Towards an Integration of Multiple Process Improvement Reference Models Based on Automated Concept Extraction

Simona Jeners, Horst Lichter, and Ana Dragomir

RWTH Aachen University, Research Group Software Construction
{jeners,lichter,adragomir}@swc.rwth-aachen.de

Abstract. A variety of process improvement reference models (IRM) such as CMMI, COBIT or ITIL support IT organizations. These reference models cover different domains (e.g. IT development, IT Services or IT Governance) but also share some similarities. There are organizations that address multiple domains and want to use different IRMs. As IRMs are described in different structures and are using different terminologies, we propose a tool based approach to extract IRMs' concepts and to normalize the terminologies. Our solution enables to semi-automatically build an integrated database of IRMs' concepts based on common meta-models and on a common terminology.

Keywords: process improvement, improvement reference models, natural language processing, CMMI.

1 Introduction

Nowadays, clients are requesting better and cheaper software products. However, the Standish Group regularly reports a high failure rate of IT-projects: 68% of IT-projects do not meet the deadlines nor achieve the requested quality or are cancelled [1]. One important factor to project success is the quality of the applied processes. Hence, more and more organizations want to establish 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 improvement reference models (IRM) such as CMMI (2006), ISO/IEC 15504 (2007), COBIT (2007) or Functional Safety IEC61508 (2010) can be considered and applied. IRMs are collections of best practices (often called procedures) based on experience and knowledge of many organizations.

The adoption and assessment of multiple IRMs bring additional benefits to organizations. The adoption allows organizations to exploit IRM synergy effects. On the one hand organizations can coordinately address different and common areas of IRMs. On the other hand the weaknesses of a single IRM can be overcome by the strengths of others. Furthermore, the assessment of the organizations' internal processes according to multiple IRMs increases the competition strength on the IT market.

D. Winkler, R.V. O'Connor, and R. Messnarz (Eds.): EuroSPI 2012, CCIS 301, pp. 205–216, 2012.

One premise for organizations to be able to exploit the synergy effects of multiple IRMs and to efficiently assess them is the availability of an integrated view of IRMs allowing to compare practices from different IRMs and to identify dependencies between them. Thus, organizations can effectively and efficiently adopt and assess multiple IRMs; the efficiency increases through an automated comparison approach.

1.1 Challenges and Goals

ISO/IEC 24744 [2] defines guidelines to "allow the combination of processes from different reference models, ease the development of new models and facilitates comparison of models". The main IRM integration challenge according to this standard is that IRMs "vary in format, content and level of prescription". In the following we detail this challenge:

Each IRM Has Its Own Structure

- Often IRMs use different names for the same structure element. For example in CMMI "procedures" are called "specific" or "generic practices" while in Functional Safety they are called "requirements". While a group of processes addressing the same topic is called "domain" in COBIT, it is entitled "category" in CMMI. This hampers the understanding of the IRMs' content.
- IRMs are written on different levels of abstraction. We found reasonable similarities between COBIT control objectives, COBIT control practices, CMMI specific-goals, generic-goals, -practices, sub-practices, SPICE practices and Functional Safety objectives and requirements. Without this mapping information, not all possible similar procedures of the IRMs can be identified.
- IRMs do not always contain all information needed to compare them on a detailed level. To identify the similarity degree of procedures, the different elements of a procedure such as inputs, outputs or roles have to be known. Only some IRMs define outputs of procedures (e.g. in CMMI under typical work products). Other elements such as inputs, roles or activities are not defined. Without suitable identification guidelines we observed that experts identify such elements differently (they map procedure content to different procedure element types).

Therefore, one goal of our approach is to define a common structure for multiple IRMs to model the IRMs' content consistently.

Each IRM Uses a Specific Terminology

- IRMs use different terms to express the same semantic concept. For example, in Functional Safety the term "hazard" and in CMMI the term "risk" is used for the same concept. SPICE uses phrases such as "risk associated with project life cycle" and CMMI uses "project risks". This hampers to understand and to compare IRMs.
- Each IRM uses a specific writing style. While in some of the IRMs verbs in their active form are used, some other IRMs make extensive use of passive, gerunds or nominalizations. This hampers understanding IRMs and identifying different terms that are semantically similar.

