Quality Indicators for Business Process Models from a Gateway Complexity Perspective

Laura Sánchez-González[1], Félix García[1], Francisco Ruiz[1], Jan Mendling[2]
[1] *Instituto de Tecnologías y Sistemas de Información, University of Castilla La Mancha, Paseo de la Universidad, nº4, 13071,Ciudad Real, España*
{Laura.Sanchez | Felix.Garcia| Francisco.RuizG}@uclm.es
[2]*WU Vienna, Augasse 2-6, 1090 Vienna, Austria, jan.mendling@wu.ac.at*

Abstract.

Context: Quality assurance of business process models has been recognized as an important factor for modeling success at an enterprise level. Since quality of models might be subject to different interpretations, it should be addressed in the most objective way, by the application of measures. That said, however, assessment of measurement results is not a straightforward task: it requires the identification of relevant threshold values, which are able to distinguish different levels of process model quality.
Objective: Since there is no consensual technique for obtaining these values, this paper proposes the definition of thresholds for gateway complexity measures based on the application of statistical techniques on empirical data.
Method: To this end, we conducted a controlled experiment that evaluates quality characteristics of understandability and modifiability of process models in two different runs. The thresholds obtained were validated in a replication of the experiment.
Results: The thresholds for gateway complexity measures are instrumental as guidelines for novice modelers. A tool for supporting business process model measurement and improvement is described, based on the automatic application of measurement, and assessment as well as derivation of advice about how to improve the quality of the model.
Conclusion: It is concluded that thresholds classified business process models in the specific level of understandability and modifiability, so these thresholds were good and useful for decision-making.

Keywords: Business process model, measure, threshold, indicator.

1 Introduction

In an organizational setting, measurement of business process (BP) models plays an important role in obtaining useful information on the direction of potential improvement [39]. Designing business process models in a way that is readily understandable is a prerequisite for the leveraging of the benefits of process improvement, as well as being crucial in the corresponding design of information

systems. Process models that are difficult to understand often contain errors such as deadlocks [30]. Good process model design can help to avoid errors right from the start. This is critical, since the propagation of errors to later stages implies exponentially-growing rework costs and efforts [52]: post-implementation errors cost more than 100 times as much as errors produced during the design stage [7].

Measurement of structural properties can be used to indicate that a model is likely to be understood well, or that it is potentially prone to errors. Complexity of business processes can be faced from different perspectives, because they are a high-level notion, made up of many different elements (splits, joins, resources, data, etc) [14]. To support business process model evaluation, several structural measures have been published to date [47], most of them focused on evaluating size or complexity. These measures can be used to evaluate static properties of process models, or, in other words, their internal quality. Typically, the significance of structural measures such as internal quality factors relies on a thorough empirical validation of their correlation with external quality attributes [56], which focus on the user perceptions [26]. These considerations stem from Software Engineering research and can be applied to business process modeling, owing to the similarities between BPs and Software [38].

Up until now, however, there are no empirically-validated thresholds available that facilitate the decision-making process related to the quality of business process models. Henderson-Sellers emphasizes their practical utility in the software engineering field by stating that "an alarm would occur whenever the value of a specific internal measure exceeded some predetermined value" [23]. Having such thresholds available is therefore highly desirable.

In this paper we focus on gateway complexity, because gateways are of central importance to the correctness of process models. As stated by Cant et al. "a conditional control structure is more complex to understand than a normal sequential section of code" [10], so it can be expected that gateway complexity greatly influences the overall complexity. Gateways define routing constraints using decision nodes, parallel execution or synchronization of the control flow. The structural measures considered in this paper related to gateway complexity are: Control-Flow Complexity (CFC), Gateway Mismatch (GM), Gateway Heterogeneity (GH), Average Gateway Degree (AGD), Maximum Gateway Degree (MGD) and Total Number of Gateways, (TNG). With regards to external quality, this paper concentrates on two of the most relevant characteristics: understandability and modifiability. The selection of understandability and modifiability is based on the importance of maintaining a good level of these characteristics in business process models, in order to adapt to the continuous changes needed to meet user requirements. A lack of understanding of process models affects modifiability tasks, which in turn directly affects the maintainability of process models [25]. Furthermore, since a process model in the design stage is concerned with requirement documentation and communication [17], these quality characteristics become very important in achieving successful implementation. It has been demonstrated in previous work that gateway measures are correlated with understandability and modifiability [46].

Based on the issues outlined above, the research question is to ask if it is possible to automatically distinguish between understandable/modifiable models and non-understandable/non-modifiable ones, by measurement of structural properties related

to gateway complexity. This article provides a twofold contribution in the context described in the preceding paragraphs.

- We identify threshold values for the set of structural measures selected. To this end, we conducted a controlled experiment involving understanding and modification tasks, the results of which are validated in a replication. As a complementary mechanism to facilitate the decision making process, we also propose the Gateway Complexity Indicator (GCI), which is defined based on the threshold values that were previously identified for the chosen gateway complexity measures.
- We present a tool for automatic support of the measurement assessment, as a proof-of-concept implementation. This tool applies a group of measures (gateway complexity measures and others) on business process models and highlights whether measures remain in the desirable range or not. If a threshold value is exceeded, the tool gives recommendations about what parts or elements of the business process model should be redesigned.

The remainder of this article is organized as follows. Section 2 provides an introduction to gateway complexity, along with techniques for threshold extraction found in literature. Section 3 explains the planning and operation of the experiment. In section 4, there is a description of how the thresholds were obtained by the application of statistical techniques on experimental data, and then the GCI indicator is defined. In Section 5, we validate our results using an experiment replication, by calculating precision, recall and accuracy measures and ROC (receiver operating characteristic) curves. Section 6 discusses the findings of this work, as well as its implications as regards the definition of modeling guidelines and recommendations. In Section 7, the tool for measurement assessment is described, with a highlight on its practical utility. Section 8 provides conclusions of this piece of research, as well as an outlook on future research.

2 Background

This section discusses the background to our research. Firstly, we introduce the evaluation of business process models quality, with a focus on gateways. Secondly, we summarize different techniques for extracting thresholds, as found in literature.

2.1 Gateway complexity of business process models

The control flow of business process models is defined using gateways, essentially. The complexity of a gateway relates to its type (XOR, AND, and OR) and to whether it is a split or a join. Splits define the decision points and points where concurrent execution is triggered, while joins specify the conditions of how paths are merged or synchronized [12]. The key idea behind the gateway complexity is to evaluate the number of mental states that have to be considered when a designer is modeling a process. Miller [34] highlighted the importance of mental states, since the mind's capacity to analyze a model would be reduced when the number of mental states becomes too big, resulting in more errors when modeling.

Split nodes are the main elements to be considered in gateway complexity. There are a number of specific aspects that are taken into account by different measures.

These include the number of inputs and outputs, the mismatch between splits and joins and the number of decision nodes of different types in a model. That being the case, the set of measures selected in this research is the following:

- Control-flow complexity (CFC) was defined by Cardoso [13] to measure the complexity of split gateways based on the number of mental states that have to be taken into account when a designer models a process.
- Gateway mismatch (GM) was defined by Mendling [30] as the sum of gateway pairs that do not match with each other, for example, when an AND-split is followed by an OR-join.
- Gateway heterogeneity (GH) was defined by Mendling [30] to quantify the frequency of different types of gateways used in a model.
- Average gateway degree (AGD) is a measure [30] to express the average number of both incoming and outgoing arcs of the gateway nodes in the model.
- Maximum gateway degree (MGD) [30] is the maximum number of incoming and outgoing arcs of a decision node in the model.
- Total Number of gateways (TNG) defined by Rolón et al. [43] is the number of decision nodes in the model.

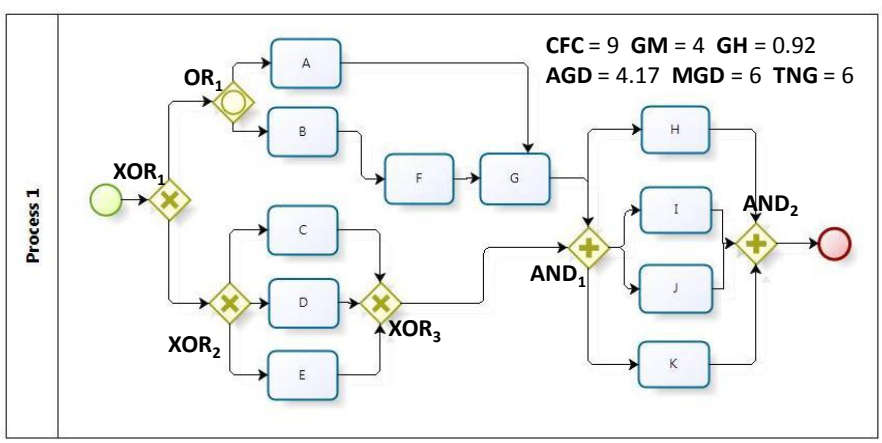


Fig. 1. Example of calculation of measures

Figure 1 shows an example of a business process model expressed in BPMN [37] and includes the value of measures in the upper-right corner. The calculation of these measures is specified in equations (1), (2), (3), (4) and (5). For the measure TNG, there is no need for an explicit equation, since it is a base measure. It is determined by counting the number of decision nodes in the model.

$$\boldsymbol{CFC(P)} = \sum_{i \in [AND-splits\ of\ P]} CFC_{AND-split}\ (i) + \sum_{j \in [XOR-splits\ of\ P]} CFC_{XOR-split}\ (j) + \sum_{k \in [OR-splits\ of\ P]} CFC_{OR-split}\ (k) = 1 + fan-out(j) + 2^{fan-out(k)} - 1 = 1 + (3 + 2) + 3 = 9 \quad \textbf{(1)}$$

$\boldsymbol{GM(P)} = GM_{XOR} + GM_{OR} + GM_{AND} = 2 + 2 + 0 = 4$ *in which* $GM_l = \left|\sum_{c \in S_l} d(c) - \sum_{c \in J_l} d(c)\right|$ *where l is the decision node type, S and J means splits and joins and d is the degree of the decision node* **(2)**

$\boldsymbol{GH(P)} = -\sum_{t \,\epsilon\, \{AND,XOR,OR\}} \frac{Ct}{C} * \log_3 \frac{Ct}{C} = -0.33 - 0.31 - 0.27$ *where Ct is total number of a specific type of decision node, and C total number of decision nodes* **(3)**

$\boldsymbol{AGD(P)} = \frac{1}{C} \sum_{C \,\epsilon\, Decision\ Nodes\ of\ P} d(c) = \frac{1}{6}\,(3 + 4 + 4 + 3 + 6 + 5) = 4.17$
where d(c) is degree of each decision node, which means number of input/outputs **(4)**

$\boldsymbol{MGD(P)} = MAX d(c)_{c \epsilon decision\ nodes\ of\ P} = 6$ **(5)**

2.2 Related work on thresholds for business process measures

The extraction of thresholds is no trivial task. It requires a theory and practical base and it should meet certain requirements. It ought [1]: a) not to be based on expert opinion but on measurement data; b) to respect the statistical properties of the measure, such as metric scale and distribution and to be resilient against outlier values; and c) to be repeatable, transparent and easy to carry out. Some authors have defined thresholds based on experience. A typical example is the value 10 for the McCabe's cyclomatic complexity measure [29]. This value relies on experience and sometimes it is difficult to defend in an objective manner. This problem of objectivity has led to different proposals which all aim to provide a theoretical foundation. The methods being used typically build on statistics such as mean and standard deviation or ROC curves. However, these statistical techniques have some weaknesses, as is summarized in Table 1.

