A Model Based Integration Approach for Reference Models

Simona Pricope
RWTH Aachen University
Research Group Software Construction
+492418021340
simona.pricope@swc.rwth-aachen.de

Horst Lichter
RWTH Aachen University
Research Group Software Construction
+492418021331
lichter@swc.rwth-aachen.de

ABSTRACT

A variety of reference models (RMs) such as CMMI, COBIT or ITIL support IT organizations to improve their processes. As these RMs cover different domains and also share some similarities, organizations may benefit from the adoption of multiple RMs. However, organizations need a systematic support to select and efficiently implement RMs. We present the MoSaIC approach for a semantic RM integration based on common meta-models to help organizations in understanding and adopting the considered RMs.

Categories and Subject Descriptors

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

General Terms

Management

Keywords

Process improvement, reference model, meta-model

1. INTRODUCTION

Nowadays, the software market is expanding and clients are requesting better and cheaper software products. The Standish Group regularly reports that the failure rate of IT-projects is still too high: 68% of IT-projects do not meet their deadlines nor achieve the requested quality or are cancelled [12]. One important impact factor to project success is the quality of the applied IT-processes because software quality heavily depends on these processes. Hence, more and more organizations are obligated to identify, structure, and improve their processes systematically. Because the process improvement road is quite long and expensive, it needs to be guided. To support process improvement different IT reference models such as CMMI [2], COBIT [3] or Functional Safety [7] can be considered and applied. RMs are collections of best practices based on experience and knowledge of many organizations. As organizations can address multiple areas (e.g. development, operations), a single RM is mostly not enough. The adoption of multiple RMs allows covering multiple areas from different IT domains. Furthermore, it allows exploiting synergy effects among them. On the one hand organizations can address coordinately different and common areas. On the other hand the weaknesses of a single RM can be overcome by the strengths of others.

Although there is free information available about each single RM, there is no integrated view that makes a collection of RMs more transparent and supports organizations in the selection and adoption of RMs. This lack of transparency is the main problem that hampers organizations to use the experience and knowledge reflected by these RMs:

Adoption of RMs is too expensive. Many organizations, specially small and medium ones, do not adopt RMs because of the high costs (e.g. COBIT compliance to SOX costs 1% of total revenue for small and medium organizations whereas just 0.1% for large organizations [1]). Organizations need a systematic support to choose from the collection of RMs only what best fits for them.

Many RMs exist for different IT domains, IT development, IT Services or IT Governance and in each of these domains several areas such as technical construction or supplier management are covered. There is no clear systematics or criteria to select the "right" RMs or parts of RMs.

RMs are too complex. On the one hand, RMs address very many areas. On the other hand, RMs are based on different structure and terminology. Each single RM defines its own specific structure and uses a specific set of terms. Hence, different terms are used for the same semantic concept. Furthermore, RMs are defined on different levels of abstractions. Best practices of RMs can be described either very generally or more concretely. All these hamper to understand and adopt RMs.

RMs overlap. Although RMs exist for different IT areas, they may address similar topics. For example, project or risk management is addressed in almost all RMs. To efficiently adopt multiple RMs an organization must be able to easily compare the selected RMs and identify their similarities and differences. Furthermore, the organizations should be aware of the essence of similar RMs' process areas or RMs' practices. The specific details of each RM should be also easy to identify if necessary.

RMs are changing. Since RMs are updated continuously and new RMs are developed organizations must keep pace with their evolution and must be able to understand and apply the changes.

Due to these problems the organizations need more transparence and support regarding these questions: (Q1) Which RMs should be considered for process improvement? (Q2) How can we efficiently implement the selected RMs?

2. GOALS AND SOLUTION APPROACH

In order to address the questions mentioned above we have developed a new approach called **MoSaIC** (**Mo**del-based **S**election of **a**pplied **I**mprovement **C**oncepts) aiming to offer more transparency and support to organizations in the process of selecting and adopting RMs or parts of them.

12th International Conference on Product Focused Software Development and Process Improvement (Profes '11). ACM, New York, NY, USA, 6-9.
doi: 10.1145/2181101.2181103

Based on the organizations' needs, entire RMs or only parts can be selected. For example, by selecting only parts of RMs, organizations can address their specific problems, will not be overwhelmed and don't have waste costs with the RMs' implementation. Furthermore, analogous to agile development the organizations can iteratively observe the results of their improvement. An objective approach to support the decision of selecting RMs is needed. For this selection approach, we first need to overcome the lack of transparency of the RMs' collection Therefore, we present in the following the MoSaIC approach to achieve transparency (hence answering question Q2). To efficiently support organizations in multiple adoption of RMs, we propose a model based integration approach addressing the following basic research questions: (Q21) *How can the complexity of RMs be reduced to facilitate the understanding of RMs?* (Q22) *How can RMs be compared effectively and how can the comparison results support organizations?* (Q23) *How can an easy update of changed RMs and an easy integration of new ones be supported*?

