Metric based Comparison of Reference Models based on Similarity

Simona Jeners, Horst Lichter, Elena Pyatkova
RWTH Aachen University, Research Group Software Construction
{simona.jeners, horst.lichter, elena.pyatkova}@rwth-aachen.de

Abstract

A variety of reference models such as CMMI, CobiT or ITIL support IT organizations to improve their processes. Although these reference models (RM) cover different domains they also share some similarities. There are organizations that address multiple domains and want to use different RMs. As RMs may overlap in some processes, we present an approach to compare RMs' procedures which is based on a common RM integration model and on similarity metrics. Our approach enables organizations to better understand RMs by identifying commonalities and specific details of the different RMs in order to avoid redundant improvement measures.

Keywords: *Reference Models, Similarity Metrics, Meta Models, Process Improvement*

1. Introduction

Nowadays, the software market is expanding and clients are requesting better, faster, and cheaper software products. However, the Standish Group regularly reports that the failure rate of IT-projects is still too high [1]. One important impact factor to project success is the quality of the applied development processes. Hence, more and more organizations are obligated to identify, structure, and improve their processes systematically. As 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 or CobiT can be considered and applied. Reference models (RMs) are collections of best practices based on experience and knowledge of many organizations. The adoption of multiple RMs allows an organization to exploit synergy effects between 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. This requires comparing RMs, i.e. comparing the procedures they define. However, this comparison raises some problems:

- **RMs define different structures and terminologies**. Because RMs are developed for different IT domains and are written by different authors, each single RM defines its own specific structure and language. This hampers the understanding and comparison of RMs.
- **RMs may address similar topics**. Although RMs are intended for different IT areas, they may address similar topics. For example, project or risk management is addressed in almost all RMs. Because procedures of RMs can be described either very generally or more concretely, organizations should recognize similar procedures to better understand and implement the abstract requirements of the more general one and to avoid redundancies. Moreover, the organizations can compare RMs to identify the differences that need to be implemented for compatibility to multiple RMs. This again requires a detailed RM comparison.

To solve the problems mentioned above, we have proposed a new model based integration approach for RMs (MoSaIC = Model based Selection of Applied Improvement Concepts), defining meta models to build an integrated view on different RMs [2]. As the comparison of RM procedures is crucial for MoSaIC, we have developed a metric based comparison approach based on a profound analysis of similarity theory and measurement methods.

The remaining of this paper is organized as follows. In the second chapter, we give an overview of the related work; especially we present relevant aspects of similarity theory. In the third chapter, we introduce our approach to calculate similarity between the RMs' procedures. Based on some excerpts of CMMI, CobiT and SPICE we finally discuss the results of our evaluation and give an overview of future work. Conclusions and a summary conclude this paper in the last chapter.

2. Related Work

There are some published approaches that consider the comparison of RMs. Ferreira, Machado and Paulk [3] define metrics to measure “size” and “complexity” of RMs. To measure the “size” the shared scope of the RM’s process areas (the number of common process areas) and their differences in the description detail is considered. Complexity is measured based on the internal coupling and the dependencies of the process areas. Therefore, this approach determines the differences between RMs based on the relations between the process areas but does not compare the content of RMs.

Content based comparisons are often provided by the owners of RMs offering mappings between process areas or practices (e.g. ISACA offers mappings between CobiT/CMMI and CobiT/ITIL). However, the mappings are often only bilateral and to certain extend subjective.

To overcome these problems, some authors try to integrate RMs using models that formalize the RMs on a fine granular level. Ferchichi and Bigand [4], Liao, Qu and Leung [5] as well as Malzahn [6] define a common structure to link RMs and reveal their similarities. For this purpose similar RM practices are connected manually. As x and y, we model the concepts of the RMs, such as activities, inputs, outputs and roles. The concept extraction and the definition of the semantic relation between them allow us to automatically calculate the similarity between the RMs.

2.1. The MoSaIC RM Integration Approach

To support organizations in adopting multiple RMs we have developed MoSaIC, a new model based approach to integrate different RMs and to select appropriate improvement concepts. It defines two meta models, the Integrated Structure Meta Model (IS Meta Model) and the Integrated Concept Meta Model (IC Meta Model) that are used to integrate the structure and the concepts of different RMs (see Fig. 1). In contrast to Ferchichi and Bigand [4] we model on a more fine grained level. Compared to the work of Malzahn [6], which also addresses this fine granularity, we base the comparison on different similarity relations and not only on “equivalence”. As we differentiate between several similarity relations we aim to get a more accurate degree of similarity between RMs resp. between procedures of RMs.