Erni and Lewerentz [18] define a maximum and minimum threshold, based on calculations with the mean and standard deviation of data. They calculate the minimum threshold by subtracting the standard deviation of the mean and the maximum threshold is calculated by adding it. However, a requirement prior to obtaining valid thresholds with this method is that the measures analyzed have to follow a normal distribution, which is considered as an important limitation. French [21] also proposes a technique for threshold extracting, based on mean and standard deviation, but using in addition the Chebyshev's inequality theorem, in order to avoid the normality restriction. The main limitation of this methodology is that for measures with high range or high variation, it will identify a smaller percentage of observations than its theoretical maximum. Other authors, Shatnawi [49] and Sanchez-Gonzalez et al.[46, 48], use the Bender method [5], which comes from epidemiological studies and is based upon the logistic regression model, which requires a binary variable. After obtaining the logistic regression equations, the method defines a Value of an Acceptable Risk Level (VARL), which is given by a suggested probability p_0 and is calculated using the logistic regression coefficients. For measure values over VARL, the risk of a poor-quality of models is higher than the p_0. However, the need for a binary variable is a limitation, as is the definition of p_0, which is defined by the engineer arbitrarily.

Author	Measure	Technique	Weaknesses
Erni and Lewerentz [18]	Software measures: Class and method complexity, coupling, cohesion	Mean and standard deviation	Data must follow a normal distribution
French [21]	Software measures	Mean and standard deviation but using additionally the Chebyshev's inequality theorem	This methodology is sensitive to a large number of outliers
Shatnawi [49]	Chidamber and Kemerer measures	Bender method [5]	A binary variable is needed and the definition of p_0 variable is arbitrary
Sanchez-Gonzalez et al.[46, 48]	Business Process measures	Bender method [5]	
Shatnawi [50]	Chidamber and Kemerer measures	ROC curves	This methodology does not succeed in deriving monotonic thresholds
Catal et al.[15]	Software measures	ROC curves	The maximum value of sensitivity/specificity sometimes does not exist
Benlarbi [6]	Chidamber and Kemerer	Linear regression	There is no empirical evidence supporting the model
Yoon et al. [55]	Software measures	k-means cluster algorithm	It requires an input parameter that affects both the performance and the accuracy of the results
Herbold et al. [24]	Control flow structuredness, coupling, size, method complexity, inheritance, staticness	Machine learning based method	The methodology produces only a binary classification
Ferreira et al [20]	Object-oriented software measures	The values found most commonly in practice	The effectiveness of the method depends on the sample size
Rosenberg [44]	Object-oriented software measures	Histogram analysis	There is no clear evidence of how these values are associated with error-probability.

Table 1 Different threshold initiatives found in literature

Other techniques are adapted to threshold extraction. That is the case of ROC curves, typically used for making decisions about diagnostics in radiology to distinguish between healthy and ill subjects in clinical medicine. Some authors used ROC curves for threshold extraction, such as Shatnawi [50] and Catal et al.[15]. To plot a ROC curve, a binary and ordinal categorization is needed. The curve is constructed with pair values of *sensitivity*, *1- specificity* and we have to select the pair

which minimizes false-positives and false-negatives at the same time. The main limitation of this technique is also the required binary variable.

Benlarbi [6] investigates the relation of measure threshold and software failures for a set of measures using a linear regression. Two error probability models are compared, with and without threshold. For the model with threshold, measure values below the threshold obtain a zero probability of error. However, no empirical evidence supports the model with threshold. On the other hand, Yoon et al. [55] used the K-means cluster algorithm to identify thresholds. Threshold values can be identified by observations that appear either in isolated clusters or far away from other observations within the same cluster. However, the algorithm requires a parameter that affects the performance and the accuracy of the results. The identification of thresholds, it must also be remembered, is manual; in general, the input parameters influence the final results.

Herbold et al. [24] used a machine learning algorithm to define an approach for the calculation of the threshold. This utilizes the learning of axis-aligned d-dimensional rectangles for the calculation. This methodology, however, produces a binary classification and can therefore only differentiate between good and bad; further shades of gray are not possible. Ferreira et al [20] proposed the extraction of thresholds derived by analyzing the statistical properties of the data obtained from a large collection of open-source programs developed in Java, thereby identifying the values most commonly used in practice. However, the effectiveness of the technique depends directly on the sample size. Finally, Rosenberg [44] uses a simple histogram to demonstrate prevailing and extreme values, but there is no clear evidence of how these values are associated with error-probability.

The research presented in this paper focuses on understandability and modifiability of business process models. All the methods discussed, which are summarized in table 1, require the dependent variable to be binary. This is a major problem for defining thresholds of process model understanding, which is typically measured on an ordinal or interval level. Moreover, most of the analyzed measures which were applied on software or another field, such as business process models, have no threshold definition. In this work, we avoid the loss of information by applying a new technique for threshold extraction based on ANOVA tests. We use some of the techniques mentioned (Bender method [5] and ROC curves) for secondary purposes, as validation of extracted thresholds. This is detailed in the following sections.

3 Experimental settings

The empirical definition of thresholds requires the availability of data. We therefore carried out a controlled experiment on understandability and modifiability, in order to obtain threshold values for gateway complexity measures. Using the GQM template for goal definition [3], the experiment goal can be defined as follows:
"**Analyzing** business process models **for the purpose of** evaluating process model understanding and modification **with respect to** their gateway complexity from **the point of view of** process model designers **in the context of** undergraduate computer engineering students".

An overview of the experiment design is presented in Figure 2 and the description of it (based on [54]) it is set out in detail below.

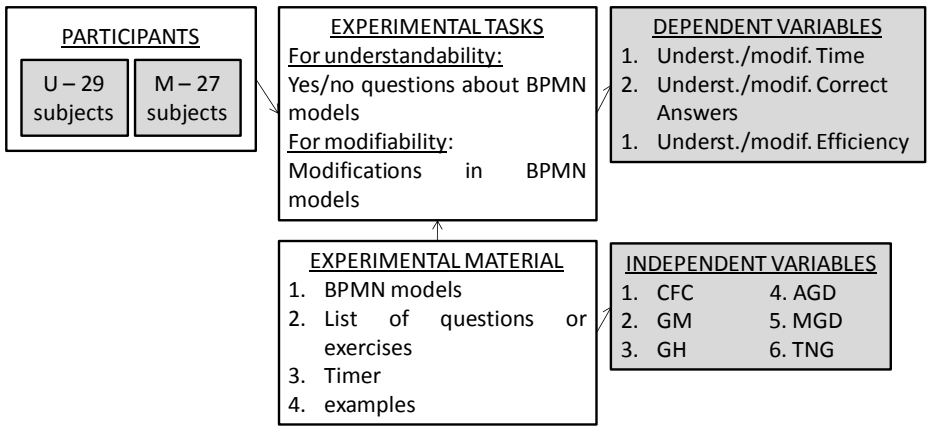


Fig. 2. Overview of the experimental design

a) Planning

The planning was carried out according to the following steps. The experiment addresses a real problem when analyzing gateway complexity of business process models, comparing measurement results with obtained thresholds, and discovering what parts of models should be redesigned. The experimental material is composed of models without realistic label names, however. This is an important requirement for neutralizing a potential effect of varying domain knowledge among subjects [41]. The **selection of subjects** has been done on the basis of convenience. In the experiment, the subjects were students of the 4th year of Computer Science degree at University of Castilla La Mancha (Spain). They had knowledge of process modeling but they did not know the specific model notation, BPMN. This being so, a seminar about the notation was carried out previously and the subjects were trained to perform the experiment successfully. With regard to **variable selection**, the independent variables are the aforementioned structural measures: CFC, GM, GH, AGD, MGD and TNG. The dependent variables are the efficiency of understandability and modifiability which were obtained by calculating four main base measures on experimental data:

- Understandability/Modifiability time: total of seconds needed for finishing the whole questions in a specific model.
- Understandability/Modifiability correct answers: total number of correct answers. For understandability, each model has four yes/no questions about the model, and for modifiability, three exercises that consist in the implementation of some modifications on the model.
- Understandability/Modifiability efficiency = correct answers / time.
- Personal assessment of complexity: a subjective answer about how difficult facing the exercises in each model was. Subjects could indicate between 1 (very easy) and 5 (very difficult).

Regarding the **instrumentation** step of the experiment planning stage, the objects were 10 BPMN models, a number of models chosen by us in an effort to avoid fatigue effects. An excerpt of the experimental material is included in Appendix A. To select

a representative subset, for each model we considered different values of gateway complexity measures. Measure values are the lowest for the first model and are highest for the tenth model and they vary (see Table 2) in small increments across the ten models. The models were therefore specifically designed for this experiment, in order to include enough variability in the measure values. Model 1 has the lowest values of the measures and Model 10 the highest. To avoid potential bias from different levels of complexity that may come about as a result of varying lengths of labels in the model elements, abstract labels were used.

Model	CFC	GM	GH	AGD	MGD	TNG
1	8	2	0	3	3	7
2	13	6	0,62	3,67	4	9
3	22	6	0,79	3,83	5	12
4	24	12	0,84	3,85	6	13
5	30	14	0,86	3,86	6	14
6	31	15	0,86	3,88	7	16
7	37	15	0,92	4,06	7	18
8	44	18	0,92	4,16	8	19
9	51	20	0,94	4,18	9	22
10	63	25	0,94	4,22	9	23
Mean	32.3	13.3	0.76	3.87	6.4	15.3
Std. deviation	16.98	7.04	0.28	0.35	2.01	5.29

Table 2 Gateway measure values in experimental material

The **hypotheses** of the experiment are the following:

$H_{0.1}$: When asking understandability questions there are no significant differences in efficiency depending on the values of the gateway complexity measures.