3. RELATED WORK

Information about the comparison of RMs, such as mappings between process areas or practices, is available (for example from ISACA – mappings between COBIT and CMMI, COBIT and ITIL). However, the mappings are often bilateral and subjective. To overcome these problems, some authors try to integrate RMs using models by formalizing them on a fine granular level.

The need of a process architecture in a multimodel context is mentioned in a series of articles from SEI [11]. This raises the awareness to define a generic and integrated model which makes RMs more transparent and support organizations to find similarities between different RMs. Basic elements mentioned in the work of SEI, like inputs, outputs, roles and their relations are part of our integration model as well.

Ferreira, Machado and Paulk [6] present an approach to achieve transparency of RMs by comparing RMs. The problems mentioned above, complexity, different abstraction levels of RMs and overlapping are also mentioned. This approach tries to solve these problems by defining metrics to compare RMs which gives a first overview of the RMs' similarities. We want to support a more detailed comparison of RMs and also offer an automatisms to compare the RMs.

Ferchichi and Bigand [5], Liao, Qu and Leung [8] as well as Malzahn [9] define a common structure to link RMs and reveal their similarities. For this purpose similar RM practices are connected manually. In contrast to the first two approaches we model on a more fine grained level. The third approach also addresses this fine granularity but does not define similarity. We differentiate between serveral similarity relations to get a more accurate degree of similarity between RMs resp. between parts of RMs.

4. MOSAIC INTEGRATION APPROACH

In the following we describe the MoSaIC way to integrate RMs. First, we motivate and give a short overview of our integration approach. Then we present the two meta-models of MoSaIC that provide the basis for a model based RM integration approach.

The main idea of MoSaIC's RM integration approach is to normalize RMs based on a joint structure and on a common set of terms. According to mega modeling theory [4], we can normalize by defining appropriate meta-models. We have analyzed published RM meta-models, e.g. the one of CMMI, extracted and added only those elements needed to achieve the goals defined at the beginning of this paper.

To model different RMs the same way we have developed a so called *Integration Structure Meta-Model* (IS Meta-Model). It defines core and additional RM element types introduced by different RMs as well as fine grained RM concept element types, such as activities, artifacts or roles. While the core and additional element types allow providing a rough overview of the most important aspects of RMs, the conceptual elements types allow the integration of concrete and abstract RMs and a detailed comparison of RMs.

A *RM concept* (concept for short) is a word or the smallest sequence of words that has a unique meaning in the context of RMs. For example "project plan" or "work breakdown structure" are concepts used in RMs. Concepts can be derived from activities, roles and artifacts of RMs.

For each RM, such as CMMI, SPICE, COBIT or ITIL, we have extracted the core, additional and conceptual information and created respective *RM Integration Structure Models* (RM-ISMs). Mappings from the single RM-ISMs to the RMs' original structures provide more information if needed.

Furthermore, RMs should be modeled using the same terminology. Here our idea is to introduce a mechanism to translate and map the terms/concepts used by each single RM to a common normative set of terms/concepts. For this purpose we have created a model containing the closure of all RM concepts, called *Integration Concept Model* (ICM). A general concept defined in the ICM can be mapped to several finer conceptual elements of one single RM or to several different RMs if they together are semantically equivalent to the general concept. The ICM concepts can be seen as dictionary entries having synonyms and explanations in the different RM-ISMs. This conforms to what is called *linguistic concordance* (an alphabetical list of the principal words used in body of work, with their immediate contexts (Wikipedia)) and supports a better understanding and avoidance of misinterpretations of the RMs' content. Furthermore, we have enhanced our meta-models by attributes and semantic relations (e.g. to model similarity between concepts) to improve their comprehension. Obviously, the ICM is the sole instance of its meta-model, the *Integration Concept Meta-Model* and links all RM-ISMs.

Figure 1 shows the most important elements of the *IS Meta-Model* represented in a UML Class diagram like notation. We have grouped the elements in three packages: **Core** contains elements mostly defined by meta-models of existing RMs; **Add-Ons** offers elements that are not always present in all RMs; **Concepts** contains elements to model concept information of RMs on a fine grained level. As our approach is centrally based on the elements of the package Concepts and the other elements can also be found in the already defined meta-models (CMMI, Functional Safety), we only concentrate on their description.

