Thresholds for Error Probability Measures of Business Process Models

Jan Mendling

Wirtschaftsuniversität Wien, Augasse 2-6, 1090 Vienna, Austria,
E-mail: jan.mendling@wu.ac.at

Laura Sánchez-González

Alarcos Research Group, TSI Department, University of Castilla La Mancha, Paseo de la Universidad, n4, 13071,Ciudad Real, Spain,
E-mail: laura.sanchez@uclm.es

Félix García

Alarcos Research Group, TSI Department, University of Castilla La Mancha, Paseo de la Universidad, n4, 13071,Ciudad Real, Spain,
E-mail: felix.garcia@uclm.es

Marcello La Rosa

Queensland University of Technology, GPO Box 2434, Brisbane QLD 4001, Australia and
NICTA Queensland Lab, PO Box 6020, St Lucia QLD 4067, Australia,
E-mail: m.larosa@qut.edu.au

Abstract

The quality of conceptual business process models is highly relevant for the design of corresponding information systems. In particular, a precise measurement of model characteristics can be beneficial from a business perspective, helping to save costs thanks to early error detection. This is just as true from a software engineering point of view. In this latter case, models facilitate stakeholder communication and software system design. Research has investigated several proposals as regards measures for business process models, from a rather correlational perspective. This is helpful for understanding, for example size and complexity as general driving forces of error probability. Yet, design decisions usually have to build on thresholds, which can reliably indicate that a certain counter-action has to be taken. This cannot be achieved only by providing measures; it requires a systematic identification of effective and meaningful thresholds. In this paper, we derive thresholds for a set of structural measures for predicting errors in conceptual process models. To this end, we use a collection of 2,000 business process models from practice as a means of determining thresholds, applying an adaptation of the ROC curves method. Furthermore, an extensive validation of the derived thresholds was conducted by using 429 EPC models from an Australian financial institution. Finally, significant thresholds were adapted to refine existing modeling guidelines in a quantitative way.

1. Introduction

Conceptual models play an important role in information systems design. It is well-known that design errors cause expensive rework when they are discovered only in late phases of the project [12]. Conceptual models are typically used at an early stage of a development project as a means of communication between different stakeholders and as a Computation Independent Model (CIM) in model-driven architecture [24]. Such models help to identify inconsistent perceptions of the yet to be built system and to specify a viable solution. The early resolution of design errors using conceptual models can thus lead to an increase in quality of the resulting software product.

Measurement is of crucial importance in providing quality assurance of a software project. In this context, DeMarco states that "you cannot control what you cannot measure" [19]. Measures establish the foundation upon which the achievement of goals can be assessed [8]. Various approaches and concepts of measurement have been applied to software engineering as a process, as well as to its inputs and to its product [23]. Moreover, empirical connections between internal measures like code complexity and external measures such as error probability have been studied already, as far back as in the early 1980s [7]. Nevertheless, most insights in this area report correlations or regression models between internal and external measures. While these statistics are informative to a software engineer in general, they do not directly help in actual decision making. Design decisions typically require a "yes" versus "no" assessment as to whether a certain change will be made. For instance, a software engineer might have to decide whether 500 lines of code are detrimental enough to warrant the decomposition of a class into multiple subclasses. What is required in such a context is a threshold value, which, if exceeded, indicates that a particular design action has to be taken.

In this paper, measurement is applied for business process models, which are typically used in the early design phase of a software project. A process model describes a set of activities and their control flow as it is supposed to be supported by a dedicated information system. Even though process modeling is one of the most heavily used modeling paradigms [17] and its importance for software quality has been widely recognized [44], up until now there has been no common understanding of threshold values that would indicate a bad process model. Against this background we provide the following contributions. Firstly, we use a collection of 2,000 business process models from practice to determine threshold values for different measures. Some of the selected measures were adapted from software engineering to business process models, such as Control-flow Complexity (CFC) (derived from Cyclomatic Complexity). Others were specifically defined for Business Process models, such as number of start or end events and Connector Mismatch. We use an approach for threshold extraction based on ROC curves. These curves have also been applied for the evaluation of object-oriented design quality such as in [52]. Secondly, we provide an extensive validation of the derived thresholds based on a case study with a large Australian financial institution. In this way, we inform empirical research on process modeling. In specific terms, we refine existing process modeling guidelines such as [42] based on the thresholds obtained. Our experimental approach is not specific to business process modeling, and can be adapted for error analysis in systems design in general.

The paper proceeds as follows. Section 2 provides an introduction to business process modeling with a focus on various measures and their empirical connection with quality aspects such as error probability. Section 3 uses a methodology for threshold derivation. Furthermore, it describes the EPC process model sample we use for our experiment, along with the resulting thresholds. In Section 4 we validate our results using a sample of EPC process models from a large Australian financial institution. Section 5 discusses the implications from this research, in particular with respect to existing process modeling guidelines. Section 6 concludes the paper with a summary and an outlook on future research.

2. Background

This section provides an overview of business process modeling and corresponding measures. Section 2.1 introduces the essential elements of a business process model. Section 2.2 provides theoretical arguments as to why certain process models are less understandable and more error-prone. Section 2.3 describes measures that correlate with understanding and error probability.

2.1. Business Process Models and Errors

Business process models capture various aspects of a business process. Typically, there is a strong emphasis on the control flow, which essentially relates to the order in which activities can be executed. In this paper we use Event-driven Process Chains (EPCs) to illustrate our argument. This choice is motivated purely by the availability to the authors of a large dataset of EPC models, which was used to conduct the experiment. Nonetheless, the results of this paper can easily be applied to other process modeling notations. EPCs define so-called functions for capturing activities of the process, and events as pre- and post-conditions to them. As with other process modeling languages like UML Activity Diagrams, BPMN or YAWL, they include so-called connectors for defining complex routing behavior [32, 39]. There are three types of connectors: XOR (exclusive branching and merging), AND (concurrent branching and synchronization), and OR (inclusive branching and synchronization). XOR-splits define decision points whose subsequent branches can be merged using an XOR-join. AND-splits introduce parallel execution that can be synchronized by downstream AND-joins.

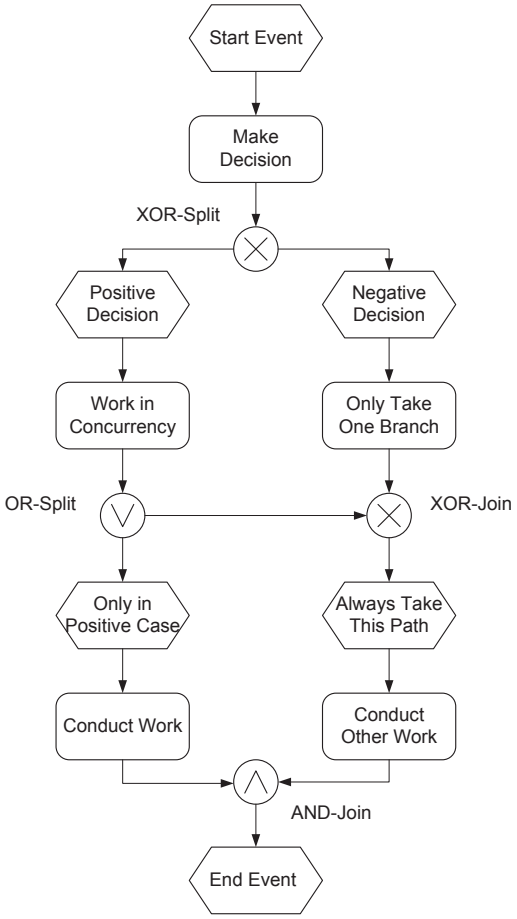

Figure 1: Example of an EPC process model

OR-splits activate one, several, or all subsequent branches based on conditions. They need to be synchronized with OR-join elements, which are difficult to implement in the general case [34, 39].

