On a Quest for Good Process Models:
The Cross-Connectivity Metric

Irene Vanderfeesten[1], Hajo A. Reijers[1], Jan Mendling[2],
Wil M.P. van der Aalst[1,2], and Jorge Cardoso[3]

[1] Technische Universiteit Eindhoven,
Department of Technology Management,
PO Box 513, 5600 MB Eindhoven, The Netherlands
{i.t.p.vanderfeesten,h.a.reijers,w.m.p.v.d.aalst}@tue.nl
[2] Queensland University of Technology,
Faculty of Information Technology,
Level 5, 126 Margaret Street, Brisbane, Australia
j.mendling@qut.edu.au
[3] SAP Research CEC, SAP AG
Chemnitzer Strasse 48, 01187 Dresden, Germany
jorge.cardoso@sap.com

Abstract. Business process modeling is an important corporate activity, but the understanding of what constitutes good process models is rather limited. In this paper, we turn to the cognitive dimensions framework and identify the understanding of the structural relationship between any pair of model elements as a hard mental operation. Based on the weakest-link metaphor, we introduce the cross-connectivity metric that measures the strength of the links between process model elements. The definition of this new metric builds on the hypothesis that process models are easier understood and contain less errors if they have a high cross-connectivity. We undertake a thorough empirical evaluation to test this hypothesis and present our findings. The good performance of this novel metric underlines the importance of cognitive research for advancing the field of process model measurement.

Keywords: business process modeling, quality metrics, connectivity, EPCs.

1 Introduction

Business process models are widely used for a variety of purposes, such as system development, training, process enactment, costing and budgeting. In many business applications their primary purpose is to act as a *means of communication* such that a process model facilitates the understanding of complex business processes among various stakeholders [16,19,26]. A process model may be used towards this end much as an architect will use a model to ascertain the views of users, to communicate new ideas, and to develop a shared understanding amongst participants. Beyond that, process models are also used as a formal

specification for the development of information systems. Altogether, it is highly desirable that process models do not contain execution errors such as deadlocks and that they are easy to understand for the involved stakeholders.

Even though theoretical quality frameworks [18] and practical modeling guidelines [2] are available for quite some time, it is only a very recent development that *empirical* insights emerge into the factors that influence the quality of process models. For instance, recent studies suggest that larger, real-world process models tend to contain more formal flaws (such as e.g. deadlocks) than smaller models [22,24]. The other study worth mentioning supports the notion that when model size is kept constant (i) a higher density of arcs between the nodes in a model and (ii) a larger number of paths through a model's logical connectors negatively affect its understandability [23].

These results are important stepping stones to what we think is a highly desirable asset for process modelers: Concrete guidelines on how to create process models in such a way that they are easy to understand for people while reducing the risk on errors. It is important to realize that a reengineering project within a multinational company may already involve the creation of thousands of process models [30]. This implies that effective modeling guidelines may lead to substantial economic benefits. This is of particular importance since most modelers are non-experts and hardly familiar with sophisticated design issues [29]. It is a considerable problem for these application areas of process modeling in practice that the current situation in understanding measurable factors of process model quality is still immature. While the mentioned experiments have progressed process model measurement, existing metrics tend to explain not more than half of the variability in a subject's understanding of process models [23]. Clearly, there is a need for a more theoretical stance to advance the design of process model metrics. In this paper, we build on insights from cognitive research into visual programming languages for the development of a new metric, the *Cross-Connectivity (CC) metric*, that aims to capture the cognitive effort to understand the relationship between any pair of process model elements.

The structure of the paper is as follows. In the next section, we will provide the motivation for the CC metric and its formalization. In Section 3 we will describe the empirical evaluation of this metric. Then, we will give an overview of related work, before giving reflections and conclusions in the final section.

2   The Cross-Connectivity Metric

Up to now, little work exists on measuring business process models that considers the cognitive effort of a model user for understanding it. One of the few examples is the research on the Control Flow Complexity (CFC) metric. In its motivation Cardoso refers to the *mental states* that may be generated by a process model and the different types of routing elements [7]. Beyond that, a recent survey into complexity metrics identifies the cognitive motivation as a potential backbone [9]. Most other existing model metrics, however, are adaptations of software

artifact quality metrics that do not dig too deep into cognitive foundations. In such cases, the theoretical basis for their application on process models is indirect at best.

To break away from this tendency, we draw inspiration from the Cognitive Dimensions Framework, as first introduced in [11]. The motivation behind this framework is to use research findings from applied psychology for assisting designers of notational systems. Designers can use the framework to evaluate their designs with respect to the impact that they will have on the users of these designs. Since its introduction, it has gained widespread adoption in the evaluation and design of information artifacts; for an overview of results, see [4].

