Coupling and process modeling
An analysis at hand of the eEPC

Daniel Braunnagel, Florian Johannsen, Susanne Leist

Department of Management Information Systems
University of Regensburg
Universitätsstraße 31
93053 Regensburg
daniel.braunnagel@wiwi.uni-regensburg.de
florian.johannsen@wiwi.uni-regensburg.de
susanne.leist@wiwi.uni-regensburg.de

Abstract: Business process modeling is a fundamental aspect in BPM initiatives. Being a central means of communication and documentation, both the quality and understandability of process models are decisive. However, the concept of process model quality is still not fully understood. The recent development has highlighted the role of coupling in models. Coupling is expected to represent an important dimension of quality for conceptual models. Still, contrary to software engineering, this perspective is hardly understood or adapted in form of metrics in process modeling. Therefore, this work collects diverse coupling metrics in the field of software engineering and transfers them to the eEPC modeling language. Once introduced and formally specified, the metrics serve for a discussion on coupling, process model quality with respect to coupling, and for their implementation.

1 Introduction

Business process modeling has gained considerable attention in BPM initiatives in recent years [MRv10;Be10;PSW08]. Process models help a business analyst in documenting and analyzing a company's business processes properly [Be10]. Based on thorough process documentation, improvement initiatives can be triggered whereas process simulation may be used for identifying weaknesses in the current process design and for evaluating alternative should-be process designs [va10]. Further, process models serve as a means for communication between stakeholders and software developers [GL06]. Therefore profound decisions on IT-investments are possible, indicating whether software is to be developed individually or standard software is to be bought for supporting a business process [Ag04;Be10]. Process models help to derive requirements software has to meet in a systematic way [BRU00].

However the described benefits of process modeling become blurred in case the process models cannot be understood by its users (see [GL06;HFL12;BRU00]). A high quality of the process models is thus decisive for BPM initiatives as well as for software development projects. Nevertheless, quality and understandability of process models are poorly understood concepts yet (see [HFL12;Mo05]). A process model is a "construction of the mind" which makes its quality hard to judge [Mo05]. As a consequence, evaluating conceptual models usually is an "art" and does not follow systematic guidelines [Mo05].

For assessing the quality of process models, a variety of quality dimensions, such as complexity, modularity, size or cohesion have been introduced and corresponding metrics have been developed (see e.g. [Va07;Me08;GL07]). Further, top-down frameworks (see e.g. [BRU00]), pragmatic guidelines (see e.g. [Si08]) and empirical studies (see e.g. [Re11]) can be found as approaches for operationalizing process model quality [MRv10]. Recently "coupling" has been presented as a quality dimension for business process modeling (see [Va07;Va08;KZB10]). While "coupling" is a well-established quality characteristic in information systems development, research has only just begun to investigate the "coupling" concept in the context of process model quality.

In the current understanding, coupling is generally defined as the connectedness of elements. It is generally used as a means to improve the understandability and maintainability of processes and respective models. [Va07] The actual way to achieve this goal, however, is subject to different implementations of the concept. As an example, *Vanderfeesten et al.* use coupling on the one hand to evaluate the variety of a process. Therefore they analyze whether or not a process allows so many alternatives that it becomes difficult to understand all of them [Va08]. On the other hand, *Vanderfeesten et al.* also use coupling as means to balance the alignment of parts of a workflow between an overly flexible or rigid structure [VRv08]. The diversity of available applications underlines the multiplicity of interpretations of the concept of coupling for process modeling.

In addition to the two above examples, a couple of further publications deal with the topic of coupling in process modelling (see section 2.1). Even though each of these publications introduces another interpretation of coupling, the currently available literature does not cover the definition extensively. As a consequence the understanding of what constitutes the quality of a process model from the perspective of coupling is limited. Also the means to measure and control the understandability and maintainability of processes or process models respectively remain limited.

The objective of the current paper is therefore to supplement the range of interpretations of coupling and its means of determining it by introducing new ways of measuring coupling in the field of process modelling.

A thorough discussion and analysis as well as a practical application of coupling in process modeling require a detailed and precise interpretation. The preferred means of the available publications (see section 2.1) are metrics, which are described either formally or semi-formally. Their specification describes precisely which elements of a process model and which connections are taken into account and how inferences on the quality of models are made upon them. Consequently this work uses metrics as means of introducing new ways to measure coupling in process modeling. Further, since metrics are necessarily language-dependent and in order to retain an insightful level of detail we focus on the modeling language eEPC.

The contributions of this paper are as follows. We supplement the current body of knowledge on coupling in process modelling with further interpretations of the concept. We therefore continue the work of discovering new factors determining the quality of processes and process models from the perspective of coupling. We provide precise definitions for each interpretation in the form of measures which are the means for a thorough discussion of what constitutes coupling in process modelling and for measuring and controlling the quality of process models.

The paper is structured as follows: In the following section we provide an overview of related work and basic terms. After introducing the methodology of transferring the metrics to EPC models (section 3), we present corresponding metrics in section 4. Section 5 explains the implementation of the metrics. The paper ends with a summary of the results, limitations and an outlook on future research.

2 Basics and Definitions

2.1 Coupling

The current literature on coupling in process modeling is preceded and influenced by literature on software engineering [Va07]. There, coupling is operationalized in the form of metrics to predict measure and control the quality of software code and its conceptual models respectively. Each metric implicitly defines a particular interpretation of coupling. E.g. one definition focuses the graph representation of software systems, i.e. the way nodes are connected by arcs, whereas another definition uses information theory to account for reused code [CK94]. Some further definitions can be found together with multiple metrics interpreting each of them (see section 3).