Alternative hypothesis, $H_{1.1}$: When asking understandability questions there are significant differences in efficiency depending on the values of the gateway complexity measures.

$H_{0.2}$: When carrying out modifiability exercises there are no significant differences in efficiency depending on the values of the gateway complexity measures.

Alternative hypothesis, $H_{1.2}$: When carrying out modifiability exercises there are significant differences in efficiency depending on the values of the gateway complexity measures.

The hypotheses have been stated according to the statistical test which will be applied (ANOVA). For $H_{0.1}$, in particular, we would like to check that, the higher the gateway complexity measure values are, the lower the efficiency of understandability will be; or, for $H_{0.2}$, the higher the gateway complexity measure values are, the lower the efficiency of modifiability will be.

In the **experimental design** step, we had 6 independent variables, so it might be thought that a factorial design should be used. That would have been impractical, however, because we would need n^6 different cases (n being the levels we decided for the independent variables). This led us to decide to consider only ten models and to try to cover a wide range of measure values. These models were not based on real

cases; activities are labeled with abstract names in order to avoid any specific domain difficulty, which is not taken into account in this experiment. For each model, there were four questions about understandability and three exercises about modifiability, with a similar complexity level. Separate runs were conducted to tackle understandability and modifiability tasks separately, the aim being to mitigate fatigue and learning effects. The order of the models was different and random for each subject, to avoid learning effects. Figure 3 summarizes the different runs. Figure A.1 and Figure A.2 show an extraction of the experimental material in the appendix.

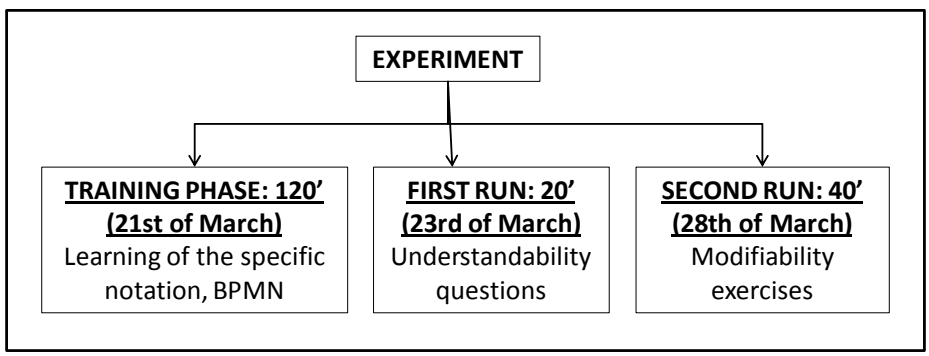


Fig. 3. Distribution of the experimental tasks of the experiment carried out in Spain

b) Operation

The operational phase was divided into three steps: preparation, execution and data validation. The **preparation step** defines the temporal experiment settings. The experiment was done in March 2011. As is shown in Figure 3, this experiment and the training took place on three different days: on the first day, a BPMN tutorial was given and some examples similar to experimental assignments were solved. On the second day, after the subjects had acquired this new knowledge, the understandability part of the experiment was done and the modifiability run was carried out on the third day. In the **execution** step, the subjects were given the 10 models in a random order and we explained how to face the tests. As we had calculated approximately in a pilot experiment, about 20 minutes were needed to finish the understandability part and 40 minutes were required for the modifiability part. Time doing exercises was measured strictly with a chronometer and each subject stopped the time counter when they had a question. Once the data was collected, we checked if the tests had been completed correctly and we discarded the cases with missing answers; this is the **data validation** step. We thus considered 28 of the 29 total subjects for the first understandability run and 25 of 27 for the modifiability run.

4 Results

In this section, experimental data gathered in the experiment are used to obtain thresholds for the gateway complexity measures. First of all, statistical tests were applied and then the results were analyzed. Details of this are given in the next two subsections.

4.1 Hypotheses testing

Firstly, we checked the hypotheses, the main idea being to demonstrate significant differences in efficiency of understandability/modifiability, taking into account the gateway complexity measures. A suitable statistical test for demonstrating differences between value groups is ANOVA [28]. This test proves whether or not the means of several groups are all equal. In our case each model was treated as a group, because there were different measure values for each model, and the purpose of ANOVA test is to detect differences in understandability or modifiability between models. In other words, there were 10 different values (ten variations) for each measure (Table 2), which divides the dataset in 10 models or groups. By using ANOVA we want to find significant jumps in efficiency between two consecutive groups; that is to say, when a small change in measure values means an important change in efficiency values.

To do all this, our method is to compare the understandability/modifiability efficiency between models or "groups", as they were called in ANOVA tests, selecting the groups with significant differences. For example, let us imagine the following situation: we start from the idea that group 1 (model 1) has the lowest measure values and group 10 (model 10) the highest measure values. Between model 1 and model 10 there are 8 other models with intermediate measure values. The ANOVA test between group 1 and group 2 indicate that there is no significant difference in efficiency, which means that the variation of measure values does not produce any significant differences in efficiency. However, the ANOVA test between group 6 and 7 indicates that there is a strong difference in efficiency, which seems to indicate the existence of a threshold between the measure values of groups 6 and 7. Selecting, for example, the CFC measure (CFC is 31 in model 6 and CFC is 37 in model 7), the ANOVA test indicates to us that a threshold value for understandability/modifiability efficiency is between 31 and 37. The description of the algorithm for threshold extraction is depicted in Algorithm 1.

Application of Algorithm 1:
The execution of Algorithm 1 implies checking the normality and homogeneity of variances of the data. The hypotheses for both statistical tests are the following:

H0,3: The dependent variable follows a normal distribution

H0,4: The dependent variable satisfies the homogeneity of variances

The results depicted in Table 3 and 4 indicate that all the groups follow a normal distribution, and that in some situations, the homogeneity of variances is satisfied; in those cases, the ANOVA test is applied. In the application of Algorithm 1 to the data, the six measures are treated as one component because the Principal Components Analysis (PCA) results indicated that all the measures are included in the same component.

Algorithm 1. Threshold calculation

Input:

nm is the number of models.

ns is the number of subjects.

m[] is a vector of the measure values, where each element m[i] corresponds to the model i and it has the following 6-tuple structure:

$m[i]=\{CFC_i, GM_i, GH_i, AGD_i, MGD_i, TNG_i\}$

where $i \in [1,nm]$.

ef[] is a bi-dimensional vector of the efficiency values where each element ef[i,j] corresponds to the model i and to the subject j, with $i \in [1,nm]$ and $j \in [1,ns]$.

Output:

lt is a list with the obtained thresholds, where each element lt[k] corresponds to a set of thresholds in the way of the following 6-tuple structure:

$t[k]=\{tCFC_k, tGM_k, tGH_k, tAGD_k, tMGD_k, tTNG_k\}$

where $k \in [1,nm-1]$. *

* nm−1 is the maximum number of thresholds among nm models

NOTE: thresholds are analyzed for each pair of consecutive models m[i] vs m[i+1].

Algorithm:

```
' Initialization
```

g[] is an intermediate vector, where each element g[i] is to store a full row of ef[] (efficiency values for all the subjects and one model)

```
for i := 1 to nm
      g[i] := {ef[i,1], … , ef[i,ns]}
next i
' Iteration comparing each pair of consecutive models i and i+1
i := 1
while i < nm
   normality1 := kolmogorovSmirnov(g[i])
   normality2 := kolmogorovSmirnov(g[i+1])
   varianceHomog = levene(g[i], g[i+1])
   if (normality1 & normality2 & varianceHomog) then
   difference = ANOVA(g[i], g[i+1])
   if difference then      ' a set of thresholds is detected
   Add m[i+1] to lt;
   end-if
   end-if
   i = i + 1
end-while
```

Normality		g1	g2	g3	g4	g5	g6	g7	g8	g9	g10
U	Z	0.442	0.544	0.743	0.545	0.722	0.432	0.757	0.564	0.989	0.603
	sig	0.990	0.929	0.638	0.928	0.674	0.992	0.615	0.908	0.282	0.860
M	Z	0.602	0.577	0.422	0.741	0.599	0.536	0.564	0.508	0.493	0.522
	sig	0.862	0.893	0.994	0.642	0.866	0.936	0.908	0.959	0.968	0.948

Table 3 Kolmogorov-Smirnov test results for checking the normality of the data

	Understandability				Modifiability			
	Levene's Test		ANOVA test		Levene's Test		ANOVA test	
	Statistic	Sig	F	Sig	Statistic	Sig	F	Sig
G1-G2	**1,47**	**0.233**	**12.81**	**0.001**	7.13	0.010	-	-
G2-G3	**0.05**	**0.82**	**5.688**	**0.021**	**2,68**	**0.108**	**17.67**	**0.000**
G3-G4	**0.006**	**0.941**	2.375	0.129	9.72	0.003	-	-
G4-G5	**0.037**	**0.848**	0.139	0.711	**0.850**	**0.361**	1.757	0.191
G5-G6	**0.428**	**0.516**	2.451	0.123	**0.183**	**0.671**	**8.124**	**0.006**
G6-G7	**0.036**	**0.850**	**6.037**	**0.017**	**0.239**	**0.627**	**11,08**	**0.002**
G7-G8	**0.004**	**0.950**	2.166	0.147	5.17	0.027	-	-
G8-G9	**1,65**	**0.204**	**51.73**	**0.000**	4.68	0.035	-	-
G9-G10	**0.036**	**0.851**	2.525	0.118	**0.03**	**0.862**	-	-

Table 4 Homogeneity of variances and ANOVA test results

According to the principal hypothesis, threshold values for process models measures were extracted and shown in Table 5. Some measures obtained two, three or four significant differences in efficiency between groups, and therefore the number of threshold values is changeable. For example, for the CFC measure 4, different thresholds were obtained, because understandability efficiency was significantly different when the measure was over 13, 22, 37 and 51.

	Understandability	Modifiability
CFC	13, 22, 37, 51	22, 31, 37
GM	6, 15, 20	6, 15
GH	0.62, 0.79, 0.92, 0.94	0.79, 0.86, 0.92
AGD	3.67, 3.83, 4.06, 4.18	3.83, 3.88, 4.06
MGD	4, 5, 7, 9	5, 7
TNG	9, 12, 18, 22	12, 16, 18

Table 5 A first approximation of threshold values for selected measures

In order to interpret the results of Table 5, they have to be completed with the correlation analysis about measures and understandability/modifiability efficiency, and the Spearman test was used. All the correlation results were significant and Spearman rho's values are the following:

- Understandability efficiency and measures CFC, GM, GH, AGD, MGD and TNG have correlation values of (-0.460, -0.452, -0.358, -0.423, -0.447 and -0.458)
- Modifiability efficiency and measures CFC, GM, GH, AGD, MGD and TNG have correlation values of (-0.238, -0.242, -0.260, -0.238, -0.216 and -0.238)

Results show that there is an inverse relationship between measures and understandability/modifiability efficiency, which means that the higher the measure values are, the lower the efficiency is. For example, as is shown in Table 5, when the CFC value is 22, the model's understandability efficiency begins to decline significantly.

