2010 36th EUROMICRO Conference on Software Engineering and Advanced Applications

Size and Complexity Attributes for Multimodel Improvement Framework Taxonomy

Andre L. Ferreira and Ricardo J. Machado
Dept. de Sistemas de Informação
Universidade do Minho, Guimarães, Portugal
{andre.ferreira,rmac}@dsi.uminho.pt

Mark C. Paulk
Institute for Software Research
Carnegie Mellon University, PA 15213 USA
mcp@cs.cmu.edu

Abstract— **Selection of best practice models is a daunting task. The number of models is considerable and the ability to compare objectively their content is not straightforward due to scope and structural variety in descriptions. The purpose of this paper is to provide a base for quantitative analysis of best practice models at the light of proposed attributes of size and complexity. We propose a characterization of size as a measure of scope coverage and detail of descriptions between models and complexity in terms of structural connectedness. We analyzed a set o best practice models popular in the Software Engineering domain and derived relative size and complexity measures of these models.**

Multimodel improvement taxonomies; Software Process Improvement; Software Engineering Management;

I. INTRODUCTION

Best practice models prescribe requirements for conducting business. They represent the knowledge derived from excellence organizations performing in specific areas. Organizations are adopting multiple best practice models to improve overall effectiveness and efficiency. Market pressure, competitiveness, regulatory compliance or the need to solve a particular issue are general business drivers for organizations to adopt multiple best practice models (hereafter used interchangeably as improvement technologies or simply models). The goal is to obtain the cumulative added value of each model.

Models adoption requires appropriate sponsorship, investment and experience and often implies dealing and implementing hundreds of requirements from a diverse set of Standards Bodies. Usually, adoption decision rests at different levels of authority and is motivated by different needs and perspectives of distinct business units. Also, models adopted are likely to accumulate over the years; typically the decision is to adopt a model after the other. These efforts, if not supported and coordinated appropriately carry significant risk of failure [1].

Adopting multiple models often results in misalignment of models implemented creating additional reconciliation effort. Redundancy or gaps between models leads to operational problems and reduced productivity. The overall picture of capability and cost of quality for each model is difficult to attain when combined into a single environment. Organizations need help in getting started effective and efficient in using quality models and adopting the proper set of models requires some guidance [2] [3] [1].

To mitigate these risks of failure and inefficiency, organizational process improvement groups need the ability to compare models effectively before these are selected for implementation. A fact is that the exercise of comparing model is not straightforward. Firstly, the number of models is considerably high across multiple domains and disciplines. Secondly, comprehensiveness of descriptions and structural differences need to be considered for effective comparison.

Selection and composition of models are two challenges that organizations need to tackle when considering the adoption of multiple models [4]. Selection occurs prior to the composition effort and represents one important step. At this stage, the most suited model or set of models is expected to be identified. Selection decisions may derive from common industry patterns of adoption or regulatory requirements. Nevertheless, the perceived value of each improvement technology is valuable information for appropriate adoption decision.

Some approaches to guide the selection and adoption process are affinity groups, taxonomies, model mappings, selections and implementation patterns and formal decision methods [5]. These approaches provide different levels of comprehensiveness for model comparison. Taxonomies enable a high level comparison, generally to compare a considerable set of models. Conversely, mappings are often used to compare two models with increased level of detail. These approaches provide a comparative qualitative analysis of models content that most of the times fail to provide a summary picture of the differences between models.

Improvement groups considering a multimodel approach often need to justify quantitatively the decision to adopt or reject a specific model to a multimodel environment. A comparative added value of a model with possible cost estimates of implementation as well as synergies with other models are not easy to derive from a qualitative analysis. Effective measures characterizing differences between models would provide additional information to inform better decision making on model adoption and/or re-engineering existing multimodel environments. This paper aims to propose an extension for a taxonomy for improvement frameworks described in [1] by characterizing attributes of size and complexity.