In process modeling, *Vanderfeesten et al.* present a definition for the concept of coupling: "Coupling is measured by the number of interconnections among modules. Coupling is a measure for the strength of association established by the interconnections from one module of a design to another. The degree of coupling depends on how complicated the connections are and on the type of connections." [Va07]. Here, coupling is generally considered as measurable and its key concept is the connections qualified by additional concepts (e.g. number, strength, etc…). As a means to improve the quality of conceptual models, reducing coupling is expected to improve the structure towards more understandable models. [Va07]

This definition founded several coupling metrics in process modeling. E.g. *Vanderfeesten et al.* present the coupling metric CP evaluating all pairs of nodes averaging their value over all pairs [VCR07]. Another metric by *Vanderfeesten et al.* is the cross connectivity metric analyzing the number of different possible paths in a process model [Va08]. Other authors use already available metrics from software engineering as starting point for their work. E.g. *Cardoso et al.* transfer metrics developed by *Halstead* (cf. [Ha77]) that use information theory to quantify code reuse [Ca06]. They further transfer metrics by *McCabe* (cf. [Mc76]) that quantify the paths through a model [Ca06]. The fan-in/fan-out metric, quantifying branches, developed by *Henry/Kafura* (cf. [HK81]), is transferred by *Mendling* (cf. [Me06]) and *Cardoso et al.* (cf. [Ca06]). Although these metrics exist, they do not exhaust the definition by *Vanderfeesten et al.* (cf. [Va07]). Further, the range of existing definitions already demonstrates how vague the current understanding of coupling is and that an extensive range of metrics with their precise definitions is necessary to render more precisely the currently fuzzy understanding. Further each distinctive metric introduces an additional application scenario. We therefore continue the previous work by transferring further metrics.

2.2 EEPC modeling

Event-Driven Process Chains (EPCs) are a popular standard for business process modeling [STA05;Me08]. EPC models can be extended by additional information in different views (e.g. data view, organization view, etc.) (see [STA05]) in which case literature then speaks of enhanced Event-Driven Process Chains (eEPCs). For the current work, relevant aspects of the eEPC can be formalized as follows (see [vOS05;Me08]).

An extended enhanced Event-Driven Process Chain (eEPC) is defined as weakly connected Graph $g = (N, A)$, fulfilling:

1. The set of nodes N is the union set of the four disjoint sets E, F, C, P and R where
   - E is the set of events $E = E_s \cup E_f \cup E_i$ and E_s, E_f and E_i are the disjoint sets of start-, final- and intermediate events with $|E_s| \geq 1$ and $|E_f| \geq 1$.
   - $F \neq \emptyset$ is the set of functions.
   - C is the set of connectors,
   - P is the set of process interfaces.
   - R is the set of resources, I encompasses the information elements: $I \subseteq R$
2. Each arc a in $A \subseteq (E \cup F \cup C \cup P \cup R) \times (E \cup F \cup C \cup R)$ connects two different nodes:
   - $|n \cdot| = 1$ for each $n \in F \cup E_i \cup E_s$ and $|\cdot n| = 1$ for each $n \in F \cup E_i \cup E_f$.
   - Resources are connected with undirected arcs.
3. Process interfaces have either an incoming or an outgoing arc: $\forall p \in P: (|\cdot p| = 1 \wedge |p \cdot| = 0) \vee (|\cdot p| = 0 \wedge |p \cdot| = 1)$

A hierarchical eEPC $eEPC_H = (G, h)$ is a set of eEPCs $g \in G$ and a partial relation $h: D \rightarrow G$ of the set D of decomposed functions or process interfaces in $Z: D \subset \bigcup_{g \in G}(P, F)$. For a node $d \in D$ where $h(d) = g$, g is called subprocess of d or process referenced by d.

The above definition covers the notation which will be used later on. A more exhaustive definition of the eEPC modeling language can be found in [vOS05;Me08].

3 Methodology

Figure 1 summarizes our methodology. First, conducting a literature review, we search for already existing coupling metrics in both, software engineering and process modeling. Second, we transfer discovered metrics from software engineering to process modeling. This step is detailed in figure 2. The work ends with discussing the results. The conceptual approach behind this work is presented in [BJ13]. There we present the idea as well as the expected results of the transfer.

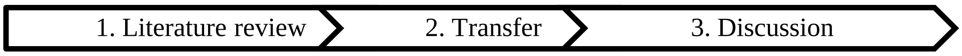

Figure 1: Methodology

For the review, the electronic databases Google Scholar, Computer.org, AISeL and Emerald Insight were queried (cf. [Co06;Vo09]). The hits, 46 peer-reviewed results were considered as relevant on the basis of their title or abstract, consist of 32 conference papers, nine journal papers, four technical reports and one book. Five sources defined metrics that are transferred and presented in this work. The remaining literature can be grouped as follows.

Use cases	[Ar07;BLS01;BS98;CZ10;El01;Go10;HCN98;LC01;Ma09;MB07;Va07;WK08]
Not transferred	[Al10;Bi10;Br98;BDW99;BDM97;Ch98;Gr09;GS08;HM95;JJ10;OTE06;Pe07;QLT06;QT09;RL92;SJ09]
Already specified	[Ca06;HK81;VCR07;Va08]
Redundant	[CYB09;Kh09;KZB10;RH97;SS05;Új10]
Transferred	[AKC99;AKC01;CK94;GS06;Ka11;Me06;PM06;RV04]

Table 1: Grouped literature review results

A first group discusses use cases, resp. consequences of high coupling. E.g. [BS98] discuss relations of coupling and run-time failures in software. The second group presents metrics that cannot be transferred to process models. E.g. [Gr09] present an approach involving runtime information which is not available in conceptual models. Third, sources discuss coupling metrics that were originally developed or transferred for eEPCs, (e.g. [Ca06;Me06;VCR07;Va08]). The existing metrics will be discussed more thoroughly in section 5. Finally, the fourth group of literature is redundant. These sources discuss metrics that are already part of the above groups. E.g. *Khlif et al.* transfer metrics to BPMN. We refer to the original description [Kh09]. A more detailed presentation of transferred and not transferred metrics can be found in [BJ13].

The remaining metrics were transferred as shown in figure 2.

Figure 2: Transfer

First, we identified the concepts of each metric's variables. Then, equivalent concepts in the eEPC notation were identified. Finally, the original concepts in the metric's definition were replaced to reformulate the metric.