For the purpose of this paper, the most important dimension of this framework consists of *the hard mental operations* that may be incurred through a particular notation, i.e. the high demand on a user's cognitive resources. Reading a process model implies some hard mental operations in this regard that behavioral relationships between model elements have to be constructed in the mind of the reader. In particular, it is quite difficult – even for experts – to understand whether pairs of activities in a model with lots of parallelism and choices are exclusive or not. Furthermore, even if activities are on a directed path, it is not directly clear on which other elements they depend if there are lots of routing elements in between them. The Cross-Connectivity (CC) metric that we define below aims to quantify the ease of understanding this interplay of any pair of model elements. It builds on the weakest-link metaphor assuming that the understanding of a relationship between an element pair can only be as easy, in the best case, as the most difficult part. Therefore, we identify suitable weights for nodes and arcs along a path between two model elements. Our assertion then is *that a lower (higher) CC value is assigned to those models that are more (less) likely to include errors, because they are more (less) difficult to understand for both stakeholders and model designers.*

Below a set of definitions is given, which together form the basis of the Cross-Connectivity metric. The term 'Cross-Connectivity' is chosen because the strength of the *connections* between nodes is considered *across* all nodes in the model. To appreciate the formalization below, it is important to note firstly that the CC metric expresses the sum of the connectivity between all pairs of nodes in a process model, relative to the theoretical maximum number of paths between all nodes (see Definition 5). Secondly, we assume that the path with the highest connectivity between two nodes determines the strength of the overall connectivity between those nodes (see Definition 4). Thirdly, the tightness of a path (i.e., degree of connectivity) is determined by the product of the valuations of the links connecting the nodes on the path (see Definition 3). So, a single weak link has its effect on the *entire* connection. Finally, differences in the types of nodes that a path consists of determine the tightness of the arcs connecting nodes (see Definitions 1 and 2). For example, an AND connector on a path gives a stronger relation than an XOR connector. At the end of the formalization, illustrative example are given of the application of the CC metric for small process models, showing how the metric can be used to select from alternatives.

Definition 1 (Weight of a Node)
Let a process model be given as a graph consisting of a set of nodes ($n_1, n_2, ... \in$ N) and a set of directed arcs ($a_1, a_2, ... \in A$). A node can be of one of two types: (i) task, e.g. $t_1, t_2 \in T$, and (ii) connector, e.g. $c_1, c_2 \in C$. Thus, $N = T \cup C$. The weight of a node n, $w(n)$, is defined as follows:

$$w(n) = \begin{cases} 1 & , \quad \text{if } n \in C \wedge n \text{ is of type AND} \\ \frac{1}{d} & , \quad \text{if } n \in C \wedge n \text{ is of type XOR} \\ \frac{1}{2^d-1} + \frac{2^d-2}{2^d-1} \cdot \frac{1}{d} & , \quad \text{if } n \in C \wedge n \text{ is of type OR} \\ 1 & , \quad \text{if } n \in T \end{cases}$$

with d the degree of the node (i.e. the total number of ingoing and outgoing arcs of the node).

There are three remarks we would like to make. In the first place, note that the definition above assumes that the process model consists of tasks and connectors. Tasks have at most one input and output arc while connectors can have multiple input and output arcs. A connector of type AND with multiple input arcs is a so-called AND-join, i.e., it synchronizes the various flows leading to the join. The OR-split connector has a behavior in-between an XOR-split (one output arc is selected) and AND-split (all output arcs are chosen). A connector can be both a join and a split (i.e. having multiple input and multiple output arcs), provided that both are of the same type. Secondly, note that we treat all model nodes as unique elements, even though their (business) semantics may be the same. In this way, for example, we support the inclusion of duplicate tasks. Finally, Definition 1 does not correspond to a concrete process modeling language with well-defined semantics. It captures those routing elements that can be expressed with standard process modeling languages such as EPCs, UML Activity Diagrams, Petri nets, BPMN, or YAWL [1].

Most of the values for $w(n)$ in Definition 1 are straightforward given the intent of this metric, e.g., arcs connected to an AND connector will have a higher weight than arcs connected to an XOR connector because the latter involves considering optionality. The only value that requires some explanation is the value for the OR connector. For the OR connector it is not clear upfront how many of the arcs will be traversed during an execution of the process, e.g., in case of an OR split with two outgoing arcs either one of the arcs can be traversed, or both of the arcs might be used. This behavior is reflected in the definition of the weight for an OR connector. The number of all possible combinations of d arcs is: $2^d - 1$. Only one of those combinations (i.e. 1 out of $2^d - 1$) is similar to the situation in which the node would have been an AND, namely the situation in which all arcs are traversed. This particular combination gets a weight of 1 (since that is the weight for an AND connector from Definition 1). Therefore, the first part of the formula for the OR connector is: $\frac{1}{2^d-1} \cdot 1 = \frac{1}{2^d-1}$. All other combinations of arcs can be seen as separate XOR nodes with weight $\frac{1}{d}$. Thus, in $2^d - 2$ out of $2^d - 1$ combinations a weight of $\frac{1}{d}$ is added, which leads to the second part of the formula.

The following definition shows that the weight of an arc is based on the weight of the corresponding nodes.