4.2 Identification of thresholds

From the results of the experiment we obtained threshold values for structural measures, which constitute alarms of poor-quality models, in terms of understandability and modifiability.

Each threshold measure value has associated linguistic labels for a correct interpretation. Linguistic labels help in the assessment and, based on the limitations of our capability of processing information published, a number of between 3 and 7 labels is advised in [34]. So we associated a linguistic label to each threshold value in each measure, as is shown in Table 6. For example, the measure CFC obtained 4 threshold values for understandability. These four values divide the CFC variable domain in five different groups: from 0 to 13, from 13 to 22, from 22 to 37, from 37 to 51 and from 51 to the infinite. The linguistic labels "fairly low", "low", "Medium", "High" and "Fairly high" to these groups, respectively. Thus, if the measure CFC is 10, a direct evaluation is given as a "fairly low measure value". Unfortunately, not all the measures obtain the same number of thresholds and the number of groups in which is divided the measure domain is different. That is the case, for example, of CFC and modifiability. With three thresholds, 4 groups are created and therefore one linguistic label should be eliminated. As the first threshold for CFC is 22, in contrast to understandability thresholds, we believed that it was more suitable to eliminate the label "fairly low".

In Table 6, thresholds obtained for each measure are also specified. The set of measures has a different number of thresholds, depending on understandability and modifiability experiments and the results of Algorithm 1. For example, regarding the CFC measure, if a model has a CFC value equal to 36, it is considered as a high value, which may compromise the understandability of the model. These threshold values are considered useful because modelers can easily gauge the measure values for the models and use the threshold values to avoid obtaining high-risk designs.

4.3 Definition of the Gateway Complexity Indicator (GCI)

The results presented in the previous section can benefit from having a single indicator to summarize them. The integration of the six measures in an indicator gives us the possibility of obtaining a unified value with a unified assessment about understandability and modifiability. This indicator could be used as a complementary mechanism to that provided by the six measures, thereby giving a global assessment.

Understandability	Modifiability	Linguistic label
Control-Flow Complexity (CFC)		
CFC ≤ 13	-	Fairly low measure value or fairly easy to understand/modify
13 < CFC ≤ 22	CFC ≤ 22	Low measure value or easy to understand/modify
22 < CFC ≤ 37	22 < CFC ≤ 31	Medium measure value or moderately difficult to understand/modify
37 < CFC ≤ 51	31 < CFC ≤ 37	High measure value or difficult to understand/modify
CFC > 51	37 < CFC	Fairly high measure value or fairly difficult to understand/modify
Gateway Mismatch (GM)		
-	-	Fairly low measure value or fairly easy to understand/modify
GM ≤ 6	GM ≤ 6	Low measure value or easy to understand/modify
6 < GM ≤ 15	6 < GM ≤ 15	Medium measure value or moderately difficult to understand/modify
15 < GM ≤ 20	15 ≤ GM	High measure value or difficult to understand/modify
GM > 20	-	Fairly high measure value or fairly difficult to understand/modify
Gateway Heterogeneity (GH)		
GH ≤ 0.62	-	Fairly low measure value or fairly easy to understand/modify
0.62 < GH ≤ 0.79	GH ≤ 0.79	Low measure value or easy to understand/modify
0,79 < GH ≤ 0.92	0.79 < GH ≤ 0.86	Medium measure value or moderately difficult to understand/modify
0.92 < GH ≤ 0.94	0.86 < GH ≤ 0.92	High measure value or difficult to understand/modify
0.94 < GH	0.92 < GH	Fairly high measure value or fairly difficult to understand/modify
Average Gateway Degree (AGD)		
AGD ≤ 3.67	-	Fairly low measure value or fairly easy to understand/modify
3.67 < AGD ≤ 3.83	AGD ≤ 3.83	Low measure value or easy to understand/modify
3.83 < AGD ≤ 4.06	3.83 < AGD ≤ 3.88	Medium measure value or moderately difficult to understand/modify
4.06 < AGD ≤ 4.18	3.88 < AGD ≤ 4.06	High measure value or difficult to understand/modify
4.18 < AGD	4.06 < AGD	Fairly high measure value or fairly difficult to understand/modify
Max. Gateway Degree (MGD)		
MGD ≤ 4	-	Fairly low measure value or fairly easy to understand/modify
4 < MGD ≤ 5	MGD ≤ 5	Low measure value or easy to understand/modify
5 < MGD ≤ 7	5 < MGD ≤ 7	Medium measure value or moderately difficult to understand/modify
7 < MGD ≤ 9	7 < MGD	High measure value or difficult to understand/modify
9 < MGD	-	Fairly high measure value or fairly difficult to understand/modify
Total Number of Gateways (TNG)		
TNG ≤ 9	-	Fairly low measure value or fairly easy to understand/modify
9 < TNG ≤ 12	TNG ≤ 12	Low measure value or easy to understand/modify
12 < TNG ≤ 18	12 < TNG ≤ 16	Medium measure value or moderately difficult to understand/modify
18 < TNG ≤ 22	16 < TNG ≤ 18	High measure value or difficult to understand/modify
22 < TNG	18 < TNG	Fairly high measure value or fairly difficult to understand/modify

Table 6 Threshold values and linguistic labels for gateway complexity measures

However, establishing the relationship between the measures is no small task. The proposed indicator is calculated by a weighted sum of all the measures. Weighted

sums have been used in other disciplines to reduce a multidimensional problem to a uni-dimensional one. The basic idea is to combine several lower-level measures to build a single, upper-level measure that may quantify different aspects of a given attribute at the same time [35]. Firstly, it is important to point out that the six measures selected to build the GCI have been demonstrated to be theoretically valid. CFC was validated in [11] according to Weyuker's [53] properties; GM, GH, AGD, MGD were validated in [30] according to the Zuse method [57]; and finally, TNG measure was validated in [42] according to Briand's theoretical framework [8]. In addition, according to the research of Poels and Dedene [40] and Fenton [19], all the measures have a ratio scale, and measures can therefore be aggregated by using weights to build the GCI indicator whose scale is also ratio. However, defining the weights associated with each measure is a difficult task, and there are different ways of facing this [35]. A first method consists of defining weights according to measurement goals. The same measurer may choose different weights under different circumstances. Another way to carry out the definition is by means of subjective weights based on theoretical or experience-based considerations and, finally, default weights can be used; typically, these may weigh all the measures equally. Using linear regression models may solve most of the problems with weighted sums.

To achieve this, the Factor Analysis, specifically a Principal Components Analysis (PCA), was conducted, using Varimax Rotation. As a result, one component was returned which groups all the measures. As can be observed in Table 7, the PCA offers component scores which are typically used to value the relative importance of each measure in the component. By combining each measure and their corresponding coefficients (components scores), it is possible to define a linear equation as a measurement approach for GCI. Since there are several methods for estimating the component scores, we have chosen the most widely-used one- the regression method [9].

Measure	Components extracted	Components scores (regression)
CFC	0.962	0.176
GM	0.970	0.177
GH	0.870	0.159
AGD	0.953	0.175
MGD	0.985	0.180
TNG	0.980	0.179

Table 7 PCA results for gateway complexity measures

As a result, the proposed formula for GCI is the following:

$$GCI = 0.176 * CFC + 0.177 * GM + 0.159 * GH + 0.175 * AGD + 0.180 * MGD + 0.179 * TNG$$

Table 8 displays different threshold values for the GCI indicator, obtained with Algorithm 1. This information can be used to give an initial interpretation of the understandability and/or modifiability of a business process model. However, understandability and modifiability are fuzzy and subjective and it is difficult to define them. In this sense, we believe it is useful to associate probabilities for considering the model to be suitable in terms of understandability and modifiability. One method for associating probabilities to thresholds was proposed by Bender.

Understandability	Modifiability	Linguistic label
GCI < 6.42		Fairly easy to understand/modify
6.422< GCI ≤ 8.77	GCI < 8.77	Easy to understand/modify
8.77 < GCI ≤ 14.5	8.77 < GCI ≤ 13.05	Moderately difficult to understand/modify
14.5 < GCI ≤ 18.9	13.05 < GCI ≤ 14.5	Difficult to understand/modify
18.9 < GCI	14.5 < GCI	Fairly difficult to understand/modify

Table 8 Thresholds for Gateway Complexity Indicator

The application of the Bender method [5] with experimental data collection presented in this article implies the definition of a binary variable, because of the logistic regression analysis. The dichotomized process is based on the idea of putting a 0 value when efficiency is higher than the median and the value of 1 when it is lower [45]. Despite the loss of information in the dichotomization (explained in the Background Section), we applied this technique with the main purpose of obtaining different probabilities in thresholds.

p_0	Gateway Complexity Indicator	
	Understandability	Modifiability
0.2	2.93	0
0.3	6.72	2.17
0.4	9.83	7.69
0.5	12.69	12.76
0.6	15.55	17.83
0.7	18.66	23.35
0.8	22.45	30.09
0.9	28.16	40.23

Table 9 Associated probabilities to thresholds for GCI

Table 9 specifies general probabilities for GCI. For example, if the GCI value is equal to 6.72, there is a 30% of probability of considering the model as non-understandable. The p-value column indicates probabilities (from zero to one) and understandability and modifiability columns indicate the thresholds for each quality attribute. It could also be interesting to know exactly the probabilities of the thresholds indicated in Table 8. This information is summarized in Table 10. This table can be interpreted as follows: if the model has a GCI value around 8.77, the model has a probability of 36.4% of being difficult to understand, or if the model is around 18.9, the model is difficult with 70.7% probability. The Bender method [5] offers complementary information about the thresholds obtained with ANOVA tests, and can clarify the assessment tasks.

Understandability		Modifiability	
Threshold	Probability	Threshold	Probability
6.42	29.1%	8.77	42.08%
8.77	36.4%	13.05	50.57%
14.05	54.8%	14.5	53.46%
18.9	70.7%	-	-

Table 10 Thresholds and related probabilities for GCI

5 Empirical validation of thresholds

In this section we present findings from applying a cross-validation to the thresholds presented in previous sections. In this case two techniques have been selected for validation purposes, such as recall, precision and accuracy measures (Section 5.1) and ROC curves (Section 5.2).