4 Coupling metrics in the context of eEPC modeling

4.1 Process Coupling

Reijers/Vanderfeesten present the Process Coupling metric (see [RV04]). Its objective is the delineation of functions that are to be executed en block. Since overly large work units turn processes inflexible and overly small work units increase the number of handovers making processes failure-prone, the balanced delineation of functions in a workflow is a means for its improvement. The functions size is measured by the number of connected information elements. [RV04]

Identify concepts. The metric was originally defined for a graph of nodes and arcs representing information elements and operations respectively. The structure focuses the processing of information elements and is called information element structure. It is delineated into partitions representing activities. The metric calculates the quotient of the number of activities actually coupled and the number of activities possibly coupled. Activities that involve one or more common information elements are considered as "coupled". [RV04]

Identify equivalent concepts. To transfer the metric to eEPC modeling language, the procedure in section 3 was used. First, involved concepts were identified which are *information element, activity* and *operation*. Equivalent concepts were identified using the original description. Information elements exist in both domains with similar meaning. Activities express behavior and possess information elements as do functions in eEPCs. Operations, expressing the way information elements are combined at a very high level of detail, could not be matched with eEPC concepts. However, the calculation does not require them.

Reformulate metric. Adapted to the eEPC language and formalization from section 2.2, process coupling for eEPCs can be calculated as follows. The degree of Process Coupling *k* is the sum of coupled pairs of information elements divided by the maximum possible number of pairs.

- $k = \begin{cases} \frac{\sum_{f_x, f_y \in F} connected\ (f_x, f_y)}{|F| * (|F| - 1)}, |F| > 1 \\ 0,\ else \end{cases}$

Functions are connected with each other if they share a common information element.

- $connected(f_x, f_y) = \begin{cases} 1, if\ (f_x, i_i) \in A \wedge (f_y, i_i) \in A \wedge (f_x \neq f_y) \\ 0, else \end{cases}$

Application. The metric quantifies the interdependence of activities regarding information elements. To achieve a low degree of coupling, one reduces the number of coupled pairs, i.e. splitting tasks in such a way that information elements are grouped in the same function, or one increases the number of functions without introducing new pairs. A process with perfectly low coupling would use any information element only once as in- or output. A process with perfectly high coupling would be such that every step in a workflow depends on one and the same information. In such a process every step would come to a halt in case this one information was missing or the one person processing the information was ill, indicating a highly inflexible process design. However, it remains difficult to interpret the difference of two values, e.g. what is the impact of 10% more coupling? In summary, the metric has a special purpose, namely to quantify the dependency degree of process steps. It allows comparing different process designs and also gives a rough indication of how good or bad a design is regarding the coupling of activities.

4.2 Coupling of a module, intramodule coupling of a module

Allen et al. present a pair of metrics, the coupling of a module and the intramodule coupling of a module. They use information theory to quantify the amount of information in the structure with a special focus on connections between eEPCs. The authors argue that the cognitive limitation of a model user is a reason for misunderstandings and erroneous application if the model exceeds this limit. Therefore, the measure is a means of controlling the amount of information in the presented model. [AKC01]

Identify concepts. The metrics focus a graph with modules that partition nodes. Nodes from different partitions can be connected. The coupling of a module assesses the graph structure connecting different modules. Therefore, the graph is reduced to arcs connecting nodes from different modules. Second, the arcs are used to build a predicate table, i.e. the incidence pattern, for each node. Third, the relative frequency of each predicate is used to calculate its entropy. Finally, the entropy values are summed up. The second metric, the intramodule coupling of a module, follows the same procedure with arcs connecting nodes within eEPCs.

Identify equivalent concepts. The transfer focuses the graphs of eEPCs. Accordingly, nodes in an eEPC, i.e. functions, connectors, resources, etc. are considered as nodes here. Further, arcs from an eEPC are considered arcs here. Modules group nodes and arcs; therefore we use an eEPC for modules. However, the eEPC notation has no arcs between eEPCs. Therefore we propose using process references and decompositions as the extension of the control flow, i.e. as arcs connecting eEPCs.

Reformulate metric. As a consequence of the previous step, the definition from section 2.2 is extended in the context of this metric by arcs between eEPCs:

B is the set of intermodule arcs:

- B_p is the set of process references from eEPCs referencing each other.
- B_{es} is the set of pairs of decomposed function and start-events of the referenced models.

- B_{ef} is the set of pairs of an end-event of a referenced eEPC and a decomposed function referencing the eEPC.
- Then B is defined as: $B = B_p \cup B_{es} \cup B_{ef}$. Each tuple in B is a directed arc called intermodule arc.
- The intermodule sub-graph S_i^* consists of all the nodes of a group of eEPCs and arcs connecting nodes from different eEPCs with nodes from an eEPC *i*.

Inc_i^* is an incidence matrix of S_i^*: $Inc_i^* = (inc_{n,a}) \in S_i^*$ with $inc_{n,a} = \begin{cases} 1, if\ n_i \subset a \\ 0\ else \end{cases}$

A pattern pat_j is a sequence of 0 and 1 of line vectors of the matrix. Its probability $Prob(pat_j)$ is its frequency over the number of distinct patterns.

The information content of a sub-graph $S_i^*$is defined as:

- $I(S_i^*) = \sum_{j=0}^{n}(-\log_2 Prob(pat_j))$

Finally, the coupling of a module is defined as:

- $Coupling(m|MS) = \sum_{i \in m} I(S_i^*)$

The metric intramodule coupling of a module follows the same steps, although instead of arcs connecting nodes from different eEPCs, with arcs connecting nodes from the same eEPC for the intramodule sub-graph S_i°. The metric is defined as:

- $IntramoduleCoupling(m_k|MS) = \sum_{i \in m_k} I(S_i^\circ)$

Application. The metrics build on information theory and calculates the entropy of arcs as means of their complexity. It is therefore an ambitious attempt to quantify the cognitive load imposed on a model reader. The authors explain that a simpler structure is better understandable and indicated by a lower metric value [AKC01]. The practical application, however, is limited. For once, the metric does not account for the amount of information stemming from the nodes semantics. Further, the metric is constructed in a way that it is essentially driven by the number of nodes. Also, without any indicator about the actual cognitive limits of model readers, any calculated metrics value remains without reference and has therefore a weak indicational value. The metrics may therefore be used to compare two alternative layouts but do not allow any inference to be drawn about minimal, optimal or maximal values. Finally, a user will face trouble trying to understand what the metric actually does and why low values are important in this case. In summary, the metric is an interesting attempt to use information theory as a means of assessing the complexity of conceptual models. Nonetheless, the lack of reference values and complicated construction make the metric difficult to apply.