Therefore, the second goal is to define guidelines to normalize the terminology (terms and writing style) of IRMs.

As a manual modeling of IRMs cannot always be done consistently the third goal is to develop a tool box supporting the expert to model the IRM's content based on a common structure and terminology.

The remaining of this paper is organized as follows. In the second chapter we will present the MoSaIC approach. Then we introduce our concept to semi-automatically extract fine grained procedure elements according to the MoSaIC approach. Evaluation results, conclusions and a summary conclude this paper in the last two chapters.

2 The MoSaIC Integration Approach

In the following we describe the MoSaIC way to integrate IRMs. Based on our previous work [3] we improved MoSaIC with additional elements based on an in-depth language analysis of IRMs' procedures. First, we give a short overview of the integration approach and describe the MoSaIC meta-models to create a common structure and terminology. Additional guidelines to achieve a common terminology are presented in the last section of this chapter.

2.1 Meta-models

To support organizations in adopting multiple IRMs we have developed MoSaIC, a new model-based approach to integrate different IRMs 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). Both are used to integrate the structure and the terminology of different IRMs.

Figure 1 depicts the purpose of both meta-models and their respective concrete models, IRM-ISMs and ICM. The different structures of IRMs are represented by different geometrical shapes while the different terminology is symbolized by different small geometrical internal shapes. For each IRM a corresponding IRM-ISM can be created (e.g. CMMI-ISM) being part of the overall MoSaIC IRM Integration Model. All ISMs are instances of the IS Meta-Model. Hence, all ISMs use the same set of element types which makes them analyzable and comparable.

There are many contributions in the literature to the integration of IRMs and their comparison. In contrast to Ferchichi and Bigand [4] and Liao, Qu and Leung [5] we model each IRM on a more fine granular level. So, we do not only model information such as categories, processes or procedures (such as e.g. in [6], [7], [8] and [9]) but also procedure elements, such as activities, roles, artifacts (outputs or inputs). Guidelines for a uniform description of IRMs [2] also call for the definition of activities and artifacts.

The notion of a *procedure concept* (concept for short) is essential for MoSaIC. A concept is a word or the smallest combination of words contained in a procedure that has a unique meaning in the context of IRMs. Examples are "project plan" or "work breakdown structure".

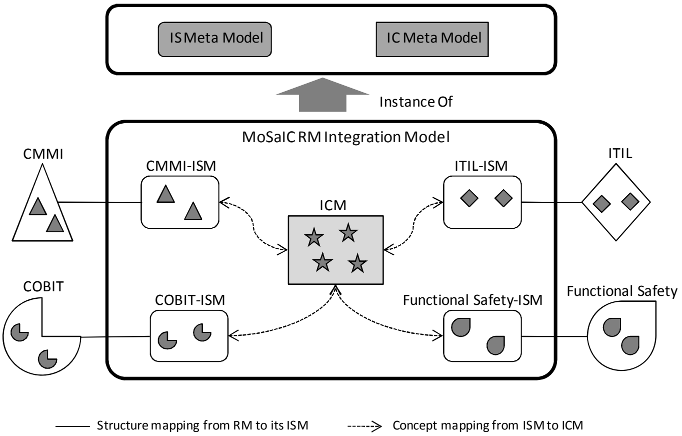


Fig. 1. Model-based integration approach of multiple IRMs

The fine model granularity allows comparing concepts of IRMs in detail. This reveals which concepts of different IRMs are similar and thus, what should an organization address more to adopt these IRMs. Malzahn [6], Soto and Münch [9] also model fine grained concepts but do not give guidelines to identify them. Based on an analysis of the writing styles applied to formulate procedures we added meta-model elements defining the different concept types on the language level (see package Language). This leads to a consistent modeling of concepts.

