2013

Coupling Metrics for EPC Models

Daniel Braunnagel
University of Regensburg, Department of Management Information Systems, Regensburg, Germany,
daniel.braunnagel@wiwi.uni-regensburg.de

Florian Johannsen
University of Regensburg, Department of Management Information Systems, Regensburg, Germany,
Florian.Johannsen@wiwi.uni-regensburg.de

Coupling Metrics for EPC Models

Daniel Braunnagel and Florian Johannsen

University of Regensburg, Department of Management Information Systems, Regensburg, Germany
{daniel.braunnagel,florian.johannsen}@wiwi.uni-regensburg.de

Abstract. Process modeling is a decisive task for modern enterprises. The effectiveness of IS development largely depends on the quality of conceptual models and their understandability. However, process model quality is still a fuzzy concept and not fully understood yet. Recently, coupling became a concept for assessing model quality, but still there is a lack of research in transferring "coupling" to business process modeling. The field of software engineering has shown the importance of measuring coupling as a means for judging the quality of a design. Therefore, this paper collects a range of coupling metrics from the field of software engineering and transfers them to event driven process chains (EPC). Further the metrics are applied to different process models and implications for the process model quality are discussed.

Keywords: Coupling, event driven process chain, process model quality

1 Introduction

Business process modeling is a decisive task for modern enterprises (see e.g. [1-2]). Business process modeling captures employees' process knowledge so that it can be used for entrepreneurial initiatives. Process models support decisions on IT-investments, the development of information systems and the improvement of processes (see [1]). Moody [3] states that the efficiency and effectiveness of IS development largely depends on the quality of conceptual models that guide IS implementation though their evaluation is more an art than a systematic procedure.

The creation of process models is as a highly subjective process [4]. Usually different persons, such as IT-employees or business analysts, are involved in the design of process models while a generally accepted approach for creating a process model is missing [5]. In addition users demand different levels of abstraction. Whereas a software engineer is interested in details concerning the control flow structure of a process to derive requirements on information systems (see e.g. [6-7]), managers usually prefer more abstract descriptions enabling strategic decision making [8-9]. For utilizing the benefits of process models, e.g. for software development, they need to be easy to understand and maintain [10].

However, the quality of business process models remains a fuzzy topic. According to Mendling et al. [2] quality frameworks, metrics, empirical surveys as well as prag-

11th International Conference on Wirtschaftsinformatik,
27th February – 01st March 2013, Leipzig, Germany

matic guidelines were introduced in recent years, dealing with quality aspects of process modeling [2]. Commonly accepted definitions of the term “process model quality” as well as standardized criteria for evaluating process models still are missing.

Frameworks, such as the guidelines of modeling (see e.g. [4], [11]) (GoM) or the SEQUAL model (see [12-13]) deliver criteria, for example “construction adequacy” (see [4]), which can be used for assessing the quality of conceptual models. However an evaluation of conceptual models based on these criteria is strongly affected by the subjectivity of the user [3]. This is because a conceptual model can only be evaluated against user’s expectations and is not to be considered as a “finished product” that can be judged on the basis of a specification [3]. In addition quality frameworks have not been widely accepted in practice and a standard has not yet emerged [3].

As Mendling et al. [2] state, manifold empirical studies on the maturation of business process modeling languages can be found (see e.g. [14-16]) [2]. Several authors (see e.g. [17-21]) introduce criteria that can be used for evaluating modeling languages. However, the object of interest in these studies is the modeling language used, not the process model itself (see [2]).

Pragmatic guidelines that can be found in literature (see e.g. [22-23]) are often too generic (e.g. “keep it simple”) [22] to support a practitioner in a modeling project (see also [2]).

In recent years, literature has focused on the development of metrics enabling an objective evaluation of process models (see e.g. [24-30]). Vanderfeesten et al. [31] assign these metrics to certain categories of process model quality. While manifold metrics for judging a process model regarding complexity or size do exist, coupling metrics for process models are still underrepresented. While coupling of modules is a well-established quality characteristic for information systems (see [32]), research has only begun to transfer this concept to business process modeling thus enabling a new perspective on process model quality. The aim of this paper is to expedite this research by introducing coupling metrics that originate in software engineering specifying them for business process modeling. Afterwards, the metrics that got transferred are applied using an example. The suggestions in this paper focus on event driven process chains (EPCs), since the interpretation of the coupling metrics will be different for different modeling languages varying in language expressiveness (see [33]).

The structure of the paper is as follows. In section 2 the basics are explained. These comprise the event driven process chains and coupling. Section 3 presents the coupling metrics found and describes how they were transferred to the EPC. Section 4 discusses the processes used as examples and the results from applying the metrics. Finally, section 5 summarizes the paper, discusses implications of the metrics for the field of process model quality and shows further work remaining.