4.3 CBO, RFC

Chidamber and Kemerer introduced the CBO and RFC metric for object-oriented systems analyzing how classes are connected. They argue that highly connected classes are hardly reusable and difficult to change [CK94].

Identify concepts. The CBO metric counts the connections of one class with other classes, the RFC metric also considers the number of methods in the source class.

Identify equivalent concepts. These metrics (and the following one) use the concepts *software program, class,* and *method.* Previously published literature transferred them in

[Va07] and [Kh09]. Further, [GR00] mapped eEPC constructs onto the ontology of [We97] and [EW05;EW09] mapped programming constructs onto [We97]. Therefore *Weber's* ontology is used as mediator to compare both domains. Table 2 summarizes the transfer for the current context.

Method. The method in object-oriented programming expresses the behavior of classes [Ar06]. For their transfer to BPMN, *Khlif et al.* suggest the analogy to tasks. Further, *Vanderfeesten et al.* propose the analogy with operation elements. Since operations have no equivalent in eEPCs, functions are the best fit (c.f. section 4.1). Ontological analyses of [GR00;EW05;EW09] suggest that functions have their ontological equivalents in transformations and therefore their object-oriented equivalent in operations. As before, the degree of detail of operations is not shown in eEPCs. The ontological equivalent of methods is lawful transformations, subsets of all possible transformations. Nonetheless, considering the lawfulness being negligible here, the analogy of method and function fits close enough. [Va07;Kh09]

Class. In object-orientation, classes group methods into logical units [Ar06]. *Khlif et al.* map classes onto processes and sub-processes [Kh09]. *Vanderfeesten et al.* relate classes to activities arguing along the hierarchy of methods and classes [Va07]. Consequently, we suggest the equivalence of classes and sub-processes, since activities are already mapped onto functions. The ontological analyses of *Green/Rosemann* and *Evermann/Wand* (see [GR00;EW05;EW09]) suggest that classes find their ontological equivalent in functional schemas. They describe the temporal order of states, as is also done by process models. The ontological concept of a "process", as mentioned by *Green/Rosemann*, could not be found. Therefore, the current mapping relates a class onto a sub-process diagram.

Software program. The software program is a set of classes [Ar06]. The concept is ignored by *Khlif et al.* However, *Vanderfeesten et al.* argue along the hierarchy of concepts to map programs onto business processes. We follow their suggestion. [Va07;Kh09]

Object orientation	eEPC notation
Software program	All eEPCs of a process
Class	Sub-process diagram
Method (private)	Function
Method (public) /Interface	Process interface, decomposition

Table 2: Conceptual mapping

Reformulate metric. CBO is calculated as the number of connections from one eEPC to another eEPC.

- $CBO \quad = |C \cup P|$

RFC counts the number of process interfaces and decomposed functions plus the number of functions in the eEPC.

- $RFC \quad = |C \cup P| + |F|$

Application. The CBO metric quantifies the number of connections a model has with another model. The RFC metric additionally takes the number of functions of a model into account. Lower numbers indicate more readable models. The metrics from *Chidamber/Kemrerer* are well known and have been subject to empirical research (c.f. [HCN98]). Their application and interpretation is easy. They do, however, capture the complexity of the models only partly, e.g. they count the number of connections but do not evaluate them, and further do not incorporate all nodes, arcs and their meaning within models. Further, information about levels that constitute "easy" or "difficult" models is not available. In summary, the metrics are an easy and transparent way to analyze the number of connections between eEPCs. Still, without any information about the levels of the metric, the interpretation of a value is difficult. It remains to compare two alternative models.

4.4 Direct Coupling, Indirect Coupling, and Total Coupling

Gui/Scotts' intention is to improve the CBO and RFC metric incorporating transitive relations [GS08].

Identify concepts. The calculation takes three steps. First, the direct coupling between two classes is calculated as the quotient of commonly used methods to all methods in the first class. Second, the indirect coupling between two classes is calculated as the product of all direct coupling values on the longest path in between. Finally, the total coupling is calculated as the quotient of the sum of all indirect coupling values and the number of pairs of classes.

Identify equivalent concepts. Building upon the metrics CBO/RFC, the transfer of concepts in table 2 can be used again. EEPCs are used for classes, references for public methods, and functions for private methods.

Reformulate metric. The direct coupling metric calculates the quotient of process references between two eEPCs *g1* and *g2* and the functions and process interfaces in eEPC *g1*. This is formalized:

- $CoupD(g_1, \mathrm{g}_2) = \frac{|D_{g_1,g_2}|}{|F_{g_1} \cup P_{g_1}|}$

For a pair of eEPCs g_1 and g_2 connected by a path π (the longest available), the indirect coupling metric calculates the product of direct coupling values on the path:

- $CoupT(g_1, g_2, \pi) = \prod_{g_3, g_4 \in \pi} CoupD(g_3, g_4)$

The metric is aggregated over all eEPCs in a system calculating the average indirect coupling among all eEPCs G:

- $WTCoup = \frac{\sum_{i,j \in G} CoupT(i,j)}{|G|^2 - |G|}$

Application. The metrics extend the CBO metric by Chdiamber/Kemerer by paths over several connected eEPCs. It presents an indicator for the length of a process model and for how many different eEPCs need to be referred to in order to understand all paths in a process, where shorter lengths (a lower value) indicate a lower complexity. The metric is more sensitive than counting the number of eEPCs, since it takes into account which part of a process is reachable after all. I.e. a low value is reached if the parts are connected linearly so that a reader can follow the eEPCs in sequence. The value will rise if the parts are connected in circles and a reader has to refer to eEPCs repeatedly to follow a path through the process.

