A Tool for Evaluating the Quality of Business Process Models

Lobna Makni, Wiem Khlif, Nahla Zaaboub Haddar, Hanène Ben-Abdallah

Mir@cl Laboratory, Faculty of Economics and Management Sciences,
Sfax University, Tunisia

{Lobna.Makni,Wiem.Khlif,Nahla.Haddar,Hanene.Benabdallah}@fsegs.rnu.tn

Abstract: Modeling business processes is an essential task when aligning, improving or automating existing business processes. To be efficient in such tasks, a business process model must be understandable, reusable and easily maintainable. For assessing the quality of a business process model, a set of quality metrics have been proposed either by adapting some mature software quality metrics, or by defining new metrics specific for business processes. The aim of this paper is to classify the quality metrics proposed so far within a framework defined in terms of design perspectives, and to implement this framework in a tool assisting in the evaluation of the quality of business process models. This tool helps the designers to select a subset of quality metrics corresponding to each design perspective and to calculate and interpret their values in order to improve the quality of their model.

1 Introduction

Quality is one of the major topics in the process modeling domain as much as it is in software modeling. Evidently, obtaining a high quality business process model (BPM) minimizes design flaws, avoids dysfunctions of the process once deployed, and helps the designer to produce a BPM that is easier to understand and maintain. The importance of BPM quality motivated several researchers to propose solutions to improve business process models. In fact, a variety of standards (cf., [SS99] [Gr03]) and frameworks (cf., [NR00] [KSW02]) has been proposed to define, manage, ensure and/or improve the quality of business process models. These standards and frameworks require/use a set of *quality metrics* was proposed to evaluate the quality of a BPM.

According to [ISO98], a quality metric is a quantitative scale and a method that can be used to determine the value taken by a characteristic of a software product. Exploiting the maturity of software quality metrics, several researchers adapted metrics from the field of software engineering for business process models (*cf.,* [CMN+06] [GL06]). The adaptation is justified by the fact that a business process model described by EPC (Event-Driven Process Chain), Petri nets, activity diagrams, BPMN (Business Process Management Notation) [OMG04], or any other modeling language manifests a great number of similarities with software models [VCM+07].

Our contribution in this paper is to propose a classification framework and a tool, baptized *BPMN Quality*, for evaluating the usability of process models. The originality and advantages of our proposition can be summarized in the following five points:

(1) The framework for classifying quality metrics accounts for BPM perspectives (*e.g.,* informational, functional, organizational and behavioral). This provides the designer with a better usage of quality metrics. Indeed, depending on his/her perspective, a designer would be examining only a subset of metrics pertinent to his/her point of view: the selected metrics are defined in terms of business process elements that he/she is interested in. In addition, once a decision is made, for instance, to restructure a BPM to improve a given metric, the designer knows the impact of his/her decision on other metrics dealing with other perspectives; that is, the BPM elements involved in the examined metric provides for traceability among the various perspectives.

(2) The *BPMN quality* tool implements all metrics proposed in the literature. Hence this tool proves for experimental evaluations of these metrics with different process model categories (administrative, workflow, etc.).

(3) *BPMN quality* takes as input a business process represented in XMI [OMG09], the OMG (Object Management Group) standard for model exchange. So, it can be integrated in any modeling tool which supports XMI interchange format.

(4) *BPMN quality* interprets results against various quality dimensions, which assists a designer in making judicious decisions.

The remainder of this article is organized as follows: In section 2, we overview quality metrics defined in the business process domain. In section 3, we first present a packaging of the BPMN meta-model [OMG04] which highlights the concepts pertinent to each perspective and the inter-dependencies among the perspectives. Secondly, we use this packaging to present our classification framework. In section 4, we present our tool *BPMN Quality* which implements the classification framework proposed in section 3 and the metrics algorithms defined in the literature. Finally, we conclude this paper with a summary of the presented work and an outline of its extensions.

2 Overview on current metrics for BPM

Several researchers have already identified the potential of business process metrics. Most of the proposed metrics were adapted from the software engineering domain. Following the software engineering categorization, we present these quality metrics divided into the three Cs categories: Coupling, Cohesion and Complexity.