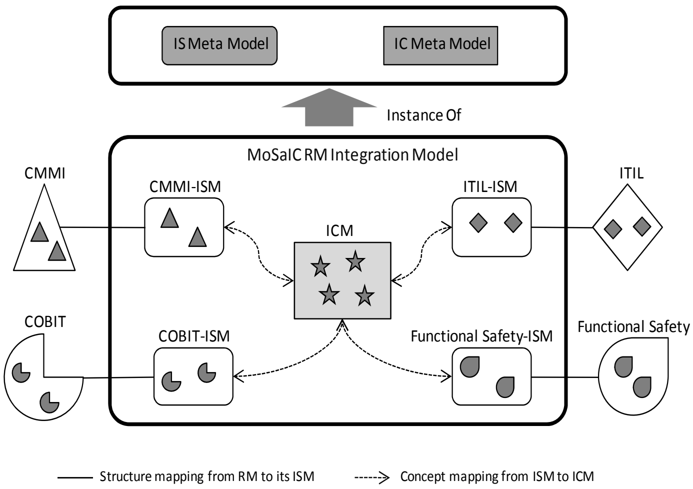

Figure 1. Model based integration approach of multiple RMs

The IS Meta Model defines a common structure to which the RMs can be reduced. The basic elements of the common structure are called *conceptual elements*. We differentiate between *activities*, *inputs*, *outputs*, *contexts* for activities and *roles*. These conceptual elements are used to model *procedures* defined by a RM. For example, the CMMI procedure SP2.7 *Establish and maintain the overall project plan* is composed of activity *Establish the overall project plan* with output *Overall project plan* and activity *Maintain the overall project plan* with input and output

Overall project plan. Each RM is modeled according to the IS Meta Model resulting in a respective RM-IS-Model.

The IC Meta Model contains the most important elements of the IS Meta Model, the conceptual elements and their semantic relations (*Equal*, *GeneralizationOf*, *ComposedOf*). For example, *Risk* and *Hazard* are equal, *Stakeholder* is a generalization of the concept *Project manager* and *Software requirements* are composed of *Functional requirements*. These concepts are identified according to the MoSaIC RM Modeling Rule Set [16] that we defined based on an in-depth analysis of CMMI, CobiT and Functional Safety.

The conceptual elements should be unique regarding their semantic interpretation thus trying to reach a normalization of the terminologies used in different RMs. Therefore, the ICM contains the closure of all semantically different concepts that appear in the RMs. The uniqueness of the ICM concepts and their traceability back to the original concepts of the RMs allows us to determine similar procedures of different RMs. For more details on the MoSaIC approach we refer to [2].

2.2. Similarity Theory

In general, similarity is an important property, because it is fundamental for our cognition. According to Goldstone and Son [9] similarity plays a key role in problem solving, remembering, prediction, and categorization. In fact, if there were no similar objects and events, an individual perceives each situation as a new one and has to learn for each particular object how to use it. The notion of similarity is applied in different domains. For instance, in geometry two objects are similar if they have the same shape; in psychology they are similar if they can be put into the same category. As there is no common definition of "similarity" we refer to the definition of Goodman [10]: *Objects are similar if they have a set of common features*.

There are several methods to determine similarity between objects. Based on measurement theory we distinguish the following four categories:

1. *Spatial methods* consider objects as points or vectors in the n-dimensional space [11]. Well-known spatial methods are the Cosine Similarity Measure or the Euclidean Distance;
2. *Feature-based methods* consider objects as a finite unsorted set of features; they calculate the similarity with respect to their features. For example, Tversky [12] combines the numbers of features that objects have in common and different to calculate their similarity.
3. *Transformational methods*, e.g. the Levenshtein Distance [13], consider the features of two objects and their order. They count the transformations needed to convert one object into the other; i.e., the smaller the number of transformations, the higher their similarity.
4. *Alignment methods* like Structure Mapping Engine [14] use features of objects and their relations to determine similarity.