The following sections are organized as: section 2 provides an overview of existing approaches available to compare improvement technologies. Section 3 proposes the definition of size and complexity measures for characterizing improvement technologies. Section 4 shows preliminary results of measuring size of Software best practice models. Section 5 concludes on the proposed approach and its possible implications.

978-0-7695-4170-9/10 $26.00 © 2010 IEEE
DOI 10.1109/SEAA.2010.54

II. Comparison of Best practices Models

The analysis on approaches for comparing models will consider, as structuring element, the characterization based on classes of comparison proposed by Halvosen and Conradi in [6], namely: comparison based on characteristics; comparison based in frameworks or bilateral mappings; and comparison based in needs mappings.

Several authors propose taxonomies based on sets of characteristics or attributes for comparison. They all share the purpose of providing a tool to help overcome the difficulty of understanding and comparing Software Process Improvement (SPI) frameworks. Paulk proposes a taxonomy with 3 major categories that comprise a total of 10 attributes which elaborate on specific topics. Halvosen and Conradi [6] taxonomy comprises 25 characteristics grouped in five categories. A simpler approach is proposed by Siviy et al. [7] by defining an affinity matrix that groups models considering 2 attributes. Also based in characteristics, Kirwan et. al. [8] define a three-category element classification to be used as taxonomy for technology composition. The classification scheme is composed of good practice elements, improvement method elements and institutionalization elements.

Heston and Phifer [2] introduce the concept of 'process DNA' and 'Quality Genes' to analyse concepts from several key industry standards. They observed that every model has a 'sweet spot' or set of business issues to which they are particularly well suitable. The selection approach is based in the concept of quality genes. They propose 18 quality genes as building blocks shared across multiple models. Each model is classified for each quality gene as *high correlation*, *some correlation* or *no correlation*. Classification is based in an analysis of each model contents to gauge its depth and coverage of each quality gene. Organizations may interpret quality genes as needs and assess which models are more suitable to meet their SPI objectives.

A framework or bilateral mapping comparison requires two models and a mapping function. The goal of the mapping function is to determine the portion of shared scope between models. Comparisons occur between architectural components and may be performed with more or less detail. Recent examples of mappings are available in [9] where CMMI-Dev [10] is mapped to ISO12207 [11] for a detailed mapping at the ISO12207 task level. Activities of ISO122207 are mapped to CMMI-Dev specific practices through ISO12207 tasks. In [12] ISO9001[15] is mapped to CMMI-Dev. The mapping occurs at the *shall statement* level. A direct relation measuring the strength of the mapping is established between *shall statements* of ISO9001 and *specific* and *generic practices* of CMMI-Dev process areas.

The approaches presented evidence operational differences when selection of models is to be carried out. Characteristics based comparison seems to be more suitable when a high-level analysis of models is required. Comparison characteristics are seen as properties or attributes that are useful to understand and compare a wide set of models. Conversely, bilateral and/or framework mappings require a deeper level of analysis. These are applicable to a pair of models, therefore are more appropriate for low level model comparisons.

In the context of the harmonization framework defined in [13], the comparison methods mentioned may be applicable in the selection and composition challenges of multiple model integration. The characteristics and needs mappings are more aligned with an initial selection stage, where the number of models is considerable and a high-level characterization may present itself more useful. Composition may benefit from a lower-level comparison by implying a deeper model analysis, allowing the identification of overlapping and gaps among models. For this reason, the framework mappings and bilateral comparison may be more suited to the composition stage. However, it may still be valuable for selection purposes, if a detailed analysis is required to choose among a reduced set of models.

Lower level comparisons require consideration of elaboration of descriptions and structural differences of models architectural components. Often, gaps exist as a direct result of different elaboration levels and structural differences may also difficult the comparison. For instance, ISO9001 uses *shall statements* to describe model requirements and CMMI-Dev uses the concept of *goal* to describe what needs to be achieved. Framework mappings and bilateral comparisons require finding a common ground for reconciliation of these differences. In the next section we address this issue of reconciliation in a quantitative level.