2.1 Complexity metrics

Informally, *complexity* measures the simplicity and understandability of a design. From this angle, [CMN+06] and [GL06], for instance, adapted McCabe's cyclomatic number [McC76] as a complexity metric for business processes called *CFC* (Control Flow Complexity). This metric evaluates the complexity introduced by the XOR, OR, and AND split constructs. The main idea behind it is the number of mental states that have to be considered when a designer develops a process. Every split in the model adds to the number of possible decisions as follows:

- Every AND-split in a model adds 1 to the *CFC* metric of the model.
- Each XOR-split with n outgoing transitions adds n to the *CFC* metric of the model.

– An OR-split with n outgoing transitions adds $2^n - 1$ to the *CFC* metric of the model.

In this way, the *CFC* of a business process *P* is calculated as follows:

$$CFC(P) = \sum_{C \in P,\ C \text{ is } AND-split} CFC(c) + \sum_{C \in P,\ C \text{ is } OR-split} CFC(c) + \sum_{C \in P,\ C \text{ is } XOR-split} CFC(c) \quad (1)$$

The greater the value of the *CFC* is, the greater the overall architectural complexity of a process is.

In addition to *CFC*, the authors in [CMN+06] propose to adapt the Halstead measures [KAH+97]. The obtained metrics are called Halstead-based Process Complexity measures (HPC). They measure the length *N*, the volume *V* and the difficulty *D* of a process as follows:

$$\text{Process Length: } N = n_1 * \log_2(n_1) + n_2 * \log_2(n_2) \quad (2)$$

$$\text{Process Volume: } V = (N_1 + N_2) * \log_2(n_1 + n_2) \quad (3)$$

$$\text{Process Difficulty: } D = (n_1 / 2) * (N_2 / n_2) \quad (4)$$

where: n_1 is the number of unique activities, splits and joins, and control-flow elements of a business process; n_2 is the number of unique data variables manipulated by the process and its activities; and N_1 and N_2 are respectively the total number of elements and data occurrences. According to the authors, these measures do not require in-depth analysis of process structures, they can predict rate of errors and maintenance effort, they are simple to calculate and they can be used for most process modeling languages.

The metric of Henry and Kafura [HK81] was also adapted to give the Interface Complexity (*IC*) [CMN+06]. It is calculated for each activity as follows:

$$IC = length * (number\ of\ inputs * number\ of\ outputs)^2 \quad (5)$$

where the length of an activity is defined depending on weather it is represented as a black box or a white box: a black box activity gives information only about its interface; thus, its length is assumed to be equal to 1. On the other hand, the length of a white box activity is equal to the number of its tasks. The advantage of the *IC* metric is that it takes into account data-driven processes [CMN+06].

Also considering activities as white boxes, other researchers define the nesting depth (*ND*) of an activity [GL06] as the number of decisions in the control flow necessary to reach this activity. Accordingly, maximum nesting depth and mean nesting depth are defined for measuring nesting depth which influences the overall complexity of a model: A high nesting depth implies a high complexity.

Furthermore, the cyclicity metric (*CYC*) counts the ratio of the number of nodes (activities and gateways) in a Cycle (*NNC*) to the total number of nodes (*TNN*) in the considered process [Me08]:

$$CYC = NNC / TNN \quad (6)$$

The modularity of business process models [GL06] was been also quantified by the *Modularization* metric which is defined as follow:

$$Modularization=(fan\text{-}in * fan\text{-}out)^2 \quad \textbf{(7)}$$

where fan-in is a count of all sub-processes that call a given sub-process and fan-out is a count of all sub-processes that are called from the sub-process under investigation. In fact, dividing a BPM into modular sub-processes can not only help to make the BPM easier to understand, but it can also lead to smaller, reusable models if modularization is used in a reasonable way [GL06].

The number of activities (tasks and sub-processes) *(NOA)* and the number of activities and control-flow elements *(NOAC)* in a process are two additional complexity metrics adapting the software engineering *LOC* (line of code) metric. They are simple and treat a process activity as a program statement [CMN+06].

Moreover, other metrics were adapted from the graph theory. For example, the authors in [CMN+06] define the Coefficient of Network Complexity *(CNC)* which is the number of arcs divided by the number of nodes (activities, joins, and splits) in a model.

For a detailed evaluation of complexity of a business model, Mendling et al. [Me08] have proposed a density metric inspired by social network analysis. They have considered minimum and maximum number of arcs for a given set of nodes. While the significance of density is promising, the experiment they carried out reveals that there are further metrics needed in addition to density.