Figure 1 illustrates the essential elements of an EPC business process model. The process is triggered by a start event. Then, a decision has to be made, yielding either a positive or a negative decision. For a *negative decision*, we only take one branch and continue via the XOR-join to a part of a process that is always taken. We conduct other work and reach an AND-join before the end event. In case of a *positive decision*, we conduct work in concurrency. The OR-split allows us to consider either one of the branches or both of them. If both branches are taken, we conduct work that can be considered only for a positive decision. The AND-join then synchronizes both branches.

The example of this process model also shows that a combination of different connectors can easily results in errors. The model cannot always terminate properly. Whenever the OR-split activates both branches, the AND-join can synchronize them and forward control towards a good completion. In any other case, the execution gets stuck at the AND-join, because control from one of the two incoming branches, which would bring the model to completion, is missing. Such an error is called a deadlock. It has been found that many process models in practice include such errors, and that often about 20% of the models have deadlocks or other behavioral problems [40]. Clearly, such deadlocks point to bad design. If a business process model is used for communication purposes and requirements analysis, a deadlock might lead to confusion in the stakeholders consulting this model. In case the process model is meant to be automatically enacted by a workflow management system, instances of this model can actually get stuck in the deadlocking state, meaning that such instances cannot further progress. This in turn may lead to an increased cost and time to complete such instances, since the process model needs first to be corrected and then re-enacted, or the deadlocking instance needs to be rectified on the fly.

2.2. Theoretical Considerations on Errors

Finding errors in a process model is not a trivial task. For some classes of process models, deadlocks and other problems can be found in an efficient way using the soundness criterion and Petri net analysis techniques [1, 22]. There are also tools in existence which fix unsound nets automatically [25]. It is generally the case, though, that the reachibility graph needs to be constructed; this is an NP-complete problem [21]. We therefore have to consider two aspects when tracing back the reasons why errors occur in process models [50, 25]: First of all, often behavioral errors cannot be discovered from the process model directly, since a combination of nodes may be responsible for that model to be unsound. Secondly, typically a considerable mental effort is required when checking a process model for correctness due to the number of transitions per trace when looking for errors.

Process models are efficient in representing the different decisions and routing conditions throughout the progress of executing a business process. In contrast to a reachability graph, they define behavior much more compactly. Research on visual programming languages emphasizes that a particular type of information is often highlighted at the expense of another one [26]. That means that there should be a fit between the task and the chosen representation [56]. The reachability graph shows errors more explicitly, yet its size can be disproportionately large in comparison to the corresponding process model. A process model, in contrast, has to be analyzed carefully to detect errors.

Apart from all we have just said, it should be remarked that humans possess only limited cognitive capabilities. Checking different execution sequences of a process model is a *hard mental operation* in terms of the cognitive dimensions framework [27]. Once there are too many branches to be considered, it is to be expected that the performance of a modeler will decrease because of the high cognitive load [53]. While the modeling expertise of a modeler is definitely relevant [50], these theoretical considerations would

lead us to expect that, in general, large, complex models are more likely to include errors than simple ones.

2.3. Related Work on Process Model Measures

Several factors have been found to be relevant factors for process model understanding and error probability. They include model purpose, problem domain, modeling notation, and layout [57, 28, 4, 48, 50]. In this paper, we focus on those factors that refer to the structure of a process model.

Research on process model measurement is inspired by prior work on software measures including lines of code, cyclomatic number, and object-oriented measures [38, 15, 23]. Early contributions in this area provide conceptual definitions of process model measures [36, 47, 45]. In the meantime, the focus of research is upon experiments and the empirical validation of measures. Cardoso reports upon the results of an experiment to correlate process measures with the perceived complexity of process models [14]. A team of researchers which includes Canfora, Rolón, and García correlate understandability and maintainability with size, complexity, and coupling of a process model [13, 5]. Further measures are defined based on cognitive considerations [54] and concepts of modularity [55, 3]. A set of measures is validated; these measures are seen as predictors of error probability in [43]. Other works demonstrate that size is an important model factor along with additional measures like structuredness [39].

3. Threshold Determination

In this section, we determine thresholds for discriminating process models of high and low error probability. Section 3.1 introduces the hypotheses for this research. Then, Section 3.2 describes our methodological approach based on logistic regression and ROC curves. Section 3.3 provides an overview of the model collection being used and the set of measures considered. Section 3.4 shows the results of the threshold calculation.

3.1. Hypothesis

The previous discussion gives us reason to assume that process models of higher complexity are more likely to have errors. In essence, this proposition builds on a cognitive argument that to analyze more complex models, more cognitive capabilities are required. Since the human brain can process only a particular amount of information at a time, there should also be a level of complexity at which the likelihood of an error is significantly higher than it is for small models. If that is the case, suitable process model measures would be able to discriminate models of high and low error probability, and a discriminating threshold for such measures should exist. Accordingly, we hypothesize as follows:

H: Thresholds of process model measures provide a significant means of discriminating models of high and low error probability.

In the following, we aim to investigate this hypothesis for various measures.

3.2. Methodology

To test hypothesis **H**, we follow a two-step approach: first, we have to estimate the discriminator function, and second, determine the thresholds. We utilize logistic regression for estimating a discriminator function (in which the p-value should be lower than 0.05) and ROC curves (in which the AUC value close to 0.5 indicated a non-valid curve) for finding thresholds. The significance of the AUC values is statistically checked using the Wilcoxon test of ranks [37].

Logistic regression is a statistical model for estimating the probability of binary choices [31]. In our case, we are interested in the binary variable *hasErrors* with a range of {*error*, *no error*}. The idea of a logistic regression is that this probability can be represented by the odds. This is the ratio of error probability divided by probability of no error. The logistic regression estimates the odds based on the logit function, which is $logit(p_i) = ln(\frac{p_i}{1-p_i}) = \beta_0 + \beta_1 x_{1,i} + \ldots + \beta_k x_{k,i}$, where β_0 is called the intercept and β_1, β_2, β_3, and so on, are called the regression coefficients of independent variables $x_{1,i}$, $x_{2,i}$, $x_{3,i}$ respectively . In our case, we will consider k process model measures as input variables and observations from i EPC process models. From the formula it follows that $p_i = \frac{e^{\beta_0+\beta_1 x_{1,i}+\ldots+\beta_k x_{k,i}}}{1+e^{\beta_0+\beta_1 x_{1,i}+\ldots+\beta_k x_{k,i}}}$. Figure 2 shows the relationship between input and dependent variables as an S-shaped curve converging to 0 for $-\infty$ and to 1 for ∞. Typically, 0.5 is used as a cut-off value for predicting either event or non-event. $Exp(\beta_k)$ gives the factor of change for the odds if the input variable β_k is incremented. $Exp(\beta_k) > 1$ increases and $Exp(\beta_k) < 1$ decreases error probability.