To develop the cross-validation, a replication of the experiment was done. The experiment and the replication are similar, but only differ in subjects. These new subjects are students of the Master of Information Systems and Master of Economics and Management Science, in the Humboldt University of Berlin (Germany). The replication was developed in July 2011 and it was divided in two runs, as described in Figure 4. In this case subjects did not need any training day to learn BPMN, because they already had enough knowledge to carry out the experiment.

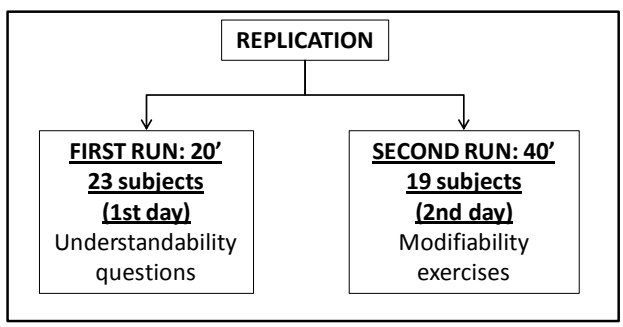


Fig. 4. Description of the replication carried out in Berlin

5.1 Validation through recall, precision and accuracy measures

The first step for the cross-validation of thresholds is based on ideas from the information retrieval field. In this field, the measure called 'precision' is defined as the ratio of true positives to the sum of true positives and false positives [36]. True positives are the number of times thresholds correctly classify a model as understandable, and false positives are the number of times thresholds erroneously classify a model as understandable. In the same context, the 'recall' measure is defined as the true positives to the sum of true positives and false negatives [36]. Finally, the 'accuracy' measure is a widespread measure of effectiveness, to evaluate a classifier's performance [33] and it is calculated as the sum of true positives and true negatives to the sum of true and false positives and true and false negatives. Precision, recall and accuracy are measures that are appropriate for computing the effectiveness of search results [2].

For calculations, we have to consider a *desirable classification* and an *actual* one. The *desirable classification* is defined by the application of thresholds; this is based on the internal quality of business process models. This classification is depicted in Table 6 and is defined with 5 levels for understandability and 4 levels for modifiability, depending on the number of thresholds extracted in previous sections (fairly low, low, medium, high, and fairly high). For example, in the experiment

material, model number 3 obtains *low gateway complexity* values or it is considered *as easy to understand*, in a desirable way. This means that we expect any model with the same measurement values to be easy to understand by a specific subject.

The *actual classification* is based on the efficiency obtained by subjects in the experiment replica, which means external quality of business process models. Efficiency values classified each case in five or four groups (depending on understandability or modifiability) based on the percentile values. That means, for understandability efficiency, the percentile of 20% classified cases on one group with the label of *fairly low*, and so on with the 40%, 60% and 80% and labels of low, medium, high and fairly high, while for modifiability we used the percentile of the 25%, 50% or median, and 75%. For example, if a specific subject obtains a low efficiency in understandability tasks in a business process model, the actual classification for that model is *difficult to understand*.

Let us suppose that the *desire classification* indicates that a specific model obtains high measure values or it is considered as difficult to understand/modify. If the *actual classification* indicates that actually the measure values are high, the prediction was right.

We considered as a *true positive* the case in which a structural measure predicts the evaluation of the quality of a specific model as fairly understandable and, finally, the understandability efficiency value of a subject which is analyzing it in the replication experiment is higher than the median. The definition of true positives/negatives and false positives/negatives is displayed in Figure 5.

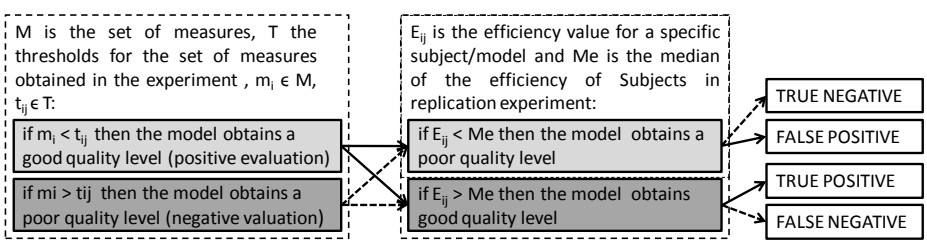


Fig. 5. Definition of the true positive/negative and false positive/negative

The precision, recall and accuracy results for the threshold validation are shown in Figures 6 and 7. These figures are used to check whether the 10 business process models of the experimental material classified as fairly low/low/medium/high/fairly high according to obtained thresholds are well-classified, in contrast to the classification based on the efficiency value of understandability/modifiability obtained by each subject in the replica.

As shown in Figures 6 and 7, we obtained better results for modifiability, due to the fact that the highest accuracy for modifiability is in the fourth tier, while the highest for understandability is in the second tier. Recall has also better results for modifiability, due to the fact that all the results are in the second tier, while the results for understandability are on the first one. Precision has similar results in both charts.

Analyzing the charts, we conclude that the best-classified cases belong to medium and fairly low groups for understandability and low and fairly high ones for modifiability. This leads us to affirm that thresholds can be useful for classifying

models in terms of the level of understandability and modifiability, due to the high values of precision, recall and accuracy.

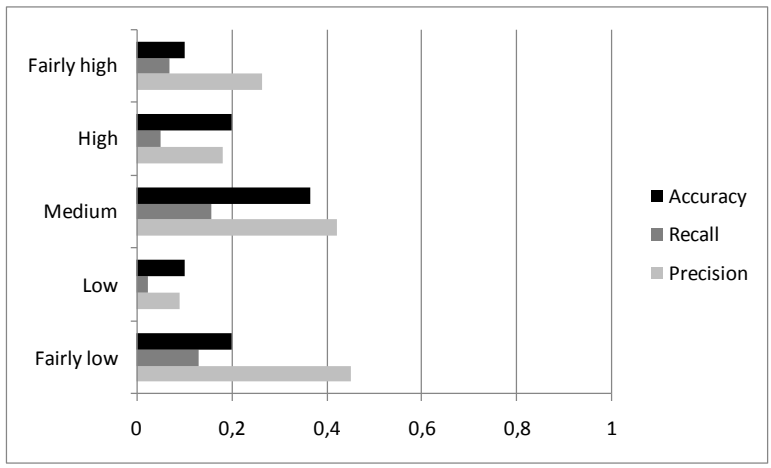


Fig. 6. Precision, Recall and Accuracy for extracted thresholds for understandability

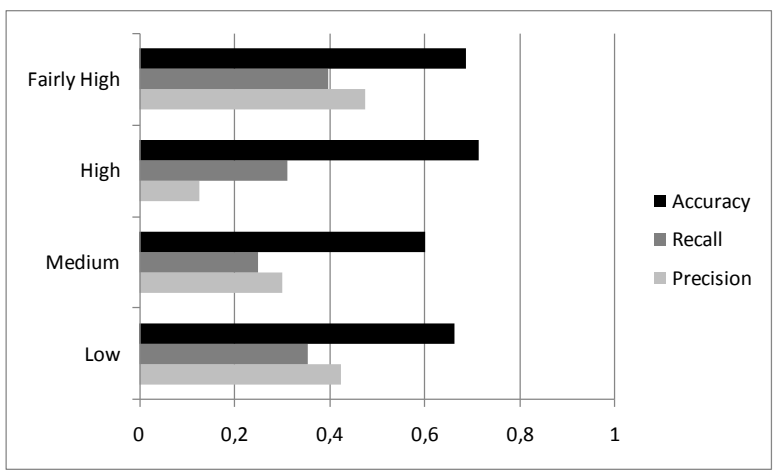


Fig. 7. Precision, Recall and Accuracy for extracted thresholds for modifiability

5.2 Validation through ROC curve analysis

While precision, recall and accuracy are suitable measures for assessing how good a classifier is, other techniques can strengthen the results. ROC curves provide a pure index of accuracy by demonstrating the limits of a test's ability to discriminate between alternative states (understandable/modifiable, non-understandable/non-modifiable) [58]. For the application of ROC curves on our experimental data, we need a categorical variable, which describes whether the model is understandable/modifiable or not, and a continuous one, in this case, the set of

measures. The final purpose of this technique is to determine how good each measure is at classifying models as understandable/modifiable or non-understandable/non-modifiable. In this case, the categories are not binary because they are classified as fairly easy to understand/modify, easy to understand/modify, moderately difficult to understand/modify, high difficult to understand/modify, and fairly difficult to understand/modify. This method calculates pairs of values as sensitivity/1-specificity, where sensitivity is calculated in a similar way to recall.

The test performance is assessed using the Area Under the Curve (AUC) and it is widely-used as a measure of performance of classification [22]. Although it is ranked between 1 and 0, an AUC < 0.5 is considered no good, 0.5<AUC <0.7 is considered as poor and only if AUC > 0.7 is it considered to be acceptable. The results obtained are shown in Table 11 and Figure 8.

Measures	Understandability		Modifiability	
	AUC	Sig	AUC	Sig
CFC	0.691	0.049	0.65	0.05
GM	0.694	0.049	0.647	0.052
GH	0.698	0.049	0.643	0.052
AGD	0.691	0.049	0.650	0.052
MGD	0.693	0.049	0.691	0.049
TNG	0.691	0.049	0.650	0.052
GCI	0.691	0.049	0.650	0.052

Table 11 AUC values for ROC curves

The most suitable result is one that provides an AUC higher or equal to 0.7 with a significance value (p-value) higher or equal to 0.05. Although most results are near these values, not all respect these conditions. We considered all significances as valid because they are 0.052 or 0.049, which is very close to 0.05. As regards AUC values, the ones which are about 0.691 (all for understandability) are considered as acceptable. The rest of them, near 0.650 (all for modifiability), are considered to be poor, but they pass the limit of 0.5 and therefore all the results are tolerable. Figure 8 shows some examples of ROC curves that depicted curves superior to the diagonal, which is a signal of good results. In conclusion, the ROC curve results reveal that thresholds for business process measures are good classifiers of models.

6 Discussion

In this section we discuss the practical utility of thresholds and the limitations of research presented here. Subsection 6.1 investigates the implications of this work for research and practice and, finally, subsection 6.2 discusses threats to validity.

6.1 Implications for research and practice

As stated in the introduction, the business process design elements related to gateway complexity are principally decision nodes and input/output sequence flows of those nodes. Thus, we aim to establish a more suitable use of these design elements, in order to avoid risking the understandability/modifiability of the business process

model. We believe this brief guideline of using decision nodes can be a help to novice modelers when modeling BP.