Beside the above metrics, a number of complexity metrics were proposed in [RRG+06] using the specification 1.1 of BPMN [OMG04]. These metrics have intuitive formulae; due to space limitations, the reader is referred to [RRG+06] for the detailed definitions.

In addition, it is worth mentioning that several complexity metrics were summarized in [AKW08] and classified into five sub-categories: size, complexity, structure, comprehensiveness and modularization. Moreover, Cardoso proposes in [Ca05] another classification of complexity metrics. He distinguishes four categories: activity complexity, control flow complexity, data flow complexity and resource complexity.

2.2 Coupling metrics

Coupling is defined as being the strength of interconnections between activities in a process model. So far, only a small number of researchers have developed coupling metrics for business process quality evaluation.

The authors in [VCR07] define coupling as being the number of interconnections between activities in a process model. The degree of coupling depends both on how complicated the connections are, and on the type of connections between activities (AND, OR, XOR). The coupling metric CP given by (8) counts all pairs of activities in a process model that are connected to each other [VCR07]:

$$CP = \frac{\sum_{t_1,t_2 \in T} connected(t_1,t_2)}{|T| * (|T|-1)} \quad (8)$$

where t_1 and t_2 are elements of the set of activities T in the BPM, m is the number of ingoing arcs to the connector, n is the number of outgoing arcs and connected (t_1, t_1) is a function that gives a weight for each branch between two activities t_1 and t_2 according to the connection type. This weight is based on the probability that the particular branch is executed. The weights for each branch can then be determined as follows:

- AND is the strongest binder, because every branch of it is followed in 100% of the cases. Thus, the probability of following a particular branch is 1.
- 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 $1/(m*n)$.
- 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. The weight of an arc therefore depends on the probability that the arc is followed. So, there are $(2n\text{-}1)$ $(2m\text{-}1)$ combinations of arcs that can be followed. One of them is the AND situation, for which the probability then is $1/$ $(2n\text{-}1)$ $(2m\text{-}1)$. All the other combinations get the weight of an XOR.

Thus, in total, the weight of an arc going from one activity to another activity via an OR connector can be calculated by [18]:

$$1/((2n-1)(2m-1))+((2n-1)(2m-1)-1)/((2n-1)(2m-1))(1/(m.\ n)) \quad (9)$$

Note that this metric takes into account only control flow and does not deal with data.
Another coupling metric is defined in [RV04], which includes data flow and counts the number of related activities for each activity. An activity is connected to another one if and only if they share one or more information elements. First, the average coupling is determined by adding up the number of connections for all activities and dividing this number by the total number of activities. To get a relative score for this metric, the average coupling is divided by the maximal number of coupling, *i.e.* the number of activities minus one.

2.3 Cohesion metrics

Cohesion metrics measure the coherence within parts of a BPM. So far, there is a very little work dealing with cohesion metrics for business processes available. The most significant one is that of Reijers and Vanderfeesten [RV04] who adapted the cohesion metric as follows: The cohesion of an activity is the product of both the relation and information cohesion. The relation cohesion quantifies how much different operations within one activity are related. It determines for each operation of an activity with how many other operations it overlaps by sharing an input or output. On the other hand, the information cohesion focuses on all information elements that are used either as input or as output by any operation within this activity. It determines how many information elements are used more than once in proportion to all the information elements used. It counts all different information elements that appear in the intersection of a pair of operations, considering all pairs. This number is divided by the total number of information elements in the activity.

In [VRM08], the authors define the cross connectivity metric (CC) which quantifies the ease of understanding and the interplay of any pair of model elements. The term 'Cross-

Connectivity' is chosen because the strength of connections between nodes is considered across all nodes in the model. As a result, CC 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.

In summary, a great number of metrics are proposed in the literature. The only classification provided for them is based on the purpose of their use which is measuring complexity, coupling or cohesion. Evidently, not all the proposed metrics consider the designed business process from a single perspective. In fact, this might hinder their usage. In the next section, first, we will organize BPM concepts in terms of their pertinent perspective; for this, we use the meta-model of the standard notation BPMN [OMG04]. Secondly, we will use this organization to classify the metrics according to different BP perspectives.