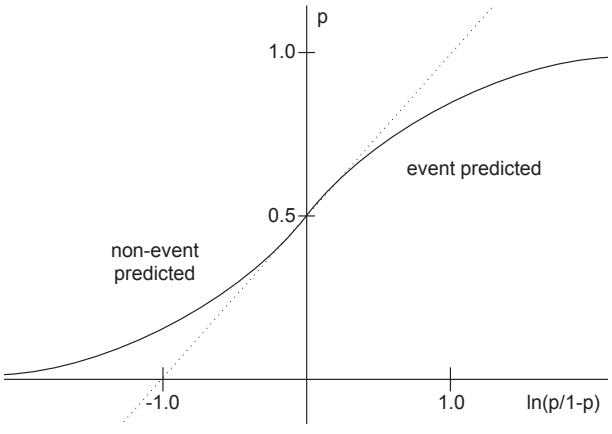

Figure 2: Illustration of a Logistic Regression

Table 1: Confusion matrix for measures and thresholds

	Actual	
Classified	Error	Non-Error
$Measure \geq threshold$	True-positives	False-positives
$Measure < threshold$	False-negatives	True-negatives

To determine thresholds we build on an approach based on Receiver Operating Characteristic (ROC) Curves. ROC curves *provide a pure index of accuracy by demonstrating the limits of a test's ability to discriminate between alternative states* (error/non-error) [59]. For the definition of a ROC curve, we need two variables: one binary (erroneous and non-erroneous model) and another continuous, which is the estimated error-probability function from the logistic regression of each measure. Each point in the ROC curve represents a pair of $sensitivity$ and $1 - specificity$. In this way, it represents the classification performance of any potential threshold. The determination of the best threshold builds on the confusion matrix (Table 1), for which sensitivity and specificity values are calculated as follows: $sensitivity$ = true positive(TP) rate = $\frac{TP}{TP+FN}$, $specificity$ = true negative(TN) rate = $\frac{TN}{FP+TN}$, where FN is *false negatives*, FP is *false positives*, and TN is *true negatives*. A *true positive* is found when the assessment of a measure value in relation to the threshold indicates that the model is likely to have errors and that in fact it does have. On the other hand, a *false positive* indicates that the model is likely to have errors and, actually, it does not have. Finally, a *false negative* indicates that the model is error-free while indeed it is not.

The test performance is assessed using the Area Under the ROC Curve (AUC). AUC is a widely used measure of performance of classification [29]. Ranging between 0 and 1, it can be used to assess how good threshold values are at discriminating between models that have errors and those that do not. There are rules of thumb for assessing the discriminative power of measures based on the AUC [31]. An AUC$<$ 0.5 is considered *no good*, *poor* if AUC$<$ 0.6, *fair* if AUC$<$ 0.7, *acceptable* if AUC$<$ 0.8, *excellent* if AUC$<$ 0.9, and *outstanding* if AUC$\leq$ 1. The standard error or p-value is estimated using a 95% confidence interval. The test checks if the AUC is significantly different from 0.5. Accordingly, our prior hypothesis can be operationalized for a measure m as

$\mathbf{H}_0^m$ Null Hypothesis: The AUC of a process model measure m is equal to 0.5.

$\mathbf{H}_A^m$ Alternative Hypothesis: The AUC of a process model measure m is significantly different from 0.5.

For those measures that are found to be valid according to the hypothesis, we can determine a threshold based on the ROC curve. As well as sensibility and specificity values, we have to consider two additional aspects in the determination. First of all, the relative costs of false results, both false negative and false positive, and the benefits of correct classifications have to be considered. Secondly, the relative proportions of the models to be discriminated have to be taken into account. High sensibility is appropriate when errors are serious and treatable, and false positives do indeed hinder the implementation of the process.

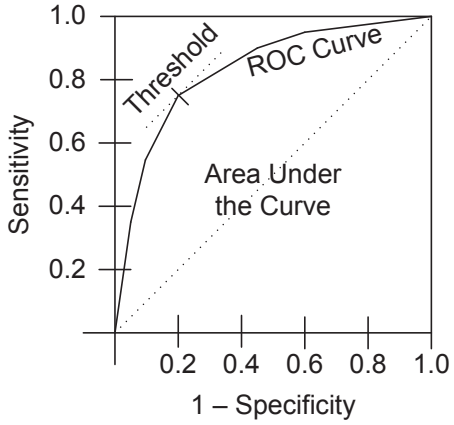

Figure 3: Illustration of a ROC Curve and Threshold

Figure 3 illustrates how the threshold is determined. We need a criterion to choose a threshold value for a measure (sensitivity, 1-specificity pair) to balance benefits and costs. The purpose is to maximize both values, i.e. sensitivity and specificity at the same time [31] minimizing false-positive and false-negatives. We use the following criterion for selecting a point in the curve as a threshold. Without domain-specific information, we assume sensitivity and specificity to be of equal importance. The best threshold can then be selected by finding the point in the curve that maximizes both sensibility and specificity. This is the point with the greatest distance from the 0.5 diagonal.

3.3. Experimental Setting

To test the hypothesis, we use a sample of business process models from practice. This sample contains 2003 EPC business process models stemming from four collections [39]. The first collection is the *SAP Reference Model*. It was developed in the 1990s as a redocumentation of the SAP R/3 system [33, p.VII]. It contains 604 models. The second collection was constructed as part of a reengineering project in the 1990s for a *German service provider* with academic supervision. It includes 381 EPC process models. The third collection was built for an *Austrian financial institution* as part of a process documentation project. It covers 935 EPCs. The fourth collection contains 83 EPCs from *consulting companies*.

We used a process analysis tool to check each model for errors [41]. This tool was also used to calculate various measures for each model. To be precise, we consider the set of process model measures formally defined in [39, pp. 117-128]. These measures can be organized into five groups:

Table 2: Measures' values for the sample process model in Figure 1

SIZE			
NODES	15	AND-SPLITS	0
ARCS	16	AND-JOINS	1
TASKS	5	XOR-SPLITS	1
START-EVENTS	1	XOR-JOINS	1
END-EVENTS	1	OR-SPLITS	1
CONNECTORS	4	OR-JOINS	0
CONNECTION			
DENSITY	0.08	CONN. COEFF.	1.07
AV.CONN.DEGREE	3	MAX.CONN.DEGREE	3
MODULARITY			
SEPARABILITY	0.2	SEQUENTIALITY	0.31
STRUCTUREDNESS	0.6	DEPTH	1
CONNECTOR INTERPLAY			
CONN. MISMATCH	4	CFC	4
CONN. HETER.	0.95		
COMPLEX BEHAVIOR			
CYCLICITY	0	TOKEN SPLIT	1

- *Size measures:* NODES, ARCS, TASKS, START-EVENTS, END-EVENTS, CONNECTORS, AND-SPLITS, AND-JOINS, XOR-SPLITS, XOR-JOINS, OR-SPLITS, OR-JOINS are all related to the number of a particular type of elements in a process model. These include counts of the number of arcs (ARCS) and nodes (NODES). The latter can be further subdivided into TASKS on the one hand and CONNECTORS on the other hand. The most specific counts are subcategories of the different types of logical connectors, like AND-SPLITS and OR-JOINS.

