Process Measurement: Insights from Software Measurement on Measuring Process Complexity, Quality and Performance

Joachim Melcher and Detlef Seese
Institut AIFB, Universität Karlsruhe (TH)
76128 Karlsruhe, Germany
{melcher|seese}@aifb.uni-karlsruhe.de

August 11, 2008

Abstract

Motivated by software (complexity) metrics, several papers about process measurement have been published in recent years proposing metrics for process complexity, quality and/or performance. Starting with an overview about these publications, we identified some weak points (e. g., missing definitions of process complexity and quality as well as a lack of validation work).

In this article, we adopt more well-established concepts from the field of software measurement to process measurement: a prediction system measurement approach avoiding a concrete definition of process complexity, measurement and prediction systems and their validation, the goal question metric paradigm for selecting process metrics and different purposes of process metrics (understand, control and improve).

The paper closes with an assessment of existing work according to the adopted concepts from software measurement. Thereby, we identified some missing aspects which could be dealt with in future work.

1 Introduction

During the previous decades, the field of software measurement has created well-established theoretical concepts for measuring software and making predictions on software quality attributes (see, e. g., [9] for an overview). Motivated by this research, several papers about measuring business processes have been published in recent years. Nevertheless, this area is rather new and much work is still to be done.

An overview about the published literature (see Section 2) shows that many process metrics adapted from software metrics were suggested. Many of them are told to measure process complexity, quality and/or performance. At the same time, missing definitions of process complexity and quality as well as a lack of validation work (compared to the number of proposed metrics) can be noticed.

The contribution of this article is an adoption of more well-established concepts from the field of software measurement to process measurement: a prediction system measurement approach avoiding a concrete definition of process complexity, measurement and prediction systems and their validation, the goal question metric paradigm for selecting process metrics and different purposes of process metrics (understand, control and improve).

The remainder of this paper is organized as follows: In Section 2, we present related work on process measurement. The question how to define and/or measure process complexity, quality and performance is dealt with in Section 3. In Section 4, we explain measurement and prediction systems and their proper validation. The application of metrics is shown in Section 5. Afterwards, we assess the existing work according to the adopted concepts from software measurement in Section 6. The paper gives a conclusion and recommends possible future work (Section 7).

2 Related Work

In [16], Lee and Yoon introduce 15 complexity metrics for Petri nets. They distinguish between *structural* (e. g., number of places and transitions, cyclomatic number) and *dynamic* (including number of markings and tokens as well as degree of parallel firing) complexity metrics.

Morasca deals with measuring internal attributes of Petri nets for concurrent software specifications [23]. He identifies size, length, complexity and coupling as interesting attributes. For each of them, he defines a set of axiomatic properties which corresponding metrics have to fulfill. Afterwards, he suggests several metrics for these four attributes and validates them against the properties.

To our knowledge, Latva-Koivisto's paper [15] is the first publication dealing with measuring the complexity of business processes. He makes some interesting remarks on how to define complexity (see Subsection 3.1). Then, he introduces several metrics for structural complexity based on graph-theory.

Inspired by McCabe's cyclomatic number for control-flow graphs of software, Cardoso recommends the control-flow complexity metric (CFC) for business processes [5]. The metric is tested by Cardoso for correlation with received complexity [6]. In [7], he offers a specialization of the CFC metric for BPEL processes.

In [4], Cardoso discusses data-flow complexity metrics for web processes in BPEL. He differentiates between data, interface and interface integration complexity. Yet, only for interface complexity, he advices a concrete metric.

Gruhn and Laue suggest complexity metrics for business processes analogous to software complexity metrics [11]. In [10], they adapt the cognitive weights metric from software engineering to business processes.

Rolón *et al.* recommend several metrics for business processes modeled in BPMN [25]. Their metrics are an adaptation and extension of the Framework for the Modeling and Evaluation of Software Processes (FMESP).

In their survey paper [3], Cardoso *et al.* propose new metrics analogous to existing metrics for software (LOC, Halstead complexity metrics and information flow metric by Henry and Kafura). Additionally, they present already published metrics like CFC [5] and the metrics of [15].