As in our case the order of the objects' features should not be considered, the transformational methods could not be applied. Furthermore, the alignment methods compare two objects that are represented as hierarchies of features related by a certain relation. As we have different relations between the features, these methods could not be applied too. The feature-based methods consider only common and different features but not features that have something in common (that are not equal but also not different). This issue is considered by the spatial methods because they regard the distance between the features of compared objects. First, Ganesan et al. [15] proposed a variant of the Cosine Distance method to consider hierarchy information of the entities and thus the similarity distance between these in the hierarchy. As ICM concepts may be related by the *GeneralizationOf*-relation they may form hierarchies as well. The similarity between entities *n* and *m* of a hierarchy considers their depth in the tree, *depth(n)* and *LCA(n,m)*, the Lowest Common Ancestor; entity of maximum depth that is ancestor of *n* and *m*.

$$Sim\ (n,m) = \frac{2\ depth\ (LCA(n,m))}{depth(n) + depth(m)}$$

$$d_{p,q} = \sqrt{\sum_{i=1}^{n} w_i\ (q_i - p_i)^2}$$

The second similarity method that we use is Weighted Euclidean Distance. For two vectors of entities $q = (q1, q2,..\ )$ and $p = (p1, p2,..\ )$ and a vector of weights $w = (w1, w2,..\ )$ the Weighted Euclidean Distance between entities q and p is defined as:

3. Determining Similarity between RM Procedures

In the following we present the MoSaIC similarity algorithm. It has two RM procedures as input parameters and returns a similarity value between 0 and 1. In order to better structure the algorithm we introduce based on the IC Meta Model a new element type, called *activity unit*. An activity unit contains conceptual elements of different types: one activity and all its associated inputs, outputs, contexts and roles. Hence, a RM procedure contains one or more activity units. Since they are the most important elements of a RM procedure, the similarity of two procedures is based on the similarity of their activity units. Activity units are similar, if their conceptual elements are similar. Hence, the basic idea of our similarity algorithm is to determine similarity on different levels.

The algorithm firstly focuses on the similarity of the conceptual elements of respective activity units (level 1). Then it computes the similarity value of the activity units (level 2). Finally, the overall similarity value of the two RM procedures is computed based on the similarity values of their activity units (level 3). Below a pseudo code description of our similarity algorithm is given.

```
computeProcedureSimilarity(PROC p1, PROC p2)
  //Generate a set of all activity unit (AU) pairs (au_i, au_j) of p1 and p2
  AUPairSet = generateAUPairs (p1, p2)

  for each (au_i, au_j) in AUPairSet do
    // Level 1
    //Create a set of all Compared Conceptual Element Pairs (CCEP).
    //Each CCEP stores two CEs of the same type and their similarity value
    CCEPSet = computeSimilarityOfCEs(au_i, au_j)
    // Level 2
    //Compute similarity of au_i and au_j (Compared AU pair; CAUP) based on
    //their CCEPSet. CAUP stores the two AUs and their similarity value.
    //Add resulting CAUP to the set CAUPSet
    CAUPSet.add(computeSimilarityOfAUs(au_i, au_j, CCEPSet)
  end for

  // Level 3
  //Compute similarity of p1 and p2 based on their CAUPSet
  return computeSimilarityOfPROCs (CAUPSet)

end
```

The functions `computeSimilarityOfXXX()`are the most interesting ones because they implement a set of dedicated *similarity metrics* that we will present in the following chapter.

4. Similarity Metrics

We propose similarity metrics for procedures, activity units and conceptual elements. All metrics return a value between 0 and 1. The metric specifications are based on the following assumptions (A1-5) and have to meet the following requirements (R1-3):

A1 Activity is the most important conceptual element type.
A2 Role, input and context are the less important conceptual element types.
A3 Output is more important than role, input and context, but less important than activity.

A4 The similarity of all part-concepts to their whole-concept in a *ComposedOf*-relation is the same.
A5 All part-concepts of a common whole-concept are not similar.
R1 Each metric should be differentiable (different inputs cause different results), comparable, reproducible (the same input always leads to the same value) and plausible (the values meet the representative condition) [16] [17].
R2 The calculated similarity values should reflect the importance of the conceptual elements.
R3 The number of conceptual elements of an activity unit should not influence its similarity value.