2 Basics

2.1 Event Driven Process Chain

The EPC was developed at the University of Saarland in cooperation with SAP AG. The EPC is known for being used as the modeling concept supporting SAP R/3 and for its use as part of the modeling framework ARIS. [34]

The event driven process chain can be defined as a graph, consisting of nodes and directional arcs (see [35-38]). The nodes may be specified as the union set of the set of events, functions, connectors, process interfaces and resources. The set of connectors is the union of the sets of and-connectors, or-connectors and xor-connectors. The set of resources is the union of several sets, with information elements being one of them. Each node in the above set is connected to at least one arc, with each arc being connected with precisely two nodes. No arc connects two functions or two events, they alternate in a path with an arbitrary number of connectors allowed in-between. Resources are connected exclusively to functions, process interfaces exclusively to events. Functions need to be connected with at least two nodes situated among the connectors or events. An EPC model is considered a graph according to the above definition. The control flow is considered the path connecting process interfaces, events, functions and connectors. A model has a beginning, consisting of events without predecessor or process interfaces without a preceding event, and an end, being events without succeeding node or a process interface without succeeding events. The control flow, however, may be continued over multiple models in case of process interfaces referencing each other. [35-38]

2.2 Coupling

The term "coupling" is most generally defined as "being connected for consideration together" [39]. Closer to the context at hand, the field of software engineering presents different more specific interpretations of coupling.

The first interpretation is based on the ontology of Bunge-Wand-Weber [40-41]. Accordingly, two things are coupled if they interact at some moment in time. This interpretation is employed by e.g. the RFC metric. This metric counts the number of methods "that can be invoked in response to a message to an object of the class" [42]. In other words, two objects of an object oriented design are considered as being "coupled" whenever one object calls a method of another object. The metric measures the degree of coupling by counting the number of methods that can be invoked.

A second interpretation focuses graph-theory. Thereby, the graph is analyzed regarding the way its elements are connected. For example, McCabe [43] builds the control flow graph of programs to calculate the cyclomatic complexity. Counting the nodes, arcs and exit nodes, the metric calculates the number of independent circuits in the control flow. The notion of coupling therefore refers to the paths through the programs code. With reference to the EPC, the weighted coupling metric [29] and the cross connectivity metric [30] were presented in this field.

A third interpretation references information theory. This interpretation aims to quantify aspects of coupling using the information content [44]. An implementation of this interpretation is presented by Halstead [45] calculating the "program length". Their program length is sensitive towards the reuse of statements and therefore considers the coupling between code modules by their reuse of code statements.

3 Coupling in the Context of EPC

To cover a wide range of different coupling interpretations, literature presenting existing metrics was searched for. The well-known literature databases Google Scholar, Computer.org (IEEE Computer Society), AISeL and Emerald Insight, that offer a wide range of different electronic sources were queried using the term pair '"coupling metrics" "business process model"' and "coupling metrics" itself. 47 results were considered as relevant and downloaded, consisting of 33 conference papers and nine journal papers that passed a peer review process. In addition, four technical reports and one book were found.

The metrics covered in these sources and their transferability to the EPC is shown in Table 1.

Table 1. Coupling Metrics

Source	Metric	Transfer	Not transferred because…
[46]	Coupling of a modular system, IntramoduleCoupling	Yes	
[47]	Coupling of a module, Intramdoule coupling of a module	Yes	
[48]	PIM	No	… requires dynamic language features (i.e. polymorphism, reflection…).
[49]	PPEP, EMC	No	… requires failure rates of components.
[42]	Depth of inheritance, No. of children	No	.. requires inheritance.
[42]	RFC, CBO	Yes	
[50]	Static / Dynamic Coupling	No	… requires inheritance.
[51]	Direct coupling,indirect coupling, total coupling	Yes	
[52]	Procedure complexity	No	… already transferred [27, 53].
[54]	Object level coupling	No	… requires inheritance / locality of data.
[55]	Interface coupling	No	… equivalent to [56].
[57]	Conceptual coupling of Services	No	… requires locality of data.
[43]	Cyclomatic complexity	No	… already transferred [53].
[58]	Conceptual coupling	Yes	
[59]	CBS … DCSS	No	… is an implementation of [42] and [43].
[60]	ASSD … ASPD	No	… requires statefullness.
[61]	CIC … AMC	No	… requires inheritance.
[56]	Process Coupling	Yes	
[29]	CP	No	… already specified for eEPC.
[30]	CC	No	… already specified for eEPC.