In [24], Reijers and Vanderfeesten introduce a heuristic for the proper size of individual activities in processes (process granularity). Activities can consist of (several) basic operations. The operations of one activity should "belong" together (highly cohesive)—while different activities should be independent from each other (loosely coupled). For that purpose, they introduce a process cohesion and a process coupling metric and furthermore a couping/cohesion ratio.

Vanderfeesten *et al.* suggest a weighted coupling metric with different weights for the different connection types [27].

Analyzing the SAP Reference Model processes with an automatic verification tool, Mendling *et al.* detect faulty EPC processes [18]. In a second step, they try to find possible predictors (based on 15 metrics) for these errors using logistic regression. In [19], Mendling proposes a density metric and repeats the regression test. In his PhD thesis [20], Mendling gives 28 metrics for EPC processes (some of them taken from [18], but many are new). Once again, he uses logistic regression to identify possible predictors for faulty processes.

In [21], Mendling *et al.* present an experiment for identifying influencing personal (theoretical knowledge and practical experience) and structural factors on process model understandability.

In [28], Vanderfeesten *et al.* introduce the cross-connectivity metric. It measures the average strength of connection between all pairs of process nodes. They empirically evaluate the metric using data of [21].

Mendling and Strembeck present a second experiment for identifying influencing factors on process understandability [22]. This time, also content related factors (task labels) are analyzed.

In [13], Jansen-Vullers *et al.* suggest a framework for process performance measures with the dimensions time, cost, quality and flexibility. For each for these dimensions, they propose a set of metrics.

3 Process Measurement

In the literature, several publications exist which try to measure process complexity using complexity metrics (e. g., [5, 15]). Yet, one also finds articles dealing with process quality and quality metrics (e. g., [26]) as well as process performance (e. g., [13]). Even though both terms ("process complexity" and "process quality") are used, proper definitions are missing.

In the rest of this section, we try to provide respective definitions, adapt a measurement approach which is also successfully used in software measurement, and show how process complexity as well as process quality (and performance respectively) fit into this measurement approach.

3.1 Process Complexity

Today, the term "complexity" is used in many domains. But only for very particular fields (e. g., complexity theory in theoretical computer science, space and time complexity of algorithms, Kolmogorov complexity), mathematical definitions exist. Generally, only "philosophical definitions" exist. The Merriam-Webster Dictionary, for example, defines the adjective "complex" as "hard to separate, analyze, or solve" (cited from [15, p. 4]).

To our knowledge, Latva-Koivisto published the first paper [15] which deals with finding a complexity measure especially for business process models. He cites

[15, pp. 4–5] some interesting ideas about complexity by Edmonds:

"This means that it [complexity] is a highly abstract construct relative to the language of representation and the type of difficulty that concerns one." [8, p. 379] "The relevant type of 'difficulty' depends somewhat upon your goals in modelling. Different kinds of difficulty will result in different measures of complexity [...]." [8, p. 381]

Latva-Koivisto states that a measure of complexity is related to [15, p. 5]:

- the use of the measure,
- the kind of difficulty associated with the use,
- the objective of the analysis, and
- the language of representation of the problem.

Cardoso defines process complexity as "the degree to which a process is difficult to analyze, understand or explain. It may be characterized by the number and intricacy of activity interfaces, transitions, conditional and parallel branches, the existence of loops, roles, activity categories, the types of data structures, and other process characteristics." [5, p. 202]

In [7, p. 36], he writes about the relation of complexity to other attributes: "A process can be measured according to different attributes. The attribute that we will target and study is the complexity associated with BPEL processes. Attributes such as time, cost, and reliability have already received some attention from researchers [...]."

3.2 Process Quality and Performance

According to Kan, quality can be defined as "conformance to requirements" or "fitness to use" [14, pp. 2]. He mentions two views of quality: the costumer's view on quality and the company's view on quality. For a costumer, quality is the "perceived value of the product he or she purchased, based on a variety of variables such as price, performance, reliability, and satisfaction" [14, pp. 3]. For a company, quality means that the customer's requirements on the product quality are fulfilled *and* that its own production costs are lower than the price for selling the product.

Adapted to process quality, one can give the following definition of process quality: For a customer, process quality means that the process' outcome (a product, a piece of information or a decision) is correct, arrives within adequate time and to an adequate price. For a company, all these factors also belong to quality—but additionally, the price for the process execution must be lower than the price which the customer is willing to pay and, furthermore, the process should be easily adaptable to changed circumstances.