4.1. Similarity between Conceptual Elements

On the first level, the function `computeSimilarityOfCEs()` computes similarity values for all pairs of conceptual elements (ce_1, ce_2) of the same type of two activity units. If there is a ce_{au1} of an activity unit and no type-equal ce_{au2} of the other activity unit, then a pair with a *null*-element is created. Obviously, the similarity of such a pair is 0. The *SimCE* similarity metric takes into account possible semantic relations between the CEs in the ICM. We define *SimCE* as follows:

$SimCE(ce_1, ce_2) = 1$ iff ce_1 and ce_2 refer to the same concept in ICM (relation *Equal*).

$SimCE(ce_1, ce_2) = 0$ iff ce_1 and ce_2 refer to different concepts in ICM and ce_1 and ce_2 are not related by any *ComposedOf* or *GeneralizationOf* relation.

$$SimCE(ce_1, ce_2) = \frac{2\, depth\,(LCA(ce_1, ce_2))}{depth(ce_1) + depth(ce_2)}$$

iff ce_1 and ce_2 refer to concepts connected by *GeneralizationOf* relations. *SimCE* is computed acc. to a variant of the Cosine Distance (see Chapter 2).
I.e., *SimCE* is high if CEs are located deeply in the *GeneralizationOf*-hierarchy (the hierarchy root does not specializes any other element) and LCA is close to both CEs.

$$SimCE(ce_1, ce_2) = \prod_{n}^{i=1} \frac{1}{partCoef \cdot |parts(l_i)|}$$

iff ce_1 and ce_2 refer to concepts that are connected by *ComposedOf* relations. Based on assumption (A5) *SimCE* is the percentage a part (ce_2) represents its whole (ce_1).
As all part-concepts have the same similarity regarding their common whole-concept (A4), *SimCE* depends on the number of the direct parts of the whole-concept. As there may be n part-of-levels l_i between ce_1 and ce_2 ($l_1 = ce_1$), *SimCE* is calculated by multiplying the similarity values of all part-of-levels between ce_1 and ce_2.
As a whole-concept may have only one part-concept, we use the coefficient *partCoef* to reflect that a whole-concept should have at least two part-concepts. I.e., $partCoef = 0.5$ iff $|parts(l_i)| = 1$, else $partCoef = 1$.

$SimCE(ce_1, ce_2) = SimCE(ce_1, ce_{int}) \cdot SimCE(ce_{int}, ce_2)$ iff ce_1 and ce_2 refer to concepts that are connected by both *ComposedOf and GeneralizationOf* relations.
SimCE is calculated according to the corresponding formulas until the intersection (ce_{int}) in the hierarchy tree and then the results are multiplied.

To better understand the *SimCE* metric, we demonstrate its application by some examples (see Fig. 2).

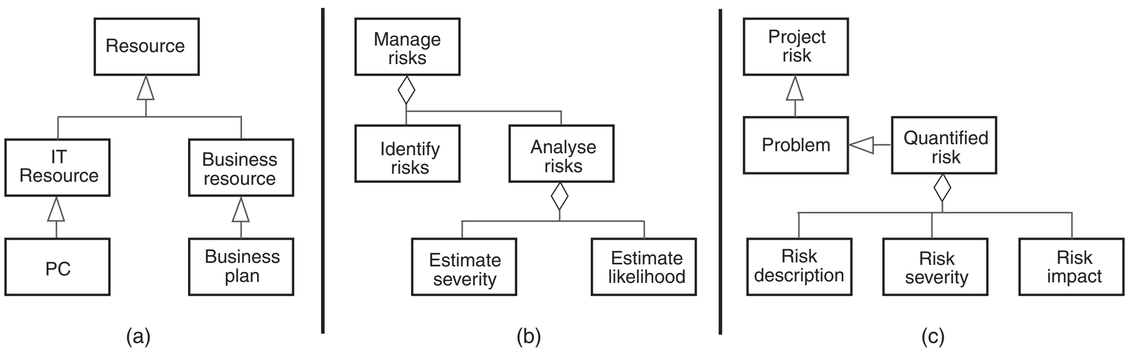

Figure 2. Examples of CE hierarchy trees

- Tree (a) represents a *GeneralizationOf*-hierarchy of output/input conceptual elements. The similarity value of *PC* and *Business Plan* is calculated as follows:

$$SimCE(PC, Business\ Plan) = \frac{2\ depth\ (Ressource)}{depth(PC) + depth(Business\ Plan)} = \frac{2\ \cdot 1}{3+3} = 0.33$$