The procedure for transferring the metrics to the EPC can be described as follows. In a first step, the concepts behind the variables of each formula are identified. The description of each concept is then used to identify equivalent concepts within EPC models. Finally, the found concepts are quantified and used to reformulate the original metrics (see Figure 1).

Fig. 1. Transfer procedure

However, in some cases metrics could not be transferred. This was the case whenever the metric comprised constructs for which no equivalent could be found in the EPC. For example, some metrics refer to the inheritance hierarchy of class objects in object oriented programming. An equivalent mechanism for inheritance among process models was not discovered. Further, since modeling takes place on the type level, metrics referring to runtime information or states could not be transferred either.

The metrics that could be transferred using this procedure are discussed in the following.

In [46], Allen et al. present two related metrics called “Coupling of a modular system” and “IntramoduleCoupling”. The motivation of these metrics is the limited capacity of the human short time memory. When the amount of information in a model breaks this limit, a user will not be able to fully realize the model, which will lead to problems in understanding the model. The metrics therefore calculate the information content of a model regarding different aspects of its graph structure. The metric “coupling of a modular system” calculates the excess entropy of the graph structure in modules. The “IntramoduleCoupling” quantifies the excess entropy of the graph structure connecting modules. Transferred to the EPC, the IntramoduleCoupling measures the repetitiveness of the patterns connecting models via process interfaces or a model hierarchy. The coupling of a modular system measures the repetitiveness of patterns in the control flow of separate models.

Allen et al. present a second pair of metrics in [47]. Adapting the prior metrics, the “coupling of a module” calculates the information content in the graph structure in each module. The “Intramodule coupling of a module” calculates the information content of the graph structure connecting the graphs. Transferred to the EPC, the coupling of a module quantifies the amount of information referring to the patterns of arcs in a model a user needs to assess in order to understand the model. The intramodule coupling, on the other hand, regards the information in the connections between models.

Chidamber/ Kemerer present two coupling metrics in [42]. The first metric, CBO, counts the number of classes one class is associated with by calling its methods or variables. The second metric RFC counts the methods in one class and all the methods in other classes that can be called from within. Transferred to the EPC, the RFC counts the functions and interfaces or hierarchies. CBO counts the number of models one model is connected to.

In [51], Gui presents three related metrics, namely the direct coupling metric, the indirect coupling metric and the total coupling metric. The direct coupling metric is calculated for two classes, it is the relation of methods and variables in the second class called by the first class divided by the total number of methods and variables in the second class. As for the second metric, indirect coupling extends the prior metric for classes without direct connection. For any pair of classes for which a connecting path via called methods or resources can be found, their indirect coupling is calculated by multiplying the direct coupling values for each pair of classes on the path. The third metric is calculated by dividing the sum of indirect coupling values of all pairs of classes by the number of potential class pairs. Transferred to the EPC, two process models are considered coupled if one model contains a process interface or hierarchical function referencing the other. Accordingly, the direct coupling metric, when transferred, divides the number of references from a second model by the number of functions and process interfaces contained in one model. The second and third metric are used like originally described.

Poshyvanyk/ Marcus [58] describe a metric using the information retrieval technique latent semantic indexing to discover a semantic structure among the textual content of source code. The metric is called conceptual coupling metric [58]. They assume that similar concepts are expressed with similar terms. Therefore calculating the co-occurrence of terms indicates how the strength of the relation between concepts. The LSI transformation of a term-document matrix containing variable text from classes as terms and from the class structure as documents presents such co-occurrence of terms. Using the strongest indicators for concepts from this transformation, a new term-document matrix can be built which is then used to calculate the similarity between classes and groups of classes. Transferred to the EPC, the term-document matrix is built using node labels as terms and models as documents. The metric then calculates the conceptual similarity between models and groups of models chosen by the user. These groups can, e.g., belong to one or more processes. The adapted metric therefore calculates the co-occurrence of terms between groups of process models. Assuming the co-occurrence is an adequate measure for a conceptual structure, this indicates the conceptual similarity of process models.

Another approach towards coupling is presented by Reijers/ Vanderfeesten [56]. The metric is defined for a so-called information element structure, which is a graph structure with nodes representing information elements and arcs representing operations. An activity can be described as a partition in the said structure. Operations are considered coupled if they are connected to a common information element. Activities are considered coupled, if they contain coupled operations. Transferred to the EPC, functions take the place of activities, information elements are used as such. Functions are considered coupled whenever they are connected to common information elements.