4.5 Conceptual coupling

Poshyvanyk/Marcus present the conceptual coupling metric that uses semantic information to calculate how far methods in object-oriented programming refer to the same semantic concept. A high semantic overlap indicates dependency causing complexity and should thus be avoided [PM06].

Identify concepts. The metric references information retrieval techniques to decompose a set of classes into semantic concepts. *Poshyvanyk/Marcus* combine vector space retrieval and latent semantic indexing on the source code of classes as text corpus. First, the source code of the methods is transformed into a term-method matrix showing the frequency of a term in a method. Second, the matrix is transformed using latent semantic indexing, analyzing which terms are highly correlated forming a semantic concept. The values allow the calculation of the distance of two classes, judging how close their concepts are (cf. [PM06]).

Identify equivalent concepts. The transfer takes special consideration of the authors' original intention. Therefore the transfer analyzes the role of the textual corpus. The role of a method is taken by an eEPC, whereas, instead of a class, the calculation is done with a group of eEPCs from the same process. In place of the terms from the source code, the redefined metric uses node labels.

Reformulate metric. Calculating the metric begins with building the term-eEPC matrix showing for each eEPC and each term its respective frequency. Second, a latent semantic analysis is applied on the matrix, reducing the matrix to its main components. The first metric, the conceptual similarity between eEPCs, CSM, uses the cosine of the vectors of two eEPCs in the reduced matrix as measure of distance.

- $CSM(g_k, g_j) = \begin{cases} \frac{\overrightarrow{g_k}' * \overrightarrow{g_j}}{\left\|\overrightarrow{g_k}\right\| * \left\|\overrightarrow{g_j}\right\|} \; if \; \frac{\overrightarrow{g_k}' * \overrightarrow{g_j}}{\left\|\overrightarrow{g_k}\right\| * \left\|\overrightarrow{g_j}\right\|} \geq 0 \\ 0 \; else \end{cases}$

The second measure is the similarity of an eEPC g with a group of eEPCs gg. Therefore, the average conceptual similarity of one eEPC with all eEPCs of the group is calculated:

- $CSMMg(g_i, gg_j) = \frac{\sum_{m_j \in mv_j} CSM(g_i, g_j)}{|g_j \in gg_j|}$

Third, the conceptual similarity of an eEPC group with another eEPC group is calculated as the average CSMMg of their eEPCs:

- $CSMgMg(gg_i, gg_j) = \frac{\sum_{m_i \in mv_i} CSMMg(g_i, gg_j)}{|g_i \in gg_i|}$

Finally, the conceptual coupling of an eEPC group can be calculated as the average coupling of a group with all other eEPC groups:

- $CCMg(gg_i) = \frac{\sum_{mg_j \in MG} CSMgMg(gg_i, gg_j)}{n-1}$

Application. The conceptual coupling metric uses an information retrieval technique that discovers semantic concepts and evaluates the degree of redundancy in the concepts, resp. terms, among eEPCs. It therefore analyzes whether either nodes are labeled ambiguously or similar tasks appear in different contexts and models. High values indicate a high semantic overlap, i.e. many common terms. The same terms reused in different contexts impair understandability. Our adaption does not define the construct of a group of eEPCs strictly, since it depends on the use case. The groups should be formed by domain, i.e. groups of processes that are supposed to deal with the same terms or not, as for example eEPCs for processes that belong together.

5 Implementation

In the previous sections the metrics were presented, transferred, re-specified, and their contribution to the assessment of process model quality was discussed. However, as can be taken from the definition of some of the metrics, the complicated calculation of some of the metrics makes their practical application tedious. E.g. the conceptual coupling metric requires a singular value decomposition of a term-model matrix over all terms used. As an application aid, we implemented the metrics presented in this work in the form of a Plug-In (available at https://svn.win.tue.nl/trac/prom/browser/Packages/CouplingMetrics/) for ProM. ProM is a framework offering several techniques for process mining and model analysis (cf. [va05]). The implementation assumes to find eEPC elements as defined but remains oblivious to their source format. To ensure the functionality, we also extended the EPML-Interface of ProM for eEPC elements that are required for the metrics. Contrary to proprietary formats such as ARIS-XML or VDX, EPML is a platform-independent XML-schema with a publicly available schema-definition (cf. [MN06]). We used the plugin with twelve different eEPC models to gain a first impression about the applicability of the metrics and the plugin. It showed that though the implementation produced values for each metric and model, their application suffers from a lack of reference. Thus it remains unclear how strong the effect onto the reader is if models perform e.g. 10% better or worse regarding a certain metric. Nonetheless, the metrics serve for the comparison of two models, giving a rough indication if one model performs better or worse than another in respect to a metric (c.f. [BJ13].).

6 Summary, limitations and outlook

This work discusses the topic of “coupling” in process modeling. Even though it is recognized as an important quality dimension (see [Va07;VRv08]) for process models it has not been explained in detail yet. Coupling metrics exist, especially in neighboring disciplines such as software engineering, based on individual and heterogeneous perceptions of coupling, while the understanding of coupling in process modeling is sparse and vague. Our research addresses this gap by analyzing and transferring ideas on “coupling” from the field of software engineering to gain a better understanding and application of this ill-defined concept. Thus our contribution is the transfer of a well-established means of controlling and managing quality from systems development to process modeling. Therefore, our work supplements the metrics allowing the measurement and management of the coupling of process models. Next to their application, the metrics provide additional definitions of the concept of coupling. They constitute elementary groundwork for the discussion of coupling in process models as well as for the fuzzy concept of process model quality and understandability in general.

However, there are limitations. First, our understanding of coupling builds on preliminary work on coupling (see section 3). Future developments regarding coupling might bring new interpretations requiring our transfer to be repeated. Second, the transfer was influenced by subjectivity regarding the interpretation of equivalent concepts. However subjectivity was mitigated by two researchers conducting the procedure and consolidating the results. Finally, we focused eEPCs to provide a reasonable level of detail. The metrics' interpretation will differ for other languages such as e.g. BPMN or UML.

In future work, the metrics will be evaluated empirically. We will analyze which metrics, and thus underlying perspectives, influence process model understandability most. Based on these insights, guidelines for producing process models that are easy to understand (regarding coupling) can be formulated. They will then be tested with practitioners and adapted to their specific needs.