Besides process complexity and quality, process performance is a third concept which can be found in process measurement literature. In [13], Jansen-Vullers *et al.* suggest a performance measurement framework consisting of the four dimensions time, cost, quality and flexibility. Quality is separated into internal and external quality in their framework. As their quality concept is practically equivalent to Kan's quality concept, and Kan's quality concept can include time, cost and flexibility, we propose to restrict on the notion "process quality" in the extensive meaning by Kan described above.

3.3 Adapted Measurement Approach from Software Measurement

All presented suggestions for a definition of process complexity show that complexity is no such property like length or mass which can be measured directly using meters or kilograms respectively. So, a more "philosophical" discussion (cf. Cardoso's definition) starts which does not bring us closer to solving our problem.

Therefore, we suggest an alternative approach: In our opinion, what is more important than complexity itself—especially for economic reasons—are the *implications* of this complexity like costs, time, duration, number of errors, changeability, flexibility, understandability, etc. (aspects of process quality according to Subsection 3.2). All these quantities have the advantage to be quantifiable and measurable.[1] The disadvantage is that they can only be measured after the process has been implemented and executed.

To overcome this problem, we suggest the adaptation of prediction systems (details will follow in Section 4)—a successful measurement approach from the area of software measurement. This is sketchily depicted in Figure 1.

A process has *internal* and *external* attributes.

Internal attributes can be measured purely in terms of the process separate from its behavior [9, p. 74]. These attributes (e. g., structural properties like the number of actions) *could* contribute to the process complexity. Numerous internal attributes are imaginable and appropriate metrics have already been proposed (especially for structural properties) or can be defined. Using these metrics, one gets corresponding metric values of the process.

External attributes can be measured only with respect to how the process relates to its environment [9, p. 74].

[1] For, e. g., costs, time, duration, number of errors, this is trivial to see. But also attributes like changeability, flexibility and understandability are measurable if we look at the costs, time, number of errors, etc. it takes to change or understand a process. Fenton and Pfleeger give some ideas for measuring maintainability in [9, pp. 354–355].

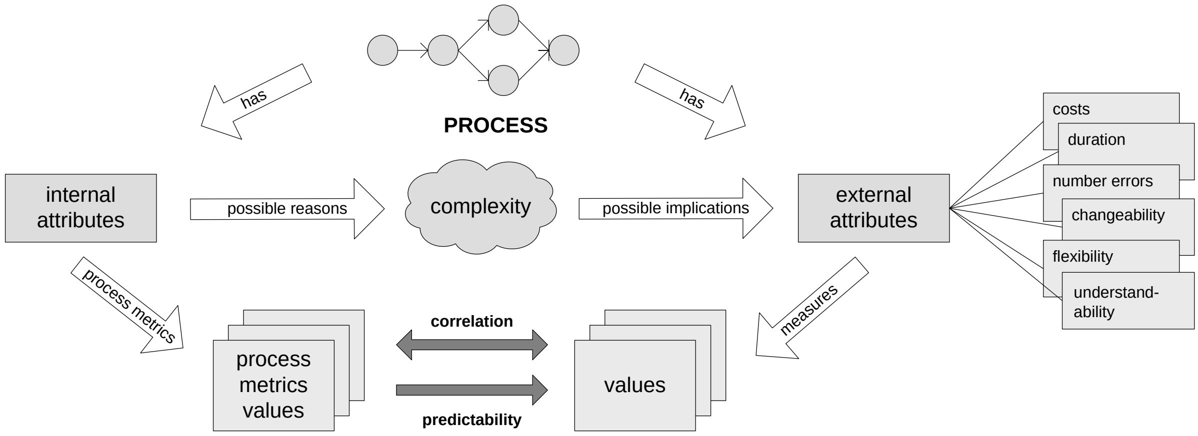

Figure 1: Prediction systems adapted to process measurement.

The external attributes like costs, time, number of errors, etc. are possibly affected by the process complexity and they are measurable. External attributes are aspects of process quality (and performance respectively).