4 Illustrating Example

The metrics described above are demonstrated using three groups of process models. Each of these groups represents the same situation, but whereas group 1 contains models without syntactical errors, group 2 and group 3 have an increasing number of errors. Figure 2 and Figure 3 show an excerpt from the same process model in group 1 and group 2. The second model contains redundant events after the first function and misses the decision and the corresponding event after “Request Schufa-Report”. Further errors (not pictured) are e.g. missing arcs after splits.

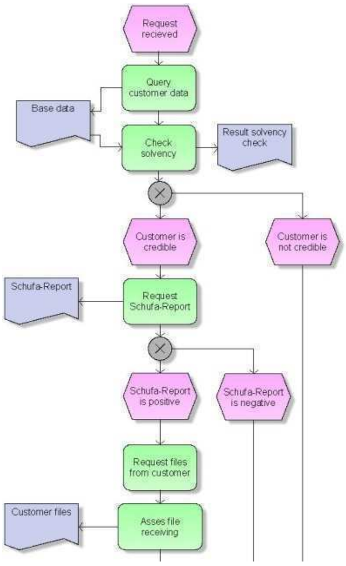

Fig. 2. Validate solvency, group 1

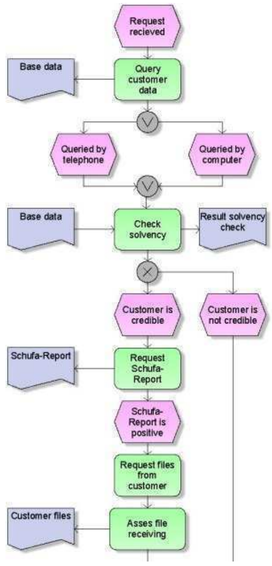

Fig. 3. Validate solvency, group 2

The first two groups of process models comprise three sub-models, the last group four sub-models. The models are quantified in Table 2.

Table 2. Example models

	Group 1	Group 2	Group 3
Credit Application			
Functions	2	2	4
Events	4	4	7
Information elements	0	0	0
Validate solvency			
Functions	13	13	9
Events	11	14	10
Information elements	11	13	7
Final decision			
Functions	9	7	5
Events	8	6	5
Information elements	2	2	4
Proceed credit application			
Functions			4
Events			6
Information elements			8

4.1 Allen et al.

The metrics of Allen et al. (see [46-47]) calculated for the models at hand are shown in Table 3.

Table 3. Metrics of Allen et al.

Metric\Group	Group 1	Group 2	Group 3
Coupling of a Module (Credit application)	33.645	33.612	50.220
IntramoduleCoupling(Final decision)	225.474	175.251	139.152
IntramoduleCoupling(Validate solvency)	458.235	587.723	359.202
IntramoduleCoupling(Credit application)	42.292	42.292	93.603
IntramoduleCoupling(Proceed credit application)			200.088
Coupling of a modular system	11.404	11.939	19.241
Intramodulecoupling of a modular system	296.953	345.767	270.727

The metrics coupling of a module hardly differ between the "group 1" and "group 2" model. The values of the "group 3" models differ due to the higher number of connections between models since this decomposition uses more models.

Regarding the models of "Final decision" and "Validate solvency" the metric suggests that "group 3" and "group 2" models are easier to assess by the model user. The values do, however, mostly result from modeling errors omitting necessary arcs in the case of "group 2" or from a smaller sub-model due to a higher degree of decomposition in the case of "group 3".

The model "Credit application" is identical in the groups of "group 1" and "group 2". In the case of "group 3", the model encompasses more nodes and arcs and therefore has a higher metric value.

The coupling of a modular system hardly differs between "group 1" and "group 2" since both their decomposition comprises the same amount of models. The last group presents a higher value due to the higher amount of sub-models.

The intramodulecoupling mostly represents the number of nodes and arcs in the corresponding sub-models.

4.2 Chidamber/ Kemerer

The metrics of Chidamber/Kemerer (see [42]) calculated for the models are shown in Table 4.

Table 4. Metrics of Chidamber/ Kemerer

Group	Model\Metric	RFC	CBO
Group 1	Credit application	2	2
	Validate solvency	13	
	Final decision	9	
Group 2	Credit application	2	2
	Validate solvency	13	
	Final decision	7	
Group 3	Credit application	4	3
	Validate solvency	9	
	Proceed credit application	4	
	Final decision	5	

The RFC and CBO mostly count objects. Accordingly, their values depend on the number of functions and process interfaces or linked sub-models respectively.

4.3 Gui/ Scott

The Gui/Scott metrics (see [51]) are calculated for the example at hand in Table 5. Since the model "Credit application" is the only model referencing sub-models, it is also the only model that can form a pair of coupled models with the remaining models.