7 References

[Ag04] Aguilar-Savén, R. S.: Business process modeling. In Int. J. of Production Economics 90 (2), 2004; pp. 129–149.

[AKC99] Allen, E. B.; Khoshgoftaar, T. M.; Chen, Y.: Measuring coupling and cohesion. In : 6th Int. Software Metrics Symp. METRICS. USA, 04.11. IEEE, 1999; pp. 119–127.

[AKC01] Allen, E.; Khoshgoftaar, T.; Chen, Y.: Measuring Coupling and Cohesion of Software Modules. In : 7th Int. Software Metrics Symp. METRICS. UK, 04.04. IEEE, 2001.

[Al10] Allier, S.; Vaucher, S.; Dufour, B.; Sahraoui, H.: Deriving Coupling Metrics from Call Graphs. In : 10th Working Conf. on Source Code Analysis and Manipulation. SCAM. Timisoara, Rumania, 12.- 13.09. IEEE, 2010; pp. 43–52.

[Ar06] Armstrong, D. J.: The quarks of object-oriented development. In Commun. ACM 49 (2), 2006; pp. 123–128.

[Ar07] Arshad, F.; Khanna, G.; Laguna, I.; Bagchi, S.: Distributed Diagnosis of Failures in a Three Tier E-Commerce System. Purdue University (ECE Tech. Reports, 354), 2007.

[BRU00] Becker, J.; Rosemann, M.; Uthmann, C. von: Guidelines of Business Process Modeling. In (van der Aalst, W. M.; Desel, J.; Oberweis, A. Eds.): Business Process Management. Springer, Berlin, Heidelberg, 2000; pp. 241–262.

[Be10] Becker, J.; Thome, I.; Weiß, B.; Winkelmann, A.: Constructing a Semantic Business Process Modelling Language for the Banking Sector. In EMISA 5 (1), 2010; pp. 4–25.

[BLS01] Beyer, D.; Lewerentz, C.; Simon, F.: Impact of Inheritance on Metrics for Size, Coupling, and Cohesion in Object-Oriented Systems. LNCS 2006, 2001; pp. 1–17.

[BS98] Binkley, A.; Schach, S.: Validation of the coupling dependency metric as a predictor of run-time failures and maintenance measures. In (Torii, K.; Futatsugi, K.; Kemmerer, R. A. Eds.): Proceedings of the 1998 Int. Conf. on Software Engineering. ICSE. Kyoto, Japan, 19.04-25.04. IEEE, 1998; pp. 452–455.

[Bi10] Birkmeier, D.: On the State of the Art of Coupling and Cohesion Measures for Service-Oriented System Design. In (Santana, M.; Luftman, J. N.; Vinze, A. S. Eds.): 16th Americas Conf. on Information Systems. ACIS. Lima, Peru, 12.08-15.08. AIS, 2010.

[BJ13] Braunnagel, D.; Johannsen, F.: Coupling Metrics for EPC Models. In (Alt, R.; Franczyk, B. Eds.): Int. Conf. on Wirtschaftsinformatik. WI2013. Leipzig, 27.02 - 01.03.2013, 2013; pp. 1797–1811

[Br98] Briand, L. C.; Daly, J.; Porter, V.; Wust, J.: A comprehensive empirical validation of design measures for object-oriented systems. In : 5th Int. Software Metrics Symposium. METRICS. Bethesda, USA, 20.03- 21.03. IEEE, 1998; pp. 246–257.

[BDW99]Briand, L. C.; Daly, J.; Wust, J.: A unified framework for coupling measurement in object-oriented systems. In IEEE Trans. Software Eng. 25 (1), 1999; pp. 91–121.

[BDM97] Briand, L.; Devanbu, P.; Melo, W.: An Investigation into Coupling Measures for C++. In (Adrion, R. W.; Fuggetta, A.; Taylor, R. N.; Wasserman, A. I. Eds.): Proc. of the 19th Int. Conf. on Software Engineering. ICSE. USA, 17- 23.05.1997; pp. 412–432.

[Ca06] Cardoso, J.; Mendling, J.; Neumann, G.; Reijers, H. A.: A Discourse on Complexity of Process Models. LNCS 4103, 2006; pp. 117–128.

[CYB09]Chen, J.; Yeap, W. K.; Bruda, S. D.: A Review of Component Coupling Metrics for Component-Based Development. In : WRI World Congress on Software Engineering. WCSE. Xiamen, China, 19.05.2009 - 21.05.2009. WRI, 2009; pp. 65–69.

[CK94] Chidamber, S. R.; Kemerer, C. F.: A Metrics Suite for Object Oriented Design. In IEEE Trans. Software Eng. 20 (6), 1994; pp. 476–493.

[Ch98] Cho, E. S.; Kim, C. J.; Kim, S. D.; Rhew, S. Y.: Static and Dynamic Metrics for Effective Object Clustering. In : 5th Asia-Pacific Software Engineering Conf. APSEC. Taipei, Taiwan, 02.12-04.12. IEEE, 1998; pp. 87- 85.

[CZ10] Chowdhury, I.; Zulkernine, M.: Can complexity, coupling, and cohesion metrics be used as early indicators of vulnerabilities? In (Shin, S. Y.; Ossowski, S.; Schumacher, M.; Palakal, M. J.; Hung, C.-C.; Chowdhury, I.; Zulkernine, M. Eds.): Proc. o. Symposium on Applied Computing. SAC. Sierre, Schweiz, 22.- 26.03. ACM, 2010; pp. 1963–1969.

[Co06] Cooper, H. M.: Synthesizing research. 3rd ed. SAGE, Thousand Oaks, USA, 2006.

[El01] El-Emam, K.: Object-Oriented Metrics: A Review of Theory and Practice. NRC-CNRC (NRC, 44190), 2001.

[EW05] Evermann, J.; Wand, Y.: Ontology based object-oriented domain modelling: fundamental concepts. In Requirements Eng 10 (2), 2005; pp. 146–160.

[EW09] Evermann, J.; Wand, Y.: Ontology Based Object-Oriented Domain Modeling. In Journal of Database Management 20 (1), 2009; pp. 48–77.

