Systematic Software Process Development: Where Do We Stand Today?

Marco Kuhrmann, Daniel Méndez Fernández, Ragna Steenweg
Technische Universität München – Software & Systems Engineering
Munich, Germany
{kuhrmann,mendezfe,steenweg}@in.tum.de

ABSTRACT

A software process metamodel (SPMM) defines a language to describe concrete software processes in a structured manner. Although agile methods gained much attention in recent years, we still need to provide process engineers with adequate tools to design, implement, publish and deploy, and manage comprehensive software processes. In response to this need, several SPMMs have been developed. It remains, however, unclear, which of those SPMMs are disseminated to which extent. In this paper, we contribute first results of a study on the state-of-the-art in the systematic development of software processes using standardized SPMMs and their corresponding infrastructure. Our results show that only a few documented standards exist and, furthermore, that among those standards only two are disseminated into practice. We focus on those standardized SPMMs, show their process ecosystem, and sketch a first picture on the state-of-the-art in SPMM-based software process development in order to foster discussions on further problem-driven research.

Categories and Subject Descriptors

D.2.9 [**Software Engineering Management**]: Software process models

General Terms

Management, Experimentation

Keywords

Software process metamodels, study, literature review

1. INTRODUCTION

Nowadays, software processes comprehend up to thousands of process elements, which need to be designed, implemented, and managed. In contrast to light-weight agile methods, comprehensive company processes thus have high requirements on process languages and process frameworks. Many companies use commercial tools to design their processes. They risk, however, proprietary process descriptions that cannot be exchanged, for instance, when working in a globally distributed project setting. For this reason, the use of standardized *software process metamodels* (SPMM), corresponding tool infrastructures, and methodologies to implement and maintain such software processes have become an indispensable means. However, an SPMM for its own is not enough to address the entire software process life cycle. To establish a systematic development of software processes, some supporting components are also required:

- A documented and formally described SPMM.
- A tool environment supporting process development.
- A methodical guideline to assist and guide process engineers in designing, developing, and publishing software processes, as well as in using the tools.

In the following, we refer to the combination of SPMMs, tools, and guidelines as *software process engineering framework* (short: process framework). Furthermore, all the aforementioned components should be subject to certain management activities, e.g. configuration management, change management, and release management to support a long-term evolution of the entire process framework.

Problem Statement. Although a large number of proposals — industrial, national, and international ones — compete for the favor of the process engineers and process users, evidence on the state of their dissemination is still missing. Existing contributions either focus on the evaluation of a specific approach or on isolated case studies with a limited scope. It remains unclear which of the available approaches are accepted in practice and which maturity they have to be effectively applied in industrial sensitive contexts.

Research Objective. To overcome the shortcoming stated above, we aim at systematically investigating the area of SPMMs to draw a big picture of the state-of-the-art in systematic software process development.

Contribution. We contribute a literature review on the state-of-the-art of SPMMs. We analyze which metamodels were published over the years and what their practical relevance is. This allows us to distill a notion of the state-of-the-art and the maturity of SPMMs to foster further problem-oriented research in this area.

Outline. In Sect. 2, we discuss work related. We introduce the study design in Sect. 3, before discussing the results of the study in Sect. 4. In Sect. 5, we conclude the paper with a discussion of our results and future work.

2. RELATED WORK

In this section, we give a brief overview of related work. Bendraou et al. [4] compare several modeling techniques for software processes, namely SPEM 1.1 and 2.0 [16], DiNitto's approach, Chou's Approach, UML4SPM [3], and the PROMENADE Approach. The study addresses concepts and capabilities of the selected modeling approaches. A comparable study is done by Henderson-Sellers and Gonzales-Perez [8], where OPEN/OPF [7, 5], SPEM, the LiveNet approach, and OOSPICE were considered. Those contributions focus on the analysis of metamodels that can be used to describe software processes in general. They pay, however, little attention to the feasibility and the practical relevance of the approaches under consideration. Another study by Ruiz-Rube et al. [18] focuses on the feasibility of the SPEM metamodel and investigates the use of this metamodel in a mapping study. In this study, the authors investigate the use and the extension of SPEM and discuss more than hundred contributions. However, the investigation of the feasibility of SPMMs was also not in scope. Martínez-Ruiz et al. [15] also focus on SPEM. In contrast to the other contributions, they focus on the opportunities to model variability of software processes using SPEM and present a case study. Martínez-Ruiz et al. are representatives for a number of contributions that focus on a particular SPMM-related aspect, e.g. [3, 2, 17, 1].