Table 5. Metrics of Gui/ Scott

Metric\Group	1: Credit application	2: Credit application	3: Credit application
Direct Coupling Metric			
Credit application	0	0	0
Validate solvency	0.07692	0.07692	0.11111
Proceed credit application			0.25
Final decision	0.11111	0.14285	0.2
Indirect coupling Metric			
Credit application	0	0	0
Validate solvency	0.07692	0.07692	0.11111
Proceed credit application			0.25
Final decision	0.111111	0.142857	0.2
Total coupling metric	0.03133	0.03663	0.04675

Since all the models in all decompositions are linked to one higher decomposition level, they all have one incoming connection and no other links. Therefore the determining variable for this metric is the number of functions in each model. Accordingly, the values mostly represent the number of activities. A higher number of activities lead to a lower metric value.

The indirect coupling metrics do not differ from the direct ones, since in this case there are no paths of more than two models.

The sum of indirect coupling values is put in relation to the maximally possible number of model pairs for the total coupling metric.

4.4 Poshyvanyk

The metric of Poshyvanyk et al. (see [58]) was used with the exemplary models, though with one limitation. Since the scenario was modeled in only one group with three alternatives, solely the conceptual similarity of models instead of the similarity of model groups were calculated (Table 6 - Table 8).

Table 6. Conceptual similarity of models, Group 1

Model\Model	Final decision	Validate solvency	Credit application
Final decision	1	0.88297	0.73799
Validate solvency	0.88297	1	0.82660
Credit application	0.73799	0.82660	1

Table 7. Conceptual similarity of models, Group 2

Model\Model	Validate solvency	Final decision	Credit application
Validate solvency	1	0.85847942	0.82395738
Final decision	0.858479424	1	0.71707854
Credit application	0.82395738	0.71707854	1

Table 8. Conceptual similarity of models, Group 3

Model\Model	Credit application	Final decision	Credit application	Validate solvency
Credit application	1	0.86429	0.86587	0.90647
Final decision	0.86429	1	0.84521	0.84859
Credit application	0.86587	0.84521	1	0.83160
Validate solvency	0.90647	0.84859	0.83160	1

As can be seen, for all groups the conceptual similarity of models declines along the rising distance in the control flow. In addition to that, the "group 1" and "group 2" models hardly differ. Indeed, since these metrics regard the co-occurrence of labels, their values do not reflect syntactical violations. The last decomposition is hardly comparable since the number of models differs.

4.5 Reijers/ Vanderfeesten

The coupling metric of Reijers/ Vanderfeesten (see [56]) is used in combination with the above process models to calculate the values in Table 9.

Table 9. Metrics of Reijers/ Vanderfeesten

Group 1	Validate solvency	0.06410
Group 2	Validate solvency	0.06410
Group 3	Proceed credit application	0.33333
	Validate solvency	0.02777

The only models containing coupled activities are "Validate solvency" in all groups, and "Proceed credit application" in the group 3 decomposition. There is no difference in the coupling among the first two decompositions since the models do not differ regarding their coupled functions. The last decomposition's models do differ since their functions are split over two models.

4.6 Discussion

The metrics transferred before were applied to three example processes. It was shown that coupling metrics from the field of software engineering can in fact be transferred to EPC models. Hence the metrics presented in this work extend the existing set of metrics (see [31]).

However, the calculation of these metrics is laborious. E.g. the metrics of Allen et al. require a separate incidence matrix for each node in a number of models. The metric of Poshyvanyk requires a singular value decomposition, which usually can only be calculated by using specialized software. Therefore, tool support is necessary for calculating these metrics. These tools need to be developed in future work.

Finally, as remarked before, the metric values hardly differ among the models of group one and two. Different reasons can be found for that. For example, the metrics of Allen et al. try to quantify the arbitrariness of patterns among the nodes of process models. However, they do not incorporate syntactical limitations, e.g. that nodes cannot be (directly) connected via two different arcs. Further, the models of group one and group two differ in syntactical errors but they are mostly equivalent structurally. The syntax, however, is ignored by many metrics, e.g. the conceptual coupling, CBO and RFC. On the other hand, group 3 differs regarding its decomposition structure which influences the metric values. In conclusion it may be said that the differences are too subtle, and the models are too small therefore resulting in little differences of the values, too. Regarding the perspective of coupling, the differences between the models are also small again resulting in small differences among the metric values. The lack of difference in the models also matches with the results of a laboratory experiment conducted with 66 students at a German university. They were asked to rate the understandability of the models on a 7-point Likert scale. The results indicated no significant difference in the understandability for all three model groups. However, the relation between understandability and coupling needs further investigation.

