A Metamodel for Representing Safety LifeCycle Development Process

Huaxi (Yulin) Zhang
IRIT, University of Toulouse
118 Route de Narbonne
31062 Toulouse Cedex 9, France
zhang@irit.fr

Brahim Hamid
IRIT, University of Toulouse
118 Route de Narbonne
31062 Toulouse Cedex 9, France
hamid@irit.fr

Damien Gouteux
IRIT, University of Toulouse
118 Route de Narbonne
31062 Toulouse Cedex 9, France
gouteux@irit.fr

***Abstract*—Metamodeling process supports the effort of creating flexible process models. The purpose of process models is to document and communicate processes and to enhance the reuse of processes. Thus, processes can be better taught and executed. Results of using metamodel process are an increased productivity of process engineers and an improved quality of the models they produce. However, most useful metamodels are activity-oriented, and the required concepts of safety lifecycle, such as validation, can not be easily modeled through these metamodels. In this paper, we propose a safety-oriented process metamodel to support all the requirements of safety control. As a proof of concept, we examine a process model that has several safety lifecycle requirements: the IEC 61508 safety lifecycle V-model standard.**

Keywords-Safety lifecycle, Development process, Modeling, Process metamodel

I. INTRODUCTION

Over the last two decades, the need for a formally defined safety lifecycle process has emerged. This is because the inevitable requirement for better processes eventually pushed control systems to a level of complexity where sophisticated electronics and programmable systems have became the optimal solution for control and safety protection [1]. The industrial processes trend to have following characters:

- Industrial processes are becoming more and more complex.
- Increasing numbers of people and organizations are involved.
- High cost in case of an unwanted spurious process trip.
- Large consequences in case the process gets out of control.

With these emergent requirements, many safety lifecycles have been proposed by different associations, like IEC (International Electrotechnical Commission) or ISA (International Society of Automation). These safety lifecycles are adopted by different domains or enterprises with some modifications to adapt different requirements (for example, domain specific requirements). However, as the fundamental differences between traditional development process and safety lifecycle are huge, such as different kinds of safety checks and the safety relationships between these checks and phases, to model these different safety lifecycles with traditional used process metamodel is not simple and direct. Most process metamodels such as SPEM (Software & Systems Process Engineering Metamodel), UMA (Unified Method Architecture), OPF (OPEN Process Framework), focus on modeling the process model with activity-oriented viewpoint to accommodate a large range of development processes. Furthermore, no process metamodel is rich enough or oriented to serve as the support of a safety lifecycle.

The goal of the paper is to present an ongoing work devoted to extend exiting framework with support for safety lifecycle development. That is, we propose a new safety lifecycle development processes technique in order to make easy their use in a building process of system/ software applications with safety support. The proposed vision is to use modeling techniques to obtain high level of abstractions in order to avoid the cost of building a process for each applications properties and/or for each domain. Reaching this purpose requires to get (1) a common representation of safety lifecycle process for several domains; (2) a process flexible structure; (3) guidelines for domain specific implementation of the process and (4) guidelines to guarantee the correctness of the process with regard to safety requirements. Thus, we propose a PPFS metamodel which response all these requirements which is developed under the European project TERESA, oriented to different concerns, namely safety lifecycle, pattern, repository, embedded system and non-/extra- functional properties. In this paper, we just concentrate on the aspect of safety lifecyle.

The remaining of this paper is organized as follows. Section II defines the context of the safety lifecycle and the problem definition is presented followed by a motivating example. Section III discusses the state of the art of process metamodels from the safety related viewpoint. Section IV outlines the PPFS process metamodel. Section V presents how the PPFS metamodel supports the safety lifecycle with its safety-related concepts. Section VI illustrates the PPFS metamodel by the IEC 61508 standard safety lifecycle V-model. Section VII concludes and draws future work directions.

