Jeners, S., O'Connor, R.V., Clarke, P., Lichter, H., Lepmets M. and Buglione, L., Harnessing Software Development Contexts to inform Software Process Selection Decisions, Software Quality Professional, Vol. 16, No. 1, December 2013.

Harnessing Software Development Contexts to Inform Software Process Selection Decisions

Simona Jeners, RWTH Aachen University, Germany

Rory V. O'Connor, Dublin City University, Ireland

Paul Clarke, Dundalk Institute of Technology, Ireland

Horst Lichter, RWTH Aachen University, Germany

Marion Lepmets, Dundalk Institute of Technology, Ireland

Luigi Buglione, ETS Montréal/Engineering IT SpA, Italy

Abstract

Software development is a complex process for which numerous approaches have been suggested. However, no single approach to software development has been met with universal acceptance, which is not surprising, as there are many different software development concerns. In addition, there are a multitude of other contextual factors that influence the choice of software development process and process management decisions. The authors believe it is important to develop a robust mechanism for relating software process decisions and software development contexts. Such an approach supports industry practitioners in their efforts to implement the software development processes vital for a particular set of contextual factors. In this paper, the authors outline a new tool-based framework for relating the complexity of software settings with the various aspects of software processes. This framework can extract the key software process concepts from process repositories, for example, from CMMI-DEV or ISO/IEC 15504-5 (a.k.a. SPICE – Software Process Improvement and Capability dEtermination). A team of software *development experts then collaborates in order to identify and validate the strength and nature of the relationship between the key process concepts and the contextual factors that are known to affect the software development process. The result of this collaboration is a prototype of a flexible model, which can be extended over time into a broader process consideration, for example, where agile processes or further specific situational factors could be added to the framework. The authors contend that a model such as the one proposed in this paper can serve as a valuable tool, assisting software developers in making decisions regarding the selection of software best practices, as well as providing general guidance for process improvement initiatives.*

Key words: best practice selection, CMMI, process repositories, situational factors

INTRODUCTION

Best practices have been documented through models and/or standards of processes for different disciplines such as software and systems engineering, information technology, or information systems. Most models and standards, such as ISO/IEC 15504-2 (SPICE) (SPICE 2010) or the CMMI-DEV (CMMI Product Team 2010), address a common purpose: to make available a set of generic processes (technical, managerial, support, and enterprise) that come from the best international practices to guide and improve organizational process, with the expected outcome to preserve, correct, and ultimately improve the quality, value, and cost-efficiency issues of the resulting products and services.

Due to the myriad of available models and standards -- henceforth referred to in this paper as process repositories (PRs) -- the authors argue that organizations have difficulties with the correct understanding and adoption of such PRs. “In the current marketplace, there are maturity models, standards, methodologies, and guidelines that can help an organization improve the way it does business" (CMMI Product Team 2010).

While there are many comparative studies on standards and models of processes (for example, on a more higher-level, such as (Sheard and Lake 1998; Minnich 2002; Mora et al. 2008; Ferreira, Machado and Paulk 2010), and on the fine granular level of practices, such as (Ferchichi and Bigand 2008; Liao, Qu, and Leung 2005; Wang et al. 1999; Malzahn 2009; Pardo et al. 2011; Soto and Munch 2008; and Jeners and Lichter 2013), which have identified core similarities and differences, only some research has been done to assist software developers and managers in making decisions regarding the selection of appropriate processes (or their alignment with specific process repositories).

The selection of appropriate processes for a project can be done by considering the project's context, and this has been widely accepted. The international standard ISO/IEC 12207 (ISO/IEC 2008, 12) states "any project is assumed to be conducted within the context of an organization," and furthermore, "this is important because a software project is dependent upon various outcomes produced by the business processes of the organization, for example, employees to staff the project and facilities to house the project." ISO/IEC 12207 also recommends that the sequencing of software development stages should be "appropriate for the project's scope, magnitude, complexity, changing needs, and opportunities." CMMI-DEV or COBIT (ISACA 2011) adopt a similar position to ISO/IEC 12207, recommending that various contexts should be considered when implementing processes."

Furthermore, other contributions also stress the role of the project context in software development process decisions. Boehm and Turner (2004, 7) suggest that when it comes to software development processes, it seems likely that the claim "one size fits all" is in fact a myth. Jones (2007, 13) further argues that the available evidence suggests that no single approach to software development "is universally deployed or even universally useful." The reason for such lack of universal utility of any single approach to software development is related to the basic requirement of a software development process: that it “should fit the needs of the project” (Feiler and Humphrey 1992, 6). Software development processes are influenced by the context in which the project operates and, therefore, the optimal software development process should be “contingent on the context” (Benediktsson, Dalcher and Thorbergsson 2006, 97) and “best fit the conditions, product, talent, and goals of the markets and organizations” (Subramanian et al. 2009). Kautz (1998) also shares this view, stating that process improvement initiatives should be “adjusted to their particular situation and…should not slavishly follow one of the comprehensive approaches.”

The aforementioned argument demonstrates that both established PRs and recognized software process academics acknowledge that a software process should be designed so as to address the context within which the process operates. Therefore, when PRs are considered as a guideline to design software processes, some PR components are more critical than others and must be intensively addressed. It therefore seems reasonable to state that this requirement can be generalized to other process improvement initiatives, such as initiatives based on COBIT, ITIL (Steinberg et al. 2011), or CMMI-SVC. However, in this paper the authors focus on the software development process improvement initiatives.

Different authors (Xu and Remesh 2007; Petersen and Wohlin 2009; Dede and Lioufko 2010; Bekkers et al. 2008; Clarke and O’Connor 2012) propose to “evaluate a wide range of contextual factors before deciding on the most appropriate process to adopt for any given project” (MacCormack and Verganti 2003). The work of Clarke and O’Connor (2012) is best grounded in the earlier related publications and offers a comprehensive listing of situational factors that can be used to characterize the context of a project. Therefore, the authors can consider these situational factors to describe the context and select appropriate practices for it. Many of these situational factors not only describe the software development context, but they are also valid for other process improvement initiatives (for example, from the “personnel” category the situational factors “personnel cohesiveness” and “customer satisfaction”). Therefore, organizations interested in PRs, such as COBIT or ITIL, can also consider these situational factors to design their processes.

The task of aligning components of PRs with situational factors is necessarily complex, as the number of PRs, their components, and situational factors are large and sometimes interrelated. This work proposes a framework -- model-based selection, adoption and assessment of improvement concepts (MoSAIC) -- to systematically support this alignment and thus, the selection of practices from multiple PRs based on the situational factors.