The last step of our approach is also the most important one: One has to show a correlation between the metric values and the values of the external attribute. If such a correlation exists, the metric can be used as a predictor for the external attribute at a much earlier time.

Internal and external attributes and a prediction system measurement approach are also successfully used in software measurement. [9, pp. 74–75].

At this point, we want to take a look back to the ideas of Edmonds and Latva-Koivisto (see Subsection 3.1). As they suggest, there is not one single complexity metric which can measure every aspect of the process complexity in our approach. Instead, several different pairs of metrics for internal and external attributes can exist forming a prediction system and so representing *one* of the existing links between *reasons* and *implications* of process complexity.

In this context, we believe Cardoso is subject to a misconception when he puts complexity at the same "level" as attributes such as time, cost and reliability (see Subsection 3.1). Instead, complexity can be the reason for these attributes.

Because of the proposed measurement approach, we recommend the notion "process metric" for the metrics measuring internal attributes instead of "complexity metric" or "complexity measure" as we do not measure complexity itself. For the metrics measuring external attributes, we recommend the notion "process quality metrics".

4 Process Measurement Validation

According to Fenton and Pfleeger, the usual meaning of measurement is "that we wish to assess some entity that already exists. This measurement for assessment is very helpful in understanding what exists now or what has happened in the past." [9, p. 42]

Based on this statement, they define measurement systems [9, p. 104]:

Definition 1 (Measurement systems) *Measurement systems are used to assess an* existing *entity by numerically characterizing one or more of its attributes.*

"However, in many circumstances, we would like to predict an attribute of some entity that does not yet exist." [9, p. 42] For example, Balasubramanian and Gupta mention that interesting process performance measurements "like process cost, cycle time, process throughput and process reliability [...] can be calculated only after process execution and are of limited use in predicting future process performance" [1, p. 680]. Consequently, they note the importance for indicators for process performance (process quality in our notation) at pre-implementation stage [1, pp. 680–681]. Cardoso emphasizes the importance "to develop methods and measurements to automatically identify complex processes and complex areas of processes" [5, p. 202].

For that second purpose of measurement, Fenton and Pfleeger define prediction systems [9, p. 104]:

Definition 2 (Prediction systems) *Prediction systems are used to predict some attribute of a* future *entity, involving a mathematical model with associated prediction procedures.*

Besides the use for *future* entities, as stated in the definition of Fenton and Pfleeger, prediction systems can also be used to predict some attribute of an *existing* entity which is measurable only in a very laborious manner.

Measurement and prediction systems have to be validated before they can be used. The different validation procedures for both systems are described in the following subsections.

4.1 Validation of Measurement Systems

The statements in this subsection apply both for process metrics and process quality metrics, unless otherwise stated.

4.1.1 Objective/Subjective Metrics

For the process quality metrics measuring external process attributes, there are two kinds of metrics: objective and subjective metrics. Objective metrics are performance-based and measure, e. g., time, costs and number of errors. Subjective metrics are perception-based and measure, e. g., how difficult does a subject rate a process.

4.1.2 Requirements of Metrics

- *reliability/consistency*: Metric values obtained by different observers of the same process have to be consistent [15, p. 3] [5, p. 202]. For mathematically defined process metrics, this is automatically fulfilled. But for process quality metrics measuring external process attributes like understandability, the exact measurement conditions are important to fulfill this requirement. Kan gives a good example [14, pp. 70–71]: If one wants to measure the height of a person, the measurements should be taken at a special time of day (e. g., always in the morning) and always barefooted. Otherwise, the metric values of the same person could vary a lot.

- *validity*: According to Kan [14, pp. 71–72], validity can be classified into *construct validity* and *content validity*. The first checks whether the metric really represents the theoretical concept to be measured (e. g., is church attendance a good metric for religiousness?). The second checks whether the metric covers the range of meanings included in the concept (e. g., a test of mathematical ability for elementary pupils cannot be limited to addition but should also include subtraction, multiplication, division and so forth).

- *computability/ease of implementation/automation*: A computer program can calculate the value of the process metric in finite time—and preferably quickly. The difficulty of the implementation of the method which computes the process metric is within reasonable limits. [15, p. 4] [5, p. 202]

  This metric requirement found in process metric literature only applies to process metrics (measuring internal process attributes) which are mathematically defined and can be computed automatically.

