A process framework for embedded systems engineering

Sofia Charalampidou, Apostolos Ampatzoglou, Paris Avgeriou
Department of Mathematics and Computer Science, University of Groningen
Groningen, The Netherlands
s.charalampidou@rug.nl, a.ampatzoglou@rug.nl, paris@cs.rug.nl

***Abstract*— Engineering of embedded systems is considered highly complex, due to the need for integrating multi-site, multi-lifecycle, multi-disciplinary, and multi-organization approaches. However, such challenges have not been comprehensively addressed in existing engineering processes. To this end, this study proposes a process framework to tackle these challenges, focusing on its meta-model, i.e. a set of elements used for process instantiation. As an exploratory validation of the proposed meta-model we conducted two focus groups with experts on embedded systems, resulting in positive feedback and industrial acceptance.**

Keywords—process framework; embedded systems; focus group

I. Introduction

Embedded systems engineering deals with a number of highly complex challenges, which, until now, have not been sufficiently addressed by process support. Specifically, there is a need for integrating multi-disciplinary, multi-lifecycle, multi-site, and multi-organization approaches, due to domain-specific characteristics.

Although such aspects are discussed in the domain of systems engineering [7], the. proposed solutions do not comprehensively address all the aforementioned challenges.

In this paper we propose and validate a process framework for the domain of embedded systems, by taking into account the abovementioned domain-specific challenges. We note that the introduction of this framework only provides a starting point for addressing these challenges in the domain of the embedded systems. A process framework is a generic, model-based mechanism, for creating project-specific process instances. The proposed process framework, from now on referenced as *PROMES Process Framework*, has been developed in the context of the PROMES project (see http://promes-itea2.eu/). According to [1], the definition of a process framework consists of three parts: (a) a meta-model defining the fundamental types of process elements, and how they inter-relate, (b) usage guidelines on how to select, integrate, and customize process element instances from the repository to produce project-specific processes, and (c) a repository of process element instances (actual descriptions of each instance). In this paper we will only cover the first, due to size limitations.

To the best of our knowledge, there are no process frameworks in the literature for the domain of embedded systems. Thus, as related work we have considered the IEEE Standard [4] that provides guidelines on defining system lifecycle processes, and Rational Unified Process for Systems Engineering (RUP-SE) [10]. Both [4] and [10] have been used as the basis for the construction of the proposed meta- model..

The rest of the paper is organized as follows: in section II we present the proposed meta-model and in section III the initial validation that we conducted through two focus groups. Finally, in Section IV we discuss the contribution of this study and present our plans for future work.

II. PROMES Process Framework

In this section, we present the meta-model of the PROMES Process Framework (see Fig. 1), along with the elements that it involves. The elements of the PROMES process framework are described as follows:

Abstract Lifecycle: Corresponds to the life cycle of a project, which starts with the beginning of a project, continues while the set of activities defined in the process framework are under progress and ends when the project has come to an end [4, 10]. *Abstract Lifecycle* is an abstract element, which acts as an interface for *Composite Lifecycle* and *Lifecycle*.

Composite Lifecycle: Corresponds to a lifecycle that includes other *Lifecycles*. As example we consider a project lifecycle, which encapsulates a software and a hardware lifecycle.

Lifecycle: The lifecycle of this type is “simple”, since it does not include other lifecycles (e.g. the waterfall lifecycle that consists of five *Development Phases*, i.e. requirements, design, implementation, verification and maintenance).

Development Phase: It is a subdivision of the *Lifecycle*, during which a unique major goal has to be fulfilled and a number of *milestones* are set to ensure its achievement [6]. An example of a development phase, borrowed by RUP, is the “inception phase”. Each *Development Phase* is connected to one *Development Activity*.

Development Activity: A development activity, either primitive or composite is related to one or more *Development Phases*. In order to complete the activity, some *input artifacts* are required and some *output artifacts* are generated [6]. A development activity is an abstract element, which cannot be instantiated in any process, but only acts as an interface for *Primitive* and *Composite Activities*. This hierarchy provides our model the flexibility to describe tasks, which can be decomposed to more than three levels.

Composite Activity: A set of *Primitive* or *Composite*

Development Activities. An example of a composite activity is software architecting, that according to [2] consists of architectural analysis, synthesis and evaluation.

Primitive Activity: A list of *tasks* that should be accomplished, in order to achieve the completion of the goal [9]. Based on the abovementioned example, a primitive activity is "architectural analysis".

