Computing coupling for business process models

Jorge Cardoso¹, Irene Vanderfeesten², Hajo A. Reijers²

¹ University of Madeira, 
Department of Mathematics and Engineering, 
9000-390 Funchal, Portugal, 
jcardoso@uma.pt

² Technische Universiteit Eindhoven, 
Department of Technology Management, 
PO Box 513, 5600 MB Eindhoven, The Netherlands 
{t.p.vanderfeesten, h.a.reijers}@tm.tue.nl

Abstract. Business process management and process models are rapidly gaining a worldwide acceptance and recognition. Recently, a new area of research has emerged, the measurement of quality metrics of process models. We believe that current modeling tools should provide more analytical capabilities, namely they should provide mechanisms to analyze the coupling, modularity and complexity of process models. In this paper, we show how the notion of coupling, which has been used successfully in software engineering for many years, can be used to evaluate the quality of process models. To test our coupling metric we have conducted a set of experiments with several SAP reference models. Our first conclusion indicates that SAP models have a very low level of coupling. This can explain the success of SAP models. A second conclusion is that coupling partially explains the error probability of process models.

1 Introduction

Quality metrics for business process models have recently gained increasing interest in the research community. We have started our work by adapting software engineering and software metrics to business process models. The first metric we have presented calculated the process complexity [5], i.e. the degree to which a process model is difficult to analyze, understand or explain. High complexity in processes may result in bad understandability, errors, defects, and exceptions, leading processes to need more time to develop, test, and maintain [5]. Therefore, excessive complexity should be avoided. Thus, it is important to develop methods and measurements to identify complex processes during the design phase of process models. Afterwards, these processes can be re-designed to reduce the complexity of related activities.

In the area of software engineering, the quality metrics help to improve a design by increasing its understandability, maintainability, error reduction, manageability and efficiency. Several researchers (Latva-Koivisto [15], Gruhn & Laue [14], Cardoso et al. [7]) have recognized the emerging importance of quality metrics in
business process modeling. Their work focuses on adapting metrics for software engineering to business process models. This can be done since there is a strong analogy between software programs and business processes, as argued before in e.g. [7,13,25]. Therefore, traditional software metrics that were designed to be applied to programs written in languages such as C++, Java, FORTRAN, etc, can be revised and adapted to analyze and study business processes characteristics, such as complexity, understandability, and maintenance [7]. A business process, possibly modeled with a language such as EPML [17], YAWL [2], or BPEL [8], can be seen as a traditional software program that has been partitioned into modules or functions (i.e. activities) that take in a group of inputs and provide some output. Several process modeling tools are available in the market, such as ARIS [3], Protos [23], and ProVision [24], which allow not only the graphical design of process models, but also allow simulating and analyzing models. For example, ARIS, probably the most complete platform available, offers users balanced scorecard and performance indicators to analyze business processes. During the business process design phase, users can model, analyze, and optimize business processes. ProVision and Protos have similar functionality allowing carrying out quantitative and qualitative analyses on costs, throughput times, and processing times. We believe that current modeling tools should provide more analytical capabilities. Namely they should provide mechanisms to analyze the coupling, complexity, and cohesion of process models.

Business process metrics would help designers to make the more correct choices when analyzing alternative design solutions. By measuring the process before and after a proposed change we can minimize the risk of the change. These metrics would allow identifying complex and error prone processes [4]. The metrics can be characterized by the number and intricacy of activity interfaces, transitions, conditional and parallel branches, the existence of loops, roles, activity categories, the types of data structures, and other characteristics [8].

In this paper, we show how the notion of coupling, which has been used successfully in software engineering for many years, can be used to quantify the complexity of an EPC business process model. Coupling is a measure for the strength of association established by the interconnections among activities. The degree of coupling depends on how complicated the connections are and also on the type of connections (XOR, AND, OR). It is hypothesized that designs with a high coupling will contain more errors than designs with lower coupling.

The paper is structured as follows. Section 2 presents the work that has been carried out to develop quality metrics for software engineering and relates it with business process engineering. This comparison is illustrated with a mapping from software metrics to process metrics. The next section gives an overview of a coupling metric that is currently being used in software engineering and explains how it can be used to compute the coupling of a process model. Also, the theory behind our general business process coupling metric is presented. Section 4 applies this general approach to EPCs. We test the coupling metric with the SAP reference model to inquire about the degree of coupling used in processes that
represent good design practices. Finally, the last section presents our conclusions and future work.