Copyright (c) IARIA, 2011. ISBN: 978-1-61208-165-6

II. Problem Statement

The main difficulty to overcome in the development of critical embedded systems is how to avoid the cost of building a process for properties of each application and/or for each domain. One way to obtain high level of abstraction is to make use of meta-modeling techniques. Informally, a process has several views with regard to the considered level of abstraction. This decomposition and separation of uses illuminates how to create, to specialize processes. This implies that a process is created at high level abstraction and then it will be transformed into more specific one. The common safety engineering meta-model will have to recognize the need to separate expertise on applications. As a result, individual application domains could have different safety engineering processes, for example, a domain where application engineers do not use model-driven engineering should have a more decoupled interaction with the modeling artifacts.

A. Safety Lifecycle: Definition and Concepts

The *safety lifecycle* can be defined as: an engineering process designed to achieve a risk-based level of safety with performance criteria that allow versatile technologies and optimal design solutions [2]. The risk-based levels are recognized as *system integrity level* (SIL). SIL measures the confidence which can be attributed on the fact that the integrity of the system functions conform with the requirements.

Many safety lifecycles are proposed, such as IEC 61508 [3], IEC 61511 [4], and ANSI/ISA S84.01 [5]. The differences between the safety lifecycle and normal development process are only the integration of safety related phases into process, but also the special concepts used to verify whether the safety lifecycle and the SIL requirements are correctly implemented and satisfied. Generally, there are four types of checks used to validate the safety lifecycle [3]:

- *Verification.* Confirmation by examination and provision of objective evidence that the intended functions have been correctly implemented and the requirements have been satisfied. and assurance that the safety analysis remains valid for the system as implemented.
- *Validation.* The activity of demonstrating that the safety-related system under consideration, before or after installation, meets in all respects the safety requirements specification for that safety-related system.
- *Functional safety audit.* Systematic and independent examination to determine whether the procedures specific to the functional safety requirements comply with the planned arrangements, are implemented effectively and are suitable to achieve the specified objectives.
- *Functional safety assessment.* Investigation, based on evidence, to judge the functional safety achieved by one or more E/E/PE safety-related systems, other technology safety-related systems or external risk reduction facilities.

Beyond these checks, the interaction and influences between the process phases should be considered. This means the safety relationships between checks and phases within one development process. To support these relationships, four basic flows should be modeled: control flow, retrieve flow, validation flow and verification flow. These flows represent the interactions and influences between checks and process phases. process, such as he verification flow, validation flow, etc. Thus, the same time with four types of checks, it also specify four types of flow relationships in process.

B. Motivating Example

Safety lifecycles are practiced in different domains or different enterprises with different kinds of versions [2]. The domain specific requirements lead different safety lifecycles, which are modified from the general or standard lifecycle to adapt their specific requirements. For example, the IEC (International Electrotechnical Commission) 61508 is today globally recognized and considered as the basic standard to evaluate the suppliers' products. IEC 61508 [3] recommends a V-model safety lifecycle, as shown in Fig. 1. How to define this kind of safety lifecycle model, such as IEC 61508 V-model, is raised as a problem. Thus, in this paper, we use IEC 61508 as a motivating example.

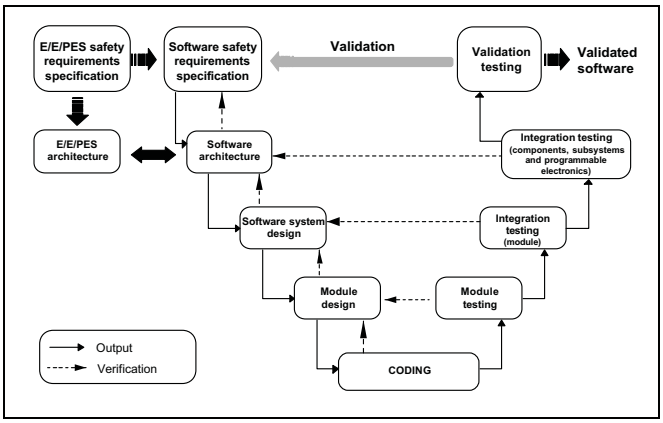

Figure 1. The V model of IEC 61508

Considering above modeling problem, we find that a safety-related metamodel, which can be applied to model these different lifecycle models, is stringently required. Thus the claim of this paper is that a safety-related process metamodel should capture the safety related process concepts to facilitate the modeling of safety-related development process. In other words, in order to model a safety lifecycle, a process metamodel should model SIL, checkpoints and different flows between checks and phases. These concepts as the minimum support and basic elements for safety-related

lifecycle, should be modeled by a process metamodel. State-of -the-art of process metamodels have been analyzed from this perspective, trying to answer the following questions:

