shrink, necessitating the development of complex control systems. To optimise the overall system design, the entire embedded control system and application design processes must be monitored. In this case, the model-based design approach demonstrates to be an effective and efficient method of comprehending product components such as commercial microcontrollers and processors, as well as algorithms and code for the operation of both microelectronic and embedded devices. Through visual prototyping and model simulation, it assists in addressing various difficulties and complexities that arise during the lifecycle of embedded application software. [1]

With the increasing complexity of Embedded System (ES) development, an increasing number of developers are turning to Model-Driven Engineering (MDE) to help with ES design [2]. MDE encourages the use of models to abstract all or a portion of the system under consideration. It enables the visualisation of ES from various perspectives and the extraction of essential properties for its modelling. Furthermore, MDE provides techniques for performing static analyses, performance analyses, simulations, and other tasks from executable models. It also allows you to separate design from implementation, which is especially useful when designing ES that consists of an application hosted by a platform. The need for early error detection and regular feedback to ensure that the system is conforming to its specifications adds to the complexity of ES design. System prototyping is one of the existing techniques for ensuring specification validation [3]. Prototypes allow for the assessment of specification fulfilment while also providing a concrete view of the system under development and the design process [4]. However, prototyping may be incompatible with model-based methodologies, where time spent creating models at various abstraction levels may be significant. [5]

Model-based design (MBD) performs verification and validation through simulation testing. It covers a wide range of disciplines, functional behaviour, and cost/performance optimisation in order to deploy a product from the initial concept of design to final validation and verification testing. Although many organisations use some form of modelling, some use simulation on an ad hoc basis, missing out on potential verification benefits. Nowadays, embedded systems shape our world. It's difficult to imagine our daily lives without it. Cell phones, home appliances, energy generators, satellites, automotive components, and so on are all examples. It becomes even more complicated when there are real-time and interface constraints. [6]

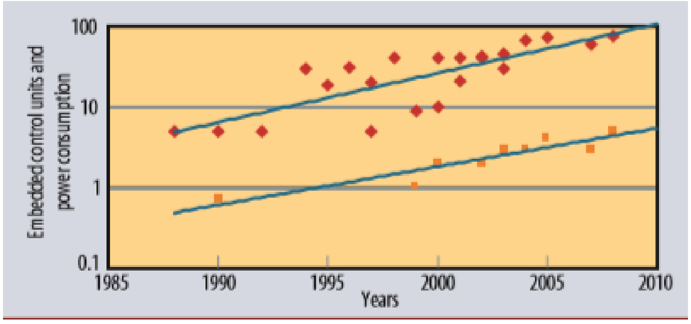


Figure 1: Automotive embedded systems

Embedded software is not a product in and of itself. It is a single component in a product that includes mechanics, electronics, and software. As a result, when developing embedded systems, various constraints must be considered. Embedded software developers should take resource constraints and environmental conditions into account when creating product-specific platforms. The need to develop specialised hardware and software platforms distinguishes IT systems from embedded systems.

Because embedded software is highly specific to its environment, software engineers face extraordinary challenges. The main source of complexity is the large number of interactions between the various system components, which share the following characteristics:

- Functionality represented by states and events
- Real-Time behavior
- Combined hardware/software systems
- High demand on availability safety, security, and interoperability
- Long lived systems in which embedded software is expected to work reliably.

Model-based design is a recommended approach for embedded hardware design firms because of the following benefits: [7]

- Allow teams to validate the overall system design specification in a shared simulation environment.
- Ease of controller design positioning on PLC hardware.
- Opens up new avenues for developing low-power controller products.

II. Review of Literature:

MDE is regarded as one of the most popular approaches in software abstraction, according to various sources (e.g., [8]). MDE helps software engineers manage the complexities of embedded software development by automating Software Development Life Cycle (SDLC) artefacts not only in implementation [9], but also in testing and documentation, by abstracting out details. There are numerous books, for example, [10], numerous conferences, and a vast body of knowledge in the application of MDE. Furthermore, because economic factors such as time-to-market necessitate a dependable development process that allows for rapid SDLC, many practitioners in various domains (e.g., consumer electronics, defence and aerospace, automotive, and telecommunication) have begun to adopt MDE. Specifically, several studies emphasise the importance of MDE in the embedded world to mitigate the effects of platform heterogeneity and complexity, in addition to validation and verification [11].

The study in [12] was a 2011 global survey of 67 participants that investigated the reasons for introducing model-based development in a single sub-domain of embedded systems (i.e., the automotive industry) while accounting for the costs and benefits. It concentrated solely on the "development" phase of the entire "engineering" (MBE) process (e.g., model-based development (MBD)). The following were the study's main findings: (1) The top three motivations for model-based development are "improvement of product quality," "development of complex functions," and "shorter development times." (2) Positive MBD experiences include "communication with other colleagues," "possibility of early simulation of the functional model," and "easier maintenance if the generated code is not changed manually," while negative MBD experiences include "high process of redesign costs" and "tool costs." and (3) MBD can save money, but only with a "well-chosen" approach and an established development process with defined interfaces and roles. Otherwise, MBD can be significantly more expensive than manual software development.