As the framework’s name suggests, the selection is only one aspect of MoSAIC. It addresses further challenges to support organizations, such as a practice comparison or the identification of dependencies between practices over the border of a single PR (Jeners and Lichter 2013). Therefore, it offers an integrated view on PRs for the adoption and assessment of one or more PRs.

Next, the authors describe the MoSAIC framework and how it supports the practice selection. Afterward they describe the steps performed to validate MoSAIC. After the validation, they summarize the benefits of the MoSAIC framework, indicating also the limitations that need to be addressed in the further work. Finally, the authors close with their conclusions in the last section.

MoSAIC FRAMEWORK

In this section the authors provide an overview of the MoSAIC framework. The framework defines meta-models to integrate different PRs and situational factors, allowing an automated selection of PR practices based on the project context. Furthermore, the authors give more details about the structure and semantics of the MoSAIC models and modelling activities. Finally, the mapping of practices to situational factors is described. Additional information about the MoSAIC framework can be found in (Jeners and Lichter 2013; Jeners, Lichter, and Dragomir 2012; Chen, Staples, and Bannerman 2008).

Overview and Parts of the Framework

The MoSAIC framework contains three main parts that are outlined in Figure 1: 1) tool-supported model operations; 2) meta-models and their respective instance models; and 3) modeling activities to build up and relate the different models: integrated structure models (ISMs) – one for each PR -- integrated concept model (ICM), and situational factor model (SFM).

The central model of MoSAIC is the ICM, which integrates the various PRs at a conceptual level, resulting in a common terminology enabling the harmonization of the previously disparate PRs and the various situational factors. This harmonization allows one to automatically identify PR practices that are needed for addressing different situations in software development settings.

Figure 2 illustrates in greater detail the structure and relations of the concrete MoSAIC models and how these models are created. First, the authors briefly introduce each PR. Then they describe how they were transformed and mapped into the respective ISMs (Part I). Afterward they present their subsequent elaboration into the central ICM (Part II). Finally, they briefly describe how the concepts stored in the ICM are related to the situational factors (Part III). Details about how ICM is related to SFM and the tool-supported model operations to map practices to situational factors are given in a further section.

PRs are organized by applying different structures as well as different terms for the same structural elements. For example, a group of processes addressing the same topic is called a *domain* in COBIT; in CMMI it is entitled *category*. Processes are called *process areas* in CMMI and *processes* in COBIT, SPICE, or ITIL. Furthermore, PRs are written on different levels of abstraction. The authors found similarities between COBIT control objectives; COBIT control practices; CMMI specific-goals, generic-goals, practices, subpractices; SPICE practices; and IEC 61508 objectives and requirements by comparing their outputs, inputs, and roles.

- **Part I:** As PRs have different structures, the authors transform each PR according to the IS meta-model to normalize its structure into a corresponding ISM, thus, *manually extracting* from the PRs elements, such as categories, processes, or practices, or practice elements such as activities, roles, artefacts (outputs or inputs), and purposes. A description and examples of these elements can be found in (Jeners, Lichter, and Dragomir 2012; Jeners and Lichter 2013). Based on an analysis of the writing styles, the authors defined further guidelines and a parsing tool to *automatically extract* these practice elements (Jeners, Lichter, and Dragomir 2012). As the automated results are not always correct due to sentence complexity, and as several important artefacts (for example, inputs that are needed to produce a certain output) are not explicitly defined in the practice text but in their description, a human expert has to validate or *manually correct* the extracted elements.

- **Part II:** To *elaborate concepts based on practice elements,* the authors map each ISM output, input, role, and purpose to the concepts in the ICM. A concept is a word or the smallest combination of words contained in a procedure that has a unique meaning in the context of PRs (for example, project plan or work breakdown structure). The ICM does not contain aggregated concepts, such as “software key stakeholder,” but contains the two basic concepts “software stakeholder” and “key stakeholder.” If a practice element is an aggregated concept, then it will be related to every ICM concept that expresses this aggregation. ICM concepts are related in ICM by generalizatioOf- and composedOf-relations and form generalizationOf-hierachies (see Jeners and Lichter 2013 for further details) to structure the ICM and support its maintainability.

- **Part III:** *Relating situational factors to concepts* creates a mapping between the SFM situational factors and the ICM concepts. MoSAIC defines four different semantic relations to support the mapping between the situational factors and the concepts. These relations also reflect the mapping strength between the situational factors and the concepts and, thus, support the automated selection of appropriate practices for a certain context characterized by a situational factor. As this mapping cannot be done automatically, human experts must be involved. The higher the number of experts involved in this mapping, the better the quality of the mapping. This, in turn, allows MoSAIC to provide better support for the selection of practices based on situational factors. The MoSAIC way to relate situational factors to concepts is the result of an intensive collaboration between a group of PR researchers.

To relate situational factors to concepts, the generalizationOf-hierarchy trees in ICM must be considered. The situational factors are related to the most abstract concepts. Therefore, the relation will also apply to its children and parents in the tree. This helps one avoid defining unnecessary relations, and thus, mapping efficiency is increased. However, when the relation does not apply for one of the children on a certain level in the generalizationOf-tree, then the situational factor must be related to all corresponding children. Similarly, the relation will also apply to their children and parents. There is also an exception. Sometimes the children and parents should not be considered (for example, the “personnel disharmony” is strongly managed by “ommitted requirement” and not by its abstract parent “requirement”). MoSAIC marks this exception and handles it in a different way

Mapping Practices to Situational Factors

One of the authors’ goals is to systematically map practices to situational factors and, thus, to select appropriate practices for a certain context characterized by situational factors.

They performed a case study in collaboration with researchers to analyze when practices are mapped to situational factors and mapped eight situational factors to all practices in CMMI-DEV of level 2 and 3 (see online supplement for further details).

Based on the discussions of the researchers and on the analysis of the manual mappings in the case study, the authors propose to map practices to situational factors based on the *modeling of relations between situational factors and concepts.*

MoSAIC mapping is based on two operations: 1) the modeling of relations between situational factors and concepts; and 2) mapping practices to situational factors operation.

First, the discussions during the collaborative workshops indicated that the mapping strength between the practices and a situational factor is derived from the mapping strength between the practice concepts and the situational factor (for example, practice "Establish and maintain a definition of required functionality and quality attributes" is strongly mapped to "application performance" because of the concept "quality attribute"). Therefore, MoSAIC uses an ordinal scale to map situational factors to concepts. In contrast to the manual mappings performed with the collaboration partners, MoSAIC uses only a three- (not four) point ordinal scale (2 = strong, 1 = medium, 0 = absent), as a differentiation between moderate and weak is too small and has no value for the process adoption. This decision was supported by several software process improvement experts from industry and research.