- Do existing process metamodels support safety lifecycles?
- If so, are these metamodels can capture all required safety concepts mentioned above explicitly?

III. State of the Art of Process Metamodel

Meta-process modeling supports the effort of creating flexible process models. The purpose of process models is to document and communicate processes and to enhance the reuse of processes. Thus, processes can be better taught and executed. Results of using meta-process models are an increased productivity of process engineers and an improved quality of the models they produce [6].

Process metamodels can be modeled from different views: activity-oriented, product-oriented and decision-oriented views [6], [7], [8]. Most process metamodels adopt the activity-oriented views, such as SPEM, UMA and OPF.

The SPEM (Software & Systems Process Engineering Metamodel) was created by the Object Management Group [9] as a *de facto*, high-level standard for processes used in object-oriented software development. The scope of SPEM is purposely limited to the minimal elements necessary to define any software and systems development process, without adding specific features for particular development domains or disciplines. The goal is to accommodate a large range of development methods and processes of different styles, cultural backgrounds, levels of formalism, lifecycle models, and communities. Thus, with SPEM, it is not easily to model all the specific concepts required by safety lifecycle.

The Unified Method Architecture (UMA) [10] has been developed within IBM [1], which is mostly used in industry to support the most important standards. The metamodel of UMA is based on SPEM, thus it has the same weakness as SPEM .

The OPEN Process Framework (OPF) is defined by OPEN [11]. Generally, it is a componentized OO development methodology underpinned by a full metamodel. The drawback of OPF is just like above twos.

Thus, we can find that these metamodels are not designed to support safety lifecycle. In some view, they permit to model the safety related concepts. With the above mentioned characters of safety lifecycle, we give a comparison between these metamodels as shown in tables I and II. Table I evaluates how these metamodels support four kinds of checks mentioned in the beginning and Table II compares these metamodels from the special required relationships of safety lifecycle. Tables I and II evaluate these metamodel from the facility of use and the easiness of comprehension via the mentioned safety concepts. The tables use four levels to evaluate these metamodel from + to ++++. From these tables, as SPEM is a general process metamodel, we can find that it is difficult to use to model safety lifecycle. UMA and OPF are better than SPEM, however they are also not designed to orient and model safety lifecycle. We can just adjust some of their concepts to represent the safety audit and safety assessment in a more general way. For the safety relationship, all these metamodels are in same level, they do not have any specific concepts to model the safety relationship, however we can still adjust their control flow concepts to safety relationship. But the semantic information of all these safety checkpoints and relationships are difficult to reserve and illustrate in these metamodels.

[1]UMA has been developed in a collaborative effort by the architects of the IBM Rational Unified Process (RUP).

Metamodel	Validation	Verification	Safety audit	Safety assessment
SPEM	+	+	+	+
UMA	++	++	++	++
OPF	++	++	+++	+++

Table I
Comparison of existing process metamodels in checkpoints

Metamodel	Control Flow	Retrieve Flow	Validation Flow	Verification Flow
SPEM	++++	++	++	++
UMA	++++	++	++	++
OPF	++++	++	++	++

Table II
Comparison of existing process metamodels in associations

Except above mentioned process metamodels, there are also other activity metamodels like OOSPICE [12], SMSDM [13]. Beyonds these, the other types of process metamodel such as decision based etc, do not orient to safety critical system development neither. As far as we know, the studied process metamodels unfortunately do not support safety related development process explicitly or facilitate the modeling of safety lifecycles. Beyonds these, many safety critical systems use safety instrument systems (SIS) to manage the safety lifecycle, however, these SIS do not have any process metamodel. Some works like [14] are proposed to model different standards and try to give recommendations during the application development using these standards. In conclusion, these existing metamodels are (1) not explicitly or directly describing the safety concepts as the first-classes and (2) not easily to use or comprehend, such as the different flows cannot be differentiated with each other. Thus, this analysis results in requirements for the process metamodel presented in this paper with following characteristics:

- Design with the viewpoint: safety-related.