- Tree (b) represents a *ComposedOf*-hierarchy of activity conceptual elements. Here the similarity value of *Manage risks* and *Estimate likelihood* is calculated as follows:

$$SimCE(Manage\ risks, Estimate\ likelihood) = \frac{1}{2} \cdot \frac{1}{2} = \frac{1}{4} = 0.25 \quad \text{//Multiplication of two part-of-levels}$$

- Tree (c) represents the combination of both. The similarity of *Project risk* and *Risk impact* is:

$$SimCE(Project\ risk, Risk\ impact) = SimCE(Project\ risk, Quantified\ risk) \cdot SimCE(Quantified\ risk, Risk\ impact) = \frac{2}{1+2} \cdot \frac{1}{3} = \frac{2}{9} = 0.22$$

4.2. Similarity between Activity Units

On the second level the function `computeSimilarityOfAUs()` calculates the similarity of two activity units au_1 and au_2 based on the *SimCE*-values for all type-equal pairs of contained conceptual elements (calculated on level 1). This is done applying a variant of the Weighted Euclidian Distance presented in Chapter 2. We consider the difference between two elements in the Euclidean Distance as the average between the *SimCE*-values of all CEs of the same type.

Let *Type = {act, out, in, role context}* be the set of all CE types. The algorithm performs the following steps:

1. For each ce_i of au_1 and au_2 the pair (ce_i, x) with the highest *SimCE*-value (best pair) is determined.
2. For each $t \in Type$ the *SimCE* average value of the best pairs is calculated (AVG_t).
3. The number of different CE types occurring in au_1 and au_2 are determined.
4. For each $t \in Type$ a type weight is calculated. According to assumptions (A1, A2, A3) we define type importance constants as follows: $IMP_{act} = 4$; $IMP_{out} = 3$; $IMP_{inp} = IMP_{role} = IMP_{context} = 1$. If one type is not present, its IMP-value is 0. Furthermore, as the number of occurring CE types should not influence the *SimAU*-value (R3), we calculate the weight for each $t \in Type$ dynamically as follows:

$$WEIGHT_t = \frac{IMP_t}{IMP_{act} + IMP_{out} + IMP_{in} + IMP_{role} + IMP_{context}}$$

The sum of the weights of all existing types in an activity unit is 1.

5. Finally, the similarity value of the activity units au_1 and au_2 is calculated as follows:

$$SimAU(au_1, au_2) = \sum_{t \in Type} WEIGHT_t \cdot AVG_t$$

4.3. Similarity between RM Procedures

On level 3 the function `computeSimilarityOfPROCs()` calculates the similarity value of two procedures p_1 and p_2 of different RMs. Again the highest *SimAU*-values for all activity units of p_1 and p_2 are determined. This leads to a set of activity unit best pairs (*AUBPSet*). The final similarity value of the two considered procedures is then calculated as the average similarity value of their activity unit best pairs.

$$SimPROC(p_1, p_2) = \frac{\sum_{aup \in AUBPSet} SimAU\big(au_{1(aup)}, au_{2(aup)}\big)}{n}$$

5. Example

In the following we explain how our algorithm and the proposed similarity metrics are applied to compute similarity between two different RM procedures. As an example we consider the following procedures:

- **P1** (CobiT 4.1, PO10.8.2): *Staff the roles based on available skills information*
- **P2** (CMMI-Dev, PP, SP 2.6): *Plan the involvement of identified stakeholders*

As Fig. 3 shows, in the ICM the activity *Plan involvement of stakeholders* of P2 is composed of the activity *Staff the roles* of P1. Furthermore, the ICM contains the activity *Identify relationships between stakeholders* which is also a part of activity *Plan involvement of stakeholders*. The outputs of both procedures *Stakeholder involvement plan* and *Staffing plan* are equal in the ICM. Obviously, both procedures contain only one activity unit.