- *Connection:* DENSITY is the ratio of the total number of arcs in a process model to the theoretically maximum number of arcs (i.e. when all nodes would be directly connected). The CONNECTIVITY COEFFICIENT is the ratio of the total number of arcs in a process model to the total number of its nodes. The AVERAGE CONNECTOR DEGREE captures the average number of both incoming and outgoing arcs of the connector nodes in the process model. The MAXIMUM CONNECTOR DEGREE expresses the maximum sum of incoming and outgoing arcs of connector nodes.

- *Modularity:* The SEPARABILITY is the ratio of the number of cut-vertices divided by the total number of nodes in the process model. The SEQUENTIALITY is the degree to which the model is constructed of pure sequences of tasks. STRUCTUREDNESS captures the extent to which a process model can be built by nesting blocks of matching split and join connectors. DEPTH is the maximum nesting of structured blocks in a process model.

- *Connector Interplay:* MISMATCH CONNECTOR is the sum of connector pairs that do not match with each other, e.g. when an AND-split is followed up by an OR-join. CONNECTOR HETEROGENITY defines the extent to which different types of connectors are used in a process model. CONTROL FLOW COMPLEXITY captures a weighted sum of all connectors that are used in a process model.

- *Complex Behaviour:* CYCLICITY captures the number of nodes in a cycle and relates it to the total number of nodes. TOKEN SPLITS gives the maximum number of paths in a process model that may be concurrently initiated through the use of AND-splits and OR-splits.

The different measures can be calculated for any EPC process model. Table 2 shows the values for the example model of Figure 2.

3.4. Threshold Calculation

We determine thresholds based on ROC curves and the Area Under the Curve. The results of testing the null hypothesis $\mathbf{H}_0^m$ for each of the measures is summarized in Table 3. All of them are significantly different from 0.5. According to the rules of thumb described previously [31], most of the measures yield an acceptable value for an AUC higher than 0.7. Several of them can be considered as excellent, if we follow the guidelines in [31]. The values have to be interpreted as follows. For instance, the measure NODES has an AUC of 0.841. This means that a model randomly selected from a group of erroneous models has 84% times more NODES than a randomly selected model from a group of non-erroneous models. The corresponding ROC curve is shown in Figure 4, along with three further charts of measures with an AUC exceeding 0.5. There are some specifics to be considered for measures of SEPARABILITY, SEQUENTIALITY, STRUCTUREDNESS and DENSITY. These measures are inversely correlated with the dependent variable error/no-error. This means that the ROC curve also has to be calculated with the inverse function $1/function(x)$. The p-value is determined based on comparing the AUC to a random curve [30]. If the p-value is low ($p_{value} < 0.05$), then it can be concluded that the Area under the ROC curve is significantly different from 0.5 and the threshold calculation is possible.

Apart from the test results, Table 3 also shows the thresholds obtained from the ROC curve analysis. The thresholds indicate which value of the corresponding measure is best able to discriminate erroneous from non-erroneous models. For example, a number of NODES higher than 31.5 or a STRUCTUREDNESS greater than 0.79 can be interpreted as indicators of poor design quality as regards error probability. Due to the involvement of humans in process modeling, one would not expect the same accuracy of predictions as in natural sciences like physics or chemistry [46]. Therefore, it is important to reflect upon the probability of errors associated with thresholds.

These probability values can be obtained via a method proposed by Bender in medical research [10]. Bender's method has been used in toxicology and occupational epidemiology studies, where it is interesting to find explanatory factors with a threshold effect on a specific response variable. The method has been adapted in other fields including software engineering [20, 11, 51].

Table 3: Thresholds identified based on ROC Curves

Measure	AUC	p-value	Threshold
CONN. HETEROGENEITY	0.874	0.011	0.4
CONN. MISMATCH	0.871	0.013	4.5
TOKEN SPLITS	0.861	0.013	7.5
CFC	0.861	0.013	4.5
NODES	0.841	0.015	31.5
DENSITY	0.831	0.016	0.033
END-EVENTS	0.824	0.023	2.5
SEQUENTIALITY	0.817	0.016	0.21
DEPTH	0.799	0.015	0.5
MAX. CONN. DEGREE	0.790	0.016	3.5
COEFF. CONNECTIVITY	0.767	0.015	1.021
STRUCTUREDNESS	0.766	0.025	0.79
SEPARABILITY	0.753	0.015	0.49
OR-SPLITS	0.739	0.024	0.5
START-EVENTS	0.736	0.024	2.5
AV. CONN. DEGREE	0.712	0.016	3.09
CYCLICITY	0.621	0.027	0.005
OR-JOINS	0.567	0.026	0.5

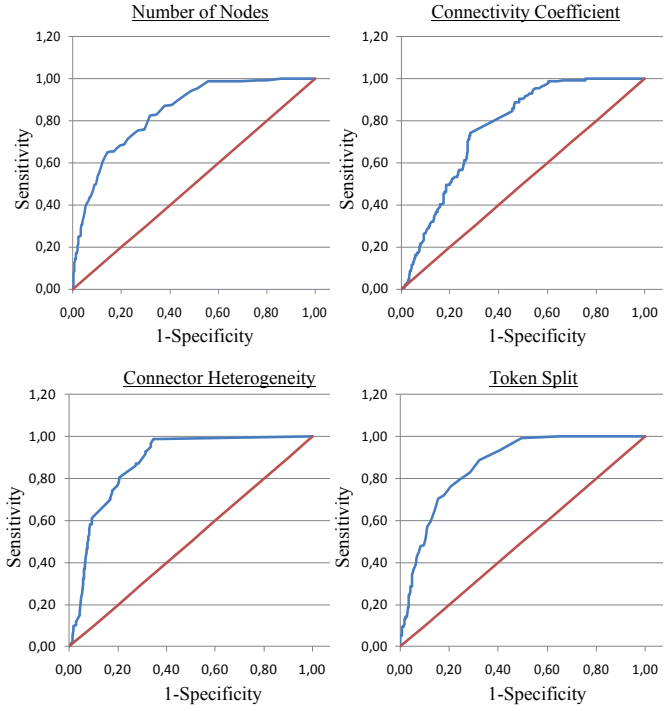

Figure 4: ROC curves about error probability measures

Following Bender, we can define a probability of finding errors in models for each range of measure values. This means "if a particular measure m yields $Y \epsilon [Y_1, Y_n]$, there is a Z% probability of finding errors in that model." Results above 89% of the measures are depicted in Table 4. The results of this method can be interpreted as the effect of a continuous risk factor m on a binary response variable Y. In our case, m is a measure and Y is the model that may have an error or not. Bender's method defines a benchmark value, which is a point of the curve where the risk of an event rises steeply. This point is called *value of an acceptable risk level* (VARL), where the acceptable risk level is given by a probability p_0. This value is calculated as follows: $VARL = p^{-}1(p_0) = \frac{1}{\beta} * (log(\frac{p_0}{1-p_0} - \alpha)$. For values of the specific measure below $VARL$, the risk of an event is lower than p_0. In the $VARL$ equation, the p_0 value can be varied from 0 to 1 in order to obtain different probabilities related to different threshold values. Table 4 shows the probability of finding errors in models associated with each thresholds. It can be interpreted as "if NODES does not exceed the threshold of 31.5, there is a 9% probability of finding errors in models".