Copyright (c) IARIA, 2011. ISBN: 978-1-61208-165-6

- Support safety-related development process with its necessary required concepts: SIL, checkpoints and safety control relationships.

IV. Overview of PPFS Metamodel

To response the above requirements of metamodel, we propose a metamodel called PPFS. This metamodel is designed under the European project TERESA. It supports several engineering concerns, namely: safety lifecycle, pattern, repository, embedded system and non-/extra- functional properties, as shown in Fig. 2. In our works, we deal with a metamodel PPFS (Process-based Pattern Fundamental Structure), which is designed to orient several engineering concerns, namely: safety lifecycle, pattern, repository, embedded system and non-/extra- functional properties, as shown in Fig. 2. In this paper, we concentrate on its safety related concern.

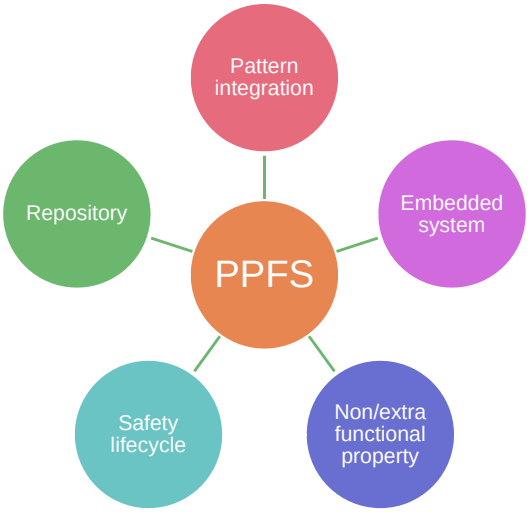

Figure 2. The characteristics of PPFS metamodel

The PPFS metamodel describes all the artifacts (and their relations) required to capture all the facets of safety-life cycle processes. It contains different packages depicted in Fig. 3 which supply different capabilities. In order to compare with other process metamodels, we give a simplified version of metamodel with necessary elements to capture safety-related concepts as shown in Fig. 4.

In this paper, we concentrate on presenting the safety-related part of the PPFS metamodel.

V. Safety Concern of PPFS Metamodel

In this section, the safety-related concepts in PPFS will be introduced, including SIL, the checks and safety relationships.

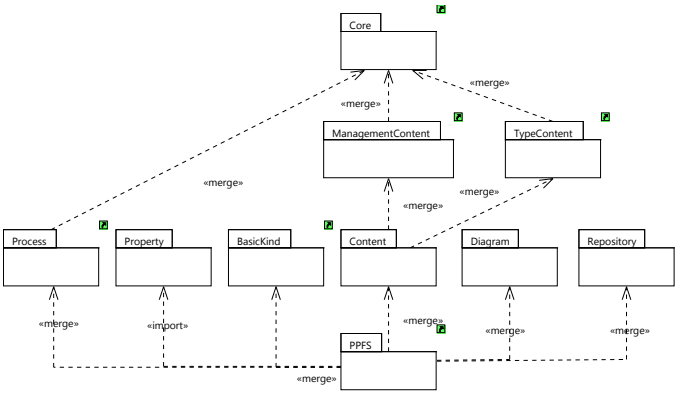

Figure 3. Structure of PPFS Metamodel

A. SIL

SIL in the PPFS metamodel is modeled as enumeration class with five levels from zero to four.