[Go10] González, L. S.; Rubio, F. G.; González, F. R.; Velthuis, M. P.: Measurement in business processes: a systematic review. In BPMJ 16 (1), 2010; pp. 114–134.

[Gr09] Green, P.; Lane, P. ; Rainer, A.; Scholz, S. B.: An Introduction to Slice-Based Cohesion and Coupling Metrics. University of Hertfordshire (Technical Report, 488), 2009.

[GR00] Green, P.; Rosemann, M.: Integrated Process Modeling. In Information Systems and E-Business Management 25 (2), 2000; pp. 73–87.

[GL06] Gruhn, V.; Laue, R.: Adopting the Cognitive Complexity Measure for Business Process Models. In: 5th IEEE Int. Conf. on Cognitive Informatics. ICCI. Beijing, 2006. IEEE, 2006; pp. 236–241.

[GL07] Gruhn, V.; Laue, R.: Approaches for Business Process Model Complexity Metrics. In (Abramowicz, W.; Mayr, H. C. Eds.): Technologies for Business Information Systems. Springer, Berlin, 2007; pp. 13–24.

[GS06] Gui, G.; Scott, P. D.: Coupling and cohesion measures for evaluation of component reusability. In (Diehl, S.; Gall, H.; Hassan, A. E. Eds.): Proc. o. Int. Workshop on Mining Software Repositories. MSR. Shanghai, China, 22.-23.03. ACM, 2006; pp. 18–21.

[GS08] Gui, G.; Scott, P. D.: New Coupling and Cohesion Metrics for Evaluation of Software Component Reusability. In : 9th Int. Conf. for Young Computer Scientists. ICYCS.

Zhang Jia Jie, China, 18.11. IEEE, 2008; pp. 1181–1186.

[Ha77] Halstead, M. H.: Elements of software science. Elsevier, New York, 1977.

[HCN98] Harrison, R.; Counsell, S.; Nithi, R.: Coupling metrics for object-oriented design. In : 5th Int. Software Metrics Symp.. METRICS. USA, 20-21.03. IEEE, 1998; pp. 150–157.

[HK81] Henry, S.; Kafura, D.: Software Structure Metrics Based on Information Flow. In IEEE Trans. Software Eng. 7 (5), 1981; pp. 510–518.

[HM95] Hitz, M.; Montazeri, B.: Measuring coupling and cohesion in object-oriented systems. In : Proc. of 3rd Int. Symp. on Applied Corporate Computing. Mexico, 25. – 27.10.1995.

[HFL12] Houy, C.; Fettke, P.; Loos, P.: Understanding Understandability of Conceptual Models. LNCS 7532, 2012; pp. 64–77.

[JJ10] Joshi, P.; Joshi, R.: Microscopic coupling metrics for refactoring. In : 10th Working Conf. on Source Code Analysis and Manipulation. SCAM. Timisoara, Rumania, 12.- 13.09. IEEE, 2010; pp. 145–152.

[Ka11] Kazemi, A.; Azizkandi, A. N.; Rostampour, A.; Haghighi, H.; Jamshidi, P.; Shams, F.: Measuring the Conceptual Coupling of Services Using Latent Semantic Indexing. In (Jacobsen, H.-A.; Wang, Y.; Hung, P. Eds.): IEEE Int. Conf. on Services Computing. SCC. Washington, USA, 04.06-09.06. IEEE, 2011; pp. 504–511.

[Kh09] Khlif, W.; Makni, L.; Zaaboub, N.; Ben-Abdalla, H.: Quality metrics for business process modeling. In (Revertia, R.; Mladenov, V.; Mastorakis, N. Eds.): Proceedings of the 9th WSEAS Int. Conf. on Applied computer science. WSEAS ACS'. Genua, Italien, 17.10.2009. World Scientific and Engineering Academy and Society, 2009; pp. 195–200.

[KZB10] Khlif, W.; Zaaboub, N.; Ben-Abdalla, H.: Coupling metrics for business process modeling. In WSEAS TRANSACTIONS on COMPUTERS 9 (1), 2010; pp. 31–41.

[LC01] Lee, A.; Chan, C. H. C.: An Exploratory Analysis of Semantic Network Complexity for Data Modeling Performance. In : 5th Pacific Asia Conf. on Information Systems. PACIS. Seoul, Korea, 20. - 22.05.2001. AIS, 2001.

[Ma09] Markovic, I.; Hasibether, F.; Sukesh, J.; Nenad, S.: Process-oriented Semantic Business Modeling. In (Hansen, R. H.; Karagiannis, D.; Fill, H.-G. Eds.): Business Services: Konzepte, Technologien, Anwendungen. Wirtschaftsinformatik. Wien, 25.02- 27.02. Österreichische Computer Gesellschaft, 2009; pp. 683–694.

[Mc76] McCabe, T.: A Complexity Measure. IEEE Trans. Softw. Eng. 2 (4) 1976; pp.308–320.

[Me06] Mendling, J.: Testing Density as a Complexity Metric for EPCs. Vienna University of Economics and Business Administration (Technical Report, JM-2006-11-15), 2006.

[Me08] Mendling, J.: Metrics for process models. Springer, Berlin, 2008.

[MN06] Mendling, J.; Nüttgens, M.: EPC markup language (EPML). In Information Systems and E-Business Management (4), 2006; pp. 245–263.

[MRv10] Mendling, J.; Reijers, H. A.; van der Aalst, W. M.: Seven process modelling guidelines (7PMG). In Information and Software Technology 52 (2), 2010; pp. 127–136.

[MB07] Meyers, T. M.; Binkley, D.: An empirical study of slice-based cohesion and coupling metrics. In ACM Trans. Softw. Eng. Methodol. 17 (1), 2007; pp. 1–27.

[Mo05] Moody, D. L.: Theoretical and practical issues in evaluating the quality of conceptual models. In Data & Knowledge Engineering 55 (3), 2005; pp. 243–276.

[OTE06] Orme, A.; Tao, H.; Etzkorn, L.: Coupling metrics for ontology-based system. In IEEE Software 23 (2), 2006; pp. 102–108.