In summary, existing contributions either focus on the application of a particular standard and provide smaller case studies with a limited scope [6], or they stay on a philosophical level[1], e.g. as found in [9] where a new notation is proposed, or in [8] where the need for the ISO 24744 [10] is motivated. Therefore, available studies focus either on general comparisons of basic concepts or on the evaluation of a particular SPMM. In this paper, we delimit from any comparison of SPMMs, their potential advantages or flaws. Instead, our intention is to focus on the analysis of the state-of-the-art in systematic process engineering using SPMMs in general. Our complete investigation, including detailed descriptions of the study design, analysis procedures, and all background data can be taken from our technical report [20].

3. STUDY DESIGN

We design the study by combining methods used for a systematic mapping study and ones used for a systematic literature review (SLR) to conduct an in-depth analysis of the previously structured publications (Kitchenham et al. [11]). After defining the research questions, we describe how we select the case and how we collect and analyze the data, before concluding with a discussion.

3.1 Research Questions

Our overall goal is to give a big picture of the state-of-the-art in systematic process engineering. To this end, we formulate the following research questions:

RQ 1 Which metamodels were published over the years?
RQ 2 What is the practical relevance of the metamodels?

To answer RQ 1, we conduct a literature review and screen the resulting contributions for software processes that are created using an SPMM. All SPMMs are collected in a list for further investigation. To answer RQ 2, we analyze the SPMMs and evaluate their practical relevance, whereas we define the term "relevance" using the following criteria:

Critria	Description
Evolution	An SPMM should have a development history and underly a proper management (e.g. configuration and change management).
Guidance	An SPMM should support process engineers with methodical guidelines for software process design, implementation, deployment, and management, and should be complemented by supporting material, e.g. coaching/training programs, certifications, knowledge exchange in a community and documentation that goes beyond plain metamodel documentation.
Tools	Since today's software processes can be of considerable size, a comprehensive tool support is necessary to create, deploy, and manage software processes or process variants.
Evidence	The feasibility of an SPMM should be proven via concrete implementations of software processes, e.g. RUP, Scrum or XP. In addition, we search for empirical evidence on the practical application. The feasibility should be proven in a scholarly fashion in academia and practice including reference implementations.

3.2 Case Selection

The case selection is opportunistic and is based on whether available approaches allow to answer our research questions. We exclude vendor-specific tools that do not deal with the widely accepted standards, but only implement proprietary models. We aim at identifying standards that are used to design software processes, which are represented as computable models and, therefore, support tool-based design, development, and management. SPMMs should be standardized and accessible to allow for a broad application.

3.3 Data Collection Procedures

We opt for an incremental data collection. The search is a combination of automated search strategies (cf. systematic literature reviews [11]) and "snow-balling" procedures. To initialize the search, we use the following query:

metamodel ***or*** *(metamodel* ***and*** *software engineering)* ***or***
(metamodel ***and*** *development process)* ***or***
(metamodel ***and*** *software development process)*

The query has been used in the following databases:

- Web of Knowledge
- ACM Digital library
- IEEE Xplore
- Google/Google scholar

Having the initial result set in place, we incrementally check the contained contributions, and, if a contribution deals with software processes and SPMMs, we further check the reference sections of the papers. After this first screening, the initial result set is cleaned and contains only contributions relevant for further investigation (primary sources). Each paper's reference section is then analyzed for further potentially relevant contributions including books and standards when cited in the paper (snow-balling). If there is a relevant contribution in a reference list, the cleaned result set is checked, whether it already contains the newly considered paper. If the newly considered paper is not in the result set,

[1]Cf. research type facets as defined by Wieringa et al. [23].