- SIL 4: the highest target and most onerous to achieve, requiring state of the art techniques (usually avoided)
- SIL 3: less onerous than SIL 4 but still requiring the use of sophisticated design techniques.
- SIL 2: requiring good design and operating practice to a level not unlike ISO 9000.
- SIL 1: the minimum level but still implying good design practice.
- SIL 0: referred to as “not-safety related” in terms of compliance.

With these five levels, the SIL attribute of process class can be set to SIL value to determine the process demand rate, which is a measure of the integrity and the stability of the process (see Fig. 4).

B. Checkpoint

Checkpoint is defined as an activity or phase which presents the safety checks in different levels of process. In other words, in the PPFS, safety checks are named checkpoints. They are used to verify whether the safety requirements are correctly implemented. To fulfill the requirements presented in Section II, we specify four kinds of checkpoints: validation, verification,safety audit and safety assessment. The structure of checkpoint and related classes is depicted in Fig. 5.

Furthermore, in order to facilitate the extension of the metamodel, the different kinds of checkpoints are defined as CheckpointKind. With this class, the checkpoint can be easily extended by different required types. The relationship is shown in Fig. 6.

In the following, we present four kinds of checkpoints predefined in the PPFS metamodel.

1) Verification: The definition of validation is a confirmation by examination and provision of objective evidence that (i) the intended functions have been correctly implemented