2 The relationship between Software Engineering and Business Process Metrics

In the software engineering field many researchers have studied and developed metrics to measure the quality of a software program design. In this section, we first give an overview of the classification of the software quality metrics found in literature. Next, we show why we think software quality metrics can also be adapted to business process modeling, by elaborating on the similarities between those two areas. Finally, we give a summary of the “state of the art” in the area of business process metrics, at the time of writing this paper.

2.1 Classification of Software Engineering quality metrics

Research on quality metrics in software engineering ranges back to the 1970’s. Well known names in this area are Fenton and Melton [12], Selby and Basili [26], Myers [20], Troy and Zweben [27], and Conte, Dunsmore and Shen [9]. The latter two references provide a classification of the quality metrics. The quality of a software design can be categorized into five design principles:

- **Coupling** - Coupling is measured by the number of interconnections among modules. Coupling is a measure for the strength of association established by the interconnections from one module of a design to another. The degree of coupling depends on how complicated the connections are and also on the type of connections. It is hypothesized that programs with a high coupling will contain more errors than programs with lower coupling.

- **Cohesion** - Cohesion is a measure of the relationships of the elements within a module. It is also called module strength. It is hypothesized that programs with low cohesion will contain more errors than programs with higher cohesion.

- **Complexity** - A design should be as simple as possible. Design complexity grows as the number of control constructs grows, and also as the size - in number of modules - grows. The hypothesis is that the higher the design complexity the more errors the design will contain.

- **Modularity** - The degree of modularization affects the quality of a design. Over-modularization is as undesirable as under-modularization. The hypothesis is that low modularity generally relates to more errors than high modularity.

- **Size** - A design that exhibits large modules or a deep nesting is considered undesirable. It is hypothesized that programs of large size will contain more errors than smaller programs.
2.2 Similarities between software programs and business processes

Based on a number of similarities between software programs and business processes, we think it is viable to adapt and revise the well-known quality metrics from software engineering to business process modeling. These similarities previously have been distinguished by [7,13,25]:

1. They both focus on information processing. Within each step, one or more outputs are produced on the basis of one or more inputs.
2. They have a similar compositional structure. A program can be split up into modules or classes. Every module consists of a number of statements, and every statement contains a number of variables and constants. Likewise, a workflow process has activities. Each activity is composed of elementary operations and each operation uses one or more pieces of information to produce new information.
3. Their dynamic execution is derived from a static structure. When instantiating either a software program or a workflow process, an execution flow of their elements takes place in accordance with their static representation. This flow may involve consecutive executions, concurrency, conditional routings, etc.

Since business process models, and workflow models in particular, have much in common with software programs it is easily understood that the software quality measures also have potential to improve business process design. For further explanation of the “state of the art” in business process modeling metrics in the next section, we adopt the presented classification from software quality metrics since there is no other classification in the BPM area yet. However, we would like to note that several researchers use some of these design principles as a way to measure complexity (e.g. size and coupling are used to determine the complexity of a process model [7,14,15,19]).

2.3 State of the art in Business Process Metrics

At this moment three surveys are available in which researchers describe the opportunities to apply software engineering metrics, and in particular complexity, to BPM. They also investigate how these software engineering metrics can be adopted to the BPM field ([7,14,15]). However, none of these metrics has been made very concrete and practical yet, they have not yet been tested and a number of limitations is identified for each metric. With the design principles of the software engineering domain in mind, we give an overview of the research that has been done so far on quality metrics in business process modeling:

– Coupling - Coupling measures the number of interconnections among the modules of the model. We have identified two studies using a coupling measure to determine the quality of a business process design. Although Mendling [19] calls his metric a complexity metric, he is actually measuring the interconnections among several tasks in a business process model based
on theory of social network analysis. Thus, his metric should be classified as a coupling metric. Moreover, Reijers and Vanderfeesten [25] also developed a similar coupling metric counting the overlap of data elements for each pair of activities. However, both metrics do not deal with how complicated the connections are (see the definition of coupling in section 2.1).

- **Cohesion** - Cohesion measures the coherence within the parts of the model. Reijers and Vanderfeesten [25] developed a cohesion metric for workflow processes that looks at the coherence of the steps within the activities of the process model.

