

A MAM net Model of Software Product Line for Project Management

Dorjgotov Sainzaya¹, Seung-Hyeok Shin² and LeeSub Lee³

^{1,2,3} Kumoh National Institute of Technology

61, YangHo-Ro, Gumi, South Korea

¹zayatsum8@gmail.com, ²shinbaad@kumoh.ac.kr, ³eesub@kumoh.ac.kr

Abstract

Software Product Line(SPL) is one of the best development method for maximizing software reuse. SPL have known as an increasing used methodology over the last years. It has been taken advantages of a lot of benefits of software development in terms of time, cost and effort. Even though SPL is extremely effective for reusability, management of the SPL is very complicated. It means that utilization of existing Project Management(PM) tool and technics might be insufficient for complex SPL project management. Due to the complexity of the artifacts relation, communications for the parallel development of core assets and product development, and unplanned work, project management of the SLP is not a typical work.

This paper proposes a MAM net basted formal model of SPL for software project management and control. The suggested model uses FORM method for metadata of project management information and PMBOK is used for project management methodology. Based on the proposed model, project management methodology enables more sophisticated and easy project management, and it can develop and design an automated software system for project management.

Keywords: *Software product line, Software engineering, Project management, MAM net model, Formal model*

1. Introduction

A project is temporary in that it has a pre-defined beginning and ending, and has a restricted scope and limited resources. Project management is a methodical approach to planning and guiding project processes from start to finish [1][2]. Also it is essential for managing complex projects in organizations, especially in large-scale organizations. Project management is nowadays a key activity area, as developments and change management, both in small and large organizations, that can be efficiently implemented in the form of projects [3].

Software Product Line(SPL) is one of the most critical way for software reuse and software project management. In SPL, software artifacts such as requirements specification, system architecture, design, components, *etc.* are reused across a family of a product line with/without some adaptations [4]. In a new product producing, reusable requirements, components, architecture, test case *etc.* are reused with some adaptation which called core assets. Therefore, reusing method will be more effective than other conventional development methods [5]. Therefore, many organizations have been trying to adopt Software Product Line Engineering (SPLE) for improving the quality and productivity of information systems and software products [6].

Received (November 10, 2017), Review Result (January 5, 2018), Accepted (January 12, 2018)

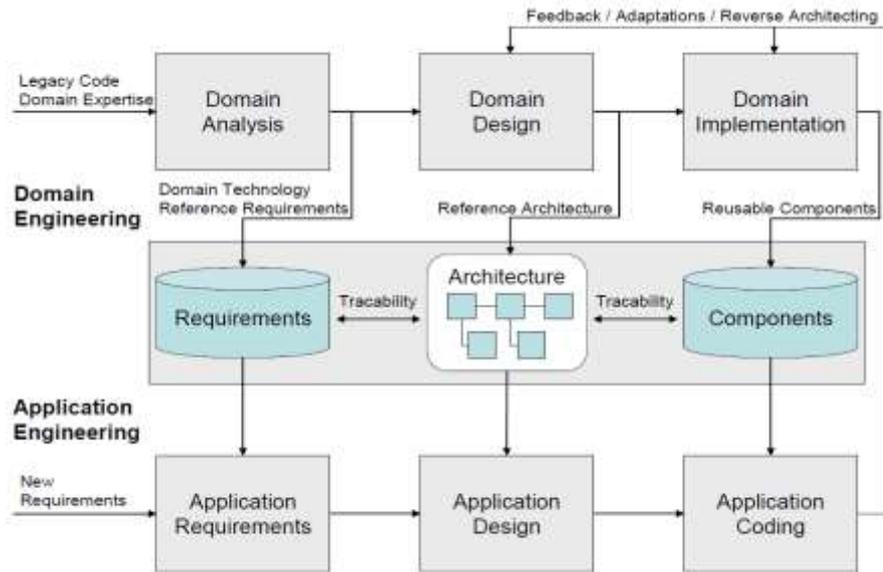


Figure 1. Software Product Line Engineering

Figure 1 shows a conventional SPLE process. [8] SPLE consists of two main compositions. In domain engineering process, a reusable core assets are built. The core assets are software artifacts that are reused in the production of customized products in a SPL. In the application engineering process, new products are produced using the reusable core assets. There are several problems in the SPLE. Firstly, there are many versions of core assets and for each core asset, there are several corresponding applications exist. Secondly for each project can be delayed due to unplanned work during the project. Certain number of defects will inevitably remain in released software products because software testing cannot demonstrate the absence of defects.