Second, the experts had difficulties indicating the mapping strength between the practices and the situational factors. A definition of medium and strong mapping strength is needed. Therefore, the authors define four relations to semantically enrich the mapping strength between concepts and situational factors:

- *Concerns:* A situational factor is strongly related to a concept if the situational factor *concerns* a concept (for example, "requirements rigidity" concerns "requirement").

- *Strongly manages*: A situational factor is strongly related to a concept if the situational factor is *strongly managed* ("requirements changeability" is strongly managed by "analyzed change request") by the adoption of the concept.

- *Manages:* The relationship between a situational factor and a concept is *medium* if the situational factor *is managed* by the adoption of the concept ("requirements changeability" is managed by "traceability matrix").

- *Influences*: The relationship between a situational factor and a concept is *medium* also when the situational factor influences the adoption of a concept ("requirements changeability" influences the "project plan").

Figure 4 visualizes an example of the mapping between concepts and situational factors based on a three-point ordinal scale and the semantical enrichment of the mapping.

The *mapping of practices to situational factors operation* uses the mappings among ISMs, ICM, and SFM to automatically identify important practices based on situational factors. Shortly, it identifies all related ICM concepts of the situational factor and then the corresponding ISM artifacts responsible for their practices. According to the relation between ICM concepts and situational factors, the practices are categorized as strong or medium.

Figure 4 depicts the steps by giving some examples of different mapping relations between situational factors and concepts (abstract or children), and between concepts and practices outputs/inputs. For example, "personnel disharmony" is strongly managed by the abstract concept "guideline for managing teams." Its children are mapped to the ISM inputs "guidelines for structure teams" or output "maintained guideline for structure teams" that are contained in the practice organizational process definition (OPD) SP1.7 of CMMI-DEV. Furthermore, the practice "requirements management" (REQM) SP1.2 of CMMI-DEV is also selected, as the "personnel disharmony" is strongly managed by the concept "committed requirements." Furthermore, the relations between "personnel disharmony" and "project goals" or "maintained goals" do not lead to the selection of practices, as CMMI-DEV does not contain inputs/outputs that are related to these concepts.

To summarize, MoSAIC systematically maps practices to situational factors by considering the practices' concepts and their different relations (concerns, strongly manages, manages, or influences) to the situational factors.

VALIDATION

In this section, the authors present the validation performed that shows that the MoSAIC framework is adequate to support the different operations, such as the mapping between practices and situational factors or comparison of practices used for the selection, adoption, and assessment of multiple PRs. Furthermore, they show that MosAIC is extendable by performing the different operations on PRs (for example, COBIT or ITIL) for further IT domains. Several experts participated in the validation activities to acquire broader feedback, thus improving MosAIC (for further details see the online supplement).

In the following, the authors describe the validation activities by referring to the MoSAIC modeling activities. Figure 5 gives an overview of these activities and the experts involved.

In the modelling activity *extract manually*, ISM elements such as PRs, categories, and practices were extracted. As PRs are described on different levels of abstractions and use different identifiers for practices, the authors validated this extraction (validation activities 4, 5, and 6). The validation showed that the different element types, such as control objectives and practices in COBIT, practices and goals in CMMI, or requirements in IEC 61508, are to some extent similar and can be compared.

In the modelling activity *extract automatically,* practice elements were extracted from the corresponding practice descriptions based on grammatical rules, on certain prepositions, and on further GATE words databases (for example, person database). In validation activities 1 and 2, three participants (one researcher and two master students) manually modelled the practice elements and compared the results with the automated extraction.

Validation activity 1 Practices written in different writing styles (containing verbs in passive, perfect continuous, or modal form; gerunds; and nominalizations), and also containing different concept types (activities, inputs, outputs, roles, and purposes), were first manually identified (14 CMMI-DEV, 18 COBIT, and 15 IEC 61508 practices). The automatically extracted results were then compared to the manual extraction. The deviation showed that the results are promising (Jeners, Lichter, and Dragomir 2012), but still need to be improved. The authors improved not only the grammatical rules the tool was based on, but also the prepositions and the GATE person database to allow a better extraction of inputs and roles.

Validation activity 2 Entire processes from CMMI-DEV, SPICE, COBIT, and IEC 61508 were randomly selected (110 practices totally). The authors worked with these PRs, as at that time the authors had experience with them. The automated extraction was compared to the manual extraction, and they calculated the deviation by using two metrics from pattern recognition and information retrieval theory: precision and recall. They achieved good results (see, for example, Table 3). As the MoSAIC parser is mainly based on the syntactical analysis of the sentences, the identification of inputs and purposes led to weaker results (for example, as prepositions are not always used to introduce an input, the parser cannot differentiate between an input and an output artefact). Thus, the semantic is important and the databases were not large enough to solve this issue by a machine. Furthermore, the language was sometimes too complex for the automated extraction (large sentences with more than 25 words).

In the modelling activity *correct manually,* the authors did not only correct the automated extraction of practice elements but also modelled the practice’s inputs and outputs that are not explicitly mentioned in the practice description. Some PRs mention or list artefacts that are needed or are produced by the practice activities (for example, in CMMI-DEV the typical work products, in SPICE the work products). Without modelling the artefacts explicitly, the authors got a high deviation between the dependencies calculated by MoSAIC and the dependencies that were identified by collaborating participants (see validation activity 3).

The modelling activity *elaborate concepts based on practice elements* creates the ICM and maps the practice elements to the ICM concepts. To validate the ICM creation, its relations to the ISMs, and to the different MoSAIC processes performed on these models, two major activities were conducted: evaluation of *dependencies between practices,* and the *manual similarity mappings between practice activities pairs.* As the MoSAIC mapping process between practices and situational factors is based on ICM and its relations to ISMs, the validation activities 7 and 8 also validate this modeling activity. In all of these validation activities, 10 different participants were involved (three consultants, two industry partners, and five academic researchers).