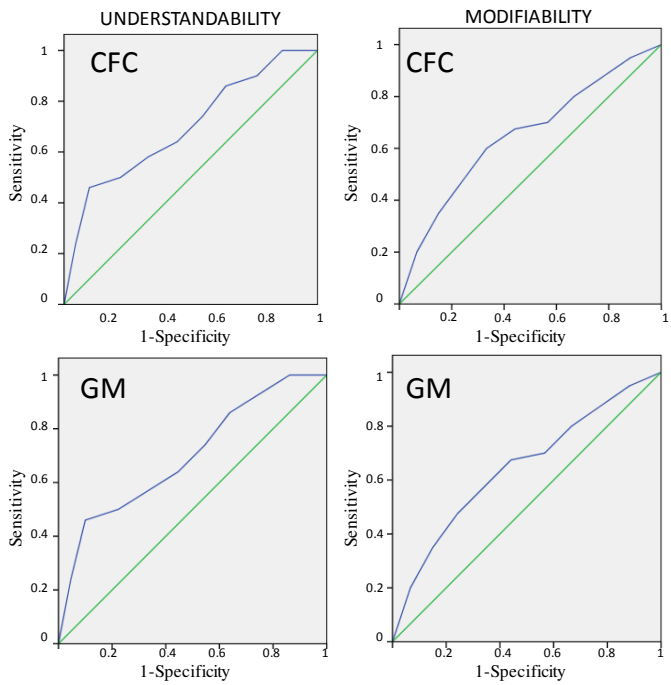


Fig. 8. Example of ROC curves of some measures

First of all, it is important to define the most suitable number of decision nodes. Following the thresholds for the TNG measure, the gateway complexity is high when the model has more than 18 decision nodes, and very high with more than 22. For this reason, we establish the number of nodes as being between 18 and 22. But it is not only the number of decision nodes that increases the complexity of the model; it is also the diversity of their types (XOR, OR and AND). Following the CFC measure, OR-split nodes create more mental states, a total of $2^n - 1$, which means that the focus of reducing gateway complexity should be in this type of decision nodes, while AND nodes imply a lower increase of complexity for models. Since heterogeneity of decision nodes is an important point in the evaluation of complexity, the thresholds for the GH measure indicate to us that more than 10 XOR decision nodes, 7 AND nodes or 4 OR nodes endanger the quality of the model. Input/output sequence flows from decision nodes are another key aspect in gateway complexity. Specifically, more than 7 input/output sequence flows increase the complexity of the model and more than 9 is not acceptable, due to the fact that the modeler would take into account a very "difficult" number of mental states. Finally, an important aspect in a good design is about the number of output/input in split/join nodes. A good design has the same output sequence flows for splits and input sequence flows for joins. To be precise, if

that difference is higher than 15, the complexity could increase too much- higher than 20 is not appropriate.

All of this information can be summarized in the following set of rules for business process modeling:

- Include no more than 18-22 decision nodes.
- Minimize the number of OR split nodes.
- Include no more than 10 XOR, 7 AND and 4 OR decision nodes.
- Each decision node should have fewer than 7-9 input/output sequence flows.
- A difference higher than 15-20 in the number of input/output sequence flows between split/join nodes is not acceptable.

These guidelines are a contribution to the previously-published guidelines by Mendling [31], Becker [4], or Sánchez-González [46, 48] which include some advice about how to model a business process without endangering its general quality. In our case we focus on gateway complexity, extending the information with various recommendations.

6.2 Threats to validity

The purpose of this section is to analyze the different threats to the validity of the experiment and the replica, in particular the conclusion and construct, as well as internal and external validity [16]. We addressed these threats during the experiment design or execution. In the following lines we outline the threats to the experiment, and highlight some improvement to be carried out as future work.

6.2.1 Threats to conclusion validity

The conclusion validity describes our ability to draw statistically-correct conclusions based on the measurements, or it may be said to be the extent to which conclusions about the existence of a statistical relationship between treatments and outcomes are warranted. In the experiment, we consider convenient samples to be: 60 values, 10 models and 53 subjects. Although subjects in the experiment are limited in number, we believe that the sample size is significant enough to allow us to check the main hypothesis (through ANOVA tests) and to obtain conclusion validity. Moreover, the independent variables chosen (measures) in the experiment are also sufficiently valid to draw a conclusion.

Other aspects can be considered as limitations in the validity of the conclusion. We should point out, for instance, that the questions are answered using a pen and paper, and that the responsibility to indicate the time taken in answering depends on subjects. These methods are sometimes not very accurate. However, conclusion validity focuses mainly on the sample size, as was indicated in [54].

6.2.2 Threats to construct validity

Construct validity is concerned with the relation between theory and observation [54], which implies to check whether the relationship between cause and effect is casual or not. Some threats to the construct validity are related to the design, and others have to do with social factors. All of the subjects knew how to fill the questionnaires, because

the experimental material included an example. Moreover, we measured understandability and modifiability, which are calculated by the measure of efficiency (dependent variable). This measure is considered objective, because it reflects the relationship between the time the subjects need to complete the questions or exercises and the correct answers obtained in those tasks. As regards social threats, it seemed to us that the subjects tried to do a good job, knowing that they could increase their final marks. All of these considerations lead us to provide objective measurement of what in this empirical study we purported to measure.

6.2.3 Threats to internal validity

Internal validity is concerned with whether the effect measured is because of changes caused by the researcher, or if it is due to some other unknown cause. In our case, it would mean that there are no significant differences in the efficiency of understandability and modifiability when the business process model is structurally more complex. A number of potential threats to internal validity are listed below.

We have tackled different aspects that could threaten the internal validity of the study, such as: differences between subjects, precision in the time values, learning effects, fatigue effects, persistence effects, subject motivation and mortality.
With respect to *differences between subjects*, all the subjects chosen had a similar knowledge of modeling, because all of them belong to the same level of their degree course. As far as *precision in the time values* is concerned, a large clock was visible to everyone, to show the time in a way that was the same for all the subjects. *Learning effects* were avoided by randomly ordering the 10 different models. *Fatigue effects* were mitigated by conducting the runs of the experiments on different days. As regards *subject motivation*, they were motivated by an assessment of the results they gave and by the fact that they were given extra points towards their final course marks. Finally, *mortality* is about the subjects who did not complete the exercises correctly, and whose experiments should be discarded. In our case, this occurred in the case of 3 of the 56 subjects; that is considered non-relevant for the study.

6.2.4 Threats to external validity

The external validity of a study describes the possibility of generalizing its results. The following threats to external validity were identified; these could limit the realism of the experiment [51]:

- Materials and tasks used: we defined non-real business process models with abstract labels, which are sometimes more difficult to interpret. However, eliminating the influence of domain knowledge with neutral labels also prevents the creation of a potential bias which could stem from varying length of natural labels [32], thus affecting the validity of the experiment. For this reason, abstract levels were used.
- Subjects: the experiments were performed by students and this might be seen as a hindrance to a generalization of the results, as compared to the performance of professionals with a strong background in business process modeling, for example. It does not seem that students perform less competently than professionals with respect to cognitive tasks [41], however. Moreover, the

particular students involved in this research were finishing their degree (Spain) or were master's students (Germany), so it appears to us that using students as subjects did not imply a serious threat to external validity.

- Environment: the modifiability part required some parts of the models to be entered or deleted using only a pen and paper. This could sometimes be more difficult than using a process modeling tool.

6.2.5 Checklist for evaluating the quality of the experiment

The last threats point to some disadvantages in obtaining valid results. It is recognized, however, that the evaluation of the quality of human-centric experiments is recognized as being a difficult and imprecise task. Some researchers realized the need to analyze the quality of experiments in an objective way. In [27], authors described an attempt to develop a procedure for evaluating the quality of experiments by means of a quality checklist. This checklist is classified into three groups:

- Questions on aims: the aims of our research are clearly stated for each experiment, specifying hypotheses in each case.
- Questions on design, data collection and data analysis: the sample size is correctly specified for each experiment, and the main parts of the experiment design are described in detail. After collecting data, these are processed with statistical techniques, obtaining objective conclusions. Limitations and experimenter bias are described, so the reader can be made aware of the scope of the experiments.
- Questions on study outcome: the results are described, specifying thresholds for each measure. These threshold values are described in more detail in subsequent sections.

Following this quality checklist for analyzing the quality of the experimental design presented here, it is possible to establish if the understandability and modifiability experiment and replication are well-designed.

7 Supporting tool for measurement assessment

Findings reported in this research have implications in business process modeling. Derived thresholds guide the modelers about what parts should be redesigned if we are to avoid endangering the understandability or modifiability of models. A previously-developed tool, named MT-BPMN (Measurement Tool for BPMN models) was enhanced for use in supporting measurement assessment, giving some advice to the modeler about how to redesign the model. In addition, this tool was used to help in the experimental process.

MT-BPMN has been developed in Java (Eclipse platform) and it can load different types of business process models in serialized formats. The main two functionalities of this tool are: to calculate measures on business process models represented in BPMN notation and to evaluate these measurement results through thresholds analysis. Consequently, MT-BPMN can serve as a guide to BPMN modelers in improving the general quality of the models. The main menu of the tool is organized as follows:

- *Project viewer*, to load different measurement projects, which resume some measurement activities performed in the past.
- *New evaluation* (see Figure 9), which sets out the measurement results of a specific model. As can be seen in the example shown on the left-hand side, the model was developed with Bizagi 1.6.0. Within this functionality, various other tabs are included for different purposes: the first tab shows the measurement results, while the second tab, *graphic representation of measures*, represents the measurement results in different charts: bar, line, areas, kiviat and sectors. Indicators can also be represented in meter and thermometer charts. These charts provide the user with the advantage of quickly analyzing the quality level of models. This is shown in Figure 10. Finally, the tab *modeling tips* show the different advice for improving the model according to measurement results. This tab is shown in Figure 11.
- *Compare results* tab is used to compare measurement results of two models, to select the best one. This option is especially useful for comparing two versions of the same process.
- *Define new measures*, to include new measures in the database. After including a new measure, it will be calculated in the next measurement application.
- *Define new modeling tips* is a special tab for including modeling tips associated with thresholds. These thresholds can be introduced previously in the database or can be introduced at that particular point in time.
- *Configuration tool* is an option which can configure the tool, extending it so it can load more input formats. To achieve this, users must specify the XML labels which represent each one of the elements of BPMN.