Table 4: Thresholds and probabilities of finding errors in models

Measure	Threshold	Probability
TOKEN SPLITS	7.5	19%
DENSITY	0.033	16%
SEQUENTIALITY	0.21	12%
NODES	31.5	9%
STRUCTUREDNESS	0.79	9%
OR-SPLITS	0.5	9%
CONN. HETEROGENEITY	0.4	8%
COEFF. CONNECTIVITY	1.021	8%
CYCLICITY	0.005	7%
SEPARABILITY	0.49	7%
AV. CONN. DEGREE	3.09	7%
CFC	4.5	7%
START-EVENTS	2.5	7%
MAX. CONN. DEGREE	3.5	6%
CONN. MISMATCH	4.5	6%
END-EVENTS	2.5	5%
DEPTH	0.5	4%

4. Threshold Validation

In this section we present findings from applying the thresholds in a cross-validation. Section 4.1 describes the validation sample. Section 4.2 shows results on the accuracy of predicting errors based on the thresholds.

4.1. Cross-Validation

For the validation of the thresholds, we used a dataset consisting of 429 EPC process models sourced from an Australian insurance company under condition of anonymity. These models capture the way the company handles insurance claims for the various insurance products they offer, e.g. house insurance, motor vehicle insurance, and worker's compensation insurance. Within this company, these models are used at a conceptual level for requirements analysis and for communication purposes. Bearing all this in mind, it should be remarked that some

Table 5: Average and standard deviation measure values of the insurance company sample

SIZE					
	μ	σ		μ	σ
NODES	27.1	28.9	AND-SPLITS	0.35	1.52
ARCS	26.3	30.9	AND-JOINS	0.07	0.30
TASKS	7.69	14.5	XOR-SPLITS	2.60	3.17
START-EV.	3.29	3.61	XOR-JOINS	1.63	1.66
END-EV.	4.14	4.58	OR-SPLITS	0	0
CONNECTORS	5.04	5.42	OR-JOINS	0	0
CONNECTION					
	μ	σ		μ	σ
DENSITY	0.06	0.04	CONN. COEFF.	0.91	0.18
AV.C.DEGREE	3.27	1.32	M.C.DEGREE	4.56	2.79
MODULARITY					
	μ	σ		μ	σ
SEPARABILITY	0.49	0.22	SEQ.	0.35	0.27
STRUCT.	0.91	0.08	DEPTH	0.62	0.61
CONNECTOR INTERPLAY					
	μ	σ		μ	σ
C. MISMATCH	3.98	5.42	CFC	6.92	8.31
C. HETER.	0.13	0.22			
COMPLEX BEHAVIOR					
	μ	σ		μ	σ
CYCLICITY	0.04	0.11	TOKEN SPLIT	0.39	1.68

of these models are at a very high-level of abstraction, while others also include information about partners, IT resources and organizational policies.

Table 5 shows the average and standard deviation measure values for this collection of EPC models. The models of the insurance are significantly larger than the models that we used for determining the thresholds: while the prior sample has on average 20 nodes, the insurance models have 27 nodes. It is also interesting to note that OR-connectors are not found in the insurance sample. This is the consequence of a design guideline, which forbids the usage of OR-connectors. Most of the insurance models are highly structured (STRUCTUREDNESS of 0.91) and contain hardly any loops (CYCLICITY of 0.04). It should also be noted that the nesting structure is rather flat (DEPTH of 0.62). We used the same process analysis tool as before to check each model for errors[41]. We found 20 models with errors such as deadlocks, which yields an error rate of 4.66%.

4.2. Prediction

We approached the cross-validation of the thresholds from an information retrieval perspective. In this field of research true and false positives as well as true and false negatives are used as the basis for calculating precision and recall measures for assessing the quality of a search result [6]. *Precision* is the ratio of true positives to the sum of true and false positives. In terms of error prediction, this is the ratio of correctly found erroneous models based on a threshold value in relation to the sum of all error predictions. *Recall* is the ratio of true positives to the sum of true positives and false negatives, i.e. the ratio of correctly found erroneous models to the sum of all erroneous models. This measure is calculated similarly to sensitivity, which is used to plot ROC curves. Although sensitivity and specificity, and precision and recall are calculated in a similar way, they have different purposes: sensitivity and specificity are used for plotting ROC curves, while recall and precision are used to check the quality of measure thresholds. Furthermore, we will discuss the *accuracy*, which is the percentage of correctly classified models.

The precision and recall result from applying the thresholds to error prediction in the insurance sample are shown in Figure 5. STRUCTUREDNESS has the best precision of 23%, followed by NODES, COEFF. CONNECTIVITY, DENSITY, CONN. HETEROGENEITY, and CONN. MISMATCH that all have a precision of 10% to 13%. This set of measures are in the middle tier in terms of recall ranging from 55% to 70%. STRUCTUREDNESS only achieves a recall of 30%. It is interesting to note that the other measures which are rather weak in precision yield high recall values. An exception is the measures of complex behavior: CYCLICITY and TOKEN SPLIT have both low precision and recall values. Most of the cases have a high recall with a low precision, which means all the models with errors were selected by thresholds but several non-erroneous models were also selected. This implies that identifying error or non-erroneous models by means of only one measure is not enough. Several measures should be considered jointly.

From an accuracy perspective, three groups can be distinguished. In the first tier, TOKEN SPLIT, STRUCTUREDNESS, CYCLICITY, and CONNECTOR HETEROGENEITY all have an accuracy greater than 80%. It must be noted though that TOKEN SPLIT did not yield any true positive. The second group including NODES, COEFFICIENT OF CONNECTIVITY, DENSITY, and CONNECTOR MISMATCH are accurate with more than 75%. All other measures have an accuracy of less than 55%.

5. Discussion

In this section, we discuss the derived thresholds and their validation. Section 5.1 investigates the implications of this work for research and practice. Finally, Section 5.2 discusses threats to validity.

5.1. Implications for Research and Practice

The findings reported in this paper have implications for the role of measures in process modeling. To be more specific, the derived thresholds bear the potential to define modeling guidelines in a more precise way.

General guidelines of process modeling such as SEQUAL [35] or the Guidelines of Modeling [9] have been available for some time. Recent work in this area has aimed to define guidelines in a more quantitative and operational way, as well as to base them on empirical evidence. The Seven Process Modeling Guidelines are a result of these efforts. These guidelines formulate the following modeling directives [42]:

G1 Use as few elements in the model as possible.

G2 Minimize the routing paths per element.