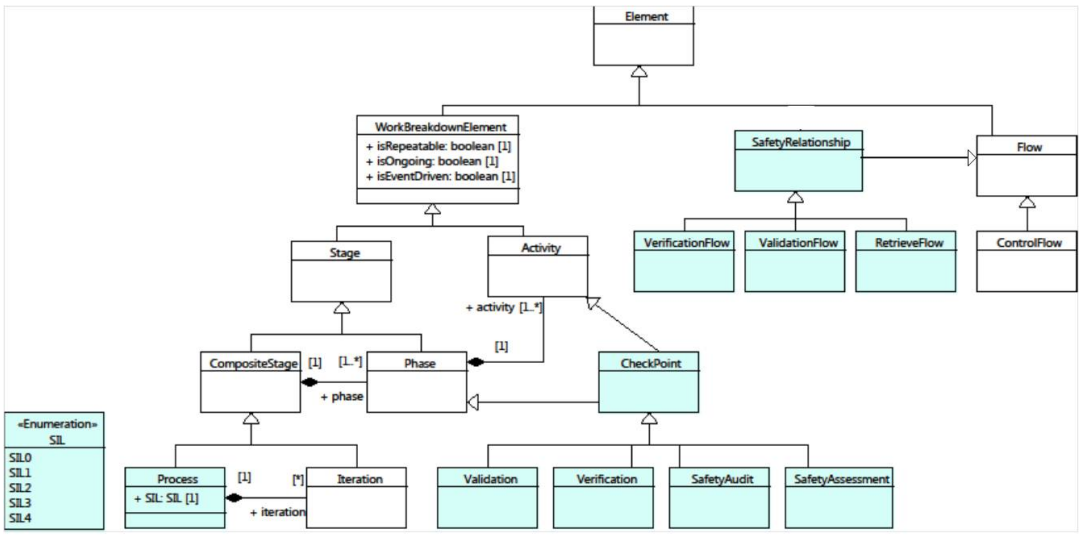

Figure 4. The structure of PPFS from safety-related viewpoint

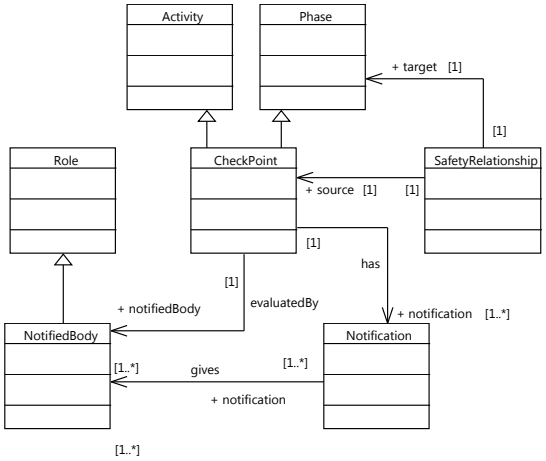

Figure 5. Structure of Checkpoint

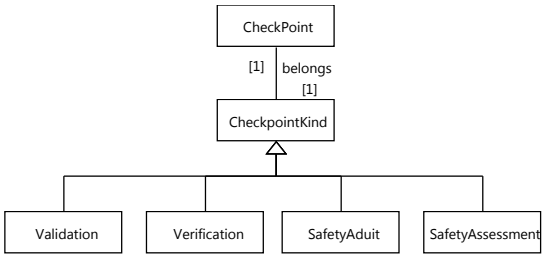

Figure 6. The types of checkpoint

and (ii) the requirements have been satisfied and (iii) assurance that the safety analysis remains valid for the system as implemented.

2) Validation: The activity of demonstrating that the safety-related system under consideration, before or after installation, meets in all respects the safety requirements specification for that safety-related system.

3) Safety audit: Safety audit defines a systematic and independent examination to determine whether the procedures specific to the functional safety requirements comply with the planned arrangements, are implemented effectively and are suitable to achieve the specified objectives. Figure. 7 gives an example of safety audit, which serves as a checkpoint and also a phase or activity.

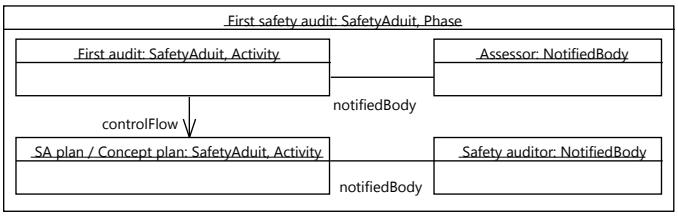

Figure 7. The example of Safety audit

4) Safety assessment: Safety assessment is defined as an investigation, based on evidence, to judge the functional safety achieved by one or more E/E/PE safety-related systems, other technology safety-related systems or external risk reduction facilities.

C. Safety Relationships

There are five kinds of safety relationships: internal verification, external verification, validation and retrieve flow. We precisely define different kinds of verification relationships in the PPFS metamodel.

1) Control Flow: is a Flow element that presents the continuation of one Work Breakdown Element to another Work Breakdown Element. The control flow presents the

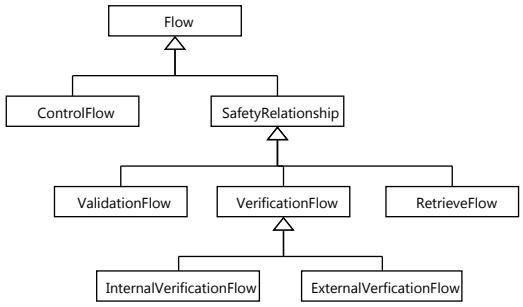

Figure 8. Structure of Flows

next work breakdown element after finishing the previous one.

2) Internal Verification Flow: is a Flow element that presents the internal verification relationship of one Work Breakdown Element to another Work Breakdown Element.

Internal Verification Flow represents the internal verification that we performed before start a new development phase. These actions must be carried out in order to check that the actions performed in the immediately previous phase have been done in a proper way. These actions are performed by a group independent to the design team and these actions are restricted to the left branch of the Safety Life Cycle (V-Model). These verification actions are shared between the safety audit team (Safety Auditor) and the team in charge of carried out the internal reviews.

3) External Verification Flow: is a Flow element that presents the external verification relationship of one Work Breakdown Element to another Work Breakdown Element.

External Verification Flow represents the normal verification performed at the right branch of Safety Life Cycle (V-Model). These actions are performed by the verification team and they start at the end of the implementation phase. The typical actions in this kind of verification are often listed below:

- Static analysis - Code coverage/Syntactic analysis
- Unit Tests
- Integration Tests
- System Tests - Validation Tests

4) Validation Flow: is a Flow that represents the validation relationship between two Work Breakdown Element. In safety lifecycle V-model, validation executes at the end of the implementation phase in V-model to confirm that the installed and commissioned SIFs meet the Safety Requirements Specification (SRS).