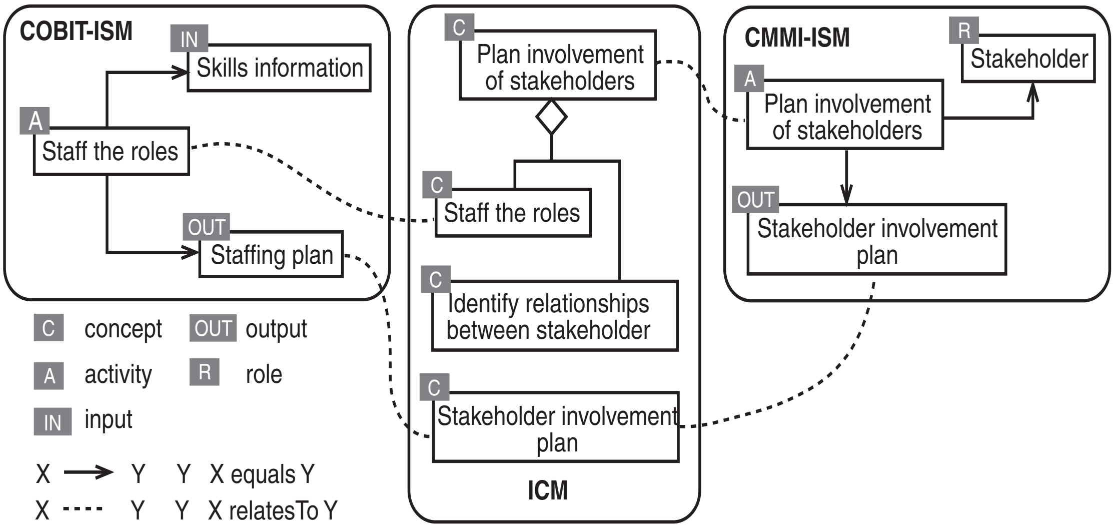

Figure 3. ISM and ICM models representing two CMMI and CobiT procedures

Level 1: All possible CE pairs of the same type are generated and their similarity values are calculated.

Activities	(Staff the roles, Plan involvement of stakeholders)	$SimCE(a_1, a_2) = \frac{1}{parts(a_2)} = \frac{1}{1 \cdot 2} = 0.5$
Outputs	(Staffing plan, Stakeholder involvement plan)	$SimCE(o_1, o_2) = 1$
Inputs	(Skills information, *null*)	$SimCE(i_1, null) = 0$
Roles	(Stakeholder, *null*)	$SimCE(r_1, null) = 0$

As there is exactly one pair for each CE type, it is also the best pair and averages are not needed.

Level 2: First, the weight value for each CE type is computed.

$WEIGHT_{out}$ = 1/3; $WEIGHT_{in}$ = 1/9; $WEIGHT_{role}$ = 1/9

$$WEIGHT_{act} = \frac{IMP_{act}}{IMP_{act} + IMP_{out} + IMP_{in} + IMP_{role} +} = \frac{4}{4+3+1+1} = \frac{4}{9}$$

The similarity value *SimAU* of the only activity unit pair is the following:

$$SimAU(au_{p1}, au_{p2}) = \frac{4}{9} \cdot \frac{1}{2} + \frac{1}{3} + \frac{1}{9} \cdot 0 + \frac{1}{9} \cdot 0 = \frac{5}{9} = 0.55$$

Level 3: As both procedures contain only one activity unit, the final similarity value for procedures P1 and P2 is: $SimPROC(P1,P2) = 0.55$

6. Evaluation

In the following we present the evaluation results of applying the proposed similarity metrics to procedures defined by CMMI, CobiT and SPICE. The evaluation was done as follows.

- First, we manually determined similar CMMI/CobiT and CMMI/SPICE procedures. As we found reasonable similarities between CobiT control objectives, CobiT control practices, CMMI specific-goals, generic-goals, -practices, sub-practices and SPICE practices, we consider these procedures for the evaluation. To detect similar CMMI/ CobiT procedures a mapping provided by the IT Governance Institute [19] was used; for CMMI and SPICE no official mapping could be used.
- Second, the defined MoSaiC Modeling Rules [20] were applied to create the ISM models and the common ICM model.
- Finally, we computed the similarity values for all procedure pairs (34 pairs: 17 CMMI-SPICE and 17 CMMI- CobiT). To compare the results to expert judgment, we mapped the similarity values to five categories: *identical [1,1]; high (0.7, 1); medium (0.3, 0.7]; low (0, 0.3]; different [0,0]*.