ICM (the only instance of its IC Meta-Model) is part of MoSaIC's IRM Integration Model as well. It defines all concepts and semantically links all IRM-ISMs by connecting related concepts across the borders of single IRMs. Compared to Malzahn [6] we base the concept comparison not only on "equivalence" but on different similarity relations (e.g. generalization) to get a more accurate degree of similarity. The uniqueness of the ICM concepts, their consistent identification and their traceability back to the original concepts of the IRMs allows to automatically identify the similarities of different IRMs.

Figure 2 shows the most important elements of the IS and IC Meta-Model. The elements of the IS Meta-Model are grouped in three packages.

- Package **Core** contains elements mostly defined by meta-models of existing IRMs. *ReferenceModel* represents an IRM and is structured by means of *Categories*. A category defines a certain topic that is addressed in one or more *ProcessAreas*. A process area addresses a topic to be improved by defining *Procedures*. By means of the *dependsOn* relation dependencies between procedures are modeled.
- Package **Concepts** contains elements to model concept information of IRMs on a fine grained level. We differentiate between *Activities, Artifacts* (*Inputs* and *Outputs*), *Contexts* for activities and *Roles*. These *ProcedureConcepts* are used to model IRM procedures. Each procedure concept from an IS Meta-Model *relatesTo* a concept in the IC Meta-Model.

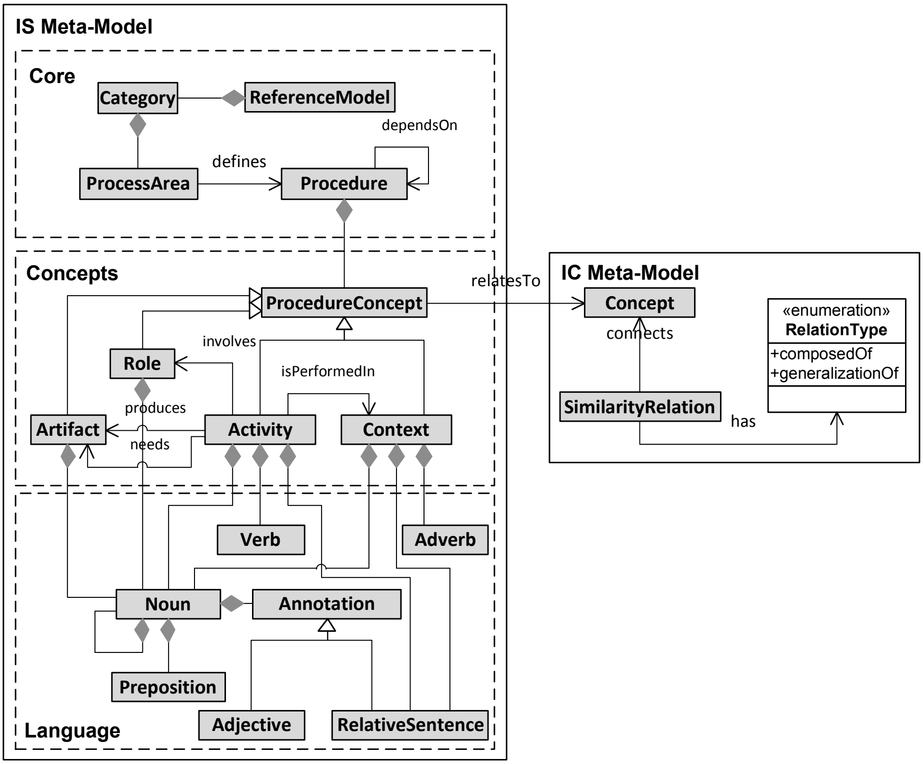


Fig. 2. Integrated Structure and Concept Meta-Model

- Package **Language** contains the syntactical elements used to describe concepts. Artifacts and roles are composed of one or more *Nouns*. Nouns can contain *Prepositions* and build composed Nouns ("records of quality assurance activities"). Nouns can be *annotated* by the use of an *Adjective* ("formal practice") or a *RelativeSentence* ("organisational structure that reflects business needs"). Activities can be composed of one verb and noun ("Create a supplier agreement") or by a verb and a relative sentence ("verify that personnel have the competencies"). *Contexts* can be composed of an *Adverb* ("Formally confirm the agreement") or a relative sentence ("Use effective methods to package the assembled product") or a noun introduced by a preposition ("Deliver with confirmation").