Validation activity 3 To validate the relations between concepts and outputs/inputs, and the MoSAIC dependency identification operation, the authors identified the practice dependencies. To evaluate the dependencies, the authors verified dependencies between 31 practices within the CMMI-DEV processes REQM, MA, CM, PPQA, and SAM based on previously established relationships (Chen, Staples, and Bannerman 2008). The authors calculated the deviation as the number of missing dependencies divided by the total number of dependencies. First, they obtained a deviation of 0.5 (every second MoSAIC result deviates by one point from the experts' result) for 54 dependencies within four CMMI-DEV-processes. Finally, the modelling of implicit inputs and outputs led to a better deviation: 0.19. A deviation was expected, as the authors did not model the artefacts that were not specified in the practice descriptions (for example, MoSAIC did not identify that CMMI-DEV MA-SP1-2 "Specify measures to address measurement objectives" is dependent on CMMI-DEV MA-SP1-4 "Specify how measurement data are analyzed and communicated," and, as the authors did not model for CMMI-DEV MA-SP1-4 the outputs "updated measures" and "updated measurement objectives" that, according to the authors of the mentioned paper, are used as inputs in CMMI-DEV MA-SP1-2).

Validation activities 4 through 6 Furthermore**,** to validate the relations between concepts and outputs/inputs and the MoSAIC comparison operation, the authors compared the various PR practices. The MoSAIC framework uses similarity metrics derived from the similarity theory to compare the PRs' practices and calculates the similarity degree of the practice activities (**see 31 – AUTHOR: NAMES ARE MISSING FOR THIS REFERENCE IN REF. LIST. PLEASE ADD**; Jeners, Lichter, and Pyatkova 2012). First, the authors manually determined similar practices from CMMI-DEV, SPICE, COBIT, CMMI-SVC, and ITIL (they used mapping materials from International Software Consulting Group (ISCN) for the comparison of CMMI-DEV with SPICE, mapping materials provided by ISACA (ISACA 2011) for CMMI-DEV with COBIT and mapping materials from their cooperation partner for CMMI-SVC and ITIL). Second, they computed the similarity degree for 161 pairs of practice activities. As the calculated similarity degree can have a value in the range of [0, 1], the authors mapped their results to five categories: [1,1] as *identical***;** **[0.67, 1)** as *high;* [**0.3, 0.67) AUTHOR: ARE THERE SUPPOSED TO BE BOTH BRACKETS AND PARANTHESES USED IN THE SAME RANGE?** as *medium;* (0, 0.3) as *low;* [0,0] as *different*. This allows them to compare their results with the participants' results. The deviation is the number of incorrect results (the result category is not equal to the participants' category) divided by the number of compared pairs. The final results (0.25 for CMMI-DEV and COBIT, 0.26 for CMMI-DEV and SPICE, and 0.0 for CMMI-SVC and ITIL) indicate that on average less than every fourth metric result deviates from the given category. The deviations are mainly caused by missing mappings between ISM practice elements and ICM concepts and missing or inaccurate relationships between the ICM concepts.

The modelling activity *relate situational factors to concepts* maps the situational factors to the ICM concepts to perform the mapping between practices and situational factors. First, the relations between concepts and situational factors were derived from the relations between practices and situational factors performed by the academic researchers. Second, the authors performed a validation by comparing the results of the mapping operation to the *manual mappings between situational factors and practices*. The manual mappings were created in a collaboration with academic researchers. For eight situational factors, they obtained a deviation of 0.02 (on average less than every tenth result deviates from the two-point scale).

The reason for this deviation is that MoSAIC considers the inputs to identify best suited practices. Some of the manual mappings did not consider the inputs. A mapping exists between a situational factor and a practice when producing certain concepts, but there was no mapping when this concept was needed by a practice. A retrospective discussion on this issue with the participants pointed out that the participants considered it important to create a concept in the present (for example, “process improvements” in CMMI-DEV-OPF-SP1-3 “Identify improvements to the organization’s processes and process assets”), but not as important if and how this concept is needed in the future (CMMI-DEV-OPF-SP2-1 “Establish and maintain process action plans to address improvements”). To know if this is an issue, a broader validation is required.

DISCUSSION

There are many approaches to the complex activity of software development, with no single approach being universally applicable. A new framework, called MoSAIC, supports the integration of different PRs and situational factors that can be used to describe the context of a project. Based on the guidelines implemented in the MoSAIC framework, the authors modelled parts of multiple PRs (all 18 processes from CMMI-DEV maturity levels 2 and 3, 10 processes from SPICE, eight processes from COBIT, and parts from ITIL and CMMI-SVC). The authors extracted more than 1000 concepts that are grouped in about 100 different generalizationOf- trees. Due to the structure of the trees -- with their most abstract concept at the top and to the categorization of the subconcepts in the ICM -- it was relatively straightforward to insert new concepts, and to find and assign them to the practice elements and situational factors. Finally, the authors modelled eight situational factors and related them to the PRs' concepts.

The MoSAIC framework has various benefits to help support an organization's work with multiple PRs:

- **It is extendable:** New/changed PRs or further situational factors can be easily integrated into MoSAIC by defining their relations to ICM concepts. If the concepts do not exist, they must be created. Integration of new PRs and situational factors becomes easier. This is because the ICM already contains many of the corresponding concepts and few new concepts have to be added. Only the relations to the ICM concepts must be defined.
- **Its maturity grows**: The more PRs are integrated into MoSAIC, the better the quality of ICM and the easier it is to integrate new PRs. The PR's information helps the user to better understand the semantic of the concepts and their relations and, thus, improve the ICM and the relations of the ISMs and of the SFM to the ICM.
- **It is maintainable:** All the models (ISMs, SFM, and ICM) are saved in MySQL and XML to allow different users to maintain the database. The MoSAIC application offers two modalities to add/remove/edit the data: 1) interact directly with the MySQL database; 2) import/export XML data into/out of the MySQL database.

- **It enables model analyses:** The MoSAIC framework allows one to run different operations on the aforementioned models depending on the interests of the organizations that work with multiple PRs. The computation of the similarity degree of two or more practices, the identification of practice dependencies, the categorization of practices according to their output, and the selection of best-suited practices are only some examples that can be implemented based on the framework (Jeners and Lichter 2013).
- **It supports PR understandability:** The MoSAIC framework integrates different PRs and, thus, more information about a certain aspect is provided where PRs are overlapping. MoSAIC can determine the overlap based on the relation between ICM and ISMs. Therefore, the ICM acts as a dictionary, where the context for each concept is described in the PRs. This can lead to a better understanding of the concepts used in the PRs.

Although the MoSAIC framework is based on plenty of information from multiple PRs and from a well-proved reference framework of situational factors affecting the software development process, there are still some limitations.