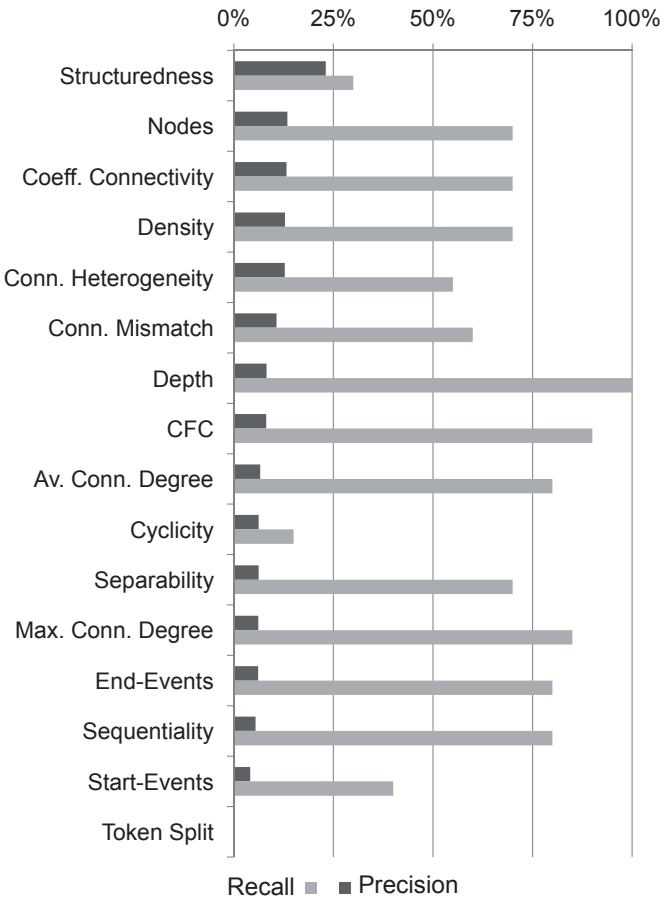

Figure 5: Precision and recall values for considered thresholds

G3 Use one start and one end event.

G4 Model as structured as possible.

G5 Avoid OR routing elements.

G6 Use verb-object activity labels.

G7 Decompose a model with more than 50 elements.

In general, the thresholds found in this paper confirm the guidelines, and are more specific. Guideline **G1** is confirmed by the threshold we found for NODES. Apparently, having a process model with fewer than 31 nodes still appears acceptable. Beyond this threshold, the probability of finding errors increases from 9% to 100% according to Bender's value at risk. In the validation, this threshold yielded 13% precision and 70% recall with an overall accuracy of classification of 78%.

Guideline **G2** suggests minimizing the routing paths of each connector. This is confirmed by the thresholds for AVERAGE CONNECTOR DEGREE and MAXIMUM CONNECTOR DEGREE, neither of which should be greater than 3. Beyond this threshold, there is an error probability of 7% and 6%, respectively. While both measures had a high recall in the validation sample of more than 80%, they provide only a limited precision of less than 10%.

Guideline **G3** recommends using one start and one end event. The thresholds we found suggest that having two start and two end events is still fine in terms of not being too error-prone. Beyond this threshold, there is still a medium error probability of 5% and 6%. Interestingly, the end-event measure provides a much better recall in the validation sample (80%) than the start-event measure (40%). This is surprising since several classes of control flow errors can be traced back directly to badly connected start events [18]. These figures apparently reflect the fact that start events are used in a structured way in the insurance sample. As a result, the start and end measures do not yield accurate results in the validation.

Guideline **G4** emphasizes the importance of structured modeling. This guideline is confirmed by the threshold of 0.79. Beyond this value, we observed an error probability of almost 10%. While STRUCTUREDNESS has a recall of only 30%, it has by far the best precision of roughly 25% for the insurance sample. The overall accuracy of prediction is greater than 90%. The central importance of this measure is therefore confirmed by our study. In order to avoid problems with structuredness, it seems desirable to use well-formed design patterns [2, 58]. This observation is further emphasized by the CONNECTOR MISMATCH measure. It has the second largest AUC value of about 87% and shows a good balance of precision and recall in the validation sample.

Guideline **G5** suggests avoiding OR-connectors. Indeed, the threshold we find confirms that the number of OR-splits and OR-joins should be below 1. OR-splits appear to be more critical depending on their AUC value. There is a 9% error probability when the threshold is passed. This makes sense since paths stemming from OR-splits need special attention in how they are synchronized. In contrast, OR-joins can be used with all kinds of splits without any harm. Indeed, we find this guideline confirmed by the fact that the modelers in the insurance company were not allowed to use OR-connectors. The criticality of connector interplay is further emphasized by the CONNECTOR HETEROGENEITY measure and its threshold of 0.4. Where OR-connectors are avoided, the maximum of this measure drops from 1 down to $-(0.5 * log_3 0.5 + 0.5 * log_3 0.5 + 0) = 0.63$. In the validation sample, this measure showed a good balance between precision and recall with an overall accuracy of 81%. Finally, there is also support for this argument from the TOKEN SPLIT measure, although its precision and recall was low. It has a high AUC of 86%, and the error probability is 19% beyond the 7.5 threshold. There was no true-positive in the insurance sample as the two models beyond this value did not have errors. Still, as all control flow errors relate to concurrency and synchronization, it appears to be a good strategy for minimizing it.

Rule **G6** refers to activity labels, which were not considered in this paper. Guideline **G7** can be refined based on the threshold found for NODES: a model should already be decomposed once it has more than 31 nodes. As a re-

Table 6: Ten Process Modeling Rules

Rule	Associated measure	Explanation
G1	Nodes	Do not use more than 31.
G2	Conn. Degree	No more than 3 inputs or outputs per connector.
G3	Start and End	Use no more than 2 start and end events.
G4.a	Structuredness	Model as structured as possible.
G4.b	Mismatch	Use design patterns to avoid mismatch.
G5.a	OR-connectors	Avoid OR-joins and OR-splits.
G5.b	Heterogeneity	Minimize the heterogeneity of connector types.
G5.c	Token Split	Minimize the level of concurrency.
G6	Text	Use verb-object activity labels.
G7	Nodes	Decompose a model with more than 31 nodes.

sult of these discussions, we give a summary of a refined list of ten process modeling rules in Table 6.

5.2. Threats to Validity

With regards to internal validity, there can be several challenges to determining appropriate threshold values. Often, not all factors can be controlled [16]. In our case, we did not have access to information on the modeling expertise of those who created the models of our sample. Accordingly, we associated the threshold values with a quantitative risk assessment [10]. It is also problematic that many statistical techniques require a significant set of input parameters to be set, which can sometimes lead to unrealistic values. The technique used in this paper, namely ROC curves with AUC, has many advantages in this regard. It is objective in the sense that a user does not need to set any parameter value. However, it also has some weaknesses. For example, one particular curve may have a larger AUC (that is apparently better) even though the alternative may show superior performance over almost the entire range of values of the classifications threshold. That said, however, it has an intuitive interpretation as a strength: *it is the average sensitivity of a classifier under the assumption that one is equally likely to choose any value of the specificity, under the assumption of a uniform distribution over specificity [29].*

In relation to external validity, we focus on verification support, navigation aids, and modeling expertise. All the models that we included in this study were created using modeling tools that do not provide explicit support for checking soundness and conducting other kinds of verification. This means that the thresholds reflect the situation where a modeler tries to create correct models without any specific tool support for it. Once efficient verification and correction aids such as [22][25] become available in modeling tools, we can assume that the thresholds are less relevant for predicting errors. It is even more likely that, they will indicate the potential rework effort that is required to yield a correct model.

Finally, even for tools without verification support there are different levels of navigation support. The importance of highlighting and layout is emphasized in various studies [57, 49]. The models we used were laid out nicely. In case, modeling tools do not support or enforce effective layout, the likelihood of encountering errors might be higher than the thresholds suggest. Indeed, most professional modeling tools include algorithms for auto-layout and help of positioning elements correctly right from the start.