Additional rules developed to transform parts of a sentence to a concept according to the IS Meta-Model and examples support experts in consistently modeling concepts [10]. The IC Meta-Model contains only concepts (activities, roles, inputs, outputs or contexts) and their semantic relations (*generalizationOf*, *composedOf*). For example, "stakeholder" is a generalization of the concept "project manager" and "software requirement" is composed of "functional requirement". The concepts should be unique regarding their semantic interpretation thus trying to reach a normalization of the terminologies used in different IRMs. Therefore, the ICM contains the closure of all semantically different concepts appearing in the IRMs.

2.2 Normalization of Writing Style

According to their personality, educational and cultural background the authors of IRMs tend to express the same ideas differently. For example, while in COBIT and Functional Safety procedures are abundant in passive sentences in CMMI procedures are written almost completely using the active form. As we aim to normalize the language of IRMs and to propose a predefined syntactical structure for the modeled elements, we analyzed some recommendations from Requirements Engineering to write clear, consistent, complete and unambiguous requirements [11], [12].

We postulate that the active form verbs should always be used instead of passive voices, modal and present continuous tenses (as recommended in literature for scientific writing[1]) and instead of nominalizations and gerunds the corresponding verb in active form should always be used.

A normalized writing style allows to identify similar concepts and to build an ICM enabling to automatically compare procedures.

3 Automated Extraction of IRM Concepts

In this chapter we present our solution to support the extraction of concepts. At first we define rules to transform a procedure to its basic concepts. Afterwards we introduce Natural Language Processing (NLP) [13] tasks to apply those rules. Finally, we present a tool chain to create an integrated model of multiple IRMs according to the MoSaIC approach.

3.1 Transformation Rules

Our approach to automatically extract concepts takes a procedure as input and has two steps: In the first *normalization step* the procedure is transformed to a normalized writing style. In the second *extraction step* the concepts are extracted (see Fig. 3).

For both steps we define a set of rules transforming a procedure into concepts. The normalization rules specify how to transform the original IRMs' procedures into the normalized writing style. The extraction rules specify how to identify the normalized concepts according to the IS Meta-Model.

[1] Literature for Scientific Writing
http://abacus.bates.edu/~ganderso/biology/resources/writing/HTWgeneral.html,
http://www.columbia.edu/cu/biology/ug/research/paper.html,
http://maic.jmu.edu/journal/index/General_Rules_for_Scientific_Writing.pdf,
http://www.ugr.es/~agcasco/tierra/Docs/kowalski_scientific_writing.pdf,
http://faculty.uca.edu/march/bio1/sciwriting/writtips.htm.

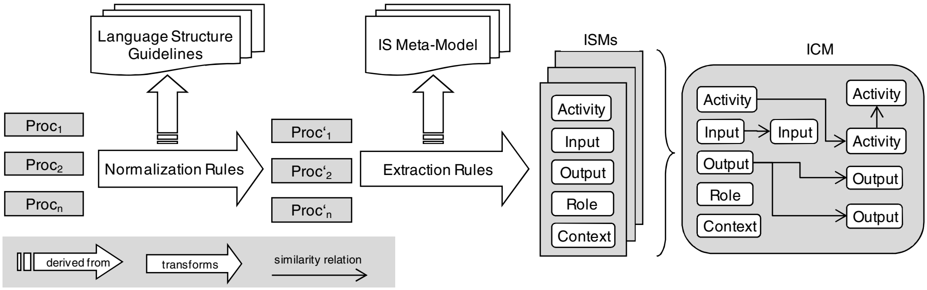


Fig. 3. Steps of normalization and concept extraction

Normalization and extraction rules transform and extract the concepts of procedures. In the definition of the rules the sign "→"means that the left side of the rule is transformed to the right side. The left and right side are specified using a variation of EBNF. But, we use the sign "+" instead of "," because in linguistics "+" is often used to concatenate grammatical structures. A representative example is the Activity Rule:

{Adverb} + Verb + Noun1 + {RelativeSentence} + {Preposition + Noun2} → Verb + Noun1

It defines that if a sentence contains a verb followed by a noun and other verb related grammatical elements (such as adverbs, relative sentences, nouns introduced by prepositions) then an activity composed of the verb and its noun is extracted. The complete set of rules with explaining examples can be found in [10].