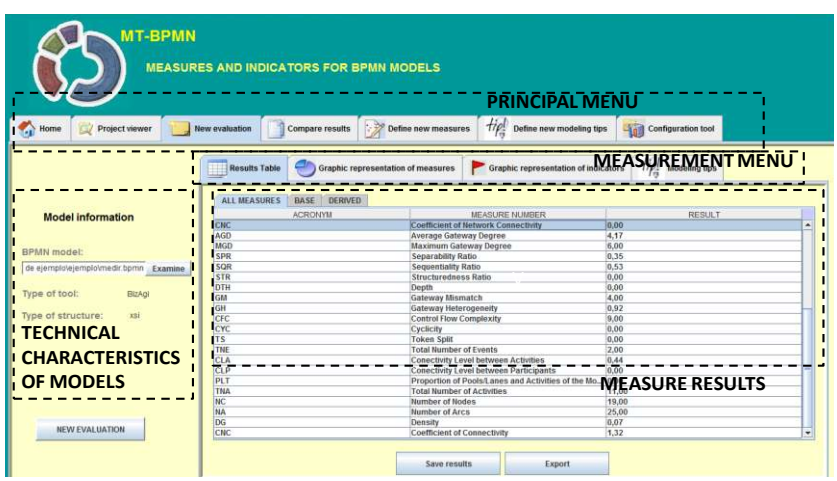


Fig. 9. Tool for measurement BPMN models

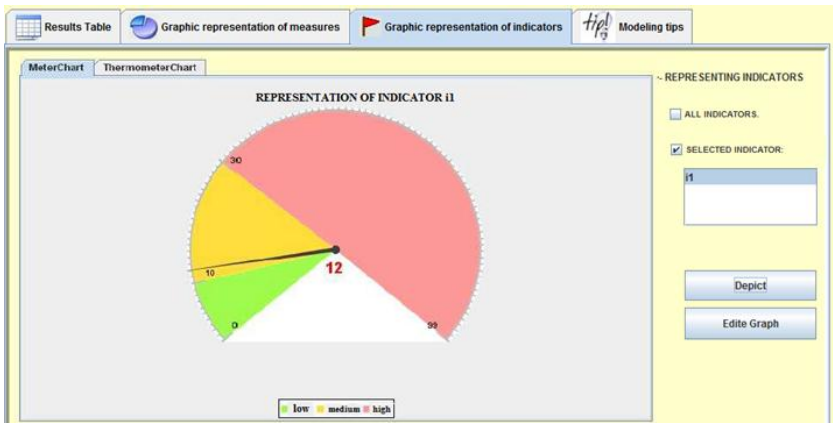


Fig. 10. Representation of an indicator using a meter chart

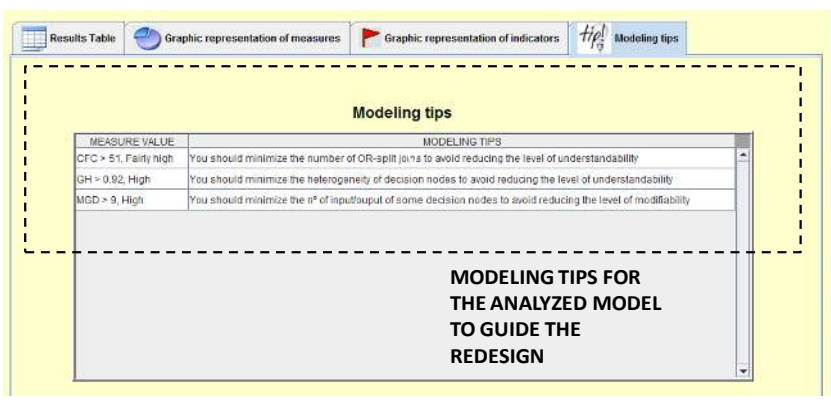


Fig. 11. Modeling tips in tool for measurement BPMN models

MT-BPMN offers a straightforward way of detecting poor-quality structures in BPMN models and it advises the modeler about how to correct them. Definition of thresholds as a decision-making technique is heavily supported by this tool and it is an automatic innovative way of improving business processes at the design stage.

8 Conclusions and future work

In this paper a set of thresholds for business process model measures have been proposed, for the assessment of the level of understandability and modifiability from the perspective of gateway complexity. A controlled experiment with two runs (one for understandability and another for modifiability) was designed and carried out by pre-graduates in a Computer Science degree, to define threshold values theoretically through the application of ANOVA tests. These thresholds were validated through a replication of the experiment, performed by postgraduate Master students. This validation was based on the estimation of precision, recall and accuracy measures and, finally, by the calculation of ROC curves. The results of the validation tasks

revealed that thresholds classified business process models at the specific level of understandability and modifiability, obtaining better results for modifiability and concluding that these thresholds were good and useful for decision-making. Moreover, thresholds for gateway complexity measures are used to define some advice for novice modelers. This advice indicates how to design a model, taking into account the number and nature of the decision nodes.

As a complementary added value of research we have also presented the MT-BPMN tool, with which it is possible to apply a set of measures on BPMN models and assess measurement results through the threshold analysis. After detecting what measures obtain bad results, this tool advises the modeler about what parts should be redesigned.

For future work we propose to validate threshold values with more experimental data, in order to strengthen the obtained results. Moreover, *quality* for business process is a very abstract term and its definition is no light task. Based on this idea, we propose to analyze business process models from other perspectives besides those already studied (understandability and modifiability). Other quality characteristics should be supported by measures and thresholds, which constitute indicators for quality assessment. Finally, a framework for continuous business process model improvement should be defined, to prevent incorrect or unexpected execution of business processes in the organization.

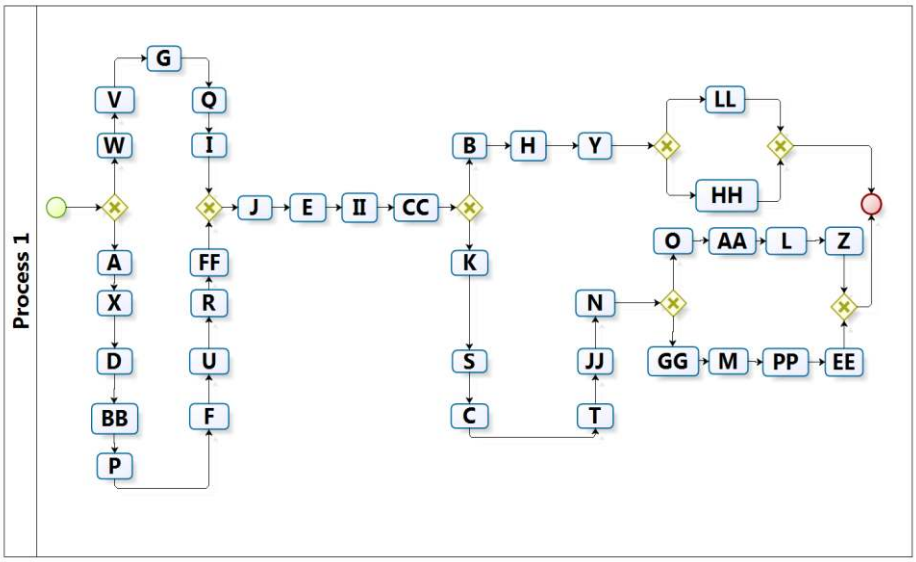
Acknowledgments. This work was partially funded by the following projects: ALTAMIRA (Junta de Comunidades de Castilla-La Mancha, Fondo Social Europeo, PII2I09-0106-2463), INGENIOSO (Junta de Comunidades de Castilla La Mancha, PEII11-0025-9533) and PEGASO/MAGO (Ministerio de Ciencia e Innovación MICINN and Fondo Europeo de Desarrollo Regional FEDER, TIN2009-13718-C02-01).

Appendix A. Experimental material

An extract of the experimental material for the understandability and modifiability experiments is shown. Figure A.1 shows an extract of the understandability experiment and Figure A.2 about the modifiability experiment.

A.1 Example of the material of the understandability experiment

Model 1



A) Answer the following questions about the model, choosing the correct option YES/NO:

STARTING TIME (indicate hh:mm:ss, e.g. 15:01:30s):___hh/___mm/___ss

1. Is it possible to execute activity C without having previously executed activity A? YES/NO

2. Is it possible to execute activity K without having previously executed activity J? YES/NO

3. Is it possible to execute activity E without having previously executed activity X? YES/NO

4. Is it possible to complete the process execution without executing activity HH? YES/NO

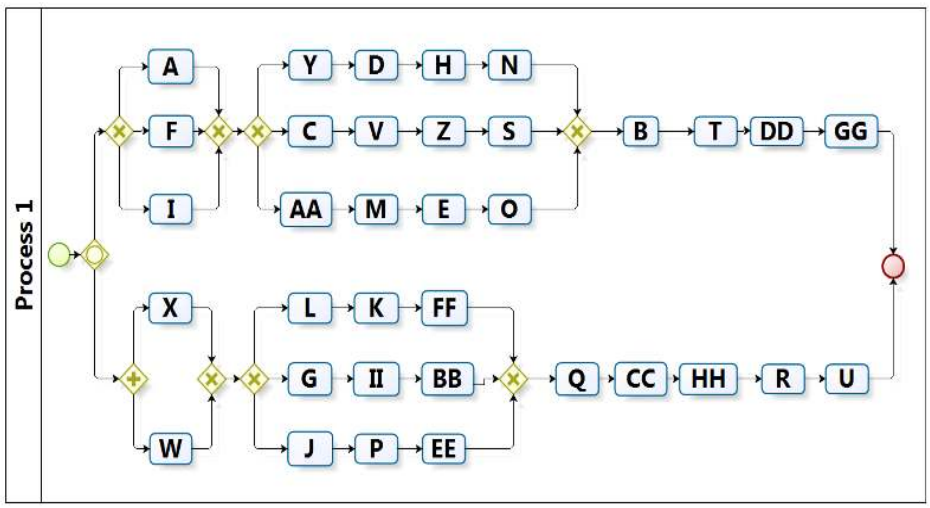
FINISHING TIME (indicate hh:mm:ss, p.e 15:01:30s):___hh/___mm/___ss

B) What, in your opinion, is the complexity of the business process model?

Fairly simple	A bit simple	Medium	Fairly complex	Very complex

A.2. Example of the material of the understandability experiment

Model 2



A) Apply the following modifications, adding, removing or modifying the gateways in the model, and changing the position of certain activities if it is necessary. Each modification is applied to the original model and not to the model modified in previous modification activities.

STARTING TIME (indicate hh:mm:ss, e.g. 15:01:30s):___hh/___mm/___ss

1. Modify the model to execute the sequence of activities L->K->FF or G->II->BB or j->P->EE or Q->CC

2. Modify the model to execute the sequence of activities AA->M and E->O in parallel.

3. Modify the model to execute activities B and T in parallel if activity N has been executed previously.

FINISHING TIME (indicate hh:mm:ss, p.e 15:01:30s):___hh/___mm/___ss

B) What, in your opinion, is the complexity of the business process model?