5 Outlook and Conclusion

The paper at hand deals with coupling metrics from software engineering. In that field coupling is a well-established concept for judging the quality of information systems. In recent years work has been done transferring the idea of coupling to business process modeling. The main motivation is to assess the fuzziness of the process model quality discipline by the quality dimension "coupling". Though only a few metrics were introduced for judging a process model regarding coupling (see e.g. [29-31]), there are different perspectives on coupling in software engineering (see section 2.2). These led to a considerable set of different metrics for coupling in software engineering.

When these metrics are transferred to business process modeling they cannot only be used to evaluate business process models but also to infer suggestions for a good process modeling style regarding coupling. The paper at hand thus contributes to this field by transferring corresponding coupling metrics to business process modeling and shows their applicability on an example.

The metrics of Allen et al. generally suggest using repetitive patterns in the structure of nodes and arcs since they are more comprehensible. The actual implementation of the metric, however, merely suggests limiting the size of each model, though no concrete limit is given. The procedures used for the CBO and RFC metrics are easier. CBO counts the number of models one model is connected with by its control flow. Consequently, the metrics suggests limiting the number of connections between models. The RFC further incorporates the size of a model. Therefore, the metric suggests creating models using a low number of functions and connections to other models. The conceptual coupling of Poshyvanyk suggests to isolate concepts in separate models and to use a distinctive vocabulary. The last metric, Process coupling of Reijers/Vanderfeesten turns out to a good value with only few functions being connected to the same information element. However, one should bear in mind that the modeler's freedom of including or omitting elements should not lead to omitting e.g. information elements solely to realize a good metric value, even though this information would be necessary for the model users.

Still, there is clearly further work to be done. The transfer procedure was influenced by subjectivity in the interpretation of equivalent constructs. This also led to alternative interpretations that need to be discussed. Further, though a range of metrics was presented, the empirical evaluation is still missing. Therefore the practical utility of these metrics remains unanswered. Further, the laborious calculation procedures should be implemented in tool support for a practical use. Furthermore, additional metrics should be searched for and the existing ones should be transferred to more modeling languages (e.g. BPMN).

References

1. Becker, J., Thome, I., Weiß, B., Winkelmann, A.: Constructing a Semantic Business Process Modelling Language for the Banking Sector. Enterprise Modelling and Information Systems Architectures 5, 4-25 (2010)
2. Mendling, J., Reijers, H., van der Aalst, W.: Seven process modeling guidelines. Information and Software Technology 52, 127-136 (2010)
3. Moody, D.L.: Theoretical and practical issues in evaluating the quality of conceptual models: current state and future directions. Data & Knowledge Eng. 55, 243-276 (2005)
4. Schütte, R., Rotthowe, T.: The Guidelines of Modeling – An Approach to Enhance the Quality in Information Models. LNCS, Vol. 1507, pp. 240-254. Springer (1998)
5. Kobler, M.: Qualität von Prozessmodellen - Kennzahlen zur analytischen Qualitätssicherung bei der Prozessmodellierung. Logos, Berlin (2010)
6. Mili, H., Tremblay, G., Jaoude, G.B., Lefebvre, É., Elabed, L., Boussaidi, G.E.: Business process modeling languages: Sorting through the alphabet soup. ACM Computing Surveys 43, 1-54 (2010)
7. Aguilar-Savén, R.S.: Business process modelling: Review and framework. International Journal of Production Economics 90, 129-149 (2004)
8. Bobrik, R., Reichert, M., Bauer, T.: View-Based Process Visualization. LNCS, Vol. 4714, pp. 88-95. Springer (2007)
9. Polyvyanyy, A., Smirnov, S., Weske, M.: Process Model Abstraction: A Slider Approach. In: Enterprise Distributed Object Computing Conference, pp. 325-331. IEEE, Munich (2008)
10. Gruhn, V., Laue, R.: Adopting the Cognitive Complexity Measure for Business Process Models. In: Yao, Y., Shi, Z., Wang, Y., Kinsner, W. (eds.): International Conference on Cognitive Informatics, pp. 236 -241. IEEE (2006)
11. Becker, J., Rosemann, M., von Uthmann, C.: Guidelines of Business Process Modeling. LNCS 1806, pp. 241-262. Springer (2000)
12. Krogstie, J., Sindre, G., Jørgensen, H.: Process models representing knowledge for action. European Journal of Information Systems 15, 91-102 (2006)