To demonstrate the application of the rules we consider as input the CMMI procedure "Training for individuals to perform their roles effectively is provided". First, the normalization rule (VerbPassive | VerbModal | VerbPresent-Continuous → VerbActiv) transforms the passive form of the verb in its corresponding active form "Provide training for individuals to perform their roles effectively". Secondly, concepts are identified using the following extraction rules:

- Activity rule ({Adverb} + Verb + Noun1 + {RelativeSentence} + {Preposition + Noun2} → Verb + Noun1) identifies the activity "provide training for individuals".
- Output rule (Verb + Noun → Noun) identifies the output "training for individuals".
- Role rule (Verb + Preposition + Noun → Noun) identifies the role "individuals".
- Context rule: (Verb + Noun + Relative Sentence → Relative Sentence) identifies the context "to perform their roles effectively".

3.2 Natural Language Processing Support

There are some well-known NLP tasks that can be used to normalize and extract concepts. *POS Tagging* and *Lemmatization* support the normalizations rules. POS tagging chunks a sentence in part of speech (POS) tokens that are "a category to which a word is assigned in accordance with its syntactic functions"[2]. In our case, the POS tagger

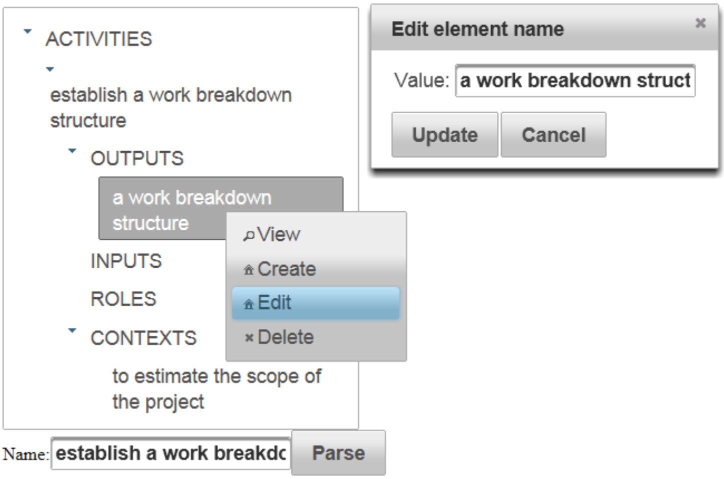
[2] "Part Of Speech", Oxford Dictionaries
http://oxforddictionaries.com/definition/part+of+speech?region=us

identifies verbs (e.g. “determine”), gerunds (e.g. “managing”). Lemmatization reduces a word to its canonical form allowing identifying the corresponding active form.

For the normalization of the writing style we use different NLP tasks to identify and extract concepts. As these transform a sentence in certain chunks, we call them *Phrase Chunking* tasks. *POS Tagging* identifies the grammatical structures of a sentence. Then *Noun Phrase Chunking* is applied to indentify noun phrases that are a combination of more nouns having a standalone semantic meaning (e.g. “work breakdown structure”). Furthermore, *Named Entity Recognition* determines all entities belonging to certain predefined categories. Such categories could be “persons”, “organizations”, “locations” and “roles”. This supports the identification of role concepts.