Thus for the project management for this complicated SPLE, a formal model of the process, artifacts, and scheduling for project management is inevitably required. For this formal model, this research surveys several project management processes, SPL models and SPL process models, and then figure out a formal model describes SPL project management.

According to the Figure 2 all of the rotating circles represent the three key activities in SPL [7]. All activities are linked together, in constant motion, and highly iterative. For example, core assets are used in products production, but the result of product development is revised core assets. The figure is neutral regarding which process is launched first. Many software product line organizations manage the activities that create core assets separately from activities that create the products. As shown in the figure, not only the development of core assets and products, but also the project management is very important in SPL. In order to control this complexity, it is essential to establish a formal model, but much research in this field has not been carried out.

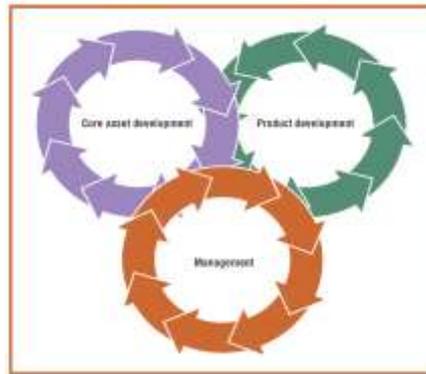


Figure 2. Software Product Line Development

One of the drawbacks of the SPL development process is that project can be delayed due to unplanned work during the project. A certain number of defects will inevitably remain in released software products not for customer, since software testing cannot demonstrate the absence of defects. These defects lead the project schedule and cost overrun. Several product projects can be undertaken at the same time independent. In that case, the following three types of unplanned work, that is depicted in Figure 3, will occur. [8]:

1. Adaptive reworks in product projects caused by residual defects in core assets.
2. Requirements changes for products.
3. Defect corrections in product projects before release.

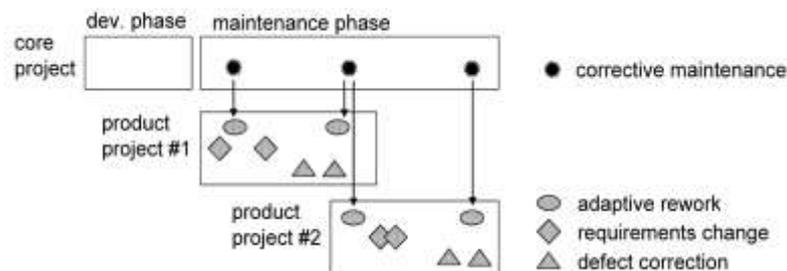


Figure 3. Assumed SPL Development and Unplanned Work

When residual defects in core assets are detected after their release to product projects, corrective maintenance is usually performed to modify the core assets. Core assets are maintained during a core asset maintenance phase to correct residual defects in core assets. [8] In order to adapt the products to the changed core asset, the core asset maintenance process brings associated rework to all ongoing products that depends on the core assets. These unplanned works make project management more complex, which makes the need for a formal model for project management more important.

2. Used Tools and Techniques

A Formal Method is a collection of mathematically based techniques used in development of computer systems. [9] It clearly describes complex systems and coordinates the connection between customers and developers. Also it provides means of guaranteeing various properties of a software system that increases its trustworthiness. [10] A formal specification is a simple description of a system using mathematical notations. Using mathematics is precise than other natural languages. Using a formal

specification language, a system can be described precisely with regards to functionality, concurrency, completeness, correctness, *etc.* [9]

There are many different types of formal method techniques used in different settings. MAM net formal model based on Petri net and dedicated to modeling software process and organization properties. [16] Process optimization and time analysis may be performed on a High Level Petri Net(HLPN) model. In this paper MAM net formal model is applied for the model construction. The MAM net is a formal model that clearly describes the processes step by step and includes the resource management such as human resource. This is particularly well suited to process management.