Activities, *Roles*, and *Artifacts* of RMs are concrete ConceptualElements. An Activity may involve Roles and is performed by one or more Roles; it usually needs and produces Artifacts. Because a certain Role or Artifact can be used in different *Procedures* of a RM, only their references are associated with *Activities*. This allows defining procedure specific information (e.g. *QualityAttribute* that characterizes these elements "formally approve the project plan"). Hence, *Activities*, *RoleRefs* and *ArtifactRefs* are the central aspects of a *Procedure*, abstractly modeled by *ProcedureElement*. However, there are some differences between the concrete *ProcedureElements*. For example, *Activities* may be *performedIn* different contexts (e.g. "approve the plan before project initiation"). As context information is conceptual information as well, *Contexts* are special *ConceptualElements* and as they are elements of procedures they are also special *ProcedureElements*. A *Context* usually explains its *Activity* (e.g. "maintain the programme by controlling the projects"), but it may also specify a temporal relation (e.g. "approve the plan before project initiation") or specify a local relation (e.g. "review requirements specification in the IT department"). The different context types are modeled by an attribute of type *ContextType*.

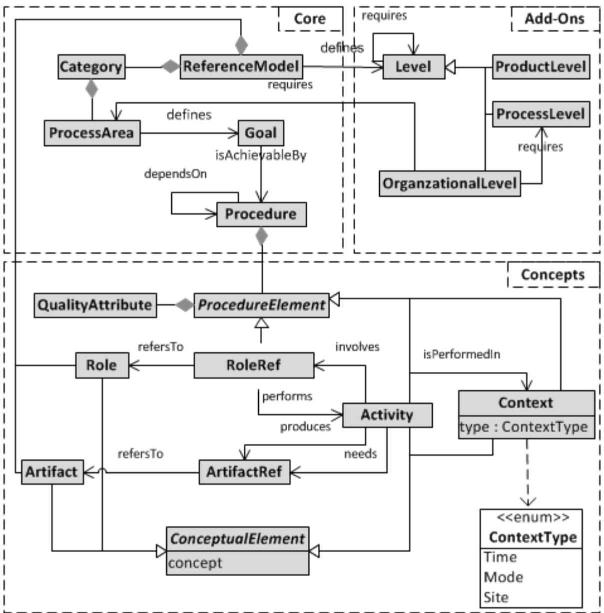

Figure 1. The Integration Structure Model

Figure 2 depicts the elements of MoSaIC's Integration Concept Meta-Model. Although it has a pretty simple structure it is sufficient to model the world of RM concepts with their relations. Obviously, a *Concept* (which is a term or a combination of terms from a RM) is the main element. A *Concept* always has a *ConceptType* and may be related to other concepts by so called *ConceptRelations* which are typed as well.

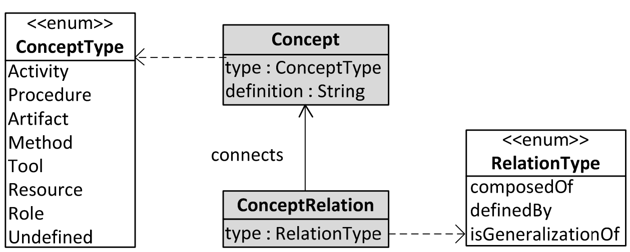

Figure 2. The Integration Concept Meta-Model

The *ConceptType* determines the role of a concept in a certain context (e.g. "work breakdown structure" may be an *Artifact* but also a *Method* depending on the context). *ConceptRelations* are used to model similarities between concepts. A concept may be *composedOf* other concepts. For example "requirements" is composed of "functional requirements" and "non-functional requirements". Furthermore, a concept may be a generalization of a more concrete concept (e.g. the concept "stakeholder" is more general the "project manager"). In addition, a concept may be *definedBy* other concepts. For example, the concept "plan the involvement of stakeholder" is defined based on the concept "stakeholder". Concepts can be *defined* by experts. Currently, initial sets of concept and concept relation types are offered. This architecture is flexible and open to introduce new concept and concept relation.

In the next section we describe the application of both meta-models.

5. APPLICATION OF META-MODELS

In order to evaluate the appropriateness of the developed meta-models we have performed a mid-size case study considering some parts of the COBIT, CMMI and Functional Safety.