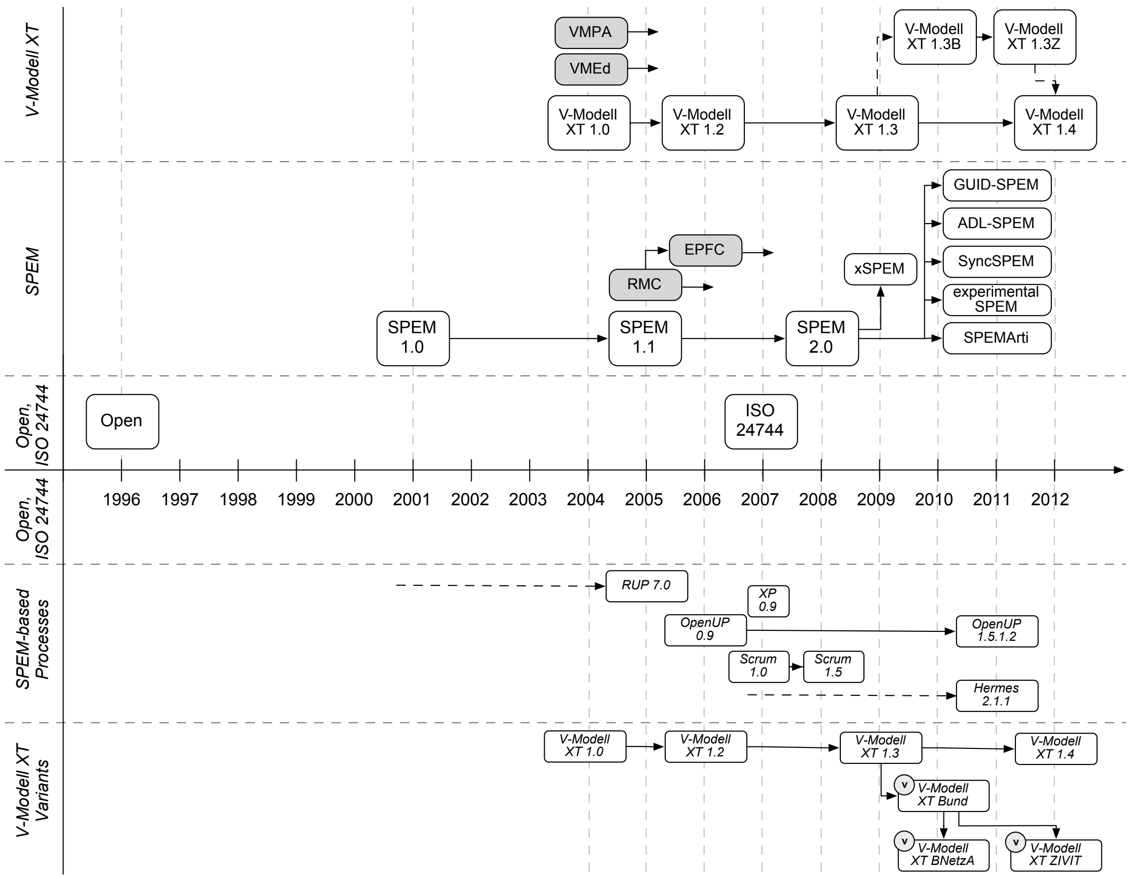

Figure 1: SPMMs, their evolution, releases of the tools, and selected reference implementations.

it is appended to the list of relevant papers. The corresponding list item is attributed with *new*. The overall procedure is repeated until all list items in the result set are marked as *checked*.

3.4 Analysis Procedures

To answer RQ 1, we perform the following procedure:
Step 1: Identify all metamodels from the collected data.
Step 2: Identify all metamodels that are computable.
Step 3: Investigate the evolution of the metamodel history.
Step 4: Create a time line from the publication dates.
The first two steps are preparatory ones in which we screen the result set and investigate the metamodels and their computability and history to be further investigated in step 3. In step 4, we create a time line for each metamodel to plot its evolution including the ones of all its variants and versions. To answer RQ 2, we check the resulting metamodels' environments according to the criteria introduced in Sect. 3.1.

3.5 Validity Procedures

As we rely on a systematic literature review, we follow the validity procedures established in this area. We furthermore rely on researcher triangulation for the in-depth analysis performed to answer our research questions. Finally, as another validity procedure holds the combination of regular SLR techniques with snow-balling to overcome the problem of a blurry terminology given in respective field.