13. Lindland, O.I., Sindre, G., Sølvberg, A.: Understanding Quality in Conceptual Modeling. IEEE Software 11, 42-49 (1994)
14. Recker, J., Indulska, M., Rosemann, M., Green, P.: Do Process Modelling Techniques Get Better? A Comparative Ontological Analysis of BPMN. In: 16th Australasian Conference on Information Systems, Sydney (2005)
15. Recker, J., Rosemann, M., Green, P., Indulska, M.: Do Ontological Deficiencies In Modeling Grammars Matter?. MIS Quarterly 35, 57-79 (2011)
16. Agarwal, S., Teas, R.K.: Cross-national applicability of a perceived quality model. Journal of Product & Brand Management 11, 213-236 (2002)
17. Remme, M.: Konstruktion von Geschäftsprozessen - Ein modellgestützter Ansatz durch Montage generischer Prozesspartikel. Gabler, Wiesbaden (1997)
18. Süttenbach, R., Ebert, J.: A Booch Metamodel. Universität Koblenz-Landau (1997)
19. Frank, U., van Laak, B.: Anforderungen an Sprachen zur Modellierung von Geschäfts-prozessen. Universität Koblenz-Landau (2003)
20. Prasse, M.: Evaluation of object-oriented modelling languages: A comparison between OML and UML. In: Schader, M., Korthaus, A. (eds.): The Unified Modeling Language – Technical Aspects and Applications. Physica, Heidelberg (1998)
21. Zelewski, S.: Eignung von Petrinetzen für die Modellierung komplexer Realsysteme - Beurteilungskriterien. Wirtschaftsinformatik 38, 369–381 (1996)
22. Bridges, G.: Top Ten Tips and Tricks For Business Process Modeling (2009)
23. Silver, B.: Ten tips for effective process modeling. BPMInstitute.org (2008)
24. Cardoso, J.: Business Process Quality Metrics: Log-Based Complexity of Workflow Patterns. LNCS, Vol. 4803, pp. 427-434. Springer (2007)
25. 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)
26. Latva-Koivisto, A.: Finding a complexity measure for business process models. Helsinki University of Technology (2001)
27. Mendling, J.: Testing Density as a Complexity Metric for EPCs. Vienna University of Economics and Business Administration (2006)
28. Mendling, J., Reijers, H.A., Cardoso, J.: What Makes Process Models Understandable? LNCS, Vol. 4714, pp. 48-63. Springer (2007)
29. Vanderfeesten, I., Cardoso, J., Reijers, H.A.: A weighted coupling metric for business process models. In: Eder, J., Tomassen, S.L., Opdahl, A., Sindre, G. (eds.): CAiSE 2007, pp. 41-44, Trondheim, Norway (2007)
30. Vanderfeesten, I., Reijers, H.A., Mendling, J., van der Aalst, W.M.P., Cardoso, J.: On a Quest for Good Process Models: The Cross-Connectivity Metric. In: Bellahsène, Z., Lénoard, M. (eds.): CAiSE 2008, pp. 480-494, Montpeiller, France (2008)
31. Vanderfeesten, I.T.P., Cardoso, J., Mendling, J., Reijers, H.A., van der Aalst, W.M.P.: Quality Metrics for Business Process Models. In: Fischer, L. (ed.): BPM and workflow handbook. Future Strategies, Lighthouse Point, USA (2007)
32. Weber, R.: Ontological foundations of information systems. Coopers & Lybrand, Melbourne (1997)
33. Recker, J., Rosemann, M., Green, P., Indulska, M.: Do ontological deficiencies in modeling grammars matter?. MIS Quarterly 35 (1), 57-79 (2011)
34. Scheer, A.-W.: ARIS - business process modeling. Springer, Berlin et al. (2000)
35. Mendling, J.: Detection and predicition of errors in EPC Business Process Models. Institute für Informationssysteme und neue Medien. WU Wien (2007)