Although each RM focuses on a specific IT domain they are similar in certain aspects. Because the aim of our case study was to explicitly show the similarities between the considered RMs and therewith their integration, we selected designated procedures that contain similar conceptual elements (CMMI procedure PP SP2.6: "*Plan the involvement of identified stakeholder*" with the typical work product "*stakeholder involvement plan*"; COBIT procedure PO10.4: "*Obtain commitment and participation from the affected stakeholders in the (..) execution of the project within the context of the overall IT-enabled investment programme*"; Functional Safety, Part 1, procedure 6.2.1b: "*Consider the identification of the persons, departments and organizations which are responsible for carrying out (..) the applicable overall, E/E/PES or software safety lifecycle phases*")

Figure 3 depicts the developed models. Each ISM contains conceptual elements, such as activities (A), artifacts (AF), or roles (R). The activities of COBIT and FS are described by context (CXT). In the figure we can easily identify the **similarities** between these procedures because similar ISM elements are connected to the same concept (C) or related concepts in the ICM. Furthermore, we can easily extract the **essence** by identifying only the general concepts: perform the activity "*plan the involvement of the stakeholder*" for the "*execution of the project*", involve the "*stakeholder*" and produce the "*stakeholder involvement plan*". If necessary the organization can easily identify the details (e.g. what are the stakeholder in FS?)

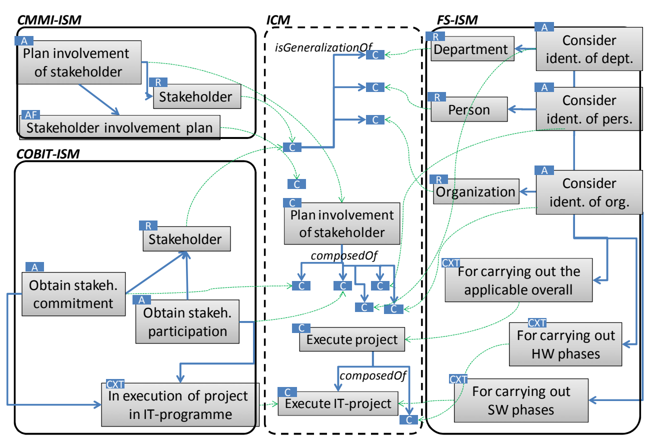

Figure 3. Excerpt of a MoSaIC RM Integration Model

6. EXPERIENCES & FUTURE WORK

In the following we describe the experiences gained with the developed RM integration approach. Furthermore, we propose some ideas towards further research and ideas concerning the application of the MoSaIC RM integration approach.

The modeling of selected parts of CMMI, COBIT and Functional Safety showed that the IS and IC Meta-Models offer a stable structure for integrating the chosen RMs. The RM specific ISMs as well as the central ICM were created manually. Because our approach is based on fine grained RM information, a large number of conceptual elements had to be modeled. Thereby, redun-

12th International Conference on Product Focused Software Development and Process Improvement (Profes '11). ACM, New York, NY, USA, 6-9.
doi: 10.1145/2181101.2181103

dant definitions of semantically equivalent concepts in the ICM had to be avoided. Furthermore, we always tried to relate new concepts to existing ones in case of similarity (by "composedBy", "isGeneralizationOf" or "definedBy" relations). However, the manual modeling was not always easy and sometimes it was difficult to model concepts consistently.

Because the manual modeling of RMs is a time consuming and error-prone process, a dedicated tool box is needed to cope with the consistency problem and to model the fine grained conceptual elements. A semiautomatic tool that is performing a syntactical and linguistic analysis of the RM documents may generate recommendations to model the conceptual elements properly. For example, prepositions like "based on" or "in line with" may require modeling a respective artifact. Because the RM documents are written very differently, modeling recommendations can not only rely on syntactic rules. For example, in some RM documents the activities of procedures are written by nouns while in others verbs are used. Hence, a plain syntactical analysis of the documents is not sufficient. Further rules are needed to transform the language used in the original documents in a "normalized" language. We will investigate if rules used to precisely write requirements [10] could be adopted to transform the original text in a "normalized" language (e.g. the passive or noun form of a verb is transformed in its active form). This may allow a semiautomatic extraction of conceptual elements and their modeling in the ISM and ICM. Furthermore, the tool box should be able to adapt and improve its generated recommendations according to modeling decisions done by RM experts.

Furthermore, we want to offer the organizations a tool for an efficient comparison of RMs that automatically compare RMs and extract the essence of similar procedures.