3.3 Tool Support

In this section we describe a tool chain to manually and semi-automatically model the IRMs’ content according to the MoSaIC approach. The tool chain takes as input a procedure and visualizes the normalized concepts in a tree (Fig. 4). Activities of procedures are presented at the root level. Attached inputs, outputs, contexts and roles are represented as children of their corresponding activities. The tree elements are editable; their type (input, output, etc) can be changed and they can be dragged and dropped to different positions in the tree. After validation and eventual modification of the proposed concepts, the user can store the elements in the IC and IS Model.



establish a work breakdown structure

a work breakdown structure

to estimate the scope of the project

Value: a work breakdown struct

Name: establish a work breakdc

Fig. 4. Screen shot from the MoSaIC concept extraction tool

Our tool chain, implementing a pipe-and-filter architecture, re-uses existing tools to perform the required NLP tasks: GATE (General Architecture for Software Engineering) [14] offers a package of tools to implement NLP tasks. Furthermore, it allows developing so called JAPE-transducers to implement the rules mentioned in section 3.1. The Dragon Toolkit [15] and RiTa.Wordnet [16] are further tools to implement the NLP task of lemmatization. Our tool chain consists of three main components (see Fig. 5).

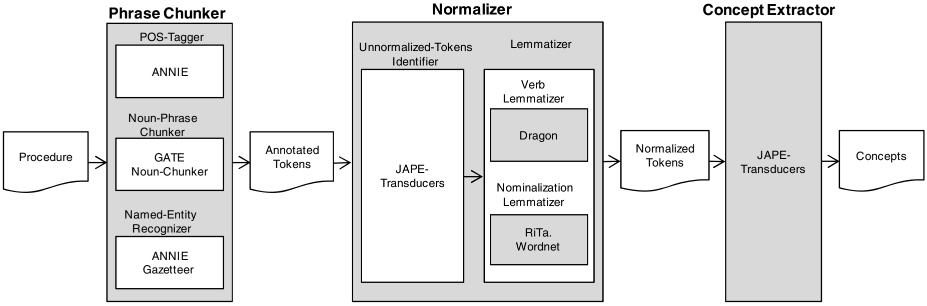


Fig. 5. Architecture of MoSaIC's NLP tool chain

The *Phrase Chunker* offers tools to split a given procedure in *Annotated Tokens*. The *POS-Tagger* (using the GATE plug-in ANNIE – based on a default lexicon and rule set resulting from the training on a large corpus taken from the Wall Street Journal) identifies tokens (such as verbs, nouns, adjectives, adverbs) and annotates them correspondingly. The *Noun-Phrase Chunker* (using GATE Noun-Chunker – an implementation of the Ramshaw and Marcus base noun phrase chunker [17]) annotates tokens as composed nouns. The *Named-Entity Recognizer* (using ANNIE Gazetter – based on a plain text file containing a list of categories and its elements) annotates tokens as category nouns.

The *Normalizer* transforms the *Annotated Tokens* that do not conform to our proposed writing style in *Normalized Tokens*. It contains two components: the *Unnormalized-Tokens Identifier* utilizes JAPE-Transducers to identify the tokens not conforming to our writing guidelines (see section 2.2). For example, in the procedure "a top-level work breakdown structure should be established" a transducer uses as input the verb tokens ("should", "be", "established") and marks these as passive activities. Another example is the recognition of nominalizations based on word suffixes, such as "ent" (e.g. deployment) or "ion" (e.g. categorization).The *Lemmatizer* (Dragon Toolkit and RiTA.Wordnet) uses as input the marks of the transducers and produces *Normalized Tokens*. Dragon Toolkit uses an English lemmatizer to extract the infinitive form of the verb (perfect continuous-verbs, gerunds and passive). Dragon also offers a Porter Stemmer. As stemmers usually produce less reliable results than lemmatizers, we preferred to use the second variant. RiTA.Wordnet uses relations (so called Cross-POS-relations) between concepts in Wordnet to identify the corresponding verb given a certain noun (nominalization).

The *Concept Extractor* uses the JAPE-Transducers to transform the normalized tokens in *IRM Concepts*. In our example, the transducer takes the noun token ("work breakdown structure"), the adjective ("top-level") and the active form of the passive activity ("establish") and mark these together as a concept activity.

4 Evaluation

In the following we present first evaluation results of applying the proposed concept extraction approach to procedures defined by CMMI, COBIT and SPICE. The evaluation was organized as follows.

- First, we manually determined procedures written in different writing styles (verbs in passive, perfect continuous or modal form; gerunds; nominalizations) and also containing different concept types (activities, inputs, outputs, roles, contexts).
- Secondly, we applied our tool chain to process such procedures (14 CMMI, 18 COBIT and 15 FS procedures) and extract the IRM concepts.
- Finally, we validated the obtained results with experts.