- **Complexity** - Complexity measures the simpleness and understandability of a design. In this area most of the research on business process metrics has been done [15,14,7] since there is also a wide range of complexity measures for software engineering. For instance, both [14] and [7] consider the adaptation of McCabes’ cyclometric number as a complexity metric to the BPM field.

- **Modularity** - Modularity measures the degree to which a design is split up into several modules. From our literature review we have to conclude that there is no research done yet on modularity metrics for business process models.

- **Size** - Size simply measures how big a model is. The size of a business process model can be measured by simple measures similar to the number of Lines of Code in software engineering metrics. Cardoso et al., Gruhn and Laue and Latva-Koivisto [7,14,15] all propose to count the number of activities to establish this measure for instance.

From this overview of the state of the art in business process metrics, we conclude that this field of research is just at its start and that there is a lot of potential for further development of business process metrics. As we have noted before, the classification is not yet very precise. Mendling uses a coupling metric as means to calculate complexity and Latva-Koivisto, Gruhn & Laue, and Cardoso et al. also use size as a measure for complexity [7,14,15]. Perhaps, this classification of business process metrics should be revised when this area is more mature.

3 A coupling metric for business processes

Coupling is considered as the most important quality metric in the area of software engineering [27]. Troy and Zweben have concluded from their experiments that coupling is a good predictor of errors in a software design. Because the research on business process metrics is very new, we decided to “build” on the knowledge from software engineering and start with the development of a metric that already has shown its importance in another area. In our approach we only look at the control flow of a process (and not at the data flow), which makes the earlier approach of Reijers and Vanderfeesten [25] unsuitable. For the other existing approach to coupling, Mendling’s density metric [19], we have identified some issues that will be solved by a new metric that we propose in this section. First, we go into more detail of the density metric and we explain the
issues. Then, we elaborate on the new coupling metric and give some illustrative examples.

3.1 The Density Metric

In [19], Mendling describes a density metric to measure complexity of process models and in particular EPCs. When looking at this complexity metric in more detail, it actually turns out to be a way to calculate the degree of coupling. In the paper, Mendling first introduces a general description of the density metric and elaborates on the definitions of the metric for three specific modeling languages (i.e. Petri-Nets, YAWL models, and EPC’s). However, each of these languages has a slightly different implementation of the general idea to calculate the density of a process model. Density is determined by the ratio of the number of arcs in the model divided by the number of all possible arcs between the elements of the process model (i.e. the number of all possible combinations of two elements in the process model). For a workflow net \[ W = (P, T, A) \] the density metric looks like:

\[
d_W = \frac{|A|}{|P| \cdot |T| + |T| \cdot |P|}
\]

with \( P \) the set of places, \( T \) the set of transitions and \( A \) the set of arcs between places and transitions.

However, we have identified two issues in this approach to calculate the degree of coupling in a process model:

1. The density metric also includes self-loops (i.e. a link from an activity to itself), while we believe coupling should be considered only between different activities.
2. Process models have connectors with different semantics (AND, OR, XOR connectors). The density metric does not consider the impact of their different meanings while we feel they may have more or less impact according to their degree of tight coupling. The definition of coupling in the software engineering field also mentions these different types of connections and their weights in the overall calculation of the coupling value for a design.

In the remainder of this paper we present our solution to the two issues mentioned above and we test our ideas on the SAP reference models [10]. Moreover, the coupling metric we propose in the next section is more general and can be used, without adaptation, for all kind of business process modeling languages.

3.2 Coupling Metric

In this section we incrementally describe the development of a coupling metric that solves the problems mentioned. First, we look at the self-loops. Starting from a different angle we have come to a similar metric like the density metric but we do not consider a self-loop as a way of coupling. In our solution, we also count the number of arcs, by looking at all connected pairs of activities,
and we divide by the total number of possible connected pairs, leaving out the pairs containing the same elements, i.e. \(|T| \ast (|T| - 1)\). This is represented by the following general equation.

\[
CP = \frac{\sum_{t_1, t_2 \in T} connected(t_1, t_2)}{|T| \ast (|T| - 1)}
\]

where

\[
connected(t_1, t_2) = \begin{cases} 
1 & \text{if } (t_1 \rightarrow t_2) \land (t_1 \neq t_2) \\
0 & \text{otherwise}
\end{cases}
\]