First, a broader involvement of experts from research and industry on the construction and validation of the MoSAIC framework would increase the quality of the data and will give better results for the selection, adoption, and assessment of PR practices. Although the modelling of the ISMs, ICM, and their connection was implicitly evaluated by the practice comparisons and dependencies, an explicit review conducted by experts would increase the framework quality and thus the quality of the practice comparison, dependencies, and selection results. Furthermore, more PRs could be integrated into the framework to increase the scope of ICM and to offer support to different organizations. Finally, the automated selection of practices was validated based on the manual mappings between the situational factors and practices. The authors think different interpretations of the CMMI-DEV practices and the too general definition of the mapping strength could lead to wrong mappings. A mapping that is based on the MoSAIC relations between concepts and situational factors that involves more experts would increase the quality of the selection results.

Second, additional situational factors from other domains could be incorporated into a later version of a framework. For example, factors that influence how IT supports the business could be integrated into the framework. These could be then be mapped to COBIT-specific concepts and allow an automated selection of COBIT practices.

Finally, it would be useful to integrate agile software development activities into MoSAIC, as this might address a larger portion of the software development community.

CONCLUSIONS

An optimal approach to software development is regarded as being dependent on a wide variety of situational factors in individual software development settings, domains, and contexts (Clarke and O’Connor 2012). In addition, there are a wide variety of software process concepts contained within a collection of diverse process repositories that practitioners can use when attempting to make key process decisions. In the absence of published guidance with respect to such complex decisions, the authors have presented a robust framework for relating software process decisions and software development contexts to harness the power of disparate conceptual activities into holistic process decisions. Early evidence from industrial application suggests that the framework is of benefit in practice (for more information see the online supplement). Therefore, the authors are convinced that the MoSAIC framework can serve as a valuable tool for software development endeavors and specifically in assisting software process managers in making decisions regarding the selection of software best practices, as well as general guidance for process improvement initiatives.

REFERENCES

SPICE. 2012. SPICE User Group 2005-2010. ISO/IEC 15504-x, Information technology -- Process assessment – Parts 1-10, 2012 2003.

CMMI Product Team. 2010. *CMMI for Development, Version 1.3 (CMU/SEI-2010-TR-033)*. Pittsburgh: Software Engineering Institute, Carnegie Mellon University.

Sheard, S. A., and J. G. Lake. 1998. Systems engineering standards and models compared. In *Proceedings of the Eighth International Symposium on Systems Engineering,* Vancouver, Canada, 589–605.

Minnich, I. 2002. IA IS 731 compared to CMMISM-SE/SW. *Syst. Eng.* 5, no. 1:62–72.

Mora, M., O. Gelman, R. O’Conner, F. Alvarez, and J. Macías-Lúevano. 2008. A conceptual descriptive-comparative study of models and standards of processes in SE, SwE, and IT disciplines using the theory of systems. *International Journal of Information Technology Systems Approach* 1, no. 2:57–85.

Ferreira, A. L., R. J. Machado, and M. C. Paulk. 2010. Quantitative analysis of best practices models in the software domain. In *Software Engineering Conference (APSEC), 2010 17th Asia Pacific*, 433 –442.

Ferchichi, A., and M. Bigand. 2008. An ontology for quality standards integration in software collaborative projects. In *Proceedings of MDISIS’08 Model Driven Interoperability Sustain. Inf. Syst. Montp. Fr.*

Liao, Li, Yuzhong Qu, and H. K. N. Leung. 2005. A software process ontology and its Application. Presented in the *Proceedings of the Int.Workshop on Future Software Technology, IWFST-2005,* Shanghai.

Wang, Y., G. King, A. Dorling, and H. Wickberg. 1999. A unified framework for the software engineering process system standards and models. In *Proceedings of the Fourth IEEE International Symposium and Forum on*, 132 –141.

Malzahn, D. 2009. Assessing - learning - improving, an integrated approach for self-assessment and process improvement systems. In *Systems 2009. ICONS ’09. Fourth International Conference on*, 126 –130. **AUTHOR: WHAT WORD GOES AFTER "ON" HERE IN THE NAME OF THE CONFERENCE. I AM CONFUSED AS TO HOW TO REFERNCE IT.**

Pardo, C., F. J. Pino, F. García, M. Piattini, M. T. Baldassarre, and S. Lemus. 2011. Homogenization, comparison and integration: A harmonizing strategy for the unification of multi-models in the banking sector. In *Product-Focused Software Process Improvement*, D. Caivano, M. Oivo, M. T. Baldassarre, and G. Visaggio, eds. Springer Berlin Heidelberg, 59–72.

Soto, M., and J. Münch. 2008. Using model comparison to maintain model-to-standard compliance. In *Proceedings of the 2008 International Workshop on Comparison and Versioning of Software Models*, New York: 35–40.

Jeners, S., and H. Lichter. 2013. Smart integration of process improvement reference models based on an automated comparison approach. In *Systems, Software and Services Process Improvement* 364, F. McCaffery, R. V. O’Connor, and R. Messnarz, eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 143–154.

ISO/IEC 2008. ISO/IEC 12207:2008, Systems and software engineering - Software life cycle processes. Geneva, Switzerland: ISO.

ISACA. 2011. COBIT®5: Process Reference Guide.

Boehm, B. W., and R. Turner. 2004. *Balancing agility and discipline: A guide for the perplexed*. Boston: Addison-Wesley.

Jones, C. 2007. Development practices for small software applications. *Software Prod. Res.*

Feiler, P. H., and W. S. Humphrey. 1992. Software process development and enactment: Concepts and definitions, 28–40.

Benediktsson, O., D. Dalcher, and H. Thorbergsson. 2006. Comparison of software development life cycles: A multiproject experiment. *IEE Proceedings - Software* 153, no. 3:87.

Subramanian, G. H., G. Klein, J. J. Jiang, and C.-L. Chan. 2009. Balancing four factors in system development projects. *Communications of the ACM* 52, no. 10:118.

Kautz, K. 1998. Software process improvement in very small enterprises: Does it pay off? *Software Process Improvement Pr.* 4, no. 4:209–226.

Steinberg, R. A., C. Rudd, S. Lacy, and A. Hanna. 2011. *ITIL service operation*. London: TSO.

Xu, P., and B. Ramesh. 2007. Software process tailoring: An empirical investigation. *Journal of Management Information Systems* 24, no. 2:293–328.

Petersen, K., and C. Wohlin. 2009. Context in industrial software engineering research, 401–404.

Dede, B., and I. Lioufko. 2010. Situational factors affecting software development process selection.