[Pe07] Perepletchikov, M.; Ryan, C.; Frampton, K.; Tari, Z.: Coupling Metrics for Predicting Maintainability in Service-Oriented Designs. In : 18th Australian Software Engineering Conf. ASWEC. Melbourne, Australia, 10.04- 13.04. IEEE, 2007; pp. 329–430.

[PSW08] Polyvyanyy, A.; Smirnov, S.; Weske, M.: Process Model Abstraction. In : Enterprise Distributed Object Computing Conf. EDOC. Munich, 15.09-19.09. IEEE, 2008.

[PM06] Poshyvanyk, D.; Marcus, A.: The Conceptual Coupling Metrics for Object-Oriented Systems. In : 22nd IEEE Int. Conf. on Software Maintenance. ICSM. Philadelphia, USA, 24.09- 27.09. IEEE, 2006; pp. 469–478.

[QLT06] Qian, K.; Liu, J.; Tsui, F.: Decoupling Metrics for Services Composition. In (Lee, R.; Ishii, N. Eds.): Proc. o. t. 5th Annual IEEE/ACIS Int. Conf. on Computer and Information Science. ICIS-COMSAR. Honolulu, Hawaii, 10.07- 12.07. IEEE; AIS, 2006; pp. 44–47.

[QT09] Quynh, P.; Thang, H.: Dynamic coupling metrics for service-oriented software. In Int. J. on Computer Science Engineering (3), 2009; pp. 282–287.

[RL92] Rajaraman, C.; Lyu, M.: Reliability and maintainability related software coupling metrics in C++ programs. In : 3rd int. symposium on software reliability engineering. ISSRE. Research Triangle Park, USA, 07.10- 10.10. IEEE, 1992; pp. 303–311.

[Re11] Recker, J.; Rosemann, M.; Green, P.; Indulska, M.: Do ontological deficiencies in modeling grammars matter? In MISQ 35 (1), 2011; pp. 57–79.

[RV04] Reijers, H. A.; Vanderfeesten, I. T.: Cohesion and coupling metrics for workflow process design. LNCS 3080, 2004; pp. 290–305.

[RH97] Rosenberg, L.; Hyatt, L.: Software quality metrics for object-oriented environments. In Crosstalk Jounal 10 (4), 1997.

[SS05] Sandhu, P. S.; Singh, H.: A Critical Suggestive Evaluation of CK Metric. In: Pacific Asia Conf. on Information Systems. PACIS. Bangkok, Thailand, 07.- 10.07.2007. AIS, 2005.

[STA05] Scheer, A.-W.; Thomas, O.; Adam, O.: Process Modeling Using Event-Driven Process Chains. In (Dumas, M.; van der Aalst, W. M.; Hofstede, A. T. Eds.): Process-aware information systems. John Wiley and Sons, 2005; pp. 119–146.

[Si08] Silver, B., 2008: Ten tips for effective process modeling. BPMInstitute.org. Available online at http://www.bpminstitute.org/resources/articles/bpms-watch-ten-tips-effective-process-modeling, checked on 9/10/2013.

[SJ09] Sunju, O.; Joongho, A.: Ontology Module Metrics. In : 2009 IEEE Int. Conf. on e-Business Engineering. ICEBE. Macau, China, 21.10-23.10. IEEE, 2009; pp. 11–18.

[Új10] Újházi, B.; Ferenc, R.; Poshyvanyk, D.; Gyimóthy, T.: New Conceptual Coupling and Cohesion Metrics for Object-Oriented Systems. In : 10th Working Conf. on Source Code Analysis and Manipulation. SCAM. Timisoara, Rumania, 12.- 13.09. IEEE, 2010.

[va10] van der Aalst, W. M.; Nakatumba, J.; Rozinat, A.; Russell, N.: Business Process Simulation. In (Vom Brocke, J.; Rosemann, M. Eds.): Handbook on Business Process Management 1. Springer, Berlin, Heidelberg, 2010; pp. 313–318.

[va05] van Dongen, B.; Medeiros, A. de; Verbeek, H.; Weijters, A.; van der Aalst, W. M.: The ProM Framework. LNCS 3536, 2005; pp. 444–454.

[vOS05] van Hee, K. M.; Oanea, O.; Sidorova, N.: Colored Petri Nets to Verify Extended Event-Driven Process Chains. LNCS 3760, 2005; pp. 183–201.

[Va07] Vanderfeesten, I. T.; Cardoso, J.; Mendling, J.; Reijers, H. A.; van der Aalst, W. M.: Quality Metrics for Business Process Models. In (Fischer, L. Ed.): BPM and workflow handbook. Future Strategies, Lighthouse Point, USA, 2007; pp. 179–191.

[VCR07] Vanderfeesten, I. T.; Cardoso, J.; Reijers, H. A.: A weighted coupling metric for business process models. CEUR Workshop Proceedings 247, 2007; pp. 41–44.

[Va08] Vanderfeesten, I. T.; Reijers, H. A.; Mendling, J.; van der Aalst, W. M.; Cardoso, J.: On a quest for good process models. LNCS 5047, 2008; pp. 480–494.

[VRv08] Vanderfeesten, I.; Reijers, H.; Aalst, W. M.: Evaluating workflow process designs using cohesion and coupling metrics. In Computers in industry 59 (5), 2008; pp. 420–437.

[Vo09] Vom Brocke, J.; Simons, A.; Niehaves, B.; Riemer, K.; Plattfaut, R.; Cleven, A.: On the Importance of Rigour in Documenting the Literature Search Process. In (Newell, S.; Whitley, E. A.; Pouloudi, N.; Wareham, J.; Mathiassen, L. Eds.): 17th European Conf. on Information Systems, ECIS 2009. ECIS. Verony, Italy, 2009; pp. 2206–2217.

[WK08] Wahler, K.; Küster, J. M.: Predicting Coupling of Object-Centric Business Process Implementations. LNCS 5240, 2008; pp. 148–163.

[We97] Weber, R.: Ontological foundations of information systems. Coopers & Lybrand, Melbourne, 1997.