Objectives:

1. Study on Automotive embedded systems
2. Define Model-Driven Approach and Development
3. Describe the hHOEi2 Approach.

III. Research Methodology:

Survey methodology is a well-established technique for obtaining a broad characterization of a specific issue by allowing the collection of various data such as opinions, perceptions, attitudes, and behaviours [13]. It has been used in a variety of fields. Surveying is an appropriate strategy for gathering empirical data from large populations.

IV. Result and Discussion:

Model-Driven Approach

Model Driven Engineering's central tenet is that "Everything is a Model." In its broadest sense, modelling is the cost-effective use of something in place of something else. It is a simplified, less expensive, and safer version of reality. It assists in avoiding the complexities and irreversibility of reality. Models are used in a variety of engineering domains, but they have only recently played a role in the development of embedded systems.

THE hHOEi^2 APPROACH

The hHOEi2 approach is presented in this section. The acronym hHOEi2 stands for Highly Heterogeneous Object-Oriented Efficient Engineering and refers to a model-based approach first proposed in [14]. hHOEi^2 incorporates a collaborative process as well as a common language for application and platform development that is based on useful concepts such as objects, associations, state machines, and message passing. Figure 1 depicts a portion of the hHOEi2 process. The hHOEi2 process is platform-based design, as the platform design flow includes the same four phases as the application design flow. This feature enables designers to iterate across multiple stacked platforms, lending the process its fractal nature [15].

Model Driven Development

Model-driven engineering (MDE) is replacing manual programming in software development.One of the most difficult challenges is managing the growing complexity of embedded software development while maintaining product quality, reducing time to market, and lowering development costs.

Model Driven Development (MDD), like any other development activity, should be guided by a process. It is extremely difficult to cover the entire development life cycle using MDE due to a lack of tool support. Quality assurance is critical in such development, particularly in safety-related systems.

Software quality can be ensured by using validation and verification techniques prior to code generation. Researchers are looking for modelling methods to specify, analyse, and verify embedded systems in a fast and accurate manner in order to improve the automation of the design from the initial requirement specification to the final system. To be reliable and effective, a model must clearly communicate its purpose and be simple to understand and develop. CASE (Computer Aided Software Engineering) tools arose as an attempt to use models to increase abstraction and automate development tasks. They provided graphical representations of requirements specifications and generated code from them automatically. To deal with the pressure to reduce time-to-market and the increasing design difficulties, new research efforts and approaches are required. MDE is the current approach for increasing design abstraction and improving model portability, interoperability, maintainability, and reusability. By creating, maintaining, and manipulating models, Model Driven Engineering aims to increase productivity and facilitate system development. A system is built by refining models, beginning with higher levels of abstraction and progressing to lower levels until code is generated. A MDE approach can imply multiple developers, modelling languages, and tools modelling the same system. [16-17]

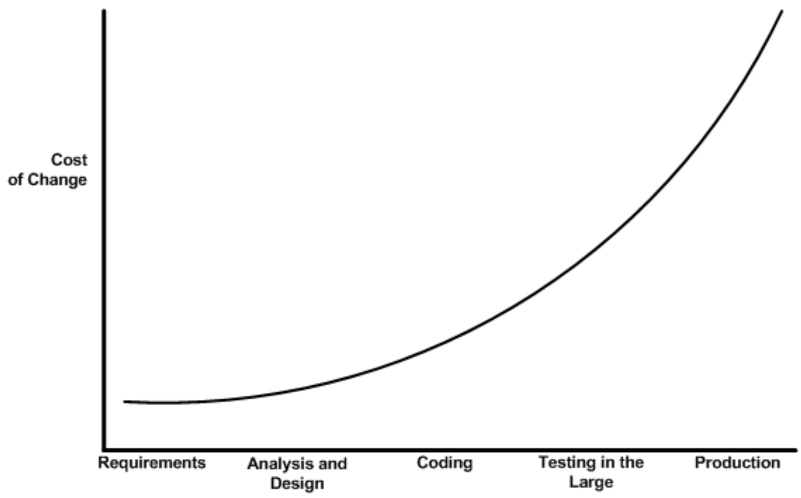


Figure 3 - Phases of SDLC vs. Cost of Bug Fixing

Model Driven Development, when combined with Model Driven Validation, will provide information before the system is implemented. It aids in the early detection of problems, lowering the cost of rework. Models, on the other hand, are typically generated manually, which takes time and effort. Because of the close hardware interaction, late defect correction costs much more in embedded systems than in other software types. The clear focus is thus on detecting defects as early as possible. Best would be during the insertion phase. [18]

Proposed Approach

A method for deploying MBV for embedded systems without affecting their real-time behaviour is underutilised. Our strategy is to provide a tool chain to aid in MBV automation, particularly during MIL. [19]

The steps in our proposed approach are as follows:

1- Model-Based Development: A model is created based on the requirements during this phase. A general overview of the system can be represented with class diagrams, and the detailed behaviour can be represented with state charts, which include states and transitions between them.