Bekkers, W., I. van de Weerd, S. Brinkkemper, and A. Mahieu. 2008. The influence of situational factors in software product management: An empirical study, 41–48.

Clarke, P., and R. V. O'Connor. 2012. The situational factors that affect the software development process: Towards a comprehensive reference framework. *Information Software Technologies* 54, no. 5, pp. 433–447.

MacCormack, A., and R. Verganti. 2003. Managing the sources of uncertainty: Matching process and context in software development. *Journal of Prod. Innov. Management* 20, no. 3:217–232.

Jeners, S., H. Lichter, and A. Dragomir. 2012. Towards an Integration of multiple process improvement reference models based on automated concept extraction. In *EuroSPI* 301, D. Winkler, R. V. O'Connor, and R. Messnarz, eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 205–216.

Chen, X., M. Staples, and P. Bannerman. 2008. Analysis of dependencies between specific practices in CMMI maturity level 2. In *Software Process Improvement* 16, R. V. O'Connor, N. Baddoo, K. Smolander, and R. Messnarz, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 94–105.

S. J. -, H. L. -, and E. P. -. 2012. Metric based comparison of reference models based on similarity. *International Journal Digit. Content Technol. Its Appl.* 6, no. 21:50–59. AUTHOR: PLEASE PROVIDE MISSING INFORMATION.

Jeners, S., H. Lichter, and E. Pyatkova. 2012. Automated comparison of process improvement reference models based on similarity metrics, 743–748.

ISACA. 2011. COBIT mapping: Mapping of CMMI for development V1.2 With COBIT 4.1.

BIOGRAPHIES

Simona Jeners studied computer science at the Computer Science Faculty, Iasi, Romania, the Konstanz University, Konstanz, Germany, and at RWTH Aachen, Germany. Since 2010 she has been a research assistant at the Department of Software Construction, at the RWTH

Aachen. She has accrued six years working in different large organizations and has supported them in their software process improvement based on different models, such as CMMI-DEV, SPICE or Functional Safety (ISO 61508). Jeners can be reached by email **at XXX.**

Rory V. O'Connor is a senior lecturer in software engineering at Dublin City University and a senior researcher with Lero, The Irish Software Engineering Research Centre. He has held research positions at both the National Centre for Software Engineering and the Centre for Teaching Computing, and has also worked as a software engineer and consultant for several European technology organizations. He is also Ireland's head of delegation to ISO/IEC JCT1/SC7. His research interests are centered on the processes whereby software intensive systems are designed, implemented, and managed.

Paul Clarke received his doctorate from Dublin City University in 2012, having previously accrued 12 years of industrial software development experience in the telecommunications, Internet, and insurance sectors. Presently, he is a research manager in the Regulated Software Research Group at Dundalk Institute of Technology, and he is also a research fellow with Lero - the Irish Software Research Centre. His research interests include software processes, software development methodologies and standards, software process improvement, economics of software engineering, and team dynamics.

Horst Lichter studied computer science and economics at Technical University Kaiserslautern, Germany. He was a research assistant at ETH Zurich and the University of Stuttgart. He was a project manager at Union Bank of Switzerland Zurich and ABB Corporate Research, Heidelberg. Since 1998 he has been a professor for computer science at RWTH Aachen University and head of the research group Software Construction.

Marion Lepmets has conducted research in process improvement and process assessment since 2000 and has been teaching process management courses at both Tallinn University of Technology and Tartu University in Estonia. Lepmets holds a doctorate degree from Tampere University of Technology (Finland).. She received a grant from Luxembourg National Research Fund (AFR) for her post-doctorate studies on process assessment impact on IT service quality that she conducted from 2010 until 2012. She has been involved in the development of software engineering standards in the International Standardization Organization (ISO/IEC JTC1 SC7) for the last six years. Lepmets is presently a senior research fellow at the Regulated Software Research Centre, based in Dundalk Institute of Technology, Ireland.

Luigi Buglione is an associate professor at the École de Technologie Supérieure (ETS) – Université du Québec, Canada, and is currently working as a measurement & process improvement specialist at Engineering IT SpA in Rome, Italy. Previously he worked as a software process engineer at the European Software Institute (ESI) in Bilbao, Spain. Buglione is a regular speaker at international conferences on software/service measurement, process improvement, and quality, and is actively part of several international/national associations on such issues. He achieved a doctorate in MIS and a degree cum laude in economics.