Activities \(t_1\) and \(t_2\) are connected \((t_1 \rightarrow t_2)\) when there is a directed arc going from \(t_1\) to \(t_2\). In Figure 1, a number of examples is given including their value for this coupling metric.

\[
\begin{align*}
\text{(a) } & cp = 0 \\
\text{(b) } & cp = 0.333 \\
\text{(c) } & cp = 1
\end{align*}
\]

**Fig. 1.** Some examples of process models and their values for the coupling metric.

Note that \(\sum_{t_1, t_2 \in T} connected(t_1, t_2)\) actually expresses the number of arcs between transitions and therefore is similar to \(|A|\) in the density metric. Here, we use a sum notation because it makes the next step to include different weights for different types of connectors easier to understand.

Secondly, we also deal with weighted connectors. We feel that the AND, OR, and XOR, can be ranked based on the degree to which they bind two activities together. This can be easily understood by looking at it from a probabilistic point of view. We chose the weight of a branch for coupling equal to the probability that this branch is executed. Because we do not know about the probabilities for execution of a certain branch in a model at runtime, we assume they are
Fig. 2. Some examples of business process model (fragments) and their value for the coupling metric.
uniformly distributed. The weights for each branch can then be determined as follows:

- the AND is the strongest binder, because every branch of the AND connector is followed in 100% of the cases. Thus, the probability of following a particular branch is 1. Figure 2(a) presents a small process model with an AND-connector. After A has been executed, always B and C have to executed as well. Therefore, the branch from A to B and the branch from A to C both have a probability of 1 to be followed (and thus a weight of 1).

- the XOR is the weakest binder, because in any case only one of the branches is followed. Thus, the probability of following a particular branch is \( \frac{1}{m \cdot n} \), where \( m \) is the number of ingoing branches and \( n \) is the number of outgoing branches. The process model in Figure 2(c) has two alternative ways: either the branch of A to B is followed, or the branch from A to C. They can never be followed both at the same time. Because of our assumption that the two branches have equal chances of being followed, their probability is \( \frac{1}{2} \). And thus, the weight of each branch in the XOR case of Figure 2(c) is \( \frac{1}{2} \).

- the OR must have a weight in between the AND and XOR, since one does not know upfront how many of the branches will be followed. It could be that they are all followed (cf. AND situation), that only one branch is followed (cf. XOR situation), but it could also well be that several arbitrary branches are followed. The weight of an arc is therefore dependent on the probability that the arc is followed. In case of an OR there are \((2^m - 1) \cdot (2^n - 1)\) combinations of arcs that can be followed. One of them is the AND situation, for which the probability then is \( \frac{1}{(2^m - 1)(2^n - 1)} \). All the other combinations \((\frac{2^m - 1}{(2^m - 1)(2^n - 1)} - 1)\) get the weight of a XOR \(\frac{1}{m \cdot n}\).

Thus, in total, the weight of an arc going from one activity to another activity via an OR connector can be calculated by:

\[
\frac{1}{(2^m - 1)(2^n - 1)} + \frac{(2^m - 1)(2^n - 1) - 1}{(2^m - 1)(2^n - 1)} \cdot \frac{1}{m \cdot n}
\]

Figure 2(b) shows an example. The weight for each connection is:

\[
\frac{1}{(2^1 - 1)(2^2 - 1)} + \frac{(2^1 - 1)(2^2 - 1) - 1}{(2^1 - 1)(2^2 - 1)} \cdot \frac{1}{1 \cdot 2} = \frac{1}{2} + \frac{2}{3} \cdot \frac{1}{2} = \frac{2}{3}
\]

In short, the probability of following a branch is used as a weight for each arc to determine the degree of coupling between two activities. We again formalize this using the general equation. However, in this case the value for \( \text{connected}(t_1, t_2) \) may be different and ranges between 0 and 1 because the weight of an arc is dependent on the probability that this arc is followed:

\[
CP = \frac{\sum_{t_1, t_2 \in T \text{ connected}(t_1, t_2)}}{|T| \cdot (|T| - 1)}
\]