Definition 2 (Weight of an Arc)
Let a process model be given by a set of nodes (N) and a set of directed arcs (A). Each directed arc (a) has a source node (denoted by $src(a)$) and a destination node (denoted by $dest(a)$).
The weight of arc a, $W(a)$, is defined as follows:

$$W(a) = w(src(a)) \cdot w(dest(a))$$

Definition 3 (Value of a Path)
Let a process model be given by a set of nodes (N) and a set of directed arcs (A). A path p from node n_1 to node n_2 is given by the sequence of directed arcs that should be followed from n_1 to n_2: $p =< a_1, a_2, ..., a_x >$. The value for a path p, $v(p)$, is the product of the weights of all arcs in the path:

$$v(p) = W(a_1) \cdot W(a_2) \cdot ... \cdot W(a_x)$$

Definition 4 (Value of a Connection)
Let a process model be given by a set of nodes (N) and a set of directed arcs (A) and let P_{n_1,n_2} be the set of paths from node n_1 to n_2. The value of the connection from n_1 to n_2, $V(n_1, n_2)$, is the maximum value of all paths connecting n_1 and n_2:

$$V(n_1, n_2) = \max_{p \in P_{n_1,n_2}} v(p)$$

If no path exists between node n_1 and n_2, then $V(n_1, n_2) = 0$. Also note that loops in a path should not be considered more than once, since the value of the connection will not be higher if the loop is followed more than once in the particular path.

Based on the above valuation of connectivity (i.e., tightness of the connection between two nodes), we define the *Cross-Connectivity metric*.

Definition 5 (Cross-Connectivity (CC))
Let a process model be given by a set of nodes (N) and a set of directed arcs (A). The Cross-Connectivity metric is then defined as follows:

$$CC = \frac{\sum_{n_1,n_2 \in N} V(n_1, n_2)}{|N| \cdot (|N| - 1)}$$

Example 1. To illustrate the use of the CC metric an example is elaborated. Figure 1 contains a process model with five tasks (i.e. $T = \{A, B, C, D, E\}$), three connectors (i.e. $C = \{XOR, AND, OR\}$) and seven directed arcs (i.e. $A = \{a_1, a_2, a_3, a_4, a_5, a_6, a_7\}$). To calculate the value for Cross-Connectivity the weight for each node is calculated first (see Table 1).

Table 1. The degrees and weights for the nodes in the process model of Figure 1

Node (n)	Degree (d)	Weight $(w(n))$
A	1	1
B	1	1
C	1	1
D	1	1
E	1	1
XOR	3	$\frac{1}{3}$
AND	3	1
OR	3	$\frac{1}{2^3-1} + \frac{2^3-2}{2^3-1} \cdot \frac{1}{3} = \frac{3}{7}$

Then, the weight for each arc is calculated:

$$W(a_1) = w(A) \cdot w(XOR) = 1 \cdot \frac{1}{3} = \frac{1}{3}$$

$$W(a_2) = w(B) \cdot w(XOR) = 1 \cdot \frac{1}{3} = \frac{1}{3}$$
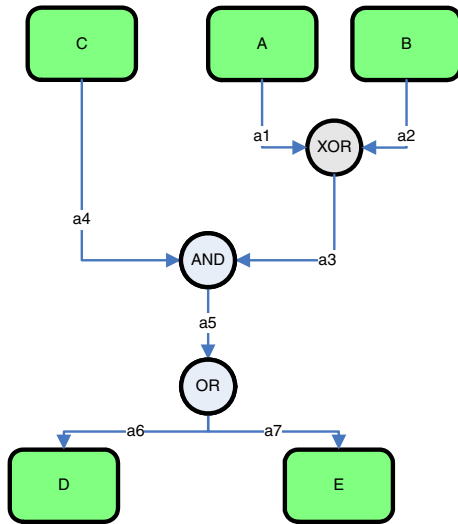
$$W(a_3) = w(XOR) \cdot w(AND) = \frac{1}{3} \cdot 1 = \frac{1}{3}$$

$$W(a_4) = w(C) \cdot w(AND) = 1 \cdot 1 = 1$$

$$W(a_5) = w(AND) \cdot w(OR) = 1 \cdot \frac{3}{7} = \frac{3}{7}$$

$$W(a_6) = w(OR) \cdot w(D) = \frac{3}{7} \cdot 1 = \frac{3}{7}$$

$$W(a_7) = w(OR) \cdot w(E) = \frac{3}{7} \cdot 1 = \frac{3}{7}$$

Fig. 1. A simple example with five tasks and three connectors. $T = \{A, B, C, D, E\}$, $C = \{XOR, AND, OR\}$, $A = \{a_1, a_2, a_3, a_4, a_5, a_6, a_7\}$.