4. RESULTS

We present our results and a first interpretation.

4.1 Contributions over Time (RQ 1)

Figure 1 shows the integrated results from the literature review. After filtering the result set, only ISO 24744 [10], OPEN [5], SPEM [16], and the V-Modell XT metamodel [21] remain for the in-depth analysis. A short description of all these metamodels as well as a discussion on the consideration of a particular metamodel can be found in [20]. Figure 1 shows the releases and publication dates of the remaing metamodels. We depict OPEN and SPEM 1.0 to be the first comprehensive metamodels published in 1995/1996 and 2001. Starting in 2004, the V-Modell XT was initially released and continuously maintained and extended up to today. In 2005, SPEM was updated (rel. 1.1), and, finally, published as an OMG standard in 2008 (rel. 2.0). In 2007, the ISO standard 24744 (SEMDM) was published. SEMDM evolves—taking into account the structure of the metamodel and the terminology used— from OPEN. The data shows that no literature is available to indicate further developments of OPEN and ISO 24744. The in-depth analysis only shows five papers that refer to these metamodels, most of them discussing concepts and solution proposals in the area of (situational) method engineering [14].

In contrast to OPEN and ISO 24744, literature suggests an evolution of SPEM and of the V-Modell XT over the years. The evolution of SPEM is straight forward until the release 2.0. Several metamodel extensions followed, e.g. to support enactment [2] or to support the deeper incorporation of artifacts [19]. Those contributions are based on SPEM 2.0, but no information is available whether those proposals will be considered when further developing SPEM.

In contrast to SPEM, we find no explicit releases of a new metamodel for the V-Modell XT. New versions are bound to new V-Modell XT variants and there is no explicit infrastructure update. A branch in the metamodel's development is caused by the development of comprehensive software process variants [20], and, furthermore, this branch was integrated in the further development of the reference process.

4.2 Practical Relevance (RQ 2)

In order to evaluate the practical relevance of an SPMM, we investigated the SPMMs according to the aforementioned criteria. The results are summarized in Table 1.

Table 1: Evaluation of the practical relevance.

Criteria	OPEN	ISO 24744	SPEM	V-Modell XT
Evolution	○	○	●	●
Guidance (t/m/s)	○/●/●	○/●/○	●/●/○	●/●/●
Tools	○	○	●	●
Evidence (a/p/r)	●/●/○	●/○/○	●/●/●	●/●/●

Evolution. Our results show no further developments for OPEN and ISO 24744, but the initial contributions (Fig. 1). On the other hand, SPEM as well as the V-Modell XT metamodel explicitly undergo an evolution. Since Fig. 1 only contains a condensed view, the complete "family trees" that draw a big picture of the metamodel evolution as well as the associated software processes can be found in our report [20].

Guidance. As shown in Table 1, we distinguish technical guidance (t) in which the handling of the infrastructure is described, methodical guidance (m), e.g. on decision making w.r.t. which modeling concepts to choose in certain settings, and supporting material (s) that addresses, e.g. training or certification. SPEM as well as the V-Modell XT provide guidance that addresses the craftsmanship part (how to install and handle the infrastructure). For SPEM-based processes, a comprehensive tutorial is available [22]; the corresponding documentation for the V-Modell XT is published as a report [21] and as a book [12]. A corresponding documentation for OPEN and ISO 24744 is not available. All SPMMs provide methodical guidance on different levels of abstraction. For instance, while the SPEM guidance is rather technical containing only a few methodological hints, the V-Modell XT guidance explicitly supports process engineers, e.g. in selecting modeling approaches. ISO 24744 and OPEN also provide some brief and technical guidance. Only the OPEN and the V-Modell XT metamodel provide rich supporting material, e.g. books on OPEN [5, 7], several books on the V-Modell XT (i.e. [12]), a complete training curriculum and a certification program for the V-Modell XT. For ISO 24744 and SPEM, documentation going beyond the metamodel documentation and online tutorials is missing.