Fairly simple	A bit simple	Medium	A bit complex	Fairly complex

References

[1]. Alves, T.L., C. Ypma, and J. Visser, *Deriving metric thresholds from benchmark data*, in *Proceedings of the 2010 IEEE International Conference on Software Maintenance*. 2010, IEEE Computer Society. p. 1-10.

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

[3]. Basili, V., G. Caldiera, and H.D. Rombach, *The goal question metric approach.* Encyclopedia of Software Engineering, Wiley, 1994.

[4]. Becker, J., M. Rosemann, and C. von Uthmann, *Guidelines of Business Process Modeling*, in *Business Process Management*. 2000, Springer Berlin / Heidelberg. p. 241-262.

[5]. Bender, R., *Quantitative Risk Assessment in Epidemiological Studies. Investigating Threshold Effects.* Biometrical Journal, 1999. **41**(3): p. 305-319.

[6]. Benlarbi, S., K. El-Emam, N. Goel, and S. N.Rai, *Thresholds for Object-Oriented Measures.* Institute for Information Technology, National Research Council Canada, 2000.

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

[8]. Briand, L., S. Morasca, and V. Basili, *Property-based Software Engineering Measurement.* IEEE Transactions on Software Engineering, 1996. **22**(1): p. 167-173.

[9]. Bryant, F.B. and P.R. Yarnold, *Principal components analysis and exploratory and confirmatory factor analysis*, in *Reading and understanding multivariate statistics*. 1995, American Psychological Association books. p. 99-136.

[10]. Cant, S.N., D.R. Jeffery, and B. Henderson-Sellers, *A conceptual model of cognitive complexity of elements of the programming process.* Information and Software Technology, 2000. **37**(7): p. 351-362.

[11]. Cardoso, J., *Evaluating the Process Control-Flow Complexity Measure.* ICWS 2005, 2005: p. 803-804.

[12]. Cardoso, J., *Approaches to Compute Workflow Complexity.* Dagstuhl Seminar, "The Role of Business Processes in Service Oriented Architectures", 2006.

[13]. Cardoso, J., *Process control-flow complexity metric: An empirical validation.* SCC '06: Proceedings of the IEEE International Conference on Services Computing, 2006: p. 167--173.

[14]. Cardoso, J., J. Mendling, G. Neumann, and H.A. Reijers, *A Discourse on Complexity of Process Models.* BPM 2006 Workshops, 2006: p. 117-128.

[15]. Catal, C., O. Alan, and K. Balkan, *Class noise detection based on software metrics and ROC curves.* Information Sciences, 2011. **181**(21): p. 4867-4877.

[16]. Cook, T.D. and D.T. Campbell, *Quasi-experimentation: Design and analysis for field settings.* Houghton Mifflin, Boston, 1979.

[17]. Dehnert, J. and W.M.P. van der Aalst, *Bridging the Gap between Business Models and Workflow Specifications.* International J. Cooperative Inf. Syst., 2004. **13**(3): p. 289-332.

[18]. Erni, K. and C. Lewerentz, *Applying Design-metrics to Object-Oriented Frameworks.* Proceedings of METRICS, 96, 1996: p. 64-74.
[19]. Fenton, N., *Software Measurement: a necessary scientific basis.* IEEE Transactions on Software Engineering, 1994. **20**: p. 199-206.
[20]. Ferreira, K.A.M., M.A.S. Bigonha, R.S. Bigonha, L.F.O. Mendes, and H.C. Almeida, *Identifying thresholds for object-oriented software metrics.* J. Syst. Softw., 2012. **85**(2): p. 244-257.
[21]. French, V.A., *Establishing software metric thresholds.* International wotkshop on software measurement, 1999.
[22]. Hand, D., *Measuring Classifier Performance: a Coherent Alternative to the Area Under the ROC curve.* Machine Learning, 2009. **77**(1): p. 103-123.
[23]. Henderson-Sellers, B., *Object-Oriented Metrics: Measures of Complexity.* Prentice-Hall, 1996.
[24]. Herbold, S., J. Grabowski, and S. Waack, *Calculation and optimization of thresholds for sets of software metrics.* Empirical Software Engineering, 2011.
[25]. ISO/IEC, *9126-1, Software engineering - product quality - Part 1: Quality Model.* 2001.
[26]. ISO/IEC, *25010, Systems and Software Engineering- System and Software product Quality Requirements and Evaluation (SQuaRE)- System and Software Quality Models.* 2011.
[27]. Kitchenham, B., D.I.K. Sjöberg, O.P. Brereton, D. Budgen, T. Dybaa, M. H., D. Pfahl, and P. Runeson, *Can we evaluate the quality of software engineering experiments?*, in *Proceedings of the 2010 ACM-IEEE International Symposium on Empirical Software Engineering and Measurement*. 2010, ACM: Bolzano-Bozen, Italy. p. 1-8.
[28]. Lindman, H.R., *Analysis of variance in complex experimental designs.* San Francisco: W. H. Freeman & Co. Hillsdale, NJ USA: Erlbaum, 1974.
[29]. McCabe, T.J., *A Complexity Measure.* IEEE Transactions on Software Engineering, 1976. **SE-2**(4): p. 308-320.
[30]. Mendling, J., *Metrics for Process Models: Empirical Foundations of Verification, Error Prediction, and Guidelines for Correctness*. 2008: Springer Publishing Company, Incorporated.
[31]. Mendling, J., H.A. Reijers, and W.M.P. van der Aalst, *Seven Process Modeling Guidelines (7PMG).* Information and Software Technology, 2010. **52**(2): p. 127-136.
[32]. Mendling, J. and M. Strembeck, *Influence Factors of Understanding Business Process Models.* Proc. of the 11th International Conference on Business Information Systems, 2008. **7**: p. 142-153.
[33]. Michie, D., D. Spiegelhalter, and C. Taylor, *Machine Learning, Neural and Statistical Classification.* 1994.
[34]. Miller, G.A., *The magical number seven or minus two: some limits on our capacity of processing information.* Psychological Rev, 1956. **63**: p. 81-97.
[35]. Morasca, S., *On the use of weighted sums in the definition of measures.* Proceedings of the 2010 ICSE Workshop on Emerging Trends in Software Metrics, 2010.

[36]. Olson, D.L. and D. Dursun, *Advanced data mining techniques.* Springer, 1st edition, 2008.
[37]. OMG. *Business Process Model and Notation (BPMN), Version 2.0*. 2011; Available from: http://www.omg.org/spec/BPMN/2.0/.
[38]. Osterweil, L., *Software processes are software too.* Proceedings of the Ninth International Conference on Software Engineering, 1987.
[39]. Park, R.E., W.B. Goethert, and W.A. Florac, *Goal-Driven software Measurement: A Guidebook.* HANDBOOK CMU/SEI-96-HB-002, 1996.
[40]. Poels, G. and G. Dedene, *DISTANCE: a framework for software measure construction*. 1999: Katholieke Universiteit Leuven.
[41]. Reijers, H.A. 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 Systems and Humans, 2011. **41**(3): p. 449-462.
[42]. Rolón, E., J. Cardoso, F. García, F. Ruiz, and M. piattini, *Analysis and Validation of Control-Flow Complexity Measures with BPMN Process Models.* The 10th Workshop on Business Process Modeling, Development, and Support, 2009.
[43]. Rolón, E., F. García, and F. Ruiz, *Evaluation Measures for Business Process Models.* Simposioum in Applied Computing SAC06, 2006.
[44]. Rosenberg, L., *Applying and interpreting object oriented metrics.* Software Technology Conference, 1998.
[45]. Royston, P., G.A. Douglas, and W. Sauerbrei, *Dichotomizing continuous predictors in multiple regression: a bad idea.* Statistics in Medicine, Wiley InterScience, 2005. **25**: p. 127-141.
[46]. Sánchez-González, L., F. García, J. Mendling, and F. Ruiz, *Quality Assessment of Business Process Models Based on Thresholds.* CoopIS 2010 - 18th International conference on Cooperative Information Systems, 2010: p. 78-95.
[47]. Sánchez-González, L., F. García, F. Ruiz, and M. Piattini, *Measurement in Business Processes: a Systematic Review.* Business process Management Journal, 2010. **16**(1): p. 114-134.
[48]. Sánchez-González, L., F. Ruiz, F. García, and J. Cardoso, *Towards Thresholds of Control Flow Complexity Measures for BPMN Models.* 26th Symposium On Applied Computing SAC 10, 2011: p. 1445-1450.
[49]. Shatnawi, R., *A Quantitative Investigation of the Acceptable Risk levels of Object-Oriented Metrics in Open-Source Systems.* IEEE Transactions on Software Engineering, 2010. **36**(2): p. 216-225.
[50]. Shatnawi, R., W. li, J. Swain, and T. Newman, *Finding Software Metrics Threshold values using ROC Curves.* Sofware Maintenance and Evolution: Research and Practice, 2009.
[51]. Sjöberg, D.I.K., B. Anda, E. Arisholm, T. Dyb, M. Jörgensen, A. Karahasanovic, E.F. Koren, and M. Vok, *Conducting Realistic Experiments in Software Engineering*, in *Proceedings of the 2002 International Symposium on Empirical Software Engineering*. 2002, IEEE Computer Society. p. 17.

[52]. Wand, Y. and C. Weber, *Research commentary: Information systems and conceptual modeling–a research agenda.* Info. Sys. Research, 2002. **13**(4): p. 363--376.
[53]. Weyuker, E.J., *Evaluating Software Complexity Measure.* IEEE Transactions on Software Engineering, 1988. **14**(9): p. 1357-1365.
[54]. Wohlin, C., P. Runeson, M. Host, M.C. Ohlsson, B. Regnell, and A. Wesslén, *Experimentation in Software Engineering: An Introduction.* Kluwer Academic, 2000.
[55]. Yoon, K.A., O.S. Kwon, and D.H. Bae, *An approach to outlier detection of software measurement data using the K.means clustering method.* IEEE computer society, 2007: p. 443-445.
[56]. Zelkowitz, M. and D. Wallace, *Experimental models for validating technology.* IEEE Computer, Computing practices, 1998.
[57]. Zuse, H., *Software Complexity: measures and methods.* Walter de gruyter and Co., New Jersey., 1991.
[58]. Zweig, M. and G. Campbell, *Receiver-Operating Characteristic (ROC) Plots: A fundamental evaluation tool in clinical medicine.* Clinical Chemistry, 1993. **39**(4): p. 561-577.