where
\[ \text{connected}(t_1, t_2) = \begin{cases} 
1 & \text{if } (t_1 \rightarrow t_2) \land (t_1 \neq t_2) \\
1 & \text{if } (t_1 \rightarrow \text{AND} \rightarrow t_2) \land (t_1 \neq t_2) \\
\frac{1}{(2^m-1)(2^n-1)} + \frac{(2^n-1)(2^n-1)-1}{(2^m-1)(2^n-1)} \cdot \frac{1}{mn} & \text{if } (t_1 \rightarrow \text{OR} \rightarrow t_2) \land (t_1 \neq t_2) \\
0 & \text{if } (t_1 \rightarrow \text{XOR} \rightarrow t_2) \land (t_1 \neq t_2) \\
0 & \text{if } (t_1 = t_2) 
\end{cases} \]

in which \( t_1 \) and \( t_2 \) are activities, \( m \) is the number of ingoing arcs to the connector, and \( n \) is the number of outgoing arcs from the connector.

Note that the idea of having different types of coupling, being weaker or stronger binders, already appeared in the area of software engineering metrics by Fenton and Melton [12]. Also note that we assume here that connectors are never directly linked to each other. There is always an activity in between.

4 Applying the coupling metric to EPCs

Having devised a theoretical metric to calculate the degree of coupling among the tasks of a process model, in this section we show how the metric can be used to empirically study the SAP Reference Model [10]. This reference model includes 604 EPC business process models [18]. In this section we carry out the same experiment described by Mendling in [19] to compare our CP metric that we have devised with the previous density metric. Mendling has found that there is some empirical evidence that the density metric could serve as a predictor for error probability. From the 604 EPC business process models from the SAP reference model, 34 have errors with respect to a relaxed soundness [11]. Based on the EPC models sample, we test the hypothesis that coupling is a determinant of error probability.

The experiment includes two independent variables: coupling (cp) and size. The variable size can take two forms: the number of nodes \((n)\) or the number of arcs \((a)\). These independent variables are used to explain the variance of the dependent variable (error) which indicates if a process model has an error. The dependent variable or response variable is a dichotomy since it takes the value 1 when a process model has an error and takes the value 0 when a process model does not have an error.

We use regression to predict an error in a process model based on the independent variables. Two regression methods can be used: linear and logistic regression. Logistic regression is more suitable to test our hypothesis since when linear regression is used for a dependent binary variable three problems may occur [22]: a) the variance of the error term is not constant, b) the error term is not normally distributed, and c) there is no restriction requiring the prediction to fall between 0 and 1. While the first two problems can be solved using weighted least-square regression, the third problem is insurmountable [22]. Moreover, unlike linear regression, logistic regression makes no assumption about the distribution of the
Fig. 3. Two example models of the SAP production reference model. In total there are 604 of these models in the SAP reference model package.
independent variables. They do not have to be normally distributed or linearly related. Nor does it assume that the dependent variable or the error terms are distributed normally. Just like linear regression, logistic regression gives each regressor a coefficient $b$ which measures the regressor’s independent contribution to variations in the dependent variable.

After data have been modeled, using logistic regression, we need to evaluate the degree of correspondence between the estimated probabilities of error in a SAP process produced by the model and the actual errors in SAP Reference Model. This evaluation can be done using goodness-of-fit statistics. The significance of the logistic model was interpreted using Hosmer-Lemeshow (H-L) and the Nagelkerke’s $R^2$ statistics. As Mendling notes, the SAP Reference Model includes only 5.6% of process models with errors, a correct classification of 94.4% is easily achieved by always predicting that a model is correct.

The Hosmer-Lemeshow statistic groups the predictions of a logistic regression model rather than the model’s predictor variable data. This statistic is a goodness-of-fit test for the null hypothesis that the difference between observed and predicted values is zero. If the H-L goodness-of-fit test statistic has $p = .05$ or less, we reject the null hypothesis that there is no difference between the observed and model-predicted values. This means the model predicts values significantly different from the observed values. Therefore, for an adequate fit, this statistic should be greater .05. Nagelkerke’s $R^2$ statistic ranges from 0 to 1 and indicates how much of the variance is explained by the model in comparison to a base model without predictors. The Wald statistic is an alternative test which is commonly used to test the significance of individual logistic regression coefficients for each independent variable (that is, to test the null hypothesis in logistic regression that a particular coefficient is zero). $\exp(B)$ indicate the change in the odds of an error associated with a one-unit change in the independent variable. Our results are presented in Table 1.