Tools. In the upper part of Fig. 1, the initial release dates of the supporting tools are assigned to the corresponding metamodel version. The results show that for SPEM the *Rational Method Composer* (RMC) and its free equivalent the *Eclipse Process Framework Composer* (EPFC) support process authoring, deployment, and management. For the V-Modell XT, two tools are part of the process framework: the Editor (VMEd) supports process authoring, deployment, and management tasks. The Project Assistant (VMPA) serves the initial project-specific tailoring incl. initial planning and the generation of document templates. We could not find comparable support for OPEN and ISO 24744.

Evidence. We differentiate academic (a) evidence, practical (p) evidence, and evidence given by a reference implementation (r). Reference implementations are shown in Fig. 1 (lower part). OPEN comes with contributions in the academic and in the practical field including solution proposals and case studies. However, apart from the process description provided in [5, 7], there still exists no reference implementation. Academic evidence can also be found for ISO 24744, e.g. in [6]. However, no industrial case study nor a reference implementation can be found also here. For SPEM and the V-Modell XT, a number of contributions evaluating the reference implementation as well as the metamodels are available, e.g. [13, 18].

5. CONCLUSION

In this paper, we analyzed the state-of-the-art in systematic software process development and showed which standards have been disseminated to which extent over the past years. We could see, for example, that OPEN and ISO 24744, although being proclaimed as an industry standard, have to the best of our knowledge no practical relevance as they remained at the level of a complex metamodel specification. They neither have an observable active community in the sense of developing new releases or tools to support process engineers and users, nor are any documented experiences available that would indicate to a certain dissemination.

In contrast, such activities are observable in the context of the V-Modell XT. The maintenance and development of the V-Modell XT is triggered by a non-profit organization that fosters the activities with a strategical perspective. It remains, however, a purely German standard with little international attention. SPEM, in turn, also proposes a rich infrastructure as done for the V-Modell XT. It is a comprehensive framework that consists of the metamodel specification, reference processes, and the supporting tool infrastructure. Due to the nature of the OMG that coordinates the activities around SPEM, the contributions have a fairly technical level. SPEM builds a backend, which is frequently used, especially in academia for exploration and prototyping. We see SPEM to be disseminated as it also comprehends reference processes that are applied in practice [18]. On the other hand, the only SPEM-based processes that seem to be continuously maintained are OpenUP (as a beverage to EPF) and Hermes (as a national Swiss standard; cf. Fig 1). The current state and future of the RUP family remains unclear; all other EPF process plug-in development seems to be stopped since 2008/2009. Current developments are mostly situated in academia, whereby it seems that much conceptual work is done, but a strategy is yet missing. It thereby remains unclear whether and how all related proposals will be considered in the further development of SPEM.

In summary, so far it seems that the only process ecosystems that are continuously maintained and developed are national standards driven by organizations in which political interests and goals are reflected and where fundings for the processes' maintenance are made available.

Therefore, software process development has reached a certain maturity resulting in various SPMMs with at least two active communities. Yet, it becomes evident that further problem-driven research is necessary, for example, regarding the demands pinpointed by available proposals. Figure 1 illustrates the domains of interests, e.g. enactment.

Future Work. As our results show, there is still a strong demand for further problem-driven research on SPMMs in particular and process frameworks in general. Investigations include, inter alia, reasoning about why selected standards remain unused, and whether and how the feasibility of those standards could be improved. Another aspect considers the fundamental modeling paradigms, i.e. the focus on artifacts and their dependencies and/or the focus on methods. The ISO-24744, for example, already breaks with commonly known and established (MOF-based) modeling paradigms by introducing the Powertype Pattern. It is, however, still unknown to which extent software processes can benefit from such initiatives. Therefore, a relevant topic is an in-depth analysis of SPMMs to understand the strengths and limitations of selected approaches, e.g. with a family of comparative studies. This finally allows us to infer a roadmap for further improvements in our field.

6. REFERENCES

[1] F. Aoussat and M. Oussalah. SPEM Extension with Software Process Architectural Concepts. *35th Annual Computer Software and Applications Conference*, 2011.

[2] R. Bendraou, B. Combemale, X. Cregut, and M. P. Gervais. Definition of an Executable SPEM 2.0. In *Asia-Pacific Software Engineering Conference*, 2007.