3 A quality metric framework for BPMN

3.1 Modeling perspectives in BPMN

The representation of models from different perspectives is a vital modeling approach to apprehend for instance the model complexity [FGH+94]. A particular perspective can be designed based on analytical criteria or subjective needs [Pe07]. For instance, a model must be capable of providing various information elements such as what activities the process comprises, who is performing these activities, when and where are these activities performed, how and why are they executed, and what data elements do they manipulate [GG01].

To represent such information, Curtis et al. propose four perspectives [CKO92]: functional, behavioral, organizational, and informational. These perspectives have been widely adopted by several researches. For instance, in [Ja94] the authors propose an approach of perspective oriented process modeling called POPM which can be used for any (basic) process modeling language.

The *functional* perspective represents what process elements (*i.e.*, activity, sub-process and tasks) are being performed, and what flows of informational entities (*e.g.*, data, artifacts, products) are relevant to these process elements [CKO92]. Given this definition, the BPMN main concept that reflects the functional perspective is *Activity*. Hence, as illustrated in Fig. 1, the BPMN meta-model part that can be used to model this perspective regroups those meta-classes related to (inheriting from) the activity meta-class.

On the other hand, the *behavioral* perspective represents *when* process elements are performed (*e.g.,* in sequence), and *how* they are performed through feedback loops, iteration, complex decision- making conditions, entry and exit criteria, and so forth [CKO92]. Therefore, the base meta-classes in this perspective are *Gateway*, *FlowNode*, *SequenceFlow*, and the elements in the package *Collaboration* from the *Analysis Model* of BPMN. As for the *organizational* perspective, it represents *where* and *by whom* process elements are performed, the physical communication mechanisms used for entity transfers, and the physical media and locations used for storing entities [LK06].