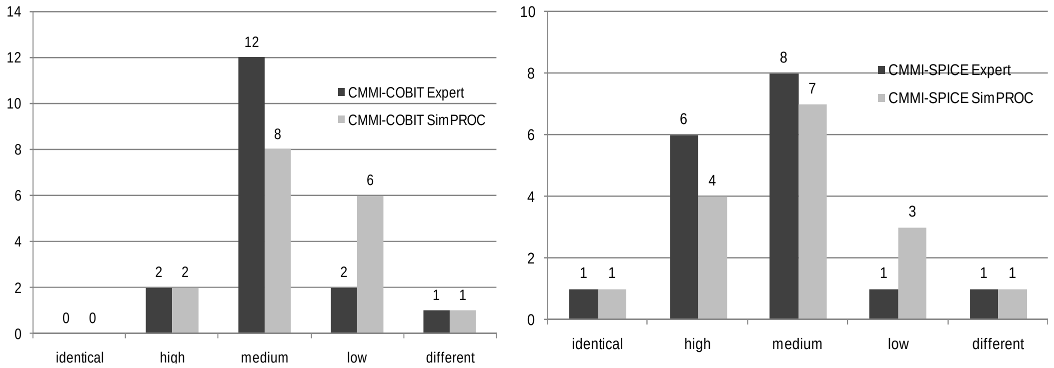

Figure 4. Evaluation results – expert judgement compared to SimPROC values

Fig. 4 summarizes the evaluation results. The charts visualize the comparison of expert judgments (EJ) with the computed *SimPROC* values (SP). They show that in most cases the *SimPROC* metric calculates similarity adequately. Below some positive examples of compared procedures are shown:

Compared Procedures	SP	EJ
SPICE ENG.2.BP2: Analyze the identified system requirements in terms of technical feasibility, risks and testability. *CMMI RD SP 3.3.3.:Analyze requirements to ensure that they are complete, feasible, realizable, and verifiable.*	0,88 (h)	h
SPICE SPL.2.BP8: The packaging for different types of media is identified. *CMMI PI SP 3.4.2.: Use effective methods to package the assembled product.*	0,58 (m)	m
COBIT PO1.3.3: Define roles and responsibilities of the stakeholders involved in the strategic planning process. *CMMI PP SP2.6: Plan the involvement of identified stakeholders.*	0,19 (l)	l

However, in some cases the results were wrong. For example, the *SimPROC* value for procedures "*PO10.7: Establish a formal, approved integrated project plan (covering business and information systems resources) to guide project execution and control throughout the life of the project*" and "*CMMI GP 2.2 Establish and maintain the plan for performing the process*" was 0.28 (*low*), while experts assess their similarity as *medium*. After an analysis we found some causes of the deviations and suggest the following improvements:

- **Similarity should be calculated for activity units, not for procedures.** Some procedures, e.g. most CMMI procedures, are compact and consist of one or only a few activity units. Other procedures, e.g. most CobiT procedures, are complex containing several activity units. Therefore, the similarity value of those procedures is low although they contain activity units that are very similar.
- **The MoSaIC Modeling Rules need to be improved.** Contexts should not be considered as standalone conceptual elements but as parts of activities, as a context concretizes an activity (e.g. SPICE SPL.2.BP13 "*The product is delivered (…) with positive conformation*"). If one procedure uses activities and not contexts to describe this concretization (e.g. CMMI SP 3.4.5 "*Deliver the product (…) and confirm receipt*") the metric does not compute the adequate similarity value.

- **The similarity calculation for activity units needs to be improved.** If an activity unit A contains a conceptual element CE and activity unit B contains all part-concepts of CE the similarity should be very high. The current metric does not reflect this adequately. Therefore, the calculation of *AVG* on level 2 should be changed, so that all pairs *(whole-concept, part-concept)* are considered a united pair. Hence, its *SimCE* value is the sum of its pairs' *SimCE* values.

The modeled procedures, the common ICM model containing all conceptual elements and semantic relations, the procedures comparison and the computed similarity values can be found in [20].

7. Conclusion and Future Research

In this paper, we presented an approach to compare procedures of different RMs based on similarity. To enable the comparison a normalization of the structure and terminology of the different RMs is needed. This was achieved by introducing two meta models, the ISMM to normalize the structure and ICMM to normalize the terminology. Based on the meta models and on an analysis of similarity methods we defined a notion of similarity and developed an algorithm that uses dedicated similarity metrics. The results obtained so far are promising.

In our future research, we want define a semi-automatic approach to extract concept elements and their semantic relations from RM descriptions. Furthermore we want to develop a dedicated tool support for all steps of the comparison approach to provide a much larger integrated model for the most popular RMS. This will offer organizations a better support to indentify similar procedures and activities in order to avoid redundant improvement measures and to save time and money.