We mentioned above that process modeling expertise is an important factor for model understanding [48, 50]. In the same vein, the thresholds obtained in this paper are dependent upon the expertise of the modelers who created the model collections of this study. While we do not have access to the demographics of the modelers involved, we have some evidence that the models have been created at least partially, by modeling experts. Both the SAP Reference Model and the models from consultancies were created to sell the models. The models of the German service provider were created with academic supervision. Hence, we may assume that the quality of the models used in this study reflect industrial standards.

6. Conclusions

In this paper a set of thresholds for business process model measures have been proposed in order to predict errors. A dataset composed of 2,003 EPC models was used to extract the thresholds systematically. Furthermore, we used a collection of 429 EPC models of an Australian financial institution for validation. As a result some meaningful thresholds were obtained for evaluating aspects that included size, connection, modularity, and connector interplay of the models. For complex behavior (`cyclicity` and `token split`) it was not possible to obtain reliable thresholds, as they yielded low precision and recall values. The obtained thresholds were applied to find quantitative support to the seven process modeling guidelines.

Mapping modeling guidelines with threshold values makes it possible to support the decision making process by suggesting improvement actions for modelers. The research presented in this paper can thus also serve as a guide to other researchers and practitioners so they can build indicators (measures with decision criteria) from validated measures. Thresholds for error prediction in EPC models can also serve as a starting point for application in practice, and they should be continuously gauged by companies according to feedback obtained from the practical experience derived from its usage. In future work, we also aim to

investigate the potential of improving precision. Current precision values may be too low for some scenarios like workflow design. In particular, we want to look into discriminant analysis, a statistical tool that might help defining indicators based on several measures. Future work will aim to further validate the thresholds by using other collections of process models from different business areas, complementing the thresholds obtained with expert opinions, and applying the approach to other business process modeling notations. In future work, we also aim to investigate the potential of improving precision. Current precision values may be too low for some scenarios like workflow design. In particular, we want to look into discriminant analysis, a statistical tool that might help defining indicators based on several measures.

Acknowledgements
This work was partially funded by the following projects: PEGASO/MAGO (Ministerio de Ciencia e Innovacin MICINN and Fondo Europeo de Desarrollo Regional FEDER, TIN2009-13718-C02-01); ALTAMIRA (Junta de Comunidades de Castilla-La Mancha, Fondo Social Europeo, PII2I09-0106-2463); ESFINGE (Ministerio de Educacin y Ciencia, Direccin General de Investigacin/Fondos Europeos de Desarrollo Regional (FEDER), TIN2006-15175-C05-05); INGENIOSO (Junta de Comunidades de Castilla La Mancha, PEII11-0025-9533) and the ARC Linkage Project "Facilitating Business Process Standardisation and Reuse" (LP110100252). NICTA is funded by the Australian Government as represented by the Department of Broadband, Communications and the Digital Economy and the Australian Research Council through the ICT Centre of Excellence program.

References

[1] W.M.P. van der Aalst, A. Hirnschall, and H.M.W. Verbeek. An Alternative Way to Analyze Workflow Graphs. In A. Banks-Pidduck, J. Mylopoulos, C.C. Woo, and M.T. Ozsu, editors, *Proceedings of the 14th International Conference on Advanced Information Systems Engineering (CAiSE'02)*, volume 2348, pages 535–552, 2002.

[2] W.M.P. van der Aalst, A.H.M. ter Hofstede, B. Kiepuszewski, and A.P. Barros. Workflow Patterns. *Distributed and Parallel Databases*, 14(1):5–51, July 2003.

[3] W.M.P. van der Aalst and K.B. Lassen. Translating unstructured workflow processes to readable BPEL: Theory and implementation. *Information and Software Technology*, 50(3):131–159, 2008.

[4] R. Agarwal, P. De, and AP Sinha. Comprehending object and process models: An empirical study. *IEEE Transactions on Software Engineering*, 25(4):541–556, 1999.

[5] E. Rolón Aguilar, F. García, F. Ruiz, and M. Piattini. An exploratory experiment to validate measures for business process models. In *First International Conference on Research Challenges in Information Science (RCIS)*, 2007.

[6] R.A. Baeza-Yates and B.A. Ribeiro-Neto. *Modern Information Retrieval*. ACM Press / Addison-Wesley, 1999.

[7] V.R. Basili and B.T. Perricone. Software errors and complexity: an empirical investigation. *Communications of the ACM*, 27(1):42–52, 1984.

[8] V.R. Basili and H. Dieter Rombach. The TAME project: Towards improvement-oriented software environments. *IEEE Transactions on Software Engineering*, 14(6):758–773, 1988.

[9] J. Becker, M. Rosemann, and C. Uthmann. Guidelines of business process modeling. In *Business Process Management, Models, Techniques, and Empirical Studies*, pages 30–49, London, UK, 2000. Springer-Verlag.

[10] R. Bender. Quantitative risk assessment in epidemiological studies investigating threshold effects. *Biometrical Journal*, 41(3):305–319, 1999.

[11] S. Benlarbi, K. Emam, N. Goel, and S. Rai. Thresholds for object-oriented measures. In *Proceedings of the 11th International Symposium on Software Reliability Engineering*, pages 24–, Washington, DC, USA, 2000. IEEE Computer Society.

[12] B.W. Boehm. *Software Engineering Economics*. Prentice-Hall, Englewood Cliffs, 1981.

[13] G. Canfora, F. García, M. Piattini, F. Ruiz, and C.A. Visaggio. A family of experiments to validate metrics for software process models. *Journal of Systems and Software*, 77(2):113–129, 2005.

[14] J. Cardoso. Process control-flow complexity metric: An empirical validation. In *Proceedings of IEEE International Conference on Services Computing (IEEE SCC 06), Chicago, USA, September 18-22*, pages 167–173. IEEE Computer Society, 2006.

[15] S.R. Chidamber and C.F. Kemerer. A metrics suite for object oriented design. *IEEE Transaction on Software Engineering*, 20(6):476–493, 1994.

[16] G.A. Churchill and R.W. Doerge. Empirical threshold values for quantitative trait mapping. *Genetics Society of America*, pages 305–319, 1995.

[17] I. Davies, P. Green, M. Rosemann, M. Indulska, and S. Gallo. How do practitioners use conceptual modeling in practice? *Data & Knowledge Engineering*, 58(3):358–380, 2006.

[18] G. Decker and J. Mendling. Process instantiation. *Data Knowl. Eng.*, 68(9):777–792, 2009.

[19] T. DeMarco. *Controlling Software Projects*. Yourdon Press, New York, 1982.

[20] K. Erni and C. Lewerentz. Applying design-metrics to object-oriented frameworks. In *Proc. of the Third International Software Metrics Symposium*, pages 25–26. Society Press, 1996.

[21] J. Esparza. Reachability in live and safe free-choice petri nets is np-complete. *Theor. Comput. Sci.*, 198(1-2):211–224, 1998.

[22] D. Fahland, C. Favre, J. Koehler, N. Lohmann, H. Völzer, and K. Wolf. Analysis on demand: Instantaneous soundness checking of industrial business process models. *Data Knowl. Eng.*, 70(5):448–466, 2011.

[23] N.E. Fenton and S.L. Pfleeger. *Software Metrics. A Rigorous and Practical Approach*. PWS, Boston, 2nd edition, 1997.