Figure 6 summarizes the results.

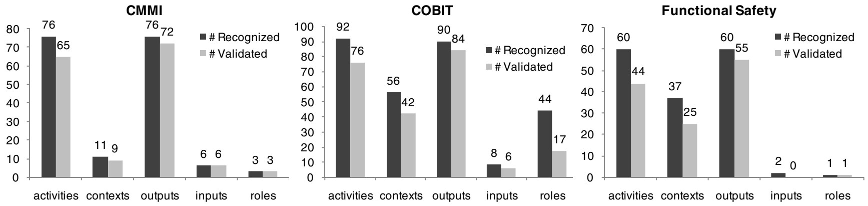


Fig. 6. Comparison of extracted and validated concepts

Obviously there were some failures in the transformation of IRM procedures to IRM concepts. After a thorough analysis we identified the following failure categories.

- **Normalization of writing styles.** Gerunds and nominalizations could not be identified always properly. Not all gerunds should be transformed to their corresponding active form (e.g. "The process is institutionalized as an optimizing process"; "optimizing" is an adjective and can be left unnormalized). Furthermore, not all nouns with the mentioned suffixes (see section 3.2) are nominalizations (e.g. "information"). Therefore, we decided only to provide concept candidates that can be corrected and modified by the user.
- **Complex syntactical structure of IRM's procedures.** Combinations of verbs and nouns connected by junctions ("and", "or") lead to an incomplete identification of activities and outputs (e.g. "Coordinate the activities and evaluation of multiple projects"). The first activity is not completely identified ("coordinate the activities" instead of "coordinate the activities of multiple projects"). This failure can be corrected by considering also junctions in the JAPE-Transducers.
- **Semantic based extraction**. The identification of inputs depends sometimes on the semantics of the verb (in "communicate policy", the concept "policy" is input and output; in "eliminate risks", the concept "risks" is only input). This failure category could be corrected by creating a dedicated verb database and by treating them separately. Furthermore, some prepositions introduce a context, an artifact or part of an artifact. It depends on the semantics of the sentence to identify the correct concept type. In "consider a mechanism for inclusion in the contract agreement" the preposition "for" introduces a context "for inclusion in the contract agreement" while in "consider a mechanism for monitoring the capability of supplier", the preposition "for" introduces a part of the output "mechanism for monitoring the capability of supplier". These cases cannot be corrected automatically and need manual interaction.

- **Not recognized grammatical tokens**. ANNIE does not recognize all verbs when performing the POS tagging. An improvement of ANNIE or its replacement by another POS tagger might alleviate this failure.
- **Small database of prepositions.** The identification of outputs, roles and inputs often depends on the prepositions that introduce the nouns. Inputs were identified as being outputs or parts of outputs/inputs were identified as roles. Our database with prepositions has to be enlarged.

Generally we observed that the quality of the proposed concept candidates is acceptable. The evaluation also showed that the syntactical structures in COBIT and Functional Safety are more complex than in CMMI.

5 Future Work and Conclusions

In this paper we presented a tool based approach to integrate multiple IRMs based on IRM concepts that can be extracted semi-automatically from IRM procedures. Based on NLP tasks, the presented tool chain allows a semi-automated modeling of IRMs concepts. Our approach supports the connection of similar IRM concepts by predefined relations and thus allows the identification of similar procedures that are composed of related concepts.

The language used in IRMs is sometimes too complex to be interpreted by machines and our tool chain does not always extract all the concepts. Some improvements can be done easily by improving the transducers that implement the transformation rules. Another improvement would be to enlarge our database by learning from the modeling decisions of the experts that are using our tool. However, our approach does not offer a fully automated way to extract concepts and to create an integrated view on different IRMs. But, it substantially supports the expert in this complex modeling task.

In the future, we will cover more areas of the most popular IRMs and we will consider also other IRMs such as SPICE or V-Model.