2- Model-based validation: As the first step in the MBV approach, a test model is created based on the model obtained in step 1. The test model in our approach is a series of statecharts, each representing one test scenario to be executed on the original model. Each test scenario is made up of a set of inputs and expected outputs.

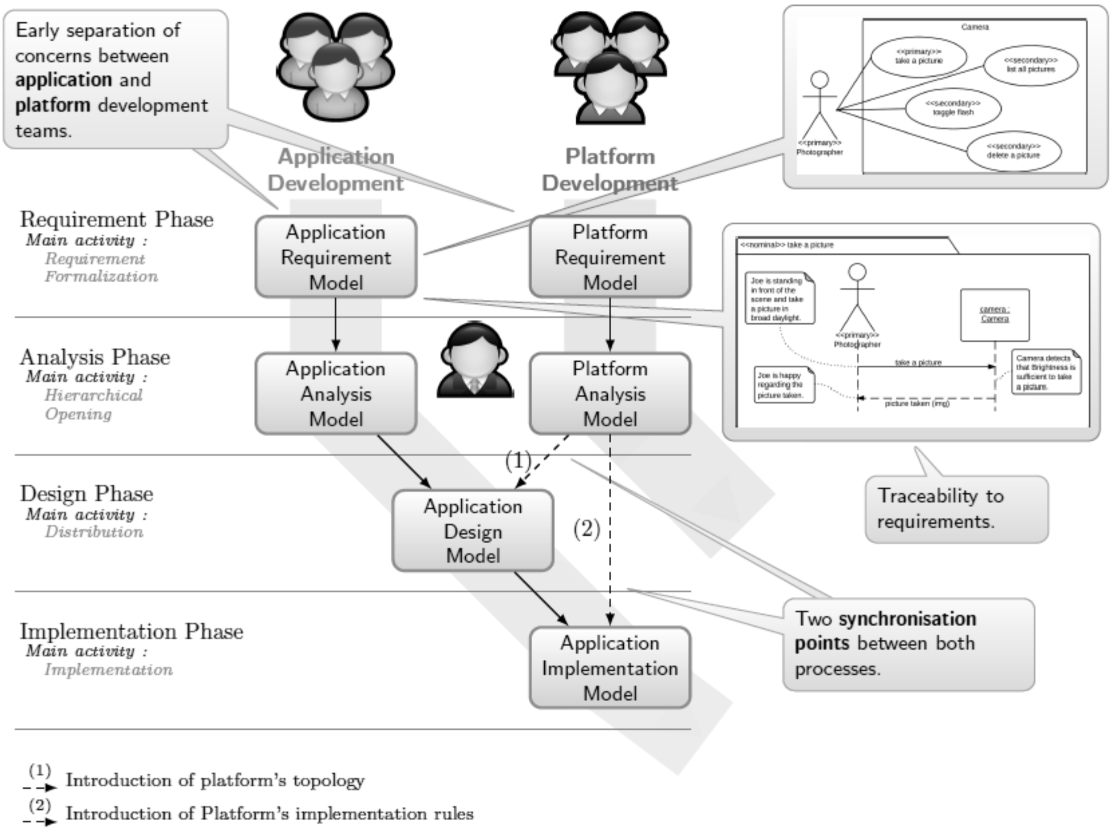
As shown in the table below, the model-based approach eliminates several time-consuming tasks from the traditional approach. It is critical to take a more time-efficient approach when developing embedded systems. The table below provides a brief comparison of the proposed approach and the traditional development workflow in terms of design, coding, and testing. Both approaches use the same requirements specification. [20]

	Model Based Development and Testing flow	Classical development and testing flow
Design	Not needed, as requirements are modeled, and code will be generated automatically	Effort of creating design document is required
Coding	Code auto generated, review and analysis efforts are minimal	Effort of source code implementation, review, and analysis is required
Testing	Only modeling effort is needed. Testing is done on the model level using proposed tool chain. Same test scenarios can be then used on generated code and on the hardware.	Effort of creating test documents, scenarios, and implementation is needed at every stage.