In our metamodel, although the validation concept comes from the safety lifecycle, we still make it generalization. That means validation can be concerned different perspective, not only just for safety, for example dependability validation, security validation etc.

5) Retrieve Flow: is a Flow that represents the retrieve relationship from checkpoints to phases or activities. The retrieve action will be proceeded when the checkpoints don't pass the examination. The process will turn back to the previous Work Breakdown Element to reexamine or redo the works. Figure 9 shows an example of retrieve flow.

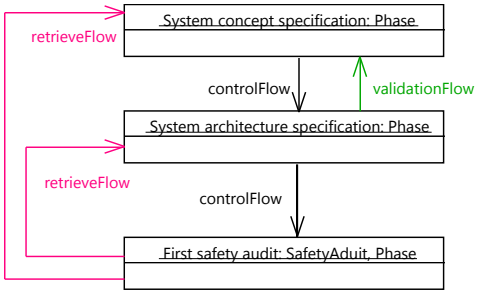

Figure 9. The example of retrieve flow

VI. An Illustration: IEC 61508 Safety Lifecycle

In this section, we try to illustrate the use of modeling framework by modeling IEC 61508 standard safety lifecycle V-model by the PPFS metamodel. Fig. 10 depicts the IEC 61508 V-model instantiated from the PPFS metamodel. From this illustration, we can easily demonstrate that it is more direct and precise using the PPFS metamodel to define the different safety lifecycle models. As software process covers the entire software development and contains almost all the necessary information of the development, thus it is difficult to present the entire process with all the information in one model. Normally, we use one process model to present the overall development in first level, and then decompose the process with different sub-models that correspond each phase of development. Fig. 10 is an example of the first level model of process.

VII. Conclusion and Future Works

This paper presented and illustrated our proposed PPFS metamodel from the safety-related viewpoint. Few process metamodel are rich enough or oriented to serve as the support of a safety lifecycle. Most process metamodels such as SPEM [9], UMA [10], OPF [11], focus on modeling the process model with activity-oriented viewpoint to accommodate a large range of development processes. As mentioned in Section III, a safety-oriented process metamodel is required. The PPFS metamodel fulfills all the required characteristics mentioned. It permits (1) to design process model from the safety-related viewpoint, (2) to support safety-related development process with SIL (safety integrity level), checkpoints and safety control relationships, (3) to facilitate modeling the domain specific safety lifecycle.

The PPFS metamodel presented in this paper is also illustrated by a case study of IEC 61508 standard safety-