To summarize, the proposed tool based approach supports the expert to semi-automatically model IRMs according to a certain language structure, thus achieving a normalization of multiple IRMs. This realizes a basis for the integration and comparison of multiple IRMs, thus supporting organizations in an effective and efficient adoption and assessment of multiple IRMs.

References

1. The Standish Group International: CHAOS Summary 2009, Chaos, pp. 1–4 (2009), `http://www.statelibrary.state.pa.us/portal/server.pt/document/690719/` (retrieved)
2. ISO/IEC TR 24774: Systems and software engineering – Life Cycle Management – Guidelines for process descriptions (2010)

3. Pricope, S., Lichter, H.: A Model Based Integration Approach for Reference Models. In: Caivano, D., et al. (eds.) 2nd Proceedings of 12th Int. Conference on Product Focused Software Development and Process Improvement, Torre Canne, Italy, June 20-22, pp. 6–9. ACM, New York (2011)
4. Ferchichi, A., Bigand, M.: An Ontology for Quality Standards Integration in Software Collaborative Projects. In: Proceedings of MDISIS 2008, Model Driven Interoperability for Sustainable Information Systems, Montpellier, France. CEUR Workshop Proceedings, vol. 340, pp. 17–30 (2008)
5. Liao, L., Qu, Y., Leung, H.: A Software Process Ontology and Its Application. In: Proceedings of IWFST 2005, Int. Workshop on Future Software Technology, Shanghai, pp 1–10 (2005)
6. Malzahn, D.: Assessing - Learning - Improving, an Integrated Approach for Self Assessment and Process Improvement Systems. In: Proceedings of ICONS 2009, Fourth International Conference on Systems, Gosier, Guadeloupe, France, pp. 126–130 (2009), doi:10.1109/ICONS.2009.31
7. Siviy, J., Kirwan, P., Marino, L., Morley, J.: Process Architecture in a Multimodel Environment. White Paper, Software Engineering Institute, CarnegieMellon (2008), http://www.sei.cmu.edu/library/abstracts/whitepapers/multimodelSeries_wp4_processArch_052008_v1.cfm
8. Wang, Y., King, G., Dorling, A., Wickberg, H.: A unified framework for the software engineering process system standards and models. In: Proceeding ISESS 1999 Proceedings of the 4th IEEE International Symposium and Forum on Software Engineering Standards, p. 132. IEEE Computer Society, Washington, DC (1999)
9. Soto, M., Münch, J.: Using Model Comparison to Maintain Model-to-Standard Compliance. In: Proceedings of the ICSE Workshop Comparison and Versioning of Software Models (CVSM 2008), Leipzig, Germany, May 17, pp. 35–40 (2008)
10. Jeners, S.: MoSaIC Reference Model Modeling Rules, Software Construction Group (2012), http://www.swc.rwth-aachen.de/DOCUMENTS/MOSAIC/EuroSPI/ModelingRules.pdf
11. Grinder, J., Bandler, R.: The Structure of Magic II: A Book about Communication and Change. Science and Behavior Books, Palo Alto (1976)
12. Pohl, K., Rupp, C.: Requirements Engineering Fundamentals – A study Guide for the Certified Professional Requirements Engineering Exam – Foundation Level – IREB compliant. Rocky Nook, Inc., Santa Barbara (2011)
13. Kao, A., Poteet, S.R.: Natural Language Processing and Text Mining. Springer, London (2007)
14. Cunningham, H., Maynard, D., Bontcheva, K., et al.: Text Processing with GATE (Version 6), University of Sheffield Department of Computer Science, April 15 (2011), http://gate.ac.uk/releases/gate-6.1-build3913-ALL/tao.pdf
15. Zhou, X., Zhang, X., Hu, X.: The Dragon Toolkit Developer Guide, http://dragon.ischool.drexel.edu/tutorial.pdf
16. Howe, D.: RiTA.Wordnet, http://www.rednoise.org/rita/wordnet/documentation
17. Ramshaw, L., Marcus, M.: Text Chunking Using Transformation-Based Learning. In: Proceedings of the Third ACL Workshop on Very Large Corpora, pp. 82–94 (1995)