Controllable Element: An abstract element that acts as an interface for all elements that can contain *Controls.*

Control: An abstract element that corresponds to any kind of control flow [8] that specifies the sequence of executing *Lifecycles, Development Phases, Development Activities,* and *Artifacts,* or to any relationship among elements.

Alternative / Loop / Optional: Elements for specifying a set of alternative / iterative / optional *Controllable Element.*

Parallel: A *Control* element for specifying a set of *Lifecycles, Development Phases,* or *Development Activities* that are executed in parallel.

Related To: A control element for creating links among a set of related *Lifecycles, Development Phases, Development Activities, or Artifacts.*

Worker: The responsibilities that an individual or a team has, that are related to a set of *Development Activities* and the way that they should be performed [9]. As an example of a worker we can consider a Software Architect or a Tester.

Artifact: A piece of information that is produced, modified, or used by a process, while targeting the final product [9]. The artifacts are used by a *Development Activity* as input/output data. An example of an artifact is the class diagram.

Composite Artifact: Corresponds to artifacts that consist of a collection of *Artifacts*. An example of a composite artifact is the Software Design Document (SDD), which contains many diagrams (among which a class diagram, which was used as an example above). Having modeled both types of artifacts (Artifacts and Composite Artifacts) with the same element, either this information would be lost (no self-reference in the model) or would lead to an inconsistent representation (self-reference in the artifact element in the model), in the sense that artifacts do not contain other artifacts.

Tool: The set of tools that are expected to be used for supporting the *Development Activities* of the process framework. [9]. An example of a tool that could be used for modeling UML diagrams is Rational Rose.

Based on the proposed meta-model, the embedded system specific challenges are handled as presented in Table I.

TABLE I. EMBEDDED SYSTEMS CHALLENGES IN METAMODEL

Challenge	Element/Attribute	Description
Multi-lifecycle	Composite Lifecycle	Different lifecycles are nested inside composite lifecycles.
Multi-discipline	Controllable Element::discipline	All lifecycles, development phases, or development activities are assigned to a discipline.
Multi-Site	Primitive Activity::site	The development activities are linked to the site where they are performed.
Multi-Organization	Primitive Activity::organization	The development activities are linked to the organization where they are performed.

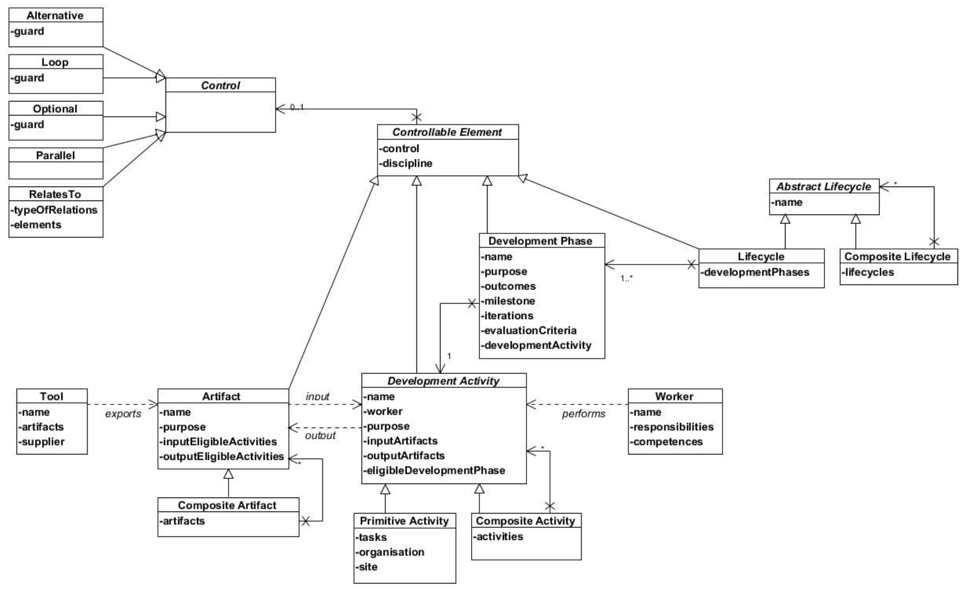

Fig. 1. PROMES Framework Meta-model

III. FOCUS GROUPS

In order to validate the requirements of the proposed meta-model and its industrial applicability, we conducted two focus groups (*analysis focus group* and *validation focus group*) with experts from the embedded systems domain. We selected focus groups as the most appropriate initial validation method, because we were aiming at investigating some new concepts, getting some additional ideas from experts, and prioritizing the next steps of developing the

PROMES process framework [5]. We note that the presented meta-model (see Fig. 1) has been constructed by accommodating the feedback that we received from both focus groups. Both focus groups were designed according to the guidelines of Kontio et al. [5].