Copyright (c) IARIA, 2011. ISBN: 978-1-61208-165-6

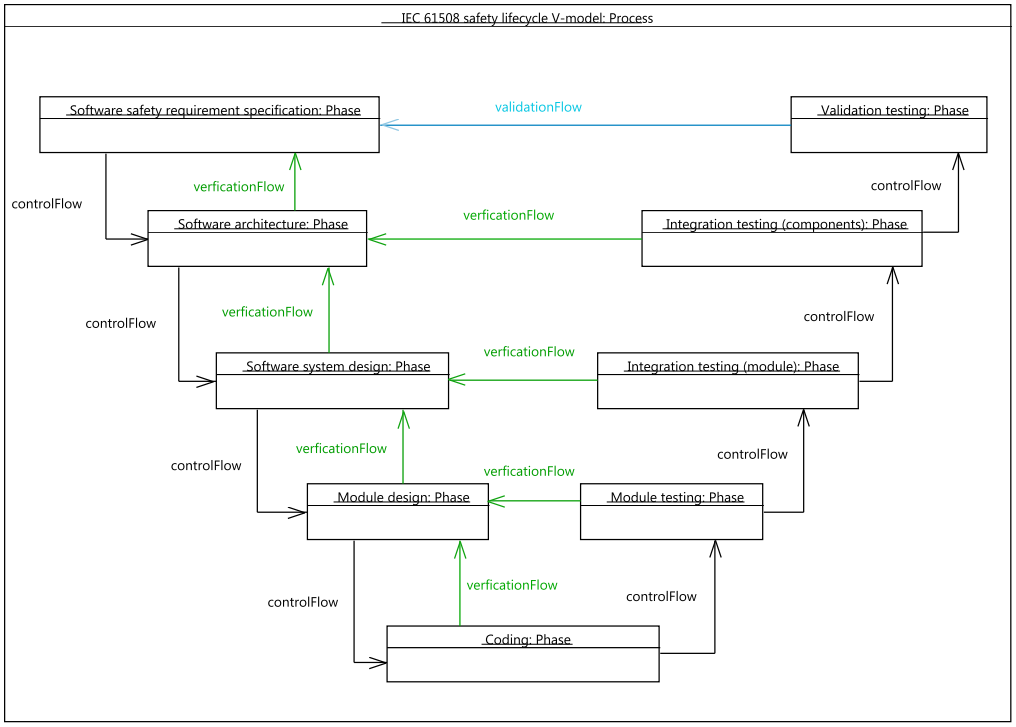

Figure 10. PPFS Metamodel instantiated by the IEC 61508 safety lifecycle.

lifecycle V-model. By this illustration, we can validate the feasibility and effectiveness of the PPFS metamodel.

As future work, we plan to extend the meta-model to refine the specifications of safety lifecycle in order to support (i) design pattern solutions, (ii) repository and (iii) extra-functional and non-functional properties.

Acknowledgements. This work is initiated in the context of SEMCO framework. It is supported by the European FP7 TERESA project and by the French FUI 7 SIRSEC project.

REFERENCES

[1] D. J. Smith and K. G. L. Simpson, *Functional Safety: A straightforward guide to applying IEC 61508 and related standards*, 2nd ed. Elsevier: Butterworth Heinemann, 2004.

[2] Exida, "Iec 61508 overview report (version 2.0)," Tech. Rep., January 2006.

[3] I. S. . IEC 61508, *Functional safety of electrical/ electronic/programmable electronic safetyrelated systems*, International Electrotechnical Commission Std., 2000.

[4] I. S. . IEC 61511, *Functional safety - Safety instrumented systems for the process industry sector*, International Electrotechnical Commission Std., 2003.

[5] A. S. S84.01, *Application of Safety Instrumented Systems for the Process Industry*, International Society for Measurement & Control Std., 1996.

[6] C. Rolland, "A comprehensive view of process engineering," in *Proceedings of the 10th International Conference on Advanced Information Systems Engineering*. London, UK: Springer-Verlag, 1998, pp. 1–24.

[7] C. Rolland, N. Prakash, and A. Benjamen, "A multi-model view of process modelling," *Requirements Engineering*, vol. 4, pp. 169–187, 1999.

[8] C. Hug, A. Front, D. Rieu, and B. Henderson-Sellers, "A method to build information systems engineering process metamodels," *J. Syst. Softw.*, vol. 82, pp. 1730–1742, October 2009.

[9] *Software & Systems Process Engineering Meta-Model Specification*, OMG, 2008.

[10] EPF. www.eclipse.org/epf.

[11] O. P. F. (OPF). http://www.opfro.org/.

[12] B. Henderson-Sellers and C. Gonzalez-Perez, "A comparison of four process metamodels and the creation of a new generic standard," *Information & Software Technology*, vol. 47, no. 1, pp. 49–65, 2005.

[13] *Standard Metamodel for Software Development Methodologies.*, Standards Australia, 2004.

[14] L. Y. C. Cheung, P. W. H. Chung, and R. J. Dawson, *Managing process compliance*. Hershey, PA, USA: IGI Publishing, 2003, pp. 48–62. [Online]. Available: http://portal.acm.org/citation.cfm?id=954321.954326

Copyright (c) IARIA, 2011. ISBN: 978-1-61208-165-6