It is necessary to describe the structure and meaning of the managed artifacts as well as how to describe the process. Artifacts are all the consequences of recyclable project performance, including source code, as well as analytical documentation, design documentation, test materials, *etc.* SPL is very complex and a lot of research has been done on it. To select the target SPL architecture various types of SPL architectural models are surveyed. [14] The surveyed architectures are Component-Oriented Platform Architecture (COPA), Family-Oriented Abstraction, Specification, and Translation(FAST), Komponentenbasierte Anwendungsentwicklung(KobrA), Quality-driven Architecture Design and quality Analysis, and Feature-Oriented Reuse Method (FORM).

Kyo C. Kang and his co-fellows in Pohang University of Science and Technology, Korea, propose a FORM as an extension to the Feature-Oriented Domain Analysis(FODA) method. Feature-oriented method for capturing commonality inside a domain. Extended also to cover architectural design and development of core assets. FORM has clearly detailed technique which can describe the project processes. Also Feature-orientation of FORM gives a common language and therefore improves communication between customers and engineers.

The management of changing schedules is a key element in project management. This element defines how and when to schedule and manage the creation, recycling, and updating of various artifacts. The two most popular methods are PMBOK [12] and PRINCE2 [13].

PMBOK is intended to encapsulate all the knowledge that relates to project management which describes what the project manager and other major stakeholders should know. PMBOK clearly aids the development of knowledge and understanding about project management. It provides a highly descriptive comprehension of the PM processes, tools, and techniques. Also it defines the collection of knowledge, generally recognized as best practices for managing Projects. It based on 5 process groups, 47 processes, 10 knowledge areas.

PRINCE2 is a process-driven project management methodology which describes what the project manager and other major stakeholders should do. PRINCE2 is a prescriptive approach for project management, complete with templates, roles, and responsibilities of multiple stakeholders. It based on 7 principles, 7 themes, and 7 processes.

The reason to choose PMBOK is PRINCE2 does not include all the knowledge areas and details specified in the PMBOK. Detailed techniques are not provided to plan and carry out project activities. Also it does not have time management process that we can implement. In contrast PMBOK provides all round guidance and knowledge on project management. It provides a complete the step-by-step process on managing a project. Also it has clearly defined time management process that we can use directly.

To summarize the tools and techniques used in this study, FORM was used to describe the type, structure and meaning of artifacts, and PMBOK was used for the SPL methodology.

3. A MAM Net Model of Software Product Line

3.1. The Scope and Assumptions of the Model

The purpose of the model is proposing how to manage the project successfully. For successful project management, important aspects are to finish a project in a timely manner and to define the processes clearly. If problem occurs in project development process, it must be handled in way that sending the problem information into their initial source. In order to focus on the purpose, the scope of our model and assumptions are defined.

The most important aspects of Project Management are time, cost, scope and quality. In order to focus on the target, the scope should be properly minimized. The first selected aspect of project management is the time scheduling to meet deadlines of a plan. The second selected aspect of SPL management is unplanned work during product development process that is not defined at the planning phase. The last one is product development schedule management. Thirdly, this model assumes that domain PM and multiple product PMs are exist separately. It means there are complex communications are required among domain PM and product PMs. During the product development process especially in implementation phase and application testing phase, unplanned work might occur due to the residual defects in core assets. In this case product PM should give information about any problems occurred with the core asset to domain PM.

Finally, the model assumes that domain project manager should create a base plan for each product engineering process. Product PM specializes the base plan according to the requirements of a production. Production plan specifies how the products are produced from the core assets. The planning process needs to be iterative and thus plans could be changed. Plans should be updated and revised as needed during their lifespan. Product builders use the core assets, according to the production plan, to produce a product that meets its own requirements. Also it has an obligation to give feedback on any problems or deficiencies encountered with the core assets. All planning processes should include establishing the plan that includes establishing estimates of the resources required to carry out the plan, having those who will be bound by the plan review it for feasibility and establishing commitments to the plan. [11] After core assets development process complete, production plan is created. The project manager can manage and control the project using the production plan which includes all of the project information, aspects and guidance, so that high quality production plan is basis of successful project.