8. References

[1] The Standish Group International, "CHAOS Summary 2009", Chaos, pp. 1 ~ 4, retrieved from http://www.statelibrary.state.pa.us/portal/server.pt/document/690719/, 2009.
[2] Simona Pricope, Horst Lichter, "A Model Based Integration Approach for Reference Models", Second Proceedings of 12th International Conference on Product Focused Software Development and Process Improvement, D. Caivano et al., Torre Canne, Italy, ACM, New York, pp. 6 ~ 9, 2011.
[3] Andre L. Ferreira, Ricardo J. Machado, Mark C. Paulk, "Quantitative Analysis of Best Practices Models in the Software Domain", Proceeding of the 17th Asia Pacific Software Engineering Conference, pp. 433 ~ 442, 2010.
[4] Anis Ferchichi, Michel Bigand, "An Ontology for Quality Standards Integration in Software Collaborative Projects", Proceedings of MDISIS'08 (Model Driven Interoperability for Sustainable Information Systems), Montpellier, FRANCE, 2008.
[5] Li Liao, Yuzhong Qu, H. Leung, "A Software Process Ontology and Its Application", Proceedings of IWFST 2005, International Workshop on Future Software Technology, Shanghai, pp. 1~10, 2005.
[6] Dirk Malzahn, "Assessing - Learning - Improving, an Integrated Approach for Self Assessment and Process Improvement Systems", Proceedings of ICONS '09, Fourth International Conference on Systems, Gosier, Guadeloupe, France, pp. 126 ~ 130, IEEE.doi:10.1109/ICONS.2009.31, 2009.
[7] Jirasak Gornmanee, Kittima Mekhabunchakij, "Translation ITIL Service Operation to Ontological Artifacts", JCIT, Vol. 6, No. 9, pp. 271 ~ 294, 2011.
[8] Suwanit Rungratri, Sasiporn Usanavasin, "Semantic based Approach Supporting CMMI Gap Analysis Process", JCIT, Vol. 7, No. 20, pp. 127 ~ 137, 2012.
[9] Robert L. Goldstone, Ji Yun Son, "Similarity", K.J. and R. Morrison, Cambridge Handbook of Thinking and Reasoning, Cambridge University Press, Cambridge, pp. 13 ~ 36, 2005.
[10] Nelson Goodman, "Problems and projects", Hackett Pub Co Inc, 1979.
[11] Patrick Groenen, "Past, Present, and Future of Multidimensional Scaling", Agrocampus Quest, Rennes, France, 2011.
[12] Amos Tversky, Itamar Gati, "Studies of similarity", Cognition and categorization, pp. 79 ~ 98, 1978.
[13] VI Lcvenshtcin, "Binary codes capable of correcting deletions, insertions, and reversals", Soviet Physics Doklady, pp. 707 ~ 710, 1966.

[14] Robert L. Goldstone, “Similarity, Interactive Activation, and Mapping”, Journal of Experimental Psychology: Learning, Memory, and Cognition, Vol. 20, No. 1, pp. 3 ~ 28, 1994.
[15] Prasanna Ganesan, Hector Garcia-Molina, Jennifer Widom, “Exploiting hierarchical domain structure to compute similarity”, ACM Transactions on Information Systems (TOIS), Vol. 21, No. 1, pp. 64 ~ 93, 2003.
[16] Norman Fenton, „Software Metrics. A Rigorous Approach”, Chapman & Hall, 1991.
[17] Jochen Ludewig, Horst Lichter, „Software Engineering – Grundlagen, Menschen, Prozesse, Techniken“, 2. Auflage, dpunkt.verlag Heidelberg, 2010.
[18] Simona Jeners, “MoSaIC Reference Model Modeling Rules”, Software Construction Group, 2012, www.swc.rwth-aachen.de/docs/MoSaIC/JDTCA/ModelingRules.pdf
[19] ISACA, “COBIT Mapping: Mapping of CMMI for Development V1.2 with COBIT 4.1”, 2011.
[20] Simona Jeners, Elena Pyatkova, “MoSaIC Reference Model Comparison Case Study”, Software Construction Group, 2012, www.swc.rwth-aachen.de/docs/MoSaIC/JDTCA/Comparison Example.pdf