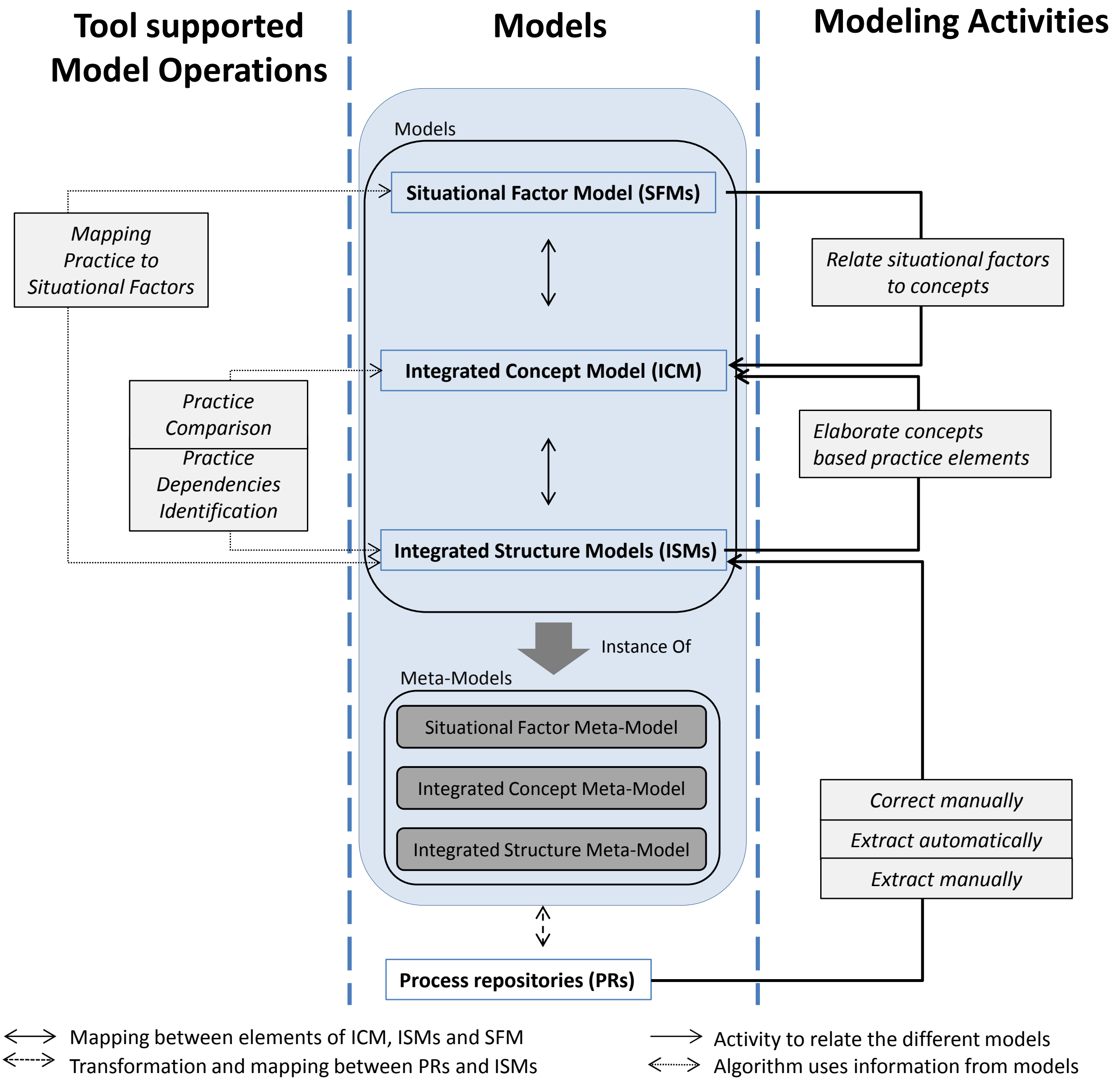

Figure 1 MoSAIC Framework Overview

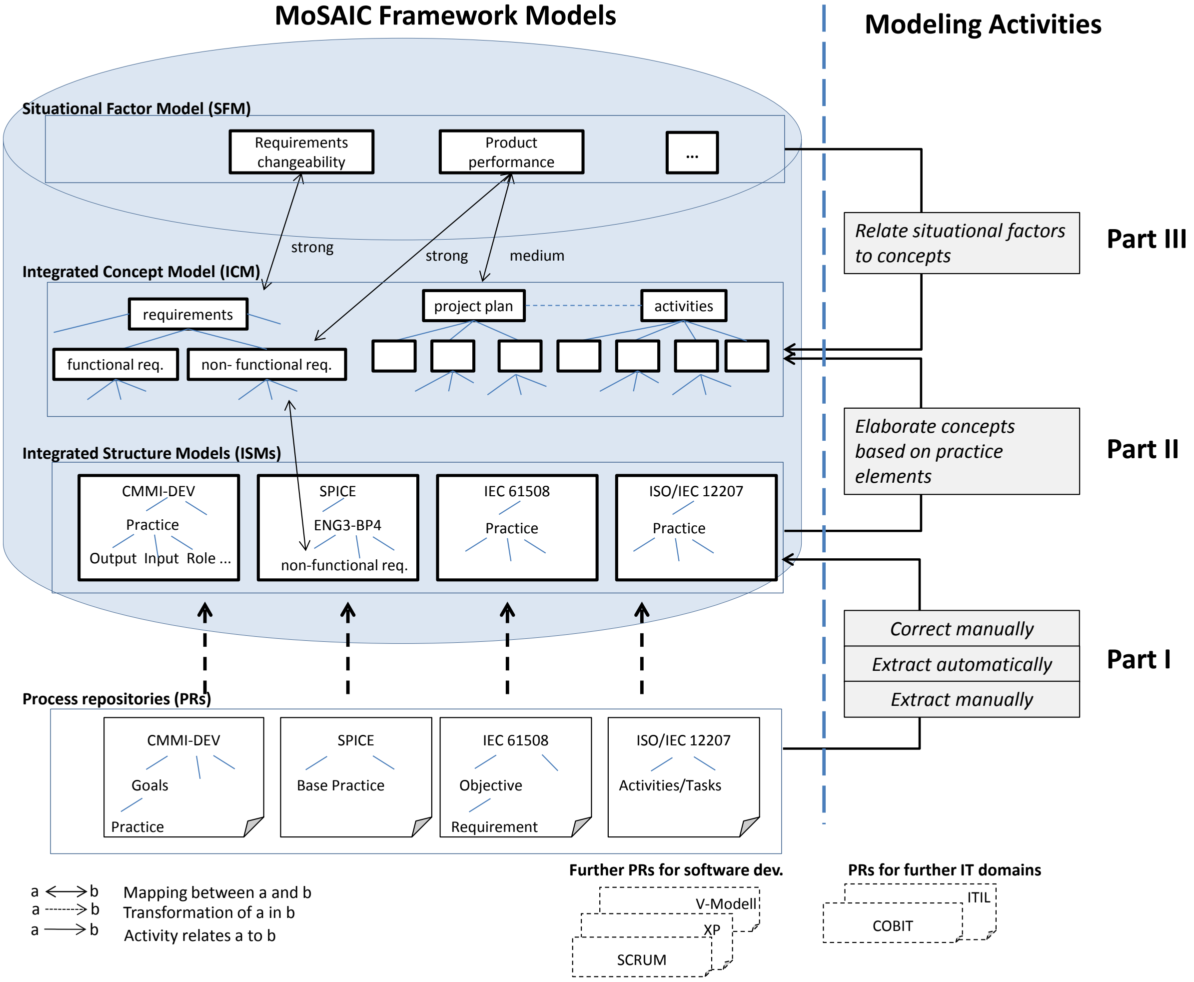

Figure 2 MoSAIC: Models and Modeling Activities

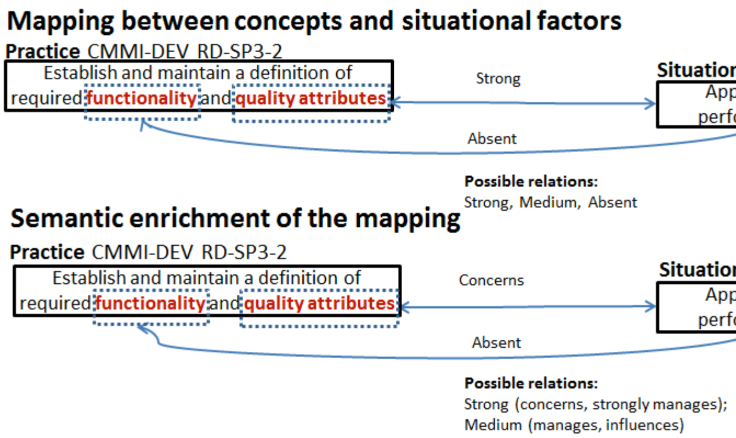

Figure 3: Example of a mapping between a concept and a situational factor

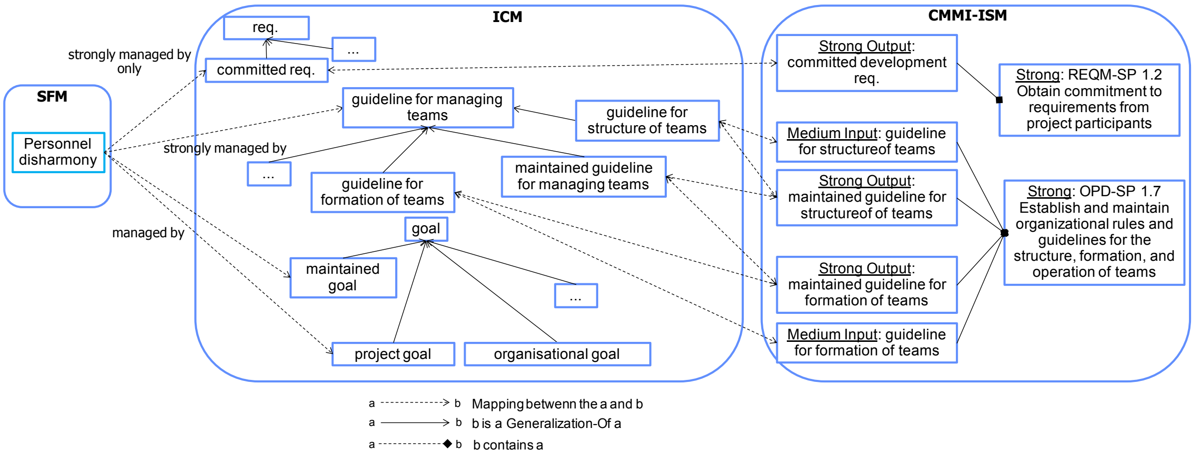

Figure 4 Examples of mapping practices to situational factors