These requirements are important as "good predictive theories follow only when we have rigorous measures of specific, well-understood attributes" [9, p. 108].

4.2 Validation of Prediction Systems

According to our adapted measuring approach (see Subsection 3.3), a proposed process metric has to be validated against a concrete external attribute (process quality metric). The goal of such a validation is to show a correlation between the process metric values and the corresponding external attribute in question. As Fenton and Pfleeger state, "rather than being a mathematical proof, validation involves confirming or refuting a hypothesis" [9, p. 104].

The validation can be done either by using existing data (e. g., from log files) or by conducting experiments (to get new data). Fenton and Pfleeger emphasize the advantages of experiments as the level of control and the level of replication are much higher [9, p. 120]. Basics about empirical investigations (e. g., experimental design among other things) can be found in [9, pp. 117–152].

As there can be different kinds of correlation (positive linear, negative linear and many forms of non-linearity) [14, pp. 77–80], scatter plots are a good method to visually search for any form of correlation (also non-linear). The next step is to use a measure of association like Spearman's rank correlation coefficient (arbitrary monotonic function) or Pearson's correlation coefficient (linear correlation). If a correlation could be found, one could also try to find an equation which mathematically describes the correlation (e. g., linear regression, multivariate regression, non-linear regression). [9, pp. 199–200]

In the field of software measurement, IEEE Standard 1061 (IEEE Standard for a Software Quality Metrics Methodology) gives a method for validating prediction systems [12, pp. 10–13] which checks among correlation also additional properties as tracking, consistency, predictability, discriminative power and reliability[2].

4.2.1 Measurement Dimensions

Predictive systems are only valid for very special conditions. According to Fenton and Pfleeger, "validation

[2]In [12], "reliability" has another meaning than the homonymous requirement for valid measurement systems (see Subsection 4.1).

must take into account the measurement's purpose; a measure may be valid for some uses but not for others" [9, p. 107].

Consequently, the conditions during validation and the later use of the prediction system must be consistent. The following four "measurement dimensions" are generally important conditions. For special cases, additional conditions may exist.

- **Process metric (internal process attribute)**
  The process metric defines the "measurement rule" for quantifying the chosen internal process attribute.
- **Process quality metric (external process attribute)**
  The external process attribute (probably affected by process complexity) whose value correlates with the process metric value.
- **Subjects**
  Which persons are involved in the measurement? Possible persons are, e. g., process designers, process analysts, programmers and end-users (i. e., the employees working in the process). As these persons have different skills and different views of the process, the values of the same external process attribute (e. g., time, costs and number of errors) can differ a lot depending on the involved persons (subjects).
- **Process phase**
  As in software engineering, a process life cycle consists of several process phases: modeling, analysis, implementation, deployment, execution, maintenance and modification of the process.

  In contrast to software engineering, process execution is an additional phase in which problems can occur. After a software program is implemented, no new errors are introduced by executing the program. But as processes are executed (at least partially) by humans, additional errors can occur while executing a process.

4.2.2 Interpretation of Process Metric Values

After having validated a prediction system, one has to identify the range or threshold between "good" and "problematic" metric values of the process metric contained in the prediction system. Only with this knowledge, one can detect problematic processes and take countermeasures.

5 Application of Metrics

Having established valid measurement and prediction systems for processes, the question arises what to do with these metrics.

In this section, we present several possible applications of metrics. They can be used both for processes that are newly implemented and for finding and dealing with "those existing processes that are good candidates for improvement and simplification, or even complete reengineering" [15, p. 3].

5.1 Selection of Metrics

As there exist numerous metrics for processes, first, one has to select proper metrics for the considered "problem". Using all available or accidentally selected metrics would just generate numerous numerical values without any purpose for the considered "problem".

Basili *et al.* propose an approach for the selection of metrics for software measurement—the *goal question metric paradigm* [2]. This approach is also applicable for process measurement and can be used both for selecting process metrics and process quality metrics.

The approach has three levels: conceptual level (goal), operational level (question) and quantitative level (metric). At the first level, a precise goal is defined. A set of questions for assessing and achieving the goal is established at the second level. At the third level, a set of metrics is assigned to each question in order to quantitatively answer the questions. The resulting GQM model has a hierarchical structure with possibly several goals, multiple questions per goal and several metrics per question. A metric can be assigned to multiple questions.