III. Architectural Attributes: Size and Complexity

Can we be compliant with model X and model Y? How much they overlap? Are there any significant differences between them? The answers are hard to attain objectively. We aim to provide means to deliver a quantitative evaluation to help answer these questions. The approach discussed in this paper will explore two concepts that we believe relevant to characterize both size and complexity of models, namely: models scope and structure.

Every model focus on a subject to which prescribes and/or describes best practices. The subject defines the scope boundary each model addresses. Scope can be broader or narrower considering its domain of applicability, e.g., Software Engineering models vary in the scope they cover and the level of detail they use in their descriptions; one may argue that ISO9001 is boarder and less detailed than CMMI-Dev. Usually, narrowing the scope often results in an increased detail of descriptions. More detail is achieved by using a higher number of descriptional components at a similar level of detail or further elaborating by creating new levels of elaboration. Hence, models can be compared according to the scope they share and by the level of detail in descriptions within this shared scope.

Another important concept relates to the structure of models. An overview performed on several models for the Software, Information Technology and Governance domains revealed the use of a hierarchical structure between the major architectural components used to describe their content. The notion of hierarchy is used to encapsulate descriptions that are further elaborated as the level in the hierarchy increases, providing additional levels of detail. For instance, CMMI-Dev uses three architectural component types (among others) to organize its content, namely: *process areas*, *specific goals* and *specific practices*. A hierarchical relation of inclusion is present: process areas are described using specific goals which are detailed using specific practices. Based on this inclusion property, an elaboration hierarchy may be derived to group model descriptions at the same logical level of detail. The inclusion of a component as part of the definition of other component indicates a new level of elaboration or detail. An elaboration hierarchy comprises an ordering of detail levels: level zero is at the refined level of elaboration. Lower levels form part of components at higher hierarchical levels.

A. Size

The characterization of size aims to translate the perspective that: the scope of a model can be measured if compared to a reference scope and that within a shared scope the amount of information may vary, introducing the concept of elaboration of descriptions within a shared scope.

One of the challenges with this perspective is the difficulty to find a reference dimension for scope to effectively compare a model scope size. The ideal scenario would be an *'include-all'* reference dimension for scope, but this may be unfeasible to define properly due to the magnitude of disciplines and domains covered in improvement frameworks.

A second challenge is how to evaluate detail of descriptions. Information is embedded in models structural components in the form of textual descriptions. Different type of architectural components and relations are established defining models architectures. One can consider the number of components each model has at specific level of detail as a model´s size measure. However, this number may not provide meaningful information on elaboration or detail. Often, a direct logical comparison between components of different models is not straightforward to derive an effective size comparison.

To overcome the first challenge one can considerer as a reference scope the scope of a model of interest. This model will provide a reference measure for scope. One may argue that this is not desirable as the perception of size may change if different models are considered. However, it also introduces the flexibility to use a reference model that best suites the interest of comparison.

The number of structural components provides a more meaningful notion of elaboration of descriptions when these relate to a shared scope. The number of components used to describe the shared scope is an indicator of different levels of elaboration used.

Therefore, two dimensions can be considered when conceptualizing size of a model: scope shared when compared to a reference model and the level of elaboration present within this shared scope by considering the number of architectural components used by each model.