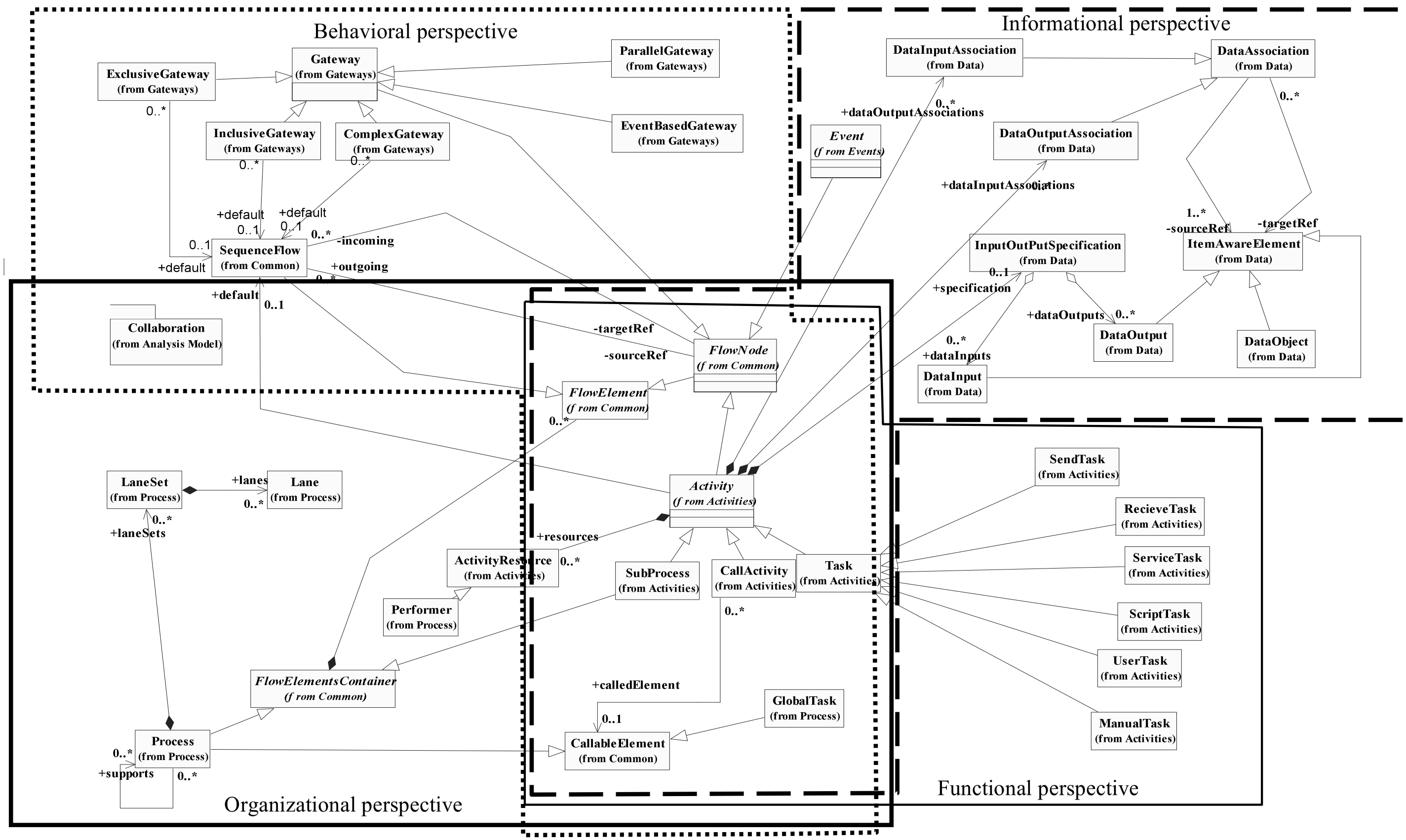

Figure 1: Perspective elements in the BPMN meta-model

Finally, the *informational* perspective represents the informational entities produced or manipulated by a process; these entities include data, artifacts, products (intermediate and end), and objects; this perspective includes both the structure of informational entities and the relationships among them [LK06]. Hence, as illustrated in the corresponding package (Fig. 1), the informational perspective is represented in terms of data and event related classes (*e.g.*, DataOutput, Event, etc.).

3.2 A classification framework

The metric classification framework which we propose is given in Fig. 2. A metric is characterized by its algorithm and the attribute *derivation* to indicate whether the metric is a derived one. Indeed, each metric may be either a direct or an indirect (i.e., a derived) metric. According to the standard 1061 [KB04], a direct (or fundamental) metric depends only on one model element. For example, *NL* (Number of lanes) is a direct metric, while *CFC* (see section 2) is derived. A direct metric provides a direct idea on the quality of one BPM element and on the interpretation of the metric value, while a derived metric captures information about more than one model element. It gives a measured value that gives indirectly a general/an aggregated view of the quality of the BPM.

The association concerns between the classes Metric and BaseElement defines those elements of BPMN (version 2.0) which are considered by this metric. For example the CFC metric must be linked to the element sequence flow. The class Metric is also linked to the class Dimension through the association informs about which associates each metric with the quality dimensions about which the metric informs. For example, the CFC metric must be combined with the dimensions comprehension and maintainability.

Moreover, Fig. 2 shows possible specializations of the class *Metric* which correspond to the three metrics categories: complexity, coupling and cohesion. The property *incomplete* tagging the generalization/ specialization relationship means that other metric categories may be added.

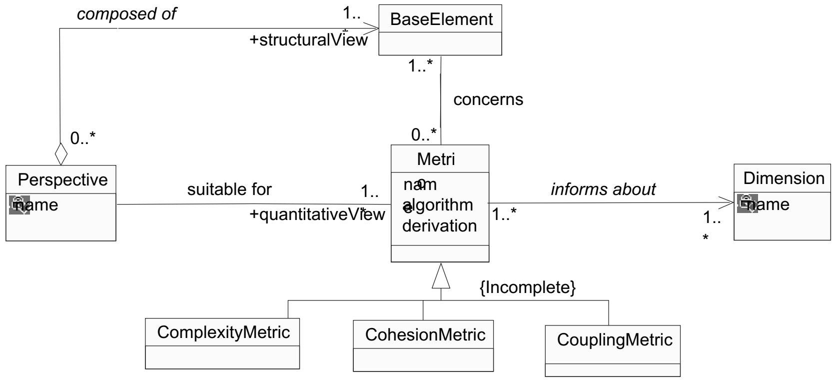

Figure 2: Quality metrics classification framework.

Finally, the third association of class *Metric* links it to the class *Perspective*. It establishes a link between each metric and all perspectives in which it may be calculated. From a perspective view point, the collection of metrics associated with it constitutes a quantitative view of that perspective.