Using this top-down approach, only useful metrics (and possibly prediction systems) for the current "problem" are selected and no unnecessary metric values are collected.

5.2 Different Measurement Purposes

For the field of software measurement, Fenton and Pfleeger mention three different measurement purposes [9, pp. 13–14] which can also be adopted to process measurement.

5.2.1 Understand

For this first purpose, a process is only measured using different selected metrics to get a better understanding about what happens within this process. Afterwards, no changes or concrete actions are conducted. Through this, the process can be compared while being modified over time (modifications not caused by process measurement!) or it can be compared with other processes within the same company.

For this purpose, only valid measurement systems are necessary.

5.2.2 Control

Here, a process is also measured and no changes on the process are made. But the remaining complexity is managed/controlled by testing "problematic" processes or process parts more intensively (e. g., test cases or inspections as in software engineering).

For this purpose, valid prediction systems are necessary.

5.2.3 Improve

For the third purpose, a process is measured. If a bad quality was measured (for existing processes) or predicted (for new processes), the process is going to be changed in order to improve the process. So, the goal is to reduce *unnecessary* complexity within the process.

One has to consider that the complexity of a process cannot be reduced arbitrarily [3, p. 117]. Here, one must distinguish between the *intrinsic* complexity of a process and the complexity of a process model (concrete process realization for this process). The chosen process model is not independent from the overall problem. So, it has a "natural" minimal complexity. This fact was already referred to by Fenton and Pfleeger for software measurement [9, p. 267].

One can compare this with an example of an analogous problem—runtime complexity of algorithms: The general problem of sorting has a (mathematically proven) minimal complexity of $\Omega(n \log n)$. The Heapsort sorting algorithm, for example, has complexity $\mathcal{O}(n \log n)$. But nevertheless, more inefficient sorting algorithms exist (e. g., Selection Sort with $\mathcal{O}(n^2)$).

But even if a reduction of complexity is possible and would probably cause higher quality, one should first compare the costs for the process change with the expected increase of incomes with this process in order to decide whether to actually implement the changes.

As the quality for the changed process is predicted within this purpose, valid prediction systems are necessary.

6 Assessment of Existing Work

In this section, we want to assess the existing work about process metrics according to the well-established methods from software measurement adopted to process measurement in this article.

Most of the proposed process metrics (measuring internal attributes) are adapted from software metrics. As they all have a mathematical definition, they fulfill the reliability/consistency and computability requirements of Subsection 4.1. So, they form valid measurement systems.

We could only find three works dealing with validating prediction systems in the literature:

- **Cardoso: Validation of control-flow complexity metric (CFC) [6]**

  Cardoso conducted a laboratory experiment and computed the Spearman rank correlation coefficient between the CFC values of processes and the subjective complexity values stated by the experiment subjects. He could show a statistically significant correlation. But it is not clear how this subjective complexity is connected to any external process attribute (process quality). So, it is no practically relevant prediction system.

- **Mendling *et al.*: Using process metrics for predicting faulty EPCs [18–20]**

  604 EPC processes of the SAP Reference Model were analyzed using the verification tool WofYAWL. This way, 34 faulty processes were identified. Multivariate logistic regression was used to predict faulty processes. As all metrics fulfill the requirements and the correlation could statistically be shown, it is a valid prediction system.

- **Cardoso, Mendling, Reijers, Strembeck, van der Aalst, Vanderfeesten: Influencing factors on process understandability [21, 22, 28]**

  In [21], Mendling *et al.* conducted a laboratory experiment and assessed the Pearson correlation coefficients (linear correlation) between several process metrics and a metric called SCORE intended to measure understandability (process quality) as well as a linear regression between the process metrics and SCORE. The SCORE metric is computed as the sum of correct answers to just eight closed and one open question about a process.

  Vanderfeesten *et al.* introduced the cross-connectivity metric (CC) [28]. It was added to the process metrics and into the data collected in [21]. No significant correlation between CC and SCORE could be found. But CC is part of a better linear regression model between process metrics and SCORE.