7. CONLUSION

In this paper we have presented MoSaIC, a model based approach to integrate reference models. The core idea is to represent each RM in a dedicated Integration Structure Model (ISM) and the common concepts in one central Integration Concept Model (ICM). The created ISMs and the ICM allow a semantic integration of RMs. As each RM-ISM contains the most important RM information, it provides a condensed overview for organizations. Furthermore, the integration of RMs eases the understanding and avoids misinterpretations of terms and concepts used in RMs (Q21), because the concepts are associated with their synonyms and contexts in the RM-ISMs. Every new RM may enrich the description of a concept and ease its comprehension. Therefore, the more RMs are integrated, the smarter the ICM will become.

The fine granularity of the models (ISMs and ICM) enables to model RMs on different levels of abstraction: abstract, concrete RMs and even internal processes can be easily integrated in the MoSaIC Integration Model. Furthermore, the integration of abstract and concrete RMs supports a better understanding of a certain area that is addressed by both of them (Q21).

Our approach allows a detailed and efficient comparison of RM procedures (Q22). The fine grained model elements, the concepts connecting the procedures, and the semantic concept relations allow identifying automatically similarities between procedures and process areas. The essence can also be identified to efficiently adopt multiple RMs. The organizations do not have to worry any more about the redundancies between the adopted parts of RMs.

Finally, changes on existing RMs or new RMs can be integrated in the MoSaIC Integration Model (Q23). This is done by creating or updating the respective ISM and adding or updating the connections between the changed or newly created ISM and the central mediator, the ICM. The integration of a new RM implies adding new concepts to the ICM if they are not already defined.

First experience and results with the presented approach are promising; we were able to effectively model the integration of some process areas of the RMs COBIT, CMMI and Functional Safety. We expect that the results of our future work will make the integration of RMs more accurate and comfortable to offer a better support to organizations for the adoption of multiple RMs.

8. REFERENCES

[1] Armstrong, R. (2008) Sox compliance: Eleven essential controls for the sme, Sarbanes-Oxley Compliance Journal 2008. http://www.s-ox.com/dsp_getFeaturesDetails.cfm?CID=2106.

[2] CMMI (2010): CMMI for Development, Version 1.3. CMU/SEI 2010-TR-033, ESC-TR-2010-033, SEI, Carnegie Mellon University, Pittsburgh, PA.

[3] COBIT(2007): Control Objectives for Information and Related Technology Version 4.1. ISACA (Information Systems Audit and Control Association).

[4] Favre, J.-M. (2006): Megamodelling and Etymology. In J.-R. Cordy, R. Lämmel, and A. Winter, editors, Transformation Techniques in Software Engineering, Dagstuhl Seminar Proceedings, No. 05161, ISSN 1862-4405.

[5] Ferchichi, A., Bigand, M. (2008): An Ontology for Quality Standards Integration in Software Collaborative Projects. Proceedings of MDISIS'08 (Model Driven Interoperability for Sustainable Information Systems), Montpellier, FRANCE, 14.

[6] Ferreira, A. L., Machado, R. J., Paulk, M. C. (2010): Quantitative Analysis of Best Practices Models in the Software Domain. Proceedings of the 17th Asia Pacific Software Engineering Conference, 433-442.

[7] IEC 61508 (2010): IEC 61608 or Functional safety of electrical/electronic/programmable electronic safety-related systems, Edition 2.0, 30 April 2010, DKE-Gremium 914.

[8] Liao, L., Qu, Y., Leung, H. K. N. (2005): A Software Process Ontology and Its Application. Proceedings of the IWFST-2005 (International Workshop on Future Software Technology, Shanghai, 1-10.

[9] Malzahn, D. (2009): Assessing - Learning - Improving, an Integrated Approach for Self Assessment and Process Improvement Systems. Proceedings of ICONS '09, the Fourth International Conference on Systems, Gosier, Guadeloupe, France, 126-130. Ieee. doi: 10.1109/ICONS.2009.31.

[10] Pohl, K., C. Rupp (2011): Requirements Engineering Fundamentals – A study Guide for the Certified Professional Requirements Engineering Exam – Foundation Level – IREB compliant. Rocky Nook, Inc, Santa Barbara.

[11] Siviy, J., Kirwan, P., Marino, L., Morley, J. (2008): Process Architecture in a Multimodel Environment. SEI White Paper, http://www.sei.cmu.edu/library/assets/multimodelSeries_wp4_processArch_052008_v1.pdf.

[12] The Standish Group International (2009): CHAOS Summary 2009. Chaos (pp. 1-4). Retrieved from http://www.statelibrary.state.pa.us/portal/server.pt/document/690719/chaos_summary_2009_pdf.

12th International Conference on Product Focused Software Development and Process Improvement (Profes '11). ACM, New York, NY, USA, 6-9.
doi: 10.1145/2181101.2181103