4 BPMN Quality Tool

Based on the classification framework presented in Section 4, we have developed a tool for assessing quality of business process models named *BPMN Quality*. It implements all the metrics defined in the literature.

BPMN Quality is developed in JAVA and is composed essentially of four modules (see Fig. 3): a BPM elements extractor, perspectives constructor, metrics calculator and an interpreter.

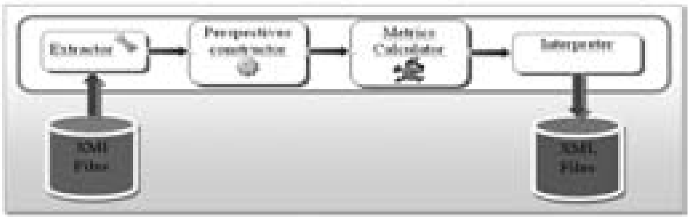

Figure 3: Functional architecture of BPMN quality.

The *extractor* receives as input a BPM transformed into XMI [10]. This transformation aims to obtain a BPM in a standard exchange format. The information extracted by this module involves all elements contained in the BPM. The use of the standard ensures that our tool can be integrated within any other modeling tool that supports this standard.
The information provided by the *extractor* is passed to the perspectives *constructor*. By taking into account the perspective chosen by the user, this module generates from the BPM data, the tree containing all elements which belong to this perspective.

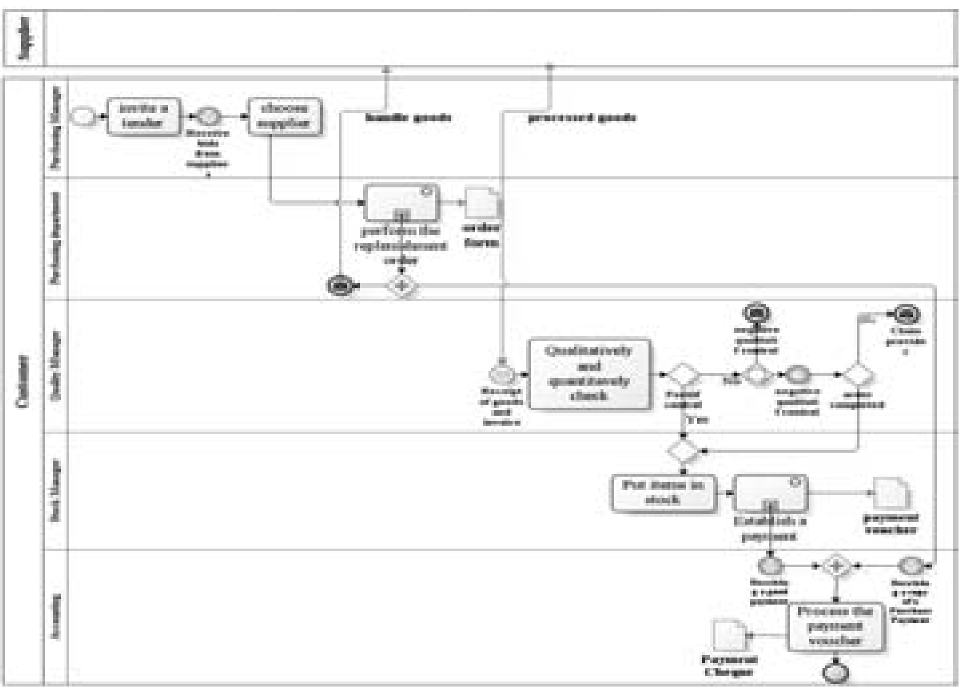

Figure 4: Supply management process.

The metric *calculator* is a module that implements all the metrics calculation algorithms. It uses the information provided by the perspectives *constructor* and the category of metric chosen by the user in order to calculate and display metrics values corresponding to these choices and saves them as a XML file. The structure of this file is the DTD issued from our classification framework.

In order to illustrate the functioning of the calculator module, we apply it to the *supply management process* model of Fig. 4. If the analyst selects the 'Complexity' metric category, the perspective type *Organizational* and the *Derived* metric type, the list of metrics which is associated to these levels will be displayed. Fig. 5 shows the interface associated with this choice and the obtained metrics values.

