What Information on Business Parameters is Required by Embedded Software Developers to do an Effective Job?

Joakim Fröberg, Stefan Cedergren, Stig Larsson

Department of Innovation, Design and Product Development
Mälardalen University, Sweden
{joakim.froberg, stefan.cedergren, stig.larsson}@mdh.se

Abstract. Embedded software design is tightly connected to the functionality and goals of the system it is used to control. For mechatronic systems such as an in-vehicle automotive system, software developers require information on the system goals including business parameters to effectively decide on architecture and functionality. This paper presents results from an case of developing a hybrid electric drive system platform, and presents the information areas that software and system engineers do perceive as important to effectively perform design. We note that business parameters are sought for and elaborate on what information is required. We analyze what these needs are and elaborate on how to address them by using methods from the literature. We conclude that the effort of developing embedded software cannot rely on statically specified business parameters; rather these would be estimated and refined by interaction throughout the development cycle.

Keywords: Embedded software, Software architecture, Pre-Study, Product-line

1 Introduction

Embedded software is an important part of a mechatronic system such as an automotive sub-system. It is state of practice for automotive suppliers to provide sub-systems such as transmissions, engines, or brakes for many customers by using a configurable platform product. The flexibility of software is utilized to achieve variability to reuse complex sub-systems in different applications, e.g. many vehicle models. For a hybrid electric drive system most parts of the mechatronic system is controlled by embedded software and the structure and functionality of the software is tightly intertwined with the functional architecture of the system. For software developers and architects, it is important to understand the many goals and uses of the platform in order to effectively develop the software system.

In developing any complex system, the pre-study phase is important to establish parameters such as system requirements, usage context, constraints, and optimization criteria. We have performed interviews with system developers of a hybrid electric drive system platform and elicited the information needs those practitioners consider important to succeed in developing such a system. From the results we have analyzed the challenges related to software development of a platform that enables variability, customization to accommodate a product line of drive systems.

1.1 A development pre-study of an automotive sub-system platform

For a platform or product line initiative, it is critical [1] to define the requirements and content of the reusable assets and to setup the development structure to achieve goals in reuse, costs, and time-to market. It is known to be an engineering problem to perform early phases of product definitions and requirements definition, but guidance and techniques are available in textbooks e.g. [4], [6].

The life-cycle of a platform product is different from that of a 'regular' product. The development of reusable assets may involve different customers, and often company strategy, complex configuration management and quality management is involved. The value and cost of platform assets are often more complex to estimate and can depend on projected production volumes, forthcoming customers, strategy, and other dynamic factors. When designing a platform, information on how to relate to the above-mentioned sources of complexity needs to be addressed.

1.2 Case characteristics

The case company has previously developed a customizable hybrid electric drive system intended for only a few similar heavy automotive applications. The goal of the development effort is now to offer customizable hybrid electric drive systems to a large number of customers with a wider scope of applications. Embedded software is involved in all parts of the system as well as in all system wide functionality.

The developed drive system shall accommodate the following functionalities: Re-generation of motion energy, optimized performance of motion actuators, optimized combustion engine control, productivity enhancement. The increased ability to control electric components compared to conventional automotive components provide the possibility to develop new functions that improve or optimize performance. The customers' willingness to buy and introduce such a drive system is heavily dependent on what fuel effectiveness and performance can be achieved for their particular applications. The resulting software system is layered in a structure from low-level control up to higher levels of decision-making and strategic functionality.

The drive system is intended for use in many, and yet to some extent unknown vehicles involve problems of flexibility and system boundary. The design needs variability and flexibility enough to accommodate vehicles with radically different architectures, system decomposition and design philosophy. Examples of design solutions that may well differ is paradigms for fault handling, diagnostics, and modes of operation. Design of an automotive subsystem will involve these types of complexity.

Developing a drive system platform product is to be performed at the same time and coordinated with development of several drive systems for specific applications. Platform development is inter-twined with each individual development project and provides and receives information and assets. As the development progress, more automotive applications will be considered. It is not fully known which applications are to be developed and thus needs support from the platform.

1.3 Objectives of study

The goal of this study is to describe the information needs in terms of business parameters in a case of developing a software platform for a complex automotive system; for what purpose and why do developers need business parameters in the development effort. Moreover, the objective is to identify known methods that can be used or could be adapted to tackle the information needs in such a pre-study phase.