[3] R. Bendraou, M.-P. Gervais, and X. Blanc. UML4SPM: An Executable Software Process Modeling Language Providing High-Level Abstractions. In *10th IEEE International Enterprise Distributed Object Computing Conference*. IEEE, 2006.

[4] R. Bendraou, J. M. Jézéquel, M. P. Gervais, and X. Blanc. A Comparison of Six UML-Based Languages for Software Process Modeling. *Software Engineering, IEEE Transactions on*, 36(5):662–675, 2010.

[5] Firesmith, D. G. and Henderson-Sellers, B. *The OPEN Process Framework: An Introduction*. Addison-Wesley Longman, 2001.

[6] Gonzalez-Perez, C. *Situational Method Engineering: Fundamentals and Experiences*, volume 244 of *The International Federation for Information Processing*, chapter Supporting Situational Method Engineering with ISO/IEC 24744 and the Work Product Pool Approach. Springer, 2007.

[7] Graham, I., Henderson-Sellers, B., and Younessi, H. *The OPEN Process Specification*. Addison-Wesley, 1997.

[8] B. Henderson-Sellers and C. Gonzalez-Perez. A comparison of four process metamodels and the creation of a new generic standard. *Information and Software Technology*, 47(1):49 – 65, 2005.

[9] B. Henderson-Sellers and C. Gonzalez-Perez. Standardizing Methodology Metamodelling and Notation: An ISO Exemplar. In *2nd International United Information Systems Conference*, 2008.

[10] Joint Technical Committee ISO/IEC JTC 1, Subcommittee SC 7. Software engineering – metamodel for development methodologies. Technical Report ISO/IEC 24744:2007, International Organization for Standardization, 2007.

[11] B. Kitchenham, O. P. Brereton, D. Budgen, M. Turner, J. Bailey, and S. Linkman. Systematic literature reviews in software engineering – A systematic literature review. *Information and Software Technology*, 51(1), 2009.

[12] M. Kuhrmann, T. Ternité, and J. Friedrich. *Das V-Modell XT anpassen*. Springer, 2011.

[13] Kuhrmann, M., Lange, C., and Schnackenburg, A. A Survey on the Application of the V-Modell XT in German Government Agencies. In *18th EuroSPI*, 2011.

[14] Kuhrmann, M., Méndez Fernández, D., and Tiessler, M. A Mapping Study on Method Engineering – First Results. In *International Conference on Evaluation & Assessment in Software Engineering*, 2013.

[15] T. Martínez-Ruiz, F. García, M. Piattini, and J. Münch. Modelling software process variability: an empirical study. *IET Software*, 5(2):172, 2011.

[16] OMG. Software & Systems Process Engineering Metamodel Specification (SPEM) Version 2.0. Technical report, Object Management Group, 2008.

[17] A. Rochd, M. Zrikem, A. Ayadi, T. Millan, C. Percebois, and C. Baron. SynchSPEM: A synchronization metamodel between activities and products within a SPEM-based Software Development Process. In *International Conference on Computer Applications and Industrial Electronics*. IEEE, 2011.

[18] I. Ruiz-Rube, J. M. Dodero, M. Palomo-Duarte, M. Ruiz, and D. Gawn. Uses and Applications of SPEM Process Models. A Systematic Mapping Study. *Journal of Software Maintenance and Evolution: Research and Practice*, 1(32), 2012.

[19] M. Silva and T. Oliveira. Towards detailed software artifact specification with SPEMArti. In *International Conference on Software and Systems Process*, 2011.

[20] Steenweg, R., Kuhrmann, M., and Méndez Fernández, D. Software Engineering Process Metamodels – A Literature Review. Technical report, TUM, 2012.

[21] Ternité, T. and Kuhrmann, M. Das V-Modell XT 1.3 Metamodell. Technical Report TUM-I0905, 2009.

[22] Tuft, B. *Eclipse Process Framework (EPF) Composer – Installation, Introduction, Tutorial and Manual*. `http://eclipse.org/epf/general/EPF_Installation_Tutorial_User_Manual.pdf`, 2010.

[23] R. Wieringa, N. Maiden, N. Mead, and R. Colette. Requirements engineering paper classification and evaluation criteria: a proposal and a discussion. *Requirements Engineering*, 11(1):102–107, 2005.