The paths between each pair of nodes are determined and the value for the connection between the pair of nodes is computed. For example node A and node D are connected through the path $< a_1, a_3, a_5, a_6 >$. In this case, this is the only path from A to D. Thus, the value of this path is the maximum value over all paths from A to D:

$$V(A,D) = v(< a_1, a_3, a_5, a_6 >) = W(a_1) \cdot W(a_3) \cdot W(a_5) \cdot W(a_6) = \frac{1}{3} \cdot \frac{1}{3} \cdot \frac{3}{7} \cdot \frac{3}{7} = \frac{1}{49}.$$

Similarly, the value for the connection from the XOR-node to the OR-node is computed:

$$V(XOR, OR) = v(< a_3, a_5 >) = W(a_3) \cdot W(a_5) = \frac{1}{3} \cdot \frac{3}{7} = \frac{1}{7}.$$

For all values, see Table 2.

Table 2. Table showing the values for the connections between all pairs of nodes

|	A	B	C	D	E	XOR	AND	OR	Total
A	0	0	0	$\frac{1}{49}$	$\frac{1}{49}$	$\frac{1}{3}$	$\frac{1}{9}$	$\frac{1}{21}$	$\frac{235}{441}$
B	0	0	0	$\frac{1}{49}$	$\frac{1}{49}$	$\frac{1}{3}$	$\frac{1}{9}$	$\frac{1}{21}$	$\frac{235}{441}$
C	0	0	0	$\frac{9}{49}$	$\frac{9}{49}$	0	1	$\frac{3}{7}$	$\frac{88}{49}$
D	0	0	0	0	0	0	0	0	0
E	0	0	0	0	0	0	0	0	0
XOR	0	0	0	$\frac{3}{49}$	$\frac{3}{49}$	0	$\frac{1}{3}$	$\frac{1}{7}$	$\frac{88}{147}$
AND	0	0	0	$\frac{9}{49}$	$\frac{9}{49}$	0	0	$\frac{3}{7}$	$\frac{39}{49}$
OR	0	0	0	$\frac{3}{7}$	$\frac{3}{7}$	0	0	0	$\frac{6}{7}$

Finally, the CC value is determined as the sum of the values for all connections, divided by the number of nodes times the number of nodes minus one:

$$CC = \frac{\frac{235}{441} + \frac{235}{441} + \frac{88}{49} + 0 + 0 + \frac{88}{147} + \frac{39}{49} + \frac{6}{7}}{8 \cdot 7} = \frac{\frac{2255}{441}}{56} \approx 0.09131$$

Now the mechanics behind the CC metric have been dealt with, it is worthwhile to explore how it can help to distinguish models that are preferable. This will clarify that our interest at this point is rather with a model's CC value *relative* to that of another model.

Consider the two models that are shown in Figure 2. Both models express the same business logic as the initial model in Figure 1, but their CC values are different. The model at the left-hand side in Figure 2 is block-structured, i.e., it differs from the initial model in the sense that the AND-join and XOR-join at the top of the model are matched by corresponding splits. Intuitively, one may expect that a block-structure will positively affect model comprehension. Indeed, as the links between the various nodes become tighter, this is expressed by a higher CC value of 0.12486 versus a value of 0.09131 of the initial model.

The model at the right-hand side in Figure 2 is different from the initial model of Figure 1 in the sense that it reorders the top connectors: It expresses
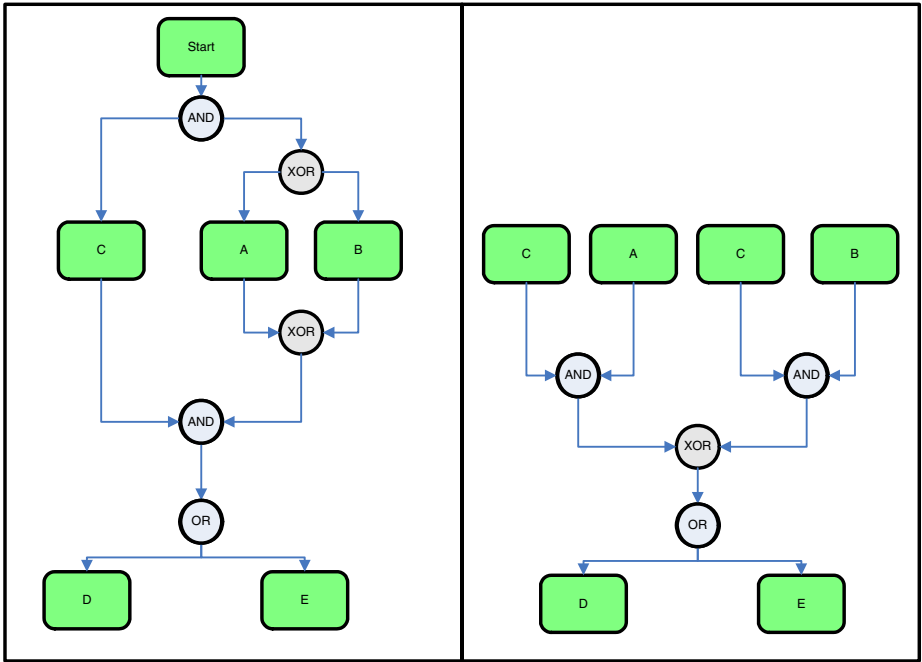
Fig. 2. Two alternatives to the example of Fig. 1

(C AND A) XOR (C AND B) instead of C AND (A XOR B), the former being the more elaborate expression with a duplicate for task C.[1] So, the model at the right-hand side can be expected to be slightly more difficult to understand than the initial model, which is supported by a lower CC value of 0.08503 versus the initial model's CC value of 0.09131.