A. Analysis Focus Group

The *analysis focus group* was performed early in the PROMES process framework development, after having designed the initial meta-model of the PROMES process framework, and after the high-level requirements were set. The participants of this focus group were Dutch industrial and research partners.

Plan the focus group: While planning the focus group our main goals were: (G1) to establish the framework's terminology, (G2) to identify additional requirements, and (G3) to verify the meta-model.

Design the focus group: We performed this focus group with domain experts, because: (a) they have knowledge of the existing domain-specific terminology as well as the requirements of the domain, and (b) they understand how existing, or even their own, processes work, so as to check if they could map to the meta-model of the proposed process framework. The involved participants[1] were both researchers (4) and practitioners (3) that had significant experience in the domain of embedded systems. The focus group was segmented into three separate sub-groups (two sub-groups with two participants and one sub-group with three participants), in order to increase the time that each individual could discuss about the topics. On the completion of the separate sub-groups, the participants gathered in plenary and discussed their findings.

Conduct the focus group: The duration of the focus group was 2 hours and it was organized as follows: (a) presentation of PROMES process framework, (b) parallel discussion in 3 sub-groups, and (c) summarizing in plenary session. The topics discussed in parallel and the mapping to the goals of the focus group were: (a) Terminology/Focus of the proposed process framework - G1, (b) Meta-model - G3, (c) Use cases for the proposed process framework - G2, and (d) Framework requirements - G2.

Analyze the data and report the results: The findings of the analysis focus group were used to update the PROMES Process Framework and provide guidance for further enhancements. Main changes can be summarized as follows:

- the term **process element** was defined as *any possible constituent of a process framework*, and the term **process component**, as *any potential instance of a process element, derived from the process framework, when instantiating a specific process* (G1).
- the focus of the PROMES Process Framework changed from software to system lifecycle (G1).

[1] Researchers were from Embedded Systems Institute. Practitioners were from Ocè, KE-Works, and Vector Fabrics.

In addition, the following needs have been identified and used as directions for future work (implemented in the current version of the framework), as follows:

- the need for adding elements that concern the **control flow** of sequential *Lifecycles*, *Development Phases* and *Development Activities* (G3).
- the need for providing a number of **variability mechanisms** that could be used for applying process instance modifications, when using the PROMES process framework (G2).
- the need for providing guidelines for the **identified use cases**: producing, customizing and checking the validity of process instances (G2).

B. Validation Focus Group

The *validation focus group* was performed in December 2013, after having completed the first version of the PROMES process framework, in which all the feedback received from the *analysis focus group* was incorporated. In this focus group the participants were industrial and research partners from Finland and The Netherlands.

Plan the focus group: The goals of this focus group were: (G1) to explore the ability of the meta-model to represent the embedded systems domain-specific challenges (multi-lifecycle, multi-discipline, multi-site, and multi-organizational engineering), and (G2) to investigate the applicability and understandability of the proposed process framework to an industrial context.

Design the focus group: The participants of the focus group were experts in the domain of embedded systems, i.e. 12 practitioners and researchers, representing two research institutes[2] and six industrial partners[3]. In this focus group no segmentation was done. We note that 7 out of the 12 participants overlapped among the two focus groups. However, since the discussed topics and goals differed, we do not consider this as threat for their outcomes.

Conduct the focus group: In total the duration of the focus group was 2 hours. The schedule in the focus group was organized as follows: (a) presentation of the PROMES process framework, and (b) plenary session, during which the industrial partners provided examples of industrial cases. During this session, the participants discussed the provided mechanisms for handling the embedded systems domain-specific challenges (G1), and how industrial processes could map to the PROMES meta-model (G2).

Analyze the data and report the results: The received feedback can be summarized as follows:

[2] The research Institutes were: Embedded Systems Institute (The Netherlands), and VTT Technical Research Centre (Finland).

[3] The industrial partners were: Ocè, KE-Works, and Vector Fabrics (The Netherlands), and Nokia Solutions and Networks, Metso, and Haloila (Finland).