  In [22], Mendling and Strembeck did another experiment examining influencing factors on process understandability. Besides correlations between personal and structural (process metrics) factors, also content related factors (task labels) were analyzed. Here, understandability is measured with six yes/no questions about the processes. Again, a linear regression model was found.

  Because of its simple definition, the content validity and reliability of the SCORE metric is questionable. It is not clear whether all aspects of process understandability are covered. The small number of asked questions and the non-systematic selection of these questions could cause that only especially easy or difficult process parts are examined by the questions. Consequently, SCORE is no valid measurement system. But this makes the whole prediction system for process understandability invalid.

We explain these points of criticism together with an experimental evaluation of our hypotheses in [17].

7 Conclusion and Future Work

In this paper, we gave a short overview about publications on process measurement. Many proposed process metrics are adapted from software metrics and are told to measure process complexity, quality and/or performance. We observed that there are no concrete definitions of process complexity and process quality in the literature. Often, both terms are even used as synonyms.

The contribution of this article is an adoption of more well-established concepts from software measurement to the field of process measurement: We adapted the prediction system measurement approach between internal and external process attributes from software measurement avoiding a concrete definition of process complexity. Additionally, we showed how process quality (and performance respectively) fit into this measurement approach. Furthermore, we pointed to the existence of measurement and prediction systems and their proper validation. Validity as additional requirement for metrics was identified as important in software measurement but is not found in process measurement literature so far. We recommended the goal question metric paradigm for the selection of process metrics and showed different purposes of process metrics (understand, control and improve).

For future work in this area, we suggest the proper validation of prediction systems using the numerous proposed process metrics. Doing so, the creation of process quality metrics measuring external process attributes that fulfill the validity requirement is especially important.

References

[1] S. Balasubramanian and Mayank Gupta. Structural metrics for goal based business process design and evaluation. *Business Process Management Journal*, 11(6):680–694, 2005.

[2] Victor R. Basili, Gianluigi Caldiera, and H. Dieter Rombach. Goal question metric paradigm. In John J. Marciniak, editor, *Encyclopedia of Software Engineering*, volume 1, pages 528–532. 1994.

[3] J. Cardoso, J. Mendling, G. Neumann, and H. A. Reijers. A discourse on complexity of process models. In Johann Eder and Schahram Dustdar, editors, *Business Process Management Workshops: Proceedings of the BPM 2006 International Workshops*, volume 4103 of *LNCS*, pages 117–128, 2006.

[4] Jorge Cardoso. About the data-flow complexity of web processes. In *Proceedings of the 6th International Workshop on Business Process Modeling, Development, and Support (BPMDS'05)*, pages 67–74, 2005.

[5] Jorge Cardoso. How to measure the control-flow complexity of web processes and workflows. In Layna Fischer, editor, *Workflow Handbook 2005*, pages 199–212. Future Strategies Inc., 2005.

[6] Jorge Cardoso. Process control-flow complexity metric: An empirical validation. In *Proceedings of the 2006 IEEE International Conference on Services Computing (SCC 2006)*, pages 167–173, 2006.

[7] Jorge Cardoso. Complexity analysis of BPEL web processes. *Software Process: Improvement and Practice*, 12(1):35–49, 2007.

[8] Bruce Edmonds. Complexity and scientific modelling. *Foundations of Science*, 5(3):379–390, 2003.

[9] Norman E. Fenton and Shari Lawrence Pfleeger. *Software Metrics: A Rigorous and Practical Approach*. International Thomson Computer Press, 2nd edition, 1996.

[10] Volker Gruhn and Ralf Laue. Adopting the cognitive complexity measure for business process models. In Yiyu Yao, Zhongzhi Shi, Yingxu Wang, and Witold Kinsner, editors, *Proceedings of the 5th IEEE International Conference on Cognitive Informatics (ICCI 2006)*, volume 1, pages 236–241, 2006.

[11] Volker Gruhn and Ralf Laue. Complexity metrics for business process models. In Witold Abramowicz and Heinrich C. Mayr, editors, *Business Information Systems: Proceedings of the 9th International Conference on Business Information Systems (BIS 2006)*, volume P-85 of *Lecture Notes in Informatics (LNI)*, pages 1–12, 2006.