3   Evaluation

In this section, we report on how the CC metric has been subjected to a thorough empirical evaluation. First, we will describe the evaluation with respect to the metric's capability to predict error probabilities in process models. Next, we will present on its suitability to explain which process models are easier to understand than others.

3.1   Validation for Error Prediction

An indication for a metric's predictive power is that it can accurately distinguish between models with errors and without errors. Because this evaluation uses a

---

[1] Recall that in the computation of the CC metric all model elements are treated as unique elements.

large set of Event-driven Process Chains (EPCs), we use the EPC soundness criterion as defined in [21] for determining whether an involved model has errors or not and assume that a decrease in CC is likely to result in more errors. Therefore, our hypothesis is:

H1: A decrease in *CC* implies an increase in error probability.

To evaluate this hypothesis, the EPCs of the *SAP Reference Model* are used. The development of the SAP reference model started in 1992 and first models were presented at CEBIT'93 [17, p.VII]. Since then, it was developed further until version 4.6 of SAP R/3, which was released in 2000. The SAP reference model includes 604 non-trivial EPCs. The advantage of considering this set of models is that there is extensive literature available that explains its creation, e.g., [17]. Furthermore, it is frequently referenced in research papers as a typical reference model and used in previous quantitative analyses, as e.g. reported in [20,22,24]. This way, our results can be compared to these related works.

As a first step, we use *correlation analysis*. In particular, we investigated to what extent the CC metric is capable to rank non-error and error models. This capability can be estimated using the rank correlation coefficient by Spearman. For CC it is -0.434. For this metric there is a strong and 99% significant correlation, which matches the expectation of the hypothesis, i.e. H1 holds.

In a second step, we use *multivariate logistic regression*. This approach estimates the coefficients B of a linear combination of input parameters for predicting event versus non-event based on a logistic function. In our case, we predict error versus non-error for the EPCs in the SAP reference model based on the CC metric and a constant. The accuracy of the estimated model is assessed based on the significance level of the estimated coefficients, the percentage of cases that are classified correctly, and the share of the variation that is explained by the regression. This share is typically measured using the Nagelkerke R^2 ranging from 0 to 1 (1 being the best possible value). The estimated coefficient should have a Wald statistic that is below 5% signalling that it is significantly different from zero. For technical details of logistic regression we refer to [13]. For applications in predicting errors in process models see [20,22,24].

We calculated a univariate logistic regression for CC first. Table 3 shows that CC alone already yields a high Nagelkerke R^2 of 0.586. The negative coefficient matches the expectation of hypothesis H1. Furthermore, we stepwise introduced other metrics to the model. We used those metrics that were found in [22] as the best combination to predict errors in EPCs. In this context, it is interesting to note that adding these metrics yields quite similar coefficients for them as in the predicting function of [22]. This suggests that the CC metric indeed measures a process model aspect that is orthogonal to metrics that have been defined before.

3.2   Validation for Understandability

To evaluate the capability of the CC metric to explain which process models are easier to understand than others, we used the empirical data described in

Table 3. Multivariate Logistic Regression Models with CC

	Parameter	Coefficient	Std.Error	Wald	Sig.	Nagelkerke	Classification
Step 1	CC	-13.813	1.229	126.386	0.000	0.586	0.791
...	...	...	...	...	...	...	...
Step 5	CC	-10.478	2.931	12.783	0.000	0.847	0.916
	Structuredness	-9.500	1.028	85.328	0.000		
	Diameter	0.139	0.032	18.829	0.000		
	Cyclicity	6.237	1.857	11.281	0.001		
	CNC	5.541	0.935	35.145	0.000		

[23]. This data was obtained in a project that aims at the analysis of the impact of both model and personal characteristics on the understandability of process models. In particular, a set of 20 model characteristics were investigated, which have been proposed and formally defined in [20].

In total, 73 students filled out a questionnaire in the fall of 2006. A set of 12 process models from practice, each having the same number of tasks (25), formed the basis of the questionnaire. As part of the models' evaluation, students were asked to answer questions like "If task K is executed for a case, can task L be executed for the same case?" The evaluation of the 12 models by the 73 students led to a total of 847 complete model evaluations. On this basis, a SCORE variable could be calculated per model as the mean sum of correct answers it received. This SCORE variable served as a way to make understandability operational.