Feedback on domain-specific challenges (G1):

- Handling multi-lifecycle engineering by providing a *Composite Lifecycle* element is acknowledged as an efficient and effective way for modeling cases where multiple lifecycles exist.
- Placing the attributes *site* and *organization* (incorporating multi-site and multi-organization engineering) in the *Development Activity* element, was considered a simple, yet effective solution by the industrial participants. The participants also acknowledged that these two attributes help from a validation point of view too, since they clearly show which site or organization is responsible for an activity, and thus owns its created (*output*) *artifacts*. This information is important in cases where a site/organization needs to use an *input artifact* that is not currently available, because it was created by another site/organization. However, there was a discussion whether such concepts deserved an element of their own, which on the other hand would make the meta-model more complex. This will be considered in future versions of the framework.
- The *discipline* attribute, which at that point was placed in the *Development Activity* element (so as to handle the multi-disciplinary nature of embedded systems), was considered useful, although in some cases it was found that redundant information was stored. For example, in the case that a complete development phase is handling a certain discipline, then all its constituent development activities had to be assigned to the same value in the discipline field. For this reason, we updated the framework based on this feedback, and we moved the attribute *disciplines* at the *ControllableElement* element, since different disciplines can be identified based on *Lifecycles, Development Phases* and *Development Activities*. Thus, if *discipline* is left blank, it inherits the discipline from its container element.

General Feedback (G2):

- The process framework is easy to use and understand by industrial practitioners.
- Although not systematically examined, the meta-model seems to map to the processes of all industrial partners.
- The use of a self-reference providing multiple levels of activities resolves the restriction of RUP [6] to have up to three levels of decomposition.

IV. Conclusions

This paper introduces the PROMES Process Framework that can assist process engineers in the domain of embedded system, to instantiate project specific processes. In order to design the meta-model of the framework, the challenges of engineering processes for the embedded systems industry (multi-lifecycle, multi-discipline, multi-site, and multi-organization) were identified. This resulted in including elements and attributes in the meta-model to address those challenges. At this point the framework does not provide a complete mechanism for handling the aforementioned challenges – we consider that as future work. However, it offers a theoretical basis for introducing more elaborate structures, methods or tools to sufficiently address them (e.g. a method for instantiating a development phase, considering the issues induced by a potential time difference between two sites, which work on successive development activities). The usefulness and applicability of the PROMES meta-model has been successfully validated by two focus groups with industrial and research experts on embedded systems. During the validation, apart from the positive feedback on the applicability of the framework, we received valuable comments that will guide our future research on the subject.

We plan to systematically evaluate the usability, the effectiveness and the accuracy of the framework, through a case study. Additionally, we plan to further investigate and improve the way we model the multi-discipline aspect. Finally, after establishing the theoretical background, we plan to implement a process element repository for supporting the use of the proposed process framework, and addressing the embedded systems domain challenges.

Acknowledgment

This research has been partially funded by the ITEA2 project 11013 PROMES. Additionally, we would like to thank the participants of the focus groups for their valuable feedback on the Process Framework.

References

[1] D. G. Firesmith and B. Henderson-Sellers, “The OPEN process framework: An introduction”, Addison-Wesley Pearson Education, London, UK, 2002

[2] C. Hofmeister, R. Nord, and D. Soni, “Applied software architecture”, Addison-Wesley Longman Publishing, Boston, MA, USA, 2002.

[3] IEEE 730:2002 Standard for software quality assurance plans, IEEE Computer Society,23 September 2002.

[4] ISO/IEC/IEEE 15288:2008 Standard for systems and software engineering - system life cycle processes, IEEE Computer Society, 31 January 2008.

[5] J. Kontio, J. Bragge, and L. Lehtola, “The focus group method as an empirical tool in software engineering”, Chapter 4 in F. Shull, J. Singer, and D. I. K. Sjøberg (editors), “Guide to Advanced Empirical Software Engineering”, Springer-Verlag New York, Inc., Secaucus, NJ, USA, 2007.

[6] P. Kruchten, “The Rational Unified Process: An introduction”, Addison-Wesley Longman Publishing, 2nd Edition, Boston, MA, USA. 2000.

[7] M. N. James, “Systems engineering guidebook: a process for developing systems and products”, Vol. 10. CRC Press, 1996.

[8] Object Management Group, “UML specification”, retrieved 12/2013 http://www.omg.org/spec/UML/2.4.1/

[9] Rational Corporation, “Rational unified process - best practices for software development teams”, IBM, online white paper, 1998.

[10] Rational Software, “Rational unified process for systems engineering – RUP SE1.1”, A Rational Software White Paper, TP 165A, 5/02, 2001.