Shared Scope. The goal of framework or bilateral mapping technique (mentioned in section 2) is to identify similarities and differences between two models. As result of applying this technique, usually, a shared scope is identified by comparing and linking a component (C_i) of a mapped model (Mp) to a component (C_j) of a reference model (Mr) of interest. A measure φ is associated to the link and represents the portion of scope defined by C_i of Mp that is shared or covered by the Cj of Mr. The relation is formalized in (1) (*np* and *nr* are the number of architectural components of Mp and Mr at a specific detail level, respectively).

$$(Ci, Cj, \varphi ij),\ \forall\ \varphi ij\ \epsilon\ [0,1]\ in\ \mathcal{R},\quad i\ \epsilon\ [0..np]\ and\ j\ \epsilon\ [0..nr] \tag{1}$$

Computing (1) for each C_i of Mp is possible to identify the number of architectural components of Mp that share a portion of scope with Mr ($\varphi ij > 0$ in (1)). The maximum value of φij for a specific C_i defines the maximum coverage obtained for that component. The summation of each max (φij) for each C_i delivers a measure of scope shared between Mp and Mr.

Elaboration of Descriptions. A mapping exercise identifies a set of links between architectural components of a mapped model (C_iMp) and components of a reference model (C_jMr). A link establishes a unidirectional bound between two architectural components of different models (see section 1 in Figure 1).

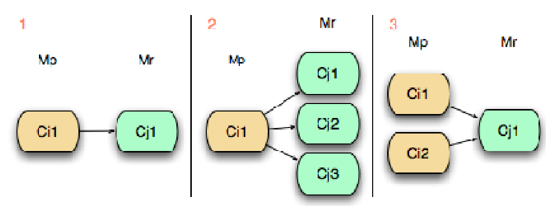

Figure 1 - Component Mappings

It is possible and often usual that a component C_i of Mp maps to more than one component C_j of Mr, establishing more than one link of type (1). The notion of elaboration of description builds upon the number of links identified between components. If a component C_i links to more than one component of C_j of Mr, we can say that Mp uses one component to describe the shared scope that is described by several components in Mr (see section 2 in Figure 1 and (2)). This is an indicator that C_i is less detailed in describing the shared scope.

$$(Ci, M_r, ni),\ ni\ \epsilon\ [1..nr] \tag{2}$$

The reverse scenario is also common (see section 3 in Figure 1 and (3)). Different components of Mp map to a single component of Mr. This indicate that C_j of Mr is less detailed in describing the shared scope.

$$(Cj, M_p, mj),\ mj \in [1..np] \quad (3)$$

It is possible to compute the elaboration factor *N* and *M* for each model by computing the central tendency values of *ni* and *mj* resulting from (2) and (3), respectively. The elaboration factor $E_f = N/M$ translates a measure of global elaboration differences between models considered ($E_f \approx 1$ indicates similar overall detail in descriptions).

B. Complexity

Complexity (or the perception of complexity) appears to be generated by three factors working in combination: variety, connectedness and disorder [14]. Often models establish internal references between architectural components. We propose structural connectedness to assess architectural complexity and deliver an objective measure of architectural components linkage of a model. Structural connectedness may be used to evaluate the systemic view of a model and compare it objectively with other models, or existing or future implemented process solutions.

IV. Comparative Analysis

An experiment was carried using the proposed approach for measuring size and complexity on software best practice models. Three mappings provided publicly by Mutafelija and Stronberg [12] were used to perform a quantitative analysis at the light of proposed size and complexity attributes. Figure 2 shows a partial result of the experiment. CMMI-Dev was used as the reference model – Mr - and ISO standards as the mapped models - Mp. In a first mapping 228 *shall statements* of ISO9001 are mapped to CMMI-Dev *practices* resulting in 83% of shared scope. In a second mapping 236 *tasks* of ISO15288 are mapped to CMMI-Dev *practices* with 77% of shared scope. In a third mapping 319 *tasks* of ISO12207 are mapped to CMMI-Dev *practices* with 74% of shared scope. Concerning elaboration of descriptions - E_f - values of 1.95, 1.38 and 1.16 were computed for ISO9001, ISO15288 and ISO12207, respectively, indicating that CMMI-Dev is more detailed, overall, than the ISO models considered.