From the earlier analysis of these results [23], the following main conclusions were drawn with respect to model characteristics:

1. From the 20 factors considered, five model factors exhibited the hypothesized relation with SCORE, i.e. (1) #OR-JOINS, (2) DENSITY, (3) AVERAGE CONNECTOR DEGREE, (4) MISMATCH, and (5) CONNECTOR HETEROGENEITY.
2. From these five model characteristics, only the correlations between DENSITY (the ratio between the actual number of arcs and the theoretical maximal number of arcs) and SCORE (-0.618) and between AVERAGE CONNECTOR DEGREE (the average number of input and output arcs of the routing elements in a model) and SCORE (-0.674) correlated significantly, with respective P-values of 0.032 and 0.016.
3. From all linear regression models on the basis of a combination of these five model factors, the regression model that only used AVERAGE CONNECTOR DEGREE displayed the best explanatory power for the variability in SCORE, with an adjusted R^2=45% (Nagelkerke's coefficient of determination).

To evaluate the CC metric, it was incorporated in the above analysis. We arrive at the following conclusions:

- Just like the five model factors that emerged from the original analysis, the CC metric displays the expected relation with SCORE.

- Unlike the density and average connector degree factors, the correlation between SCORE and the CC metric (0.549) is *not* significant at a 95% confidence interval as the P-value of 0.065 slightly exceeds the 0.05 confidence interval.
- A regression model with a *much better* explanatory power for the variation in SCORE could be developed by *including* the CC metric: the adjusted R^2 from the original model increased from 45% to 76% in the new regression model. In particular, by combining the #OR-JOINS, DENSITY, AVERAGE CONNECTOR DEGREE, MISMATCH factors and the CC metric this result could be achieved. A visualization of this regression model can be seen in Figure 3.
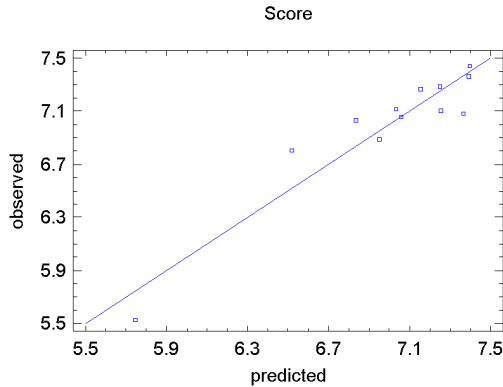
Fig. 3. Linear regression model explaining the mean SCORE for the 12 process models

What this analysis suggests is that CC on its own is slightly less powerful as an indicator for process model understandability than the two best candidate metrics available, but that it can deliver a superior explanation of the variation in understandability across models when combined with existing metrics.

4   Related Work

This section briefly describes the related work for business process metrics. In essence, related work can be organized in two categories: process model metrics inspired by software measurement and experimental work on process model metrics. In this section, we focus in particular on metrics that consider overall structural aspects of the process model beyond simple count metrics. For an overview of process model metrics in general refer to [9,12,20,33].

The early development of process model metrics is greatly inspired by and based on software quality metrics. These metrics aim at obtaining program designs that are less error-prone, easier to comprehend and easier to maintain. A survey of existing software metrics can be found in e.g. [14,35]. A number of studies demonstrate the significant correlation of software quality metrics with errors in the software design (e.g. [3,5,15,31,32]). In the tradition of this work, there are some works in the 1990s that are mainly rooted in software quality measurement.

Daneva et al. [10] introduce a set of complexity indicators for EPCs based on the visual attributes of the model: *function cohesion*, *event cohesion* and *cohesion of a logical connector*. From their validation with 11 EPCs they conclude that their metrics help to identify error-prone model fragments. Morasca proposes a set of simple metrics for software specifications designed with Petri-nets [25]. He identifies size, length, structural complexity, and coupling as interesting attributes of a design without striving for an empirical validation. The works by Reijers and Vanderfeesten extend this research stream by introducing a coupling-cohesion metric for guiding the design of a workflow process [27,28]. This approach is based on the data flow in a business process and uses the network structure of the product as a starting point rather than the process model. In [34], Vanderfeesten, Cardoso, and Reijers propose a weighted coupling metric, which puts a weight to the different types of connections between two activities in the process model. While this metric lacks a thorough cognitive motivation, it was used as a blueprint for the CC metric. Cardoso has developed a Control Flow Complexity (CFC) metric [7] which was validated against *Weyuker's* complexity axioms [6] and tested with respect to their correlation with perceived complexity [8]. In contrast to the CC metric, it does not consider the connections between different model elements, but focuses on routing elements in isolation.

Mendling et al. take an experimental approach towards process model metrics that is driven by the explanatory power of a metric in an empirical setting. In [20,22] Mendling et al. have tested 28 business process metrics (including size, density, structuredness, coefficient of connectivity, average connector degree, control flow complexity, and others) as error predictors on a set of over 2000 process models from different samples. All metrics, except for density and the maximum degree of a connector, are confirmed to be correlated to error-proneness as expected. Another result of this study is a logistic regression model is able to classify 90% of the process models correctly. Finally, a survey on understandability of process models is reported by Mendling, Reijers and Cardoso in relation to the set of metrics mentioned in the previous study [23]. The main results of this survey are already described in Section 3.2.