By including these, most of key activities could be covered.

3.2. The Key Process Steps for Defining the Model

The model is designed based on the PMBOK time management processes. The project schedule management with PMBOK consists of six main processes as shown in Figure 4. According to these steps, various artifacts defined in FORM are produced with MAM net format. In PMBOK, as a part of the planning process, the project manager develops a schedule management plan. It specifies how the schedule will be developed and managed. Also it includes the scheduling methodology, scheduling tool for developing and controlling the schedule. Using the MAM net model, the project progress can be accurately described, easily grasped, and the response to the change can be clearly handled.

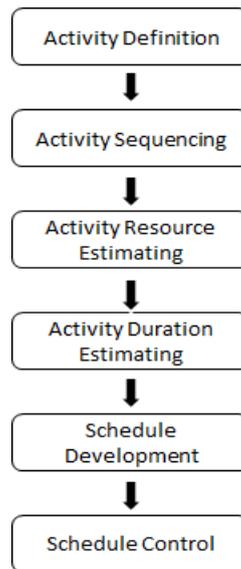


Figure 4. Project Time Management Process

The formal model can be represented with expressions and diagrams. The activities of MAM net diagram are shown in Figure 5. It shows final formal specification result of the project time management process.

If there are several artifacts in activity with similar type, comma separated list is used. It means that if there are two input artifacts in a process, we can express such as {p_r, b_p}. The output artifact of process step is going to become input artifact of successor process step. They are connected by the arcs and dotted arcs. If an output of process step connected to a Boolean decision which means true or false, a dotted arc is used.

Artifacts can be obtained from Domain Engineering and Product Engineering process results. The artifacts that is created from the Domain engineering is u_r which means user requirements, f_m which means feature model and b_p which means base plan. The artifacts created in Application engineering are p_r which means product requirements, p_p which means product plan, u_p_p which means updated product plan, s_f_m which means selected feature model, d_p which means defected plan, u_p_p which means updated product plan, p_a which means product architecture, p_c which means product code, def_p which means defected product, u_p_c which means updated product code, p which means product and r_p which means released product.

MAM net expression depicts a more formal way that is suitable for computer based processing. The model can be specified as follows:

The model is defined as

$$M \Rightarrow \{ \langle \text{Activity}, R, IP, OP, P \rangle \},$$

where

$$\text{Activity} = \{ \text{activity_name} \},$$

$$R = \{ r^1, r^2, \dots, r^n \} \text{ which denotes the type of resources,}$$

$$IP = \{ \text{input_artifacts} \},$$

$$OP = \{ \text{output_artifacts} \}$$

$$P = \{ q^1, q^2, \dots, q^n \} \text{ which denotes a set of process steps identifiers.}$$

The type of resource requests to Agent Allocation Unit represented by

$$R = \{ \text{Integer}, \text{Integer}, \text{Integer}, \text{Integer}, \text{Integer}, P \}$$

Human agents are represented by

$A = \{\text{Domain PM, Product PM, Customer, Developer, Tester}\}.$
All of the activities are sequenced, denoted by
 $P \in \{q^1, q^2, \dots, q^n\}$ identifier.

To arrange inputs and outputs of each activity, it should be represented as needed.

The input/output artifacts of model are represented by

$IP = \{\text{input_artifacts}\}, OP = \{\text{output_artifacts}\}.$

MAM net model represented as follows by activity name, input artifacts, output artifacts and identifier of activity.