The following conclusion can be drawn from the table. The Wald statistic indicate that all coefficients (CP, n and a) have an impact on error probability since they are significantly different from zero. The Hosmer-Lemeshow statistic goodness-of-fit test for the null hypothesis that the difference between observed and predicted values is zero is less than 5% which is not desirable. The Nagelkerke $R^2$ statistics fails to clearly explain the variance of the dependent variable. As a results, we believe that all coefficients (CP, n and a) can only partially explain the probability of a process model to exhibit an error. This suggests that additional experiments need to be carried out to determine other coefficients that may influence the occurrence of errors. Compared to the results obtained by Mendling when using the density metric to predict errors in the SAP Reference Model [19], the coupling metric shows a similar capability to predict errors. Nevertheless, the CP generates a model with a lower difference between the observed and the model predicted values, since H-L goodness-of-fit test yields a greater value.

One interesting aspect to study is to describe statistically the coupling of the 604 SAP process models. This study can bring further insights on how design-
Table 1.

<table>
<thead>
<tr>
<th>Coefficient</th>
<th>CP Exp(B)</th>
<th>CP Wald Sig.</th>
<th>CP and nodes Exp(B)</th>
<th>CP and nodes Wald Sig.</th>
<th>CP and arcs Exp(B)</th>
<th>CP and arcs Wald Sig.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Constant</td>
<td>0.394</td>
<td>7.893</td>
<td>1.085</td>
<td>12.850</td>
<td>0.141</td>
<td>9.728</td>
</tr>
<tr>
<td>CP</td>
<td>0.000</td>
<td>20.136</td>
<td>0.190</td>
<td>13.872</td>
<td>0.000</td>
<td>9.872</td>
</tr>
<tr>
<td>( n )</td>
<td>-</td>
<td>-</td>
<td>0.000</td>
<td>5.265</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>( n )</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>1.015</td>
<td>3.708</td>
</tr>
<tr>
<td>Hosmer-Lemeshow</td>
<td>0.585%</td>
<td>0.733%</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Nagelkerke ( R^2 )</td>
<td>0.203%</td>
<td>0.230%</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Correct Classification</td>
<td>94.4%</td>
<td>94.7%</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Correct Error Predictor</td>
<td>0</td>
<td>2</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
</tbody>
</table>

ERS usually construct their process and what are their associated levels of coupling. Our idea is to define recommended coupling values that can be used to monitor and control the development of process models. After carrying several nonparametric tests using Kolmogorov-Smirnov goodness-of-fit test we concluded that coupling did not followed a normal, uniform, Poisson, or exponential distribution. Therefore, we constructed a histogram (frequency graph) and visually identified that coupling follows a Weibull distribution for the SAP process models. Coupling had a mean of 0.0914 and a standard deviation of 0.0815. Weibull distributions are generally used to represent manufacturing and delivery times in industrial engineering problems, weather forecasting, reliability engineering and failure analysis. We can safely state that SAP designers construct process models with a coupling that follows a Weibull distribution.

5 Conclusion

The same way software engineering metrics have an important role in software development, business process metrics have also an imperative responsibility in business process modeling. Since the research on metrics for business process models is a rather new area, we present the state of the art of business process metrics. Having this survey in mind we present a new metric to evaluate the degree of coupling in a process model. This metric measures the number of interconnections among the activities of a model. The basic principal behind our coupling metric is that a process model with a high number of interconnections has a higher degree of coupling among activities. The metric is particularly useful to determine if a process model is difficult to be adapted to meet new requirements, since processes with a higher coupling are more difficult to change. To show the practical use of our coupling metric, we show its use with EPC business process models and tested it to characterize the processes of the SAP reference model. The results of this experiment gave us the evidence that SAP models have a very low level of coupling with a mean of 0.0914 and a standard deviation of 0.0815. This can explain the success and widespread use of SAP models. Furthermore, the coupling of the processes studied follows a Weibull
distribution. This can indicate that engineers tend to design process models with a lower level of coupling. Finally, we tested the coupling metric to predict errors in the SAP reference model. We found that coupling partially explains the error probability of process models. Nevertheless, we feel that additional experiments in this area need to be carried out to determine other factors that cause errors.

Acknowledgement

This research is partly supported by the Technology Foundation STW, applied science division of NWO and the technology programme of the Dutch Ministry of Economic Affairs. We would also like to thank Délia Gouveia for conducting the statistical analysis described in this paper.

References