Previously, we did a study to propose a method to find critical information in a pre-study phase for developing a complex automotive platform [9]. Part of the result was a listing of information areas that practitioners deemed critical to be able to design the system in an effective way. Strikingly, two out of five information needs expressed by developers were related to business parameters. In this paper, we use the results from our previous study, and describe and analyze the need for business parameters for embedded software developers involved in complex mechatronic automotive systems.

The method to perform the study was based on interviews. First we examined the development effort by performing exploratory interviews with eight line managers and project managers. The interviews were structured as open discussion around what challenges are faced and what the system is to do. The answer were documented and compiled into a problem formulation document that was reviewed by the interviewees. Secondly, we analyzed the problem formulation and listed the areas where some piece of information is missing in order to effectively develop the platform. Information needs were identified in five information areas around which practitioners expressed some level of concern. We elaborate on the problem of eliciting or estimating the information by studying related literature.

2 Challenges and information needs

The study gave us a listing of challenges as perceived by the interviewees in the case. We compiled all challenges that were given into six different categories.

Challenge 1. Eliciting all the complex requirements of any given heavy-vehicle application. An automotive product can be used in a number of different usage scenarios e.g. transporting goods on flat ground, or lifting material in a mine.

Challenge 2. Methods and process guidance for coordinated development of multiple drive systems. The method to systematically manage the development effort is reportedly a challenge. The overall idea is to reuse assets from a common drive system platform. The decisions on what and how to develop reusable assets must be kept on a strategic level outside the individual development projects.

Challenge 3. Deciding content and system boundary of the platform. A specific problem is the problem of deciding the scope and boundary of the drive system platform. What should be developed customizable to allow reuse and adapting to many applications, and what should be developed specifically in each case?

Challenge 4. Organizing people. A model for deciding on organization seems needed.
Challenge 5. Methods to estimate scope of applications. Many vehicles are imaginable as recipients of a hybrid drive system. Knowing what application will be targeted would enable selection of systems concepts. It is expressed that it would be difficult to take design decisions without these facts as optimization criteria.
Challenge 6. Methods to estimate business criteria. The idea with reuse is to minimize development effort for subsequent uses of the same technology. Interviewees express that the criteria for how much development effort is allowed, in order to be considered business worthy must be made known in order to plan and execute platform development.

We analyzed each of the six challenges and noted what information seems needed to enable the developers to do their job effectively according to their statements. We classified the information needs into five categories:

Info need 1. System content/boundary. In order to proceed with development of a platform hybrid electric drive system, information on the functional boundary of the platform is needed. Functional content and functional interface is requested.
Info need 2. Application scope. Information on what applications are to be developed and supported by the platform in the future. This can be based on the current customer base and company strategy.
Info need 3. Business model. A business model describes the mode of performing business; what is to be sold; who is the customer. For instance it is possible to sell engineering hours, components, or up-time. Technical decisions are made, based on assumptions around what business model is going to be employed.
Info need 4. Business criteria. Information on what constraints are imposed by the business context. There is a limit to what costs and development times can be accepted in order to meet business criteria. This information is dependent on predicted production volumes, investments, level of reuse etc. These business criteria parameters are likely dynamic over time.
Info need 5. Roles & responsibility. Information is needed on roles and responsibilities involved in decision-making in the development effort.

2.1 Discussion and related work

We note that two out of five information needs that practitioners point out are related to business parameters. We want to elaborate the need for business parameters expressed by info need 3 and 4 above: Business model, and Business criteria.

The developers state that the business model and business criteria must be made known for them to select technical concept and corresponding software architecture and controls strategy. For example, the time and cost limits for development may disqualify expensive or time-consuming concepts. At the same time these business parameters and constraints are in fact dependent on the performance and architecture of the system. For example, a higher time and cost limit is viable if the resulting fuel efficiency is correspondingly better. We conclude that the developers demand to know business parameters as if they were static is not viable; instead the business parameters and design decisions need to be increasingly accurately specified in an interaction between the company's business and development functions. Business and technical concept must be refined in parallel and both need better methods to perform estimations of outcome. At the same time, changing business parameters is bound to cause uncertainty when developing the system, as changing requirements would do.

We look at the need for business parameters and elaborate on which information is viable to define or estimate.

Developers desire knowledge about business parameters, but at the same time, the same parameters seems unrealistic to know beforehand. So, how should either the business parameters or the technical design be done? If the range and scope of the vehicles to employ the drive system platform were known and static, there would still be the issue to decide what functional boundary of the platform is suitable. Boundary would be selected based on cost comparison between development for reuse and one-shot development. In a real case the functional scope of the product line, the system boundary, the production volume, are all dynamic parameters that are not necessarily predictable beforehand.