MAM net diagram of the Figure 5 can be described by the above definition as follows:

$M \Rightarrow \{\text{User requirement analysis, } r1, \{u_r\}, \{p_r\}, q1 \}.$

$M \Rightarrow \{\text{Product planning, } r2, \{p_r, b_p\}, \{p_p\}, q2 \}$

$M \Rightarrow \{\text{Feature model selection, } r3, \{u_p_p, p_p, f_m\}, \{d_p, s_f_m\}, q3\}.$

$M \Rightarrow \{\text{Update Plan, } r4, \{d_p\}, \{u_p_p\}, q4\}.$

$M \Rightarrow \{\text{Architecture selection \& adaptation, } r5, \{s_f_m\}, \{p_a\}, q5\}.$

$M \Rightarrow \{\text{Component adaptation \& coding, } r6, \{p_a\}, \{p_c\}, q6\}.$

$M \Rightarrow \{\text{Testing, } r7, \{p_c\}, \{p\}, q7\}.$

$M \Rightarrow \{\text{Product release, } r8, \{p\}, \{r_p\}, q8\}.$

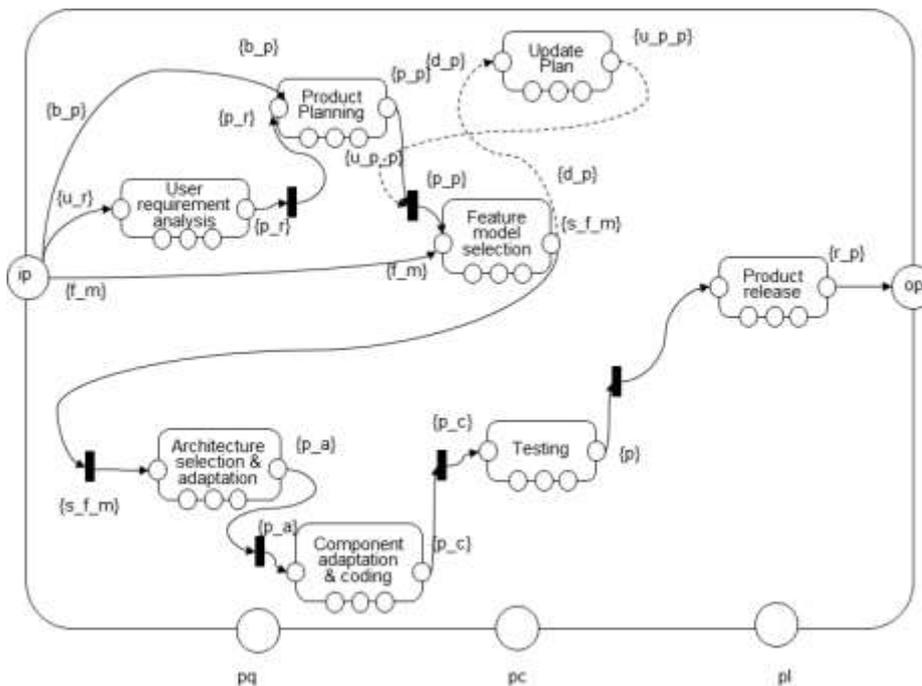


Figure 5. The Sequencing of Activities by MAM Net Diagram

4. Conclusion

Due to the complexity of the artifacts relation, communications for the parallel development of core assets and product development, and unplanned work, project management of the SPL is not a typical work. The existing project management method is not sufficient for the SPL development method.

In order to solve the problem, a formal model of SPL development is proposed. The suggested model uses FORM method for metadata of project management information MAM net model is used to express sequence of the model, and PMBOK is used for

project management methodology. Since the model is formal it can describe most of the important information clearly. Thus with the model, management activity could be conducted in accurate manner and efficiently. Furthermore, formal definitions could provide a clear basis for the project management automation system.

Future work will include the development document management system to manage complex artifacts and schedule for project management for SPL

Acknowledgements

This paper was supported by the Kumoh National Institute of Technology Research Grant. This article is a revised and expanded version of a paper entitled “A Formal Model of Software Product Line for Project Management” presented at SERSC NGCIT 2017 on August 16-18, 2017 at Liberty Central Saigon Riverside Hotel, Ho Chi Minh City, Vietnam.