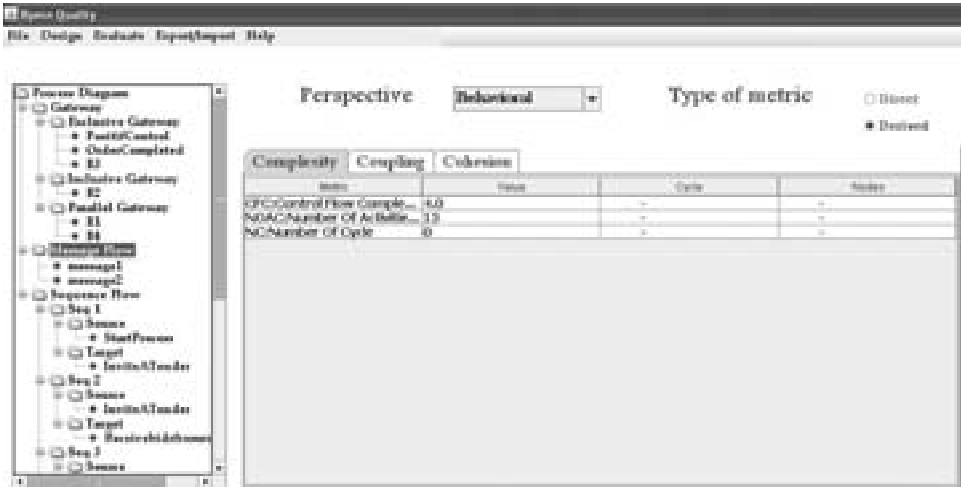

Figure 5: Interface associated with the choice of the analyst.

On the other hand, the *interpreter* module compares the obtained results to threshold values of metrics introduced by the user. According to the quality dimensions associated with the chosen metrics, the *interpreter* gives an evaluation of the quality of the business process under analysis. The comparison process is carried out as follows:

- Through the interface presented in Fig. 6, the *interpreter* displays for each of the quality dimensions maintainability, comprehension, reuse and redesign the associated metrics according to our classification framework (see Fig. 2). The relationships between a quality dimension and the metrics upon which it depends are deduced from literature (cf., [CMN+06] [GL06] [VCM+07]). Then the designer can choose one or more metrics that have to be included in the model.
- The *interpreter* proposes a priority order between the chosen metrics. This order is deduced from the researches presented in literature [CMN+06] [GL06] [VCM+07] based on the relationship between the metrics and the dimensions. In addition, it reflects the fact that a metric is used more than another in order to evaluate the BPM quality for the considered dimension. However, the order proposed by the *interpreter* can be redefined by the designer.
- The *interpreter* allows the user to introduce for each metric a threshold that reflects its optimal value. This value may be found as a result of empirical studies.

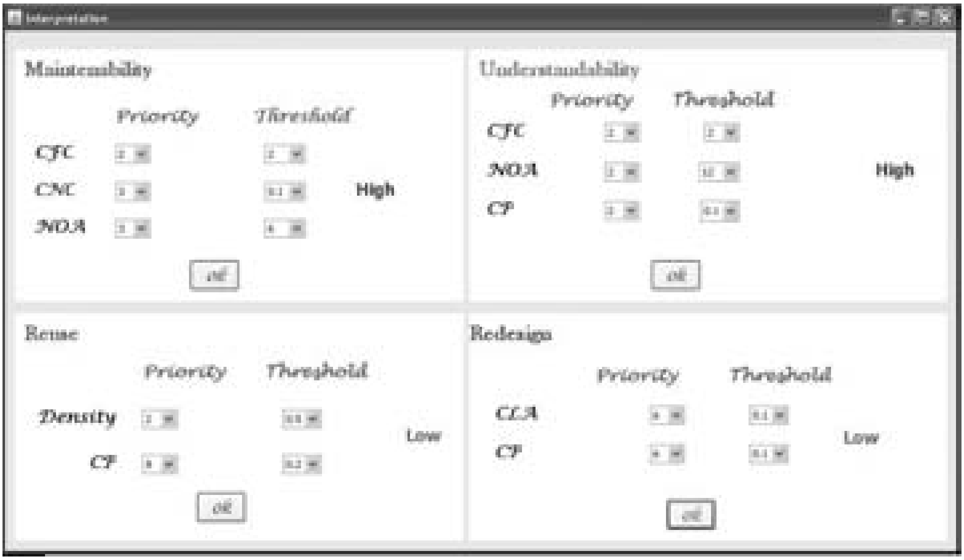

Figure 6: Results interpretation.