[24] A. Fouad, K. Kanyaru, J. Mathenge, and J. Sheridan. Embedding requirements within model-driven architecture. *Software Quality Control*, 19:411–430, June 2011.

[25] M. Gambini, M. La Rosa, S. Migliorini, and A.H.M. ter Hofstede. Automatic error correction of business process models. In *BPM*. Springer, 2011.

[26] T.R.G. Green. Conditional program statements and their comprehensibility to professional programmers. *Journal of Occupational Psychology*, 50:93–109, 1977.

[27] T.R.G. Green and M. Petre. Usability Analysis of Visual Programming Environments: A Cognitive Dimensions Framework. *Journal of Visual Languages and Computing*, 7(2):131–174, 1996.

[28] J. Hahn and J. Kim. Why are some diagrams easier to work with? effects of diagrammatic representation on the cognitive integration process of systems analysis and design. *ACMTCHI: ACM Transactions on Computer-Human Interaction*, 6, 1999.

[29] D. Hand. Measuring classifier performance: a coherent alternative to the area under the roc curve. *Machine Learning*, 77(1):103–123, October 2009.

[30] J.A. Hanley and B.J. McNeil. The meaning and use of the area under the receiver operating characteristic (roc) curve. *Radiology*, 143:29–36, 1982.

[31] D. Hosmer and S. Lemeshow. *Applied logistic regression (Wiley*

Series in probability and statistics). Wiley-Interscience Publication, September 2000.

[32] G. Keller, M. Nüttgens, and A.-W. Scheer. Semantische Prozessmodellierung auf der Grundlage "Ereignisgesteuerter Prozessketten (EPK)". Heft 89, Institut für Wirtschaftsinformatik, Saarbrücken, Germany, 1992.

[33] G. Keller and T. Teufel. *SAP(R) R/3 Process Oriented Implementation: Iterative Process Prototyping*. Addison-Wesley, 1998.

[34] E. Kindler. On the semantics of EPCs: Resolving the vicious circle. *Data & Knowledge Engineering*, 56(1):23–40, 2006.

[35] J. Krogstie, G. Sindre, and H. Jorgensen. Process models representing knowledge for action: a revised quality framework. *Eur. J. Inf. Syst.*, 15(1):91–102, February 2006.

[36] G.S. Lee and J.M. Yoon. An empirical study on the complexity metrics of petri nets. *Microelectronics and Reliability*, 32(3):323–329, 1992.

[37] S.J. Mason and N.E. Graham. Areas beneath the relative operating characteristics (roc) and relative operating levels (rol) curves: Statistical significance and interpretation. *Quarterly Journal of the Royal Meteorological Society*, 128(584):2145–2166, 2002.

[38] T.J. McCabe. A complexity measure. *IEEE Transactions on Software Engineering*, 2(4):308–320, 1976.

[39] J. Mendling. *Metrics for Process Models: Empirical Foundations of Verification, Error Prediction, and Guidelines for Correctness*, volume 6 of *Lecture Notes in Business Information Processing*. Springer, 2008.

[40] J. Mendling. Empirical studies in process model verification. *LNCS Transactions on Petri Nets and Other Models of Concurrency II, Special Issue on Concurrency in Process-Aware Information Systems*, 2:208–224, 2009.

[41] J. Mendling, G. Neumann, and W.M.P. van der Aalst. Understanding the occurrence of errors in process models based on metrics. In R. Meersman and Z. Tari, editors, *OTM Conference 2007, Proceedings, Part I*, volume 4803 of *Lecture Notes in Computer Science*, pages 113–130. Springer, 2007.

[42] J. Mendling, H. A. Reijers, and W. M. P. van der Aalst. Seven process modeling guidelines (7pmg). *Inf. Softw. Technol.*, 52:127–136, February 2010.

[43] J. Mendling, H.M.W. Verbeek, B.F. van Dongen, W.M.P. van der Aalst, and G. Neumann. Detection and Prediction of Errors in EPCs of the SAP Reference Model. *Data & Knowledge Engineering*, 64(1):312–329, January 2008.

[44] D.L. Moody. Theoretical and practical issues in evaluating the quality of conceptual models: current state and future directions. *Data & Knowledge Engineering*, 55(3):243–276, 2005.

[45] S. Morasca. Measuring attributes of concurrent software specifications in petri nets. In *METRICS '99: Proceedings of the 6th International Symposium on Software Metrics*, pages 100–110, Washington, DC, USA, 1999. IEEE Computer Society.

[46] S. Morasca and G. Ruhe. Introduction: Knowledge discovery from empirical software engineering data. 9(5), 1999.

[47] M.E. Nissen. Redesigning reengineering through measurement-driven inference. *MIS Quarterly*, 22(4):509–534, 1998.

[48] J. Recker and A. Dreiling. Does it matter which process modelling language we teach or use? an experimental study on understanding process modelling languages without formal education. In *18th Australasian Conference on Information Systems*, pages 356–366, 2007.

[49] H. Reijers, T. Freytag, J. Mendling, and A. Eckleder. Syntax highlighting in business process models. *Decision Support Systems*, 51(3):339–349, 2011.

[50] H. Reijers and J. Mendling. A study into the factors that influence the understandability of business process models. *IEEE Transactions on Systems, Man, and Cybernetics, Part A*, 41(3):449–462, 2011.

[51] R. Shatnawi. An investigation of ck metrics thresholds. *ISSRE Supplementary Conference Proceedings*, 2006.

[52] R. Shatnawi, W. Li, J. Swain, and T. Newman. Finding software metrics threshold values using roc curves. *Journal of Software Maintenance and Evolution: Research and Practice*, 22(1):1–16, 2010.

[53] J. Sweller. Implications of cognitive load theory for multimedia learning. *The Cambridge handbook of multimedia learning*, pages 19–30, 2005.

[54] I. Vanderfeesten, H.A. Reijers, J. Mendling, W.M.P. Aalst, and J. Cardoso. On a quest for good process models: The cross-connectivity metric. pages 480–494, 2008.

[55] J. Vanhatalo, H. Völzer, and F. Leymann. Faster and more focused control-flow analysis for business process models through sese decomposition. In B.J. Krämer, K.-J. Lin, and P. Narasimhan, editors, *Service-Oriented Computing - ICSOC 2007, Fifth International Conference, Vienna, Austria, September 17-20, 2007, Proceedings*, volume 4749 of *Lecture Notes in Computer Science*, pages 43–55. Springer, 2007.

[56] I. Vessey. Cognitive Fit: A Theory-Based Analysis of the Graphs Versus Tables Literature*. *Decision Sciences*, 22(2):219–240, 1991.

[57] C. Ware, H. Purchase, L. Colpoys, and M. McGill. Cognitive measurements of graph aesthetics. *Information Visualization*, 1(2):103–110, 2002.

[58] P. Wohed, W.M.P. van der Aalst, M. Dumas, A.H.M. ter Hofstede, and N. Russell. On the Suitability of BPMN for Business Process Modelling. In S. Dustdar, J.L. Fiadeiro, and A. Sheth, editors, *Business Process Management, 4th International Conference, BPM 2006*, volume 4102 of *Lecture Notes in Computer Science*, pages 161–176. Springer-Verlag, September 2006.

[59] M. Zweig and G. Campbell. *Clinical Chemistry*, 39(4):561–577, 1993.