We would like to devise a method to elicit or estimate the information that is needed. What methods exist? System Engineering texts e.g. [3], [4], [5] propose in a too general way, how to find business context. Methods for scoping are described by both Clements and Northrop [1] and Bosch [2]. Clements and Northrop [1] offer a guide as how to perform scoping of the coming applications. In order to get a method for precise estimations it seems likely that a specific method be devised for this company. Company strategy and customers may affect the estimates.

Thornell [7] describes different modes of performing business, but there is no guidance as to how to select a business model given a set of case parameters. The explicit act of modeling [8] the business parameters may help communication. The problem seems to be complex and our interviewee's express somewhat different notions on how business parameters are understood. One solution to unify developer team notions would supposedly be to model business parameters.

In a technical note from SEI, Chastek et al. [10] describes how a production strategy for product lines connects the business strategy with a roadmap for how to develop (produce) core assets and products. However, the report shows a method and an example, and no case study using the method is presented. In the case studies done, e.g. [11] and [12], several of the information needs discussed above are addressed. However, the need for business information is neglected in these studies and need as we have seen in our study more attention. The BAPO framework, presented by van der Linden et al. [13], is a model describing different aspects of product lines including business properties. An example illustrating the use is presented [13], but no case. Future work will include applying the BAPO model to the described case and other case studies to understand if there are general conclusions that can be drawn.

3 Conclusion

First, we acknowledge that software development and the selection of software architecture are tightly connected to the goals and functionality of the mechatronic system. Late involvement of software design is bound to create a lacking estimate of engineering effort that is needed by the other roles; managers and engineers of other disciplines. When software is involved in controlling most or all parts of a system, the software functionality and architecture makes for the same type of considerations as the overall system design.

We describe the case and present what information needs was reported. Similar cases of developing embedded software for complex automotive subsystems may benefit from our guidance if the information needs seem similar. We have described some of the problems in choosing and estimating business parameters and the interaction that needs to take place between business and technical roles.

Practitioners in our case express a need for knowing business parameters in order to effectively perform software and system design. In planning and executing a development effort similar to our case, consideration should be made to business model, business criteria.

Elaborating on future work we provide pointers to some relevant literature that may aid in successfully selecting and estimating business parameters.

Recognizing that business parameters are critical in an early phase of developing a software system for control of complex mechatronics could be one key to leverage complexity in an advanced product line effort.

Acknowledgments. This work has been supported by Swedish Energy Agency.

References

1. Clements, P., Northrop, L.: Software Product Lines-Practices and Patterns, Addison Wesley (2001)
2. Bosch, J.: Staged Adoption of Software Product Families. In: Software Process Improvement and Practice, pp. 125--142 (2005)
3. Department of Defense. Systems Engineering Fundamentals. Defense Acquisition University Press (2001)
4. Sage, A.P.: Systems Engineering, John Wiley & Sons, Wiley Series in Systems Engineering (1992)
5. Kotonya, G., Sommerville, I.: Requirements Engineering, John Wiley & Sons, Worldwide Series in Computer Science (1998)
6. CMU/SEI. A Systems Engineering Capability Maturity ModelSM, SECMM-95-01, CMU/SEI-95-MM-003, www.sei.cmu.edu/reports/95mm003.pdf (1995)
7. Thornell, H.: Spetsföretag. Uppsala Publishing House. (2007)
8. Weilkiens, T.: System Engineering with SysML/UML – Modelling, Analysis, Design. The MK/OMG Press (2006)
9. Fröberg, J., Cedergren, S., Larsson, S.: Eliciting Critical Information in a Pre-Study Phase of Developing a Drive System Platform for Automotive Applications, Systems Engineering and Engineering Management for Sustainable Global Development (2011)
10. Chastek, G. J., Donohoe, P., McGregor, J. D.: Formulation of a Production Strategy for a Software Product Line, ed: Carnegie Mellon University (2009)
11. Strobl, S., Bernhart, M., Grechenig, T.: An experience report on the incremental adoption and evolution of an SPL in eHealth. ICSE Workshop on Product Line Approaches in Software Engineering, Cape Town, South Africa (2010)
12. Takebe, Y., Fukaya, N., Chikahisa, M., Hanawa, T., Shirai, O.: Experiences with software product line engineering in product development oriented organization. 13th International Software Product Line Conference, San Francisco, California (2009)
13. van der Linden, F., Bosch, J., Kamsties, E., Känsälä, K., Obbink, H.: Software Product Family Evaluation. SPLC (2004)