36. Mendling, J.: Metrics for process models: Empirical foundations of verification, error prediction, and guidelines for correctness. Springer, Berlin (2008)
37. van Hee, K.M., Oanea, O., Sidorova, N.: Colored Petri Nets to Verify Extended Event-Driven Process Chains. In: Meersman, R., Tari, Z., Hacid, M.-S., Mylopoulos, J., Pernici, B., Babaoglu, Ö., Jacobsen, H.-A., Loyall, J.P., Kifer, M., Spaccapietra, S. (eds.): On the Move to Meaningful Internet Systems 2005: CoopIS, DOA, and ODBASE. LNCS, Vol. 3760, pp. 183–201. Springer, Berlin Heidelberg (2005)
38. Keller, G., Nüttgens, M., Scheer, A.-W.: Semantische Prozeßmodellierung auf Grundlage Ereignisgesteuerter Prozeßketten (EPK). In: Scheer, A.-W. (ed.): Veröffentlichungen des Instituts für Wirtschaftsinformatik, Vol. 89, Saarbrücken (1992)
39. Gove, P.B.: Webster's third international dictionary of the English language. Merriam-Webster, Springfield (1986)
40. Wand, Y., Weber, R.: An ontological model of an information systems. IEEE Transactions on Software Engineering 16, 1282–1292 (1990)
41. Wand, Y., Weber, R.: Mario Bunge's ontology as a formal foundation for information systems concepts. In: Weingartner, P., Dorn, G.W.J. (eds.): Studies on Mario Bunge's Treatise. Rodopi, Atlanta (1990)
42. Chidamber, S.R., Kemerer, C.F.: A Metrics Suite for Object Oriented Design. IEEE Transactions on Software Engineering 20, 476–493 (1994)
43. McCabe, T.J.: A Complexity Measure. IEEE Transactions on Software Engineering 2, 308–320 (1976)
44. Shannon, C.E.: A mathematical theory of communication. The bell system technical journal 27, 379–423 (1948)
45. Halstead, M.H.: Elements of software science. Elsevier, New York (1977)
46. Allen, E.B., Khoshgoftaar, T.M., Chen, Y.: Measuring coupling and cohesion: An information-theory approach. In: 6th International Software Metrics Symposium, pp. 119–127. IEEE Computer Society (1999)
47. Allen, E.B., Khoshgoftaar, T.M., Chen, Y.: Measuring Coupling and Cohesion of Software Modules: An Information-Theory Approach. In: 7th International Software Metrics Symposium. IEEE (2001)
48. Allier, S., Vaucher, S., Dufour, B., Sahraoui, H.: Deriving Coupling Metrics from Call Graphs. In: 10th IEEE Working Conference on Source Code Analysis and Manipulation, pp. 43–52. IEEE (2010)
49. Arshad, F., Khanna, G., Laguna, I., Bagchi, S.: Distributed Diagnosis of Failures in a Three Tier E-Commerce System. ECE Technical Reports. Purdue University (2007)
50. 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 Conference, pp. 87- 85. IEEE Computer Society (1998)
51. Gui, G., Scott, P.D.: Coupling and cohesion measures for evaluation of component reusability. In: Diehl, S., Gall, H., Hassan, A.E. (eds.): Proceedings of the 2006 International Workshop on Mining Software Repositories, pp. 18–21. ACM Press, New York (2006)
52. Henry, S., Kafura, D.: Software Structure Metrics Based on Information Flow. IEEE Transactions on Software Engineering 7, 510–518 (1981)
53. Cardoso, J.: Process control-flow complexity metric: An empirical validation. In: IEEE International Conference on Services Computing (SCC'06), pp. 167-173 (2006)
54. Hitz, M., Montazeri, B.: Measuring coupling and cohesion in object-oriented systems. In: Proceedings of 3rd International Symposium on Applied Corporate Computing (1995)

55. Wahler, K., Küster, J.M.: Predicting Coupling of Object-Centric Business Process Implementations. In: Dumas, M., Reichert, M., Shan, M.-C. (eds.): Business process management. LNCS, Vol. 5240, pp. 148–163. Springer, Berlin (2008)
56. Reijers, H.A., Vanderfeesten, I.T.P.: Cohesion and coupling metrics for workflow process design. In: Desel, J., Pernici, B., Weske, M. (eds.): Business process management. LNCS, Vol. 3080, pp. 290–305. Springer, Berlin (2004)
57. 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 International Conference on Services Computing, pp. 504–511. IEEE Computer Society (2011)
58. Poshyvanyk, D., Marcus, A.: The Conceptual Coupling Metrics for Object-Oriented Systems. In: 22nd IEEE International Conference on Software Maintenance, pp. 469–478. IEEE Computer Society (2006)
59. Quynh, P.T., Thang, H.Q.: Dynamic coupling metrics for service-oriented software. International Journal on Computer Science Engineering 3 (5), 282–287 (2009)
60. Qian, K., Liu, J., Tsui, F.: Decoupling Metrics for Services Composition. In: Lee, R., Ishii, N. (eds.): Proceedings of the 5th Annual IEEE/ACIS International Conference on Computer and Information Science (ICIS 2006) and 1st IEEE/ACIS International Workshop on Component-Based Software Engineering, Software Architecture and Reuse (COMSAR 2006), pp. 44–47. IEEE Computer Society (2006)
61. Rajaraman, C., Lyu, M.R.: Reliability and maintainability related software coupling metrics in C++ programs. In: 3rd international symposium on software reliability engineering, pp. 303–311. IEEE Computer Society (1992)