[12] IEEE Computer Society. IEEE Standard 1061 - IEEE Standard for a software quality metrics methodology, 1998.

[13] M. H. Jansen-Vullers, M. W. N. C. Loosschilder, P. A. M. Kleingeld, and H. A. Reijers. Performance measures to evaluate the impact of best practices. In *Proceedings of the 8th Workshop on Business Process Modeling, Development, and Support (BPMDS '07)*, 2007.

[14] Stephen H. Kan. *Metrics and Models in Software Quality Engineering*. Addison-Wesley, 2nd edition, 2002.

[15] Antti M. Latva-Koivisto. Finding a complexity measure for business process models. Research report, Helsinki University of Technology, Systems Analysis Laboratory, 2001.

[16] Gang Soo Lee and Jung-Mo Yoon. An empirical study on the complexity metrics of petri nets. *Microelectronics and Reliability*, 32(3):323–329, 1992.

[17] Joachim Melcher and Detlef Seese. Towards validating prediction systems for process understandability: Measuring process understandability. In *Proceedings of the 10th International Symposium on Symbol and Numeric Algorithms for Scientific Computing (SYNASC 2008): Workshop on Workflow and Process Management (WfPM '08)*, 2008. to be published.

[18] J. Mendling, M. Moser, G. Neumann, H. M. W. Verbeek, B. F. van Dongen, and W. M. P. van der Aalst. A quantitative analysis of faulty EPCs in the SAP reference model. BPM Center Report BPM-06-08, 2006.

[19] Jan Mendling. Testing density as a complexity metric for EPCs. Technical Report JM-2006-11-15, Vienna University of Economics and Business Administration, 2006.

[20] Jan Mendling. *Detection and Prediction of Errors in EPC Business Process Models*. PhD thesis, Vienna University of Economics and Business Administration, 2007.

[21] Jan Mendling, Hajo A. Reijers, and Jorge Cardoso. What makes process models understandable? In Gustavo Alonso, Peter Dadam, and Michael Rosemann, editors, *Business Process Management: Proceedings of the 5th International Conference BPM 2007*, volume 4714 of *LNCS*, pages 48–63, 2007.

[22] Jan Mendling and Mark Strembeck. Influence factors of understanding business process models. In Witold Abramowicz and Dieter Fensel, editors, *Business Information Systems: Proceedings of the 11th International Conference BIS 2008*, volume 7 of *Lecture Notes in Business Information Processing*, pages 142–153, 2008.

[23] Sandro Morasca. Measuring attributes of concurrent software specifications in petri nets. In *Proceedings of the 6th IEEE International Software Metrics Symposium*, pages 100–110, 1999.

[24] Hajo A. Reijers and Irene T. P. Vanderfeesten. Cohesion and coupling metrics for workflow process design. In Jörg Desel, Barbara Pernici, and Mathias Weske, editors, *Business Process Management: Proceedings of the Second International Conference BPM 2004*, volume 3080 of *LNCS*, pages 290–305, 2004.

[25] Elvira Rolón, Francisco Ruiz, Félix García, and Mario Piattini. Applying software metrics to evaluate business process models. *CLEI Electronic Journal*, 9(1), 2006.

[26] Irene Vanderfeesten, Jorge Cardoso, Jan Mendling, Hajo A. Reijers, and Wil van der Aalst. Quality metrics for business process models. In Layna Fischer, editor, *BPM and Workflow Handbook 2007*, pages 179–190. Future Strategies Inc., 2007.

[27] Irene Vanderfeesten, Jorge Cardoso, and Hajo A. Reijers. A weighted coupling metric for business process models. In Johann Eder, Stein L. Tomassen, Andreas L. Opdahl, and Guttorm Sindre, editors, *Proceedings of the CAiSE'07 Forum at the 19th International Conference on Advanced Information Systems Engineering*, pages 41–44, 2007.

[28] Irene Vanderfeesten, Hajo A. Reijers, Jan Mendling, Wil M. P. van der Aalst, and Jorge Cardoso. On a quest for good process models: The cross-connectivity metric. In Zohra Bellahsène and Michel Léonard, editors, *Advanced Information Systems Engineering: Proceedings of the 20th International Conference CAiSE 2008*, volume 5074 of *LNCS*, pages 480–494, 2008.