V. Conclusion

In this paper we proposed an approach to measure size and complexity of best practice models in a context where selection of models can benefit from explicit insights in terms of size and complexity. We tried to capture the notion of size considering the concept of shared scope between models and differences in elaboration of descriptions. We considered complexity as a measure of component architectural interconnectedness. Our approach to measure size and complexity is strongly dependent on the models mapping technique strengths and weaknesses.

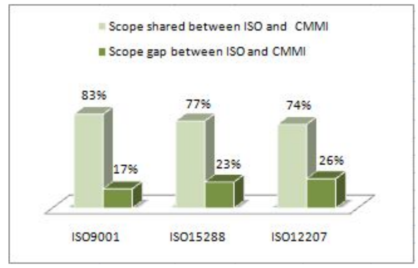

Figure 2 - Coverage of ISO standards by CMMI-Dev

Our experiment revealed that ISO standards share a high percentage of scope with CMMI-Dev and that CMMI-Dev is well ahead of ISO standards in terms of prescribing relations between architectural components. The proposed approach enables a novel comparative quantitative perspective of models scope size and complexity. This view can be used by improvement groups for developing a systemic quantitative analysis on model comparison and justify their adoption with quantitative information. Additionally, for Standard Bodies and Organizations prescribing best practices models allows building comparative charts of their models, which may be used to evolve or reconcile content with other existing models.

References

[1] M. C. Paulk, "A Taxonomy for Improvement Frameworks," in Fourth World Congress for Software Quality, 2008.

[2] K. M. Heston, and W. Phifer, "The multiple quality models paradox: how much best practice is just enough?," *Software Process: Improvement and Practice*, 2009.

[3] J. Ferguson, and S. Sheard, "Leveraging your CMM efforts for IEEE/EIA 12207," *Software, IEEE,* vol. 15, no. 5, pp. 23-28, 1998.

[4] J. Siviy, P. Kirwan, J. Morley *et al.*, "Maximizing your Process Improvement ROI through Harmonization," *http://www.sei.cmu.edu/process/research/prime.cfm (Accessed March 2010)*, 2008.

[5] J. Siviy, P. Kirwan, L. Marino *et al.*, "Strategic Technology Selection and Classification in Multimodel Environments" *http://www.sei.cmu.edu/process/research/prime.cfm (Accessed March 2010)*, 2008.

[6] C. Halvorsen, and R. Conradi, "A Taxonomy to Compare SPI Frameworks," *Software Process Technology*, pp. 217-235, 2001.

[7] J. Siviy, M. Penn, and R. W. Stoddard, *CMMI and Six Sigma Partners in Process Improvement*: Addison-Wesley, 2008.

[8] P. Kirwan, J. Siviy, L. Marino *et al.*, "Improvement Technology Classification and Composition in Multimodel Environments," *http://www.sei.cmu.edu/process/research/prime.cfm (Accessed March 2010)*, 2008.

[9] F. J. Pino, M. T. Baldassarre, M. Piattini *et al.*, "Harmonizing maturity levels from CMMI-DEV and ISO/IEC 15504," *Software Process: Improvement and Practice,* 2009.

[10] M. B. Chissis, and M. Konrad, "CMMI for Development, Version 1.2," *Addison-Wesley*, 2006.

[11] ISO/IEC 12207 - Systems and software engineering - Software life cycle processes, ISO*/IEC and IEEE*, 2008.

[12] B. Mutafelija, and H. Stromberg, "ISO 9001:2000 to CMMI v1.2 Map," *http://www.sei.cmu.edu/cmmi/casestudies/mappings/cmmi12-iso.cfm (Accessed March 2010)*, 2009.

[13] J. M. Siviy, and P. Kirwan, "SEI Approach to Harmonization," *SEI Webinar Series* 2008.

[14] D. K. Hitchins, *Advanced Systems: Thinking, Engineering, and Management* Arthech House, 2003.

[15] ISO/IEC, 9001 Quality Management Systems - Requirements, 2000.