While the metrics used in these experiments are motivated theoretically, most of them are not explicitly rooted in cognitive research. The CC metric considers hard mental operations as defined in the cognitive dimensions framework [11] as the main factor that drives understanding a process model.

5   Conclusion

In this paper, we motivated, formalized, and validated the Cross-Connectivity metric for process models. The metric expresses how tightly the nodes in a process model are connected building on a weakest-link metaphor. The definition of the metric builds on the assumption that a higher value is associated with an easier understanding of the model, which implies as a consequence a lower error-probability. As follows from our evaluation of this metric for both these aspects, it performs similarly well as the best available alternative model metrics. On top

of that, our results suggest that the CC metric adds a new cognitive perspective on process model quality, which helps to deliver a better explanatory power when it is combined with the existing ones.

In reflecting on the development of business process model metrics, it is fair to say that it is a research area in development. Initially, proposals for process model metrics were highly conceptual, on the basis of the perhaps tempting idea that if metrics are useful to analyze software programs it should be equally applicable for process models. By now, we have progressed to the stage where model metrics are put to the test for determining their effectiveness in reality. The good performance of the CC metric clearly shows that a more cognitive theoretical stance is needed to advance the field of process model measurement.

Overall, feedback from empirical validations has improved the quality of process model metrics: The metrics proposed in recent works, e.g. [20,22] and this paper, perform much better in explaining the variation of understanding and occurrence of errors in process models. In our future work, we will continue towards further improvements. In particular, we aim at evaluating model quality metrics on a wider scale, by considering larger sets of real-world models. In order to achieve that, we are collaborating with consultancy companies that practice process modeling on a day-to-day basis for their clients. Since most empirical research has been done with EPC models, we are very much interested in BPMN and Petri-net process models. Furthermore, we are investigating additional factors that contribute to a comprehensive understanding of process model quality as, for example, the visual layout a process model graph and the importance of preliminary knowledge about the domain that is captured in the model. As the ultimate goal of our research, we envision the development of a set of concrete guidelines for process modelers, substantiated by solid theoretical foundations and empirical evidence, which will help to create better process models in practice.

Acknowledgement

References

1. van der Aalst, W.M.P., ter Hofstede, A.H.M., Kiepuszewski, B., Barros, A.P.: Workflow Patterns. Distributed and Parallel Databases 14(1), 5–51 (2003)
2. Becker, J., Rosemann, M., von Uthmann, C.: Guidelines of Business Process Modeling. In: van der Aalst, W.M.P., Desel, J., Oberweis, A. (eds.) Business Process Management. LNCS, vol. 1806, pp. 30–49. Springer, Berlin (2000)
3. Bieman, J.M., Kang, B.-K.: Measuring Design-level Cohesion. IEEE Transactions on Software Engineering 24(2), 111–124
4. Blackwell, A.F.: Ten Years of Cognitive Dimensions in Visual Languages and Computing. Journal of Visual Languages and Computing 17(4), 285–287 (2007)

5. Card, D.N., Church, V.E., Agresti, W.W.: An Empirical Study of Software Design Practices. IEEE Transactions on Software Engineering 12(2), 264–271
6. Cardoso, J.: Control-flow Complexity Measurement of Processes and Weyuker's Properties. In: Proceedings of the 6th International Enformatika Conference (IEC 2005), pp. 213–218. International Academy of Sciences (2005)
7. Cardoso, J.: How to Measure the Control-flow Complexity of Web Processes and Workflows. In: Fischer, L. (ed.) Workflow Handbook 2005, Future Strategies, Lighthouse Point (2005)
8. Cardoso, J.: Process Control-flow Complexity Metric: an Empirical Validation. In: IEEE International Conference on Services Computing (IEEE SCC 2006), pp. 167–173. IEEE Computer Society Press, Los Alamitos (2006)
9. Cardoso, J., Mendling, J., Neumann, G., Reijers, H.A.: A Discourse on Complexity of Process Models. In: Eder, J., Dustdar, S. (eds.) BPM Workshops 2006. LNCS, vol. 4103, pp. 115–126. Springer, Heidelberg (2006)
10. Daneva, M., Heib, R., Scheer, A.-W.: Benchmarking Business Process Models. IWi Research Report 136, Institute for Information Systems, University of the Saarland, Germany (1996)
11. Green, T.R.G., Petre, M.: Usability Analysis of Visual Programming Environments: A 'Cognitive Dimensions' Framework. Journal of Visual Languages and Computing 7(2), 131–174 (1996)
12. Gruhn, V., Laue, R.: Complexity Metrics for Business Process Models. In: Proceedings of the 9th international conference on business information systems (BIS 2006). Lecture Notes in Informatics, vol. 85 (2006)
13. Hosmer, D., Lemeshow, S.: Applied Logistic Regression, 2nd edn. Wiley & Sons, Chichester (2000)
14. Kafura, D.: A Survey of Software Metrics. In: ACM 1985: Proceedings of the 1985 ACM annual conference on The range of computing: mid-80's perspective, pp. 502–506. ACM Press, New York (1985)
15. Kang, B.-K., Bieman, J.M.: A Quantitative Framework for Software Restructuring. Journal of Software Maintenance 11, 245–284 (1999)
16. Kawalek, P., Kueng, P.: The Usefulness of Process Models: A Lifecycle Description of how Process Models are used in Modern Organisations. In: Siau, K., Wand, Y., Parsons, J. (eds.) Proceedings of the Second CAiSE/IFIP8.1 International Workshop on Evaluation of Modelling Methods in Systems Analysis and Design, pp. 1–12 (1997)
17. Keller, G., Teufel, T.: Sap R/3 Process Oriented Implementation: Iterative Process Prototyping. Addison-Wesley Longman Publishing Co., Inc, Boston (1998)
18. Krogstie, J., Sindre, G., Jørgensen, H.: Process Models Representing Knowledge for Action: a Revised Quality Framework. European Journal of Information Systems 15(1), 91–102 (2006)
19. Lindsay, A., Downs, D., Lunn, K.: Business processes: attempts to find a definition. Information and Software Technology 45(15), 1015–1019 (2003)
20. Mendling, J.: Detection and Prediction of Errors in EPC Business Process Models. PhD thesis, Vienna University of Economics and Business Administration, Vienna, Austria (May 2007)
21. Mendling, J., van der Aalst, W.M.P.: Formalization and Verification of EPCs with OR-Joins Based on State and Context. In: Krogstie, J., Opdahl, A., Sindre, G. (eds.) CAiSE 2007 and WES 2007. LNCS, vol. 4495, pp. 439–453. Springer, Berlin (2007)