Jeners, S., O'Connor, R.V., Clarke, P., Lichter, H., Lepmets M. and Buglione, L., Harnessing Software Development Contexts to inform Software Process Selection Decisions, Software Quality Professional, Vol. 16, No. 1, December 2013.

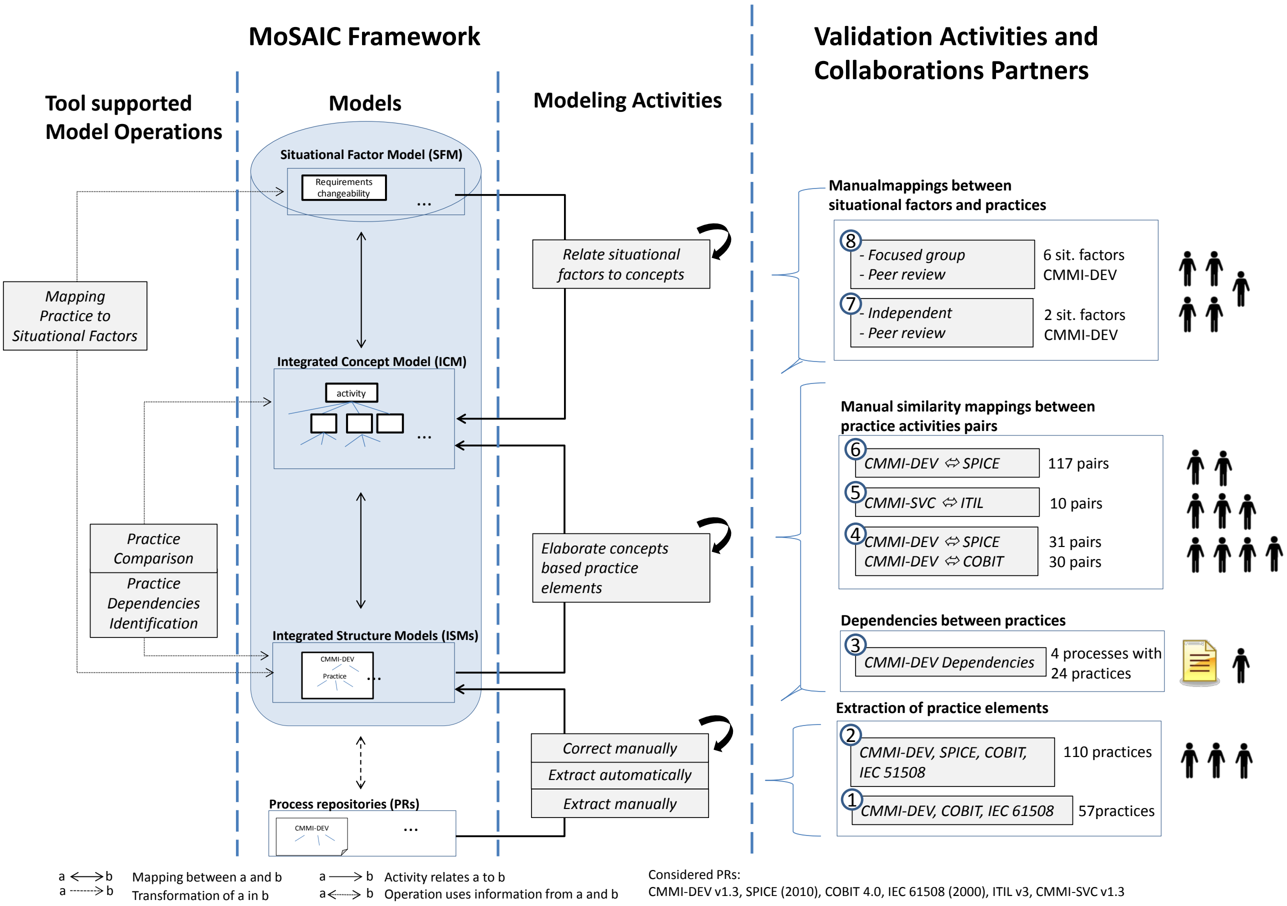

Figure 5 MoSAIC validation scenario

Metric \ Concept Type	Activity	Output	Input	Role	Purpose
PRECISION	89%	91%	85%	85%	96%
RECALL	84%	82%	65%	85%	60%

Table 3 Deviation results for the automated parsing using precision and recall metrics