References

- [1] C. Traue, “Benefits Realization”, <http://www.traue.com/pm/>, (2008).
- [2] , Z. Vukovic. “Project Management-Software Development Methodology”, *Interdisciplinary Management Research*, vol. 1, (2004), pp.146-169.
- [3] I. Intorsureanu, R. Mihalca, A. Uta and A. Andreescu, “A Design of an Integrated Document System for Project Management”, *Proceedings of the International Conference on Computer Systems and Technologies and Workshop for PhD Students in Computing*. ACM, Rousse, Bulgaria, (2009).
- [4] A. Irshad Khan, M. Alam and W. Al Jedaibi, “Variability Management in Software Development using FeatureIDE: A Case Study”, *International Journal of Scientific & Engineering Research*, vol.6, no.1, (2015).
- [5] A. Guendouz and D. Bennouar, “Component-Based Specification of Software Product Line Architecture”, *International Conference on Advanced Aspects of Software Engineering ICAASE*, Constantine, Algeria, (2014).
- [6] S.-H. Kim and J.-A. Kim, “Consistency Checking Rules of Variability between Feature Model and Elements in Software Product Lines”, *Korea Information Processing Society*, vol.3, no.1, (2014), pp.1-6
- [7] P. C. Clements, L. G. Jones, L. M. Northrop and John D. McGregor, “Project Management in a Software Product Line Organization”, *IEEE Computer Society*, vol.22, no.5, (2005), pp.54-62.
- [8] M. Nonaka, L. Zhu, M. Ali Babar and M. Staples, “Project Cost Overrun Simulation in Software Product Line Development”, *Product-Focused Software Process Improvement*, Springer Berlin Heidelberg, (2007), pp.330-344.
- [9] S. Wolff, “Scrum Goes Formal: Agile Methods for Safety-Critical Systems”, *Proceedings of the First International Workshop on Formal Methods in Software Engineering: Rigorous and Agile Approaches*, USA, (2012), pp.23-29
- [10] D. Clarke, N. Diakov, R. Hähnle, E. B. Johnsen, G. Puebla, B. Weitzel and P. Y. H Wong, “HATS-A Formal Software Product Line Engineering Methodology”, *Proceedings of the First Intl. Workshop on Formal Methods in Software Product Line Engineering (FMSPLE 2010)*, Jeju Island, South Korea, vol.2, (2010), p.8.
- [11] K. C. Kang, S. Kim, J. Lee, K. Kim, E. Shin and M. Huh, “FORM: A feature-oriented reuse method with domain-specific reference architectures”, *Annals of Software Engineering*, NJ, USA, vol.5, no.1, (1998), pp.143-168.
- [12] IT Knowledge Portal, “Association of Modern Technologies Professionals”, <http://www.itinfo.am/eng/project-management-body-of-knowledge-pmbok-guide/>, (2016).

Authors



Sainzaya Dorjgotov, she is a M.S. student in department of computer engineering at Kumoh National Institute of Technology, Gyeongsangbuk-do, Korea. She received her B.S. degree in department of computer engineering from School of Information and Communication Technology, Ulaanbaatar, Mongolia in 2012. Her research interest focuses on Project Management.



Seung-Hyeok Shin, he is currently an associate professor in Department of Applied Mathematics at Kumoh National Institute of Technology. He received B.S. in Applied Mathematics, the M.S and Ph.D. in Computer Engineering degrees from degree from Kumoh National Institute of Technology in 1998, 2000 and 2016 respectively. His current research interests are data visualization on the wireless IoT and cryptographic algorithms based on the number theories.



Lee-Sub Lee, he is an Associate professor of Department of Computer Engineering at the Kumoh National Institute of Technology Gyeongsangbuk-do, Korea. He received B.S. in Mathematics and M.S. degree in Computer Engineering from Sogang University, Seoul, Korea. He received his Ph.D. in Computer Engineering from Korea University, Seoul, Korea. He has worked as a senior researcher at Samsung SDS 1990 to 2004. His research work has been on the Software Engineering and Database System. His recent interest focuses on Software Testing, Bigdata.