Table 1 - proposed vs classical flow

Analysis Phase

During the first phase, requirements are defined for a black box system (an application for the first flow and a platform for the second). This system is opened and detailed in terms of communicating objects and object behaviours during the analysis phase. State machines are used to capture behaviours. Both development teams can carry out the hierarchical opening activity at the same time. Each opening activity aims to meet the requirement formalised in the preceding requirement phase. [21-22]



hHOEi^2: a Collaborative Top-Down Process for Embedded System Design

V. Conclusion:

We propose an MDE-based process for ES development in this paper. Throughout the development process, this process precisely defines the development tasks and their impact on the models. We define iterations width and depth for the process in particular to allow for fine-grained and consistent development planning. The process's short and well-defined iterations bridge the gap between rapid prototyping, ad hoc methods, and regular development processes. In addition, we describe the hHOEi2 process and its formalisation language.

References:

[1]. M. Broy, "The 'Grand Challenge' in Informatics: Engineering Software- Intensive Systems," Computer, vol. 39, no. 10, pp. 72–80, Oct. 2006.

[2]. "Seamless model driven systems engineering based on formal models," in Formal Methods and Software Engineering, ser. Lecture Notes in Computer Science, K. Breitman and A. Cavalcanti, Eds. Springer Berlin Heidelberg, Dec. 2009, vol. 5885, pp. 1–19.

[3]. F. Kordon and Luqi, "An introduction to rapid system prototyping," Software Engineering, IEEE Transactions on, vol. 28, no. 9, pp. 817– 821, Sep. 2002.

[4]. P. Tessier, S. Gerard, C. Mraidha, F. Terrier, and J. M. Geib, "A component-based methodology for embedded system prototyping," in Rapid Systems Prototyping, 2003. Proceedings. 14th IEEE International Workshop on, Jun. 2003, pp. 9–15.

[5]. J. Hugues, M. Perrotin, and T. Tsiodras, "Using mde for the rapid prototyping of space critical systems," in 19th IEEE/IFIP International Symposium on Rapid System Prototyping, Jun. 2008, pp. 10–16.

[6]. M. -L Boukhanoufa, A. Radermacher, F. Terrier, "Towards a Model-Driven Engineering approach for developing adaptive real-time embedded systems", Pages: 261-266, NOTERE 2010.

[7]. P. Liggesmeyer, M. Trapp, "Trends in Embedded Software Engineering", Volume: 26 , Issue: 3, Pages: 19-25, 2009

[8]. J. Hutchinson, J. Whittle, and M. Rouncefield, "Model-driven engineering practices in industry: Social, organizational and managerial factors that lead to success or failure," Science of Computer Programming, vol. 89, Part B, pp. 144-161, 2014.

[9]. P. Liggesmeyer and M. Trapp, "Trends in Embedded Software Engineering," Software, IEEE, vol. 26, pp. 19-25, 2009.

[10]. R. France and B. Rumpe, "Model-driven Development of Complex Software: A Research Roadmap," presented at the Future of Software Engineering, 2007.

[11]. B. P. Douglass, Real Time UML: Advances in the UML for Real-time Systems: Addison-Wesley, 2004.

[12]. 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.

[13]. F. Shull, J. Singer, and D. I. K. Sjoberg, Guide to Advanced Empirical Software Engineering: Springer-Verlag New York, Inc., 2007.

[14]. N. Hili, C. Fabre, S. Dupuy-Chessa, and S. Malfoy, "Efficient Embedded System Development: A Workbench for an Integrated Methodology,"in ERTS2 2012, Toulouse, France, Feb. 2012.

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

[16]. F. Ciccozzi, "Model-Driven Engineering of Embedded Real Time Systems", Research Planning Course 2009/2010
[17]. C. Shih, C. T. Wu, C. Y. Lin, H. P. A. Hsiung, N. L. Hsueh, C. H. Chang, C. S. Koong, W. C. Chu, "A Model Driven multicore software development environment for embedded systems", Volume: 2, COMPSAC 2009
[18]. H. Kashif, M. Mostafa, H. Shokry, S. Hammad, "Model-Based Embedded Software Development Flow", IDT 2009
[19]. L. Tan, J. Kim, O. Sokolsky, I. Lee, "Model-based Testing and Monitoring for Hybrid Embedded Systems", IRI 2004
[20]. B. Boehm, V. R. Basili, "Software defect reduction, top 10 list", vol. 34, no. 1, pp. 135-137, January 2001.
[21]. P. Tessier, S. Gerard, C. Mraidha, F. Terrier, and J. M. Geib, "A component-based methodology for embedded system prototyping," in Rapid Systems Prototyping, 2003. Proceedings. 14th IEEE International Workshop on, Jun. 2003, pp. 9–15.
[22]. C. Bunse, H.-G. Gross, Peper, Christian, "Applying a model based approach for embedded systems development", EUROMICRO 2007