22. Mendling, J., Neumann, G., van der Aalst, W.M.P.: Understanding the Occurrence of Errors in Process Models based on Metrics. In: Meersman, R., Tari, Z. (eds.) OTM 2007, Part I. LNCS, vol. 4803, pp. 113–130. Springer, Heidelberg (2007)
23. Mendling, J., Reijers, H.A., Cardoso, J.: What Makes Process Models Understandable? In: Alonso, G., Dadam, P., Rosemann, M. (eds.) BPM 2007. LNCS, vol. 4714, pp. 48–63. Springer, Berlin (2007)
24. Mendling, J., Verbeek, H.M.W., van Dongen, B.F., van der Aalst, W.M.P., Neumann, G.: Detection and Prediction of Errors in EPCs of the SAP Reference Model. Data and Knowledge Engineering 64(1), 312–329 (2008)
25. Morasca, S.: Measuring Attributes of Concurrent Software Specifications in Petrinets. In: Proceedings of the 6th International Symposium on Software Metrics, pp. 100–110. IEEE Computer Society, Los Alamitos (1999)
26. Ould, M.A.: Business Processes: Modelling and Analysis for Re-engineering and Improvement. Wiley, Chichester (1995)
27. Reijers, H.A.: A Cohesion Metric for the Definition of Activities in a Workflow Process. In: Proceedings of the 8th CAiSE/IFIP8.1 International workshop on Evaluation of Modeling Methods in Systems Analysis and Design (EMMSAD 2003), pp. 116–125 (2003)
28. Reijers, H.A., Vanderfeesten, I.T.P.: Cohesion and Coupling Metrics for Workflow Process Design. In: Desel, J., Pernici, B., Weske, M. (eds.) BPM 2004. LNCS, vol. 3080, pp. 290–305. Springer, Berlin (2004)
29. Rosemann, M.: Potential Pitfalls of Process Modeling: Part A. Business Process Management Journal 12(2), 249–254 (2006)
30. Rosemann, M.: Potential Pitfalls of Process Modeling: Part B. Business Process Management Journal 12(3), 377–384 (2006)
31. Selby, R.W., Basili, V.R.: Analyzing Error-Prone System Structure. IEEE Transactions on Software Engineering 17, 141–152 (1991)
32. Shen, V.Y., Yu, T.-J., Thebaut, S.M., Paulsen, L.R.: Identifying Error-Prone Software. IEEE Transactions on Software Engineering 11, 317–324 (1985)
33. Vanderfeesten, I., Cardoso, J., Mendling, J., Reijers, H.A., van der Aalst, W.M.P.: Quality Metrics for Business Process Models. In: Fischer, L. (ed.) BPM and Workflow Handbook 2007, Future Strategies, USA, May 2007, pp. 179–190 (2007)
34. Vanderfeesten, I., Cardoso, J., Reijers, H.A.: A Weighted Coupling Metric for Business Process Models. In: Eder, J., Tomassen, S.L., Opdahl, A., Sindre, G. (eds.) Proceedings of the CAiSE 2007 Forum, CEUR Workshop Proceedings, vol. 247, pp. 41–44 (2007)
35. Xenos, M., Stavrinoudis, D., Zikouli, K., Christodoulakis, D.: Object-Oriented Metrics - A Survey. In: Proceedings of the FESMA 2000, Federation of European Software Measurement Associations, pp. 1–10 (2000)