– If all necessary values are provided by the user, the *interpreter* computes for each quality dimension D_k the sum σ_k of the pertinent metric values computed by the *calculator* module. The sum is weighted by the metric priorities. Formally, σ_k is calculated as follows:

  - Let $m_1, m_2, \ldots, m_n$ be the metrics associated to a given quality dimension and $p_1, p_2, \ldots, p_n$ the assigned priorities (a high value corresponds to a high priority). The weight α_i of each metric is calculated by the following formula:

$$\alpha_i = \frac{p_i}{\sum_{i=1}^{n} p_i} \tag{9}$$

  - Let $v_1, \ldots, v_n$ be the metrics values determined by the *calculator* module. Then:

$$\sigma_k = \sum_{i=1}^{n} \alpha_i v_i \tag{10}$$

– After computing the weighted sum, the *interpreter* computes for the same quality dimension D_k the sum $\overline{\sigma}_k$ of the metrics thresholds values which are fixed by the designer. These values are weighted by the metric priorities α_i. Formally, $\overline{\sigma}_k$ is calculated as follows:

  - Let $t_1, \ldots, t_n$ the thresholds associated respectively by the designer to the metrics $m_1, \ldots, m_n$. Then:

$$\overline{\sigma}_k = \sum_{i=1}^{n} \alpha_i t_i \tag{11}$$

– Finally the comparison between σ_k and $\overline{\sigma}_k$ provides an assessment (*high*, *medium* or *low*) of the quality dimension D_k. Fig. 6 shows the interpretation of the metric values obtained for the dimensions maintainability, redesign, reuse, and understandability.

5 Conclusion

In this paper, we have presented *BPMN Quality*, a tool for assessing quality of business process models. This tool implements a classification framework organizing in three-levels the quality metrics of BPM. At the first level, the metric belongs to one of three categories: complexity, coupling and cohesion. At the second level, for each category, we associate metrics to perspectives (functional, organizational, informational and behavioral). Thanks to this second level, our framework helps the designer to select the suitable subset of quality metrics dealing with his/her perspective. At the third level, we classify quality metrics into direct and indirect (derived). On the one hand, direct metrics give a direct idea on the quality of BPM; on the other hand, derived metrics provide an aggregate value that gives indirectly a general view on the BPM quality. It is important to note that our framework is extensible, so further perspectives and metric types may be considered.

Our future work focuses on adding another plug-ins to *BPMN Quality* such as an optimization module in order to alert the designer of potential impacts of their decisions upon the various perspectives.

References

[AKW08] Abdul, G.A, Koh T.W, and Wong P.W: Complexity Metrics for Measuring the Understandability and Maintainability of Business Process Models using GQM. International Journal of Computer Science and Network Security, VOL.8 No.5, 2008.

[Ca05] Cardoso, J.: How to Measure the Control-flow Complexity of Web processes and Workflows, Workflow Handbook 2005, ISBN:0-9703509-8-8; pp. 199-212, 2005.

[CKO92] Curtis, B., Kellner, M. Over, J. Process Modeling. Communication of the ACM, 35(9), 1992.

[CMN+06] Cardoso, J.; Mendling, J.; Neuman, J.; Reijers, H.A.: A discourse on complexity of process models, In: Eder, J.; Dustdar, S. et al, editors, BPM 2006 workshops. Lecture Notes in Computer Science 4103, Springer-Verlag, Berlin; pp. 115-126, 2006.

[FGH+94] Finkelstein, A.; Gabbay, D.; Hunter, A.; Kramer, J.; Nuseibeh, B.: Inconsistency handling in multi-perspective specifications. IEEE Transactions on Software Engineering, 20; pp. 569-578, 1994.

[GG01] George M. Giaglis, A: taxonomy of business process modeling and information systems modeling techniques, International Journal of Flexible Manufacturing Systems, Springer,2001.

[GL06] Gruhn, V.; Laue, R.: Complexity metrics for business process models, In: Witold Abramowicz and Heinrich C. Mayer, editors, 9th international conference on business information systems, vol. 85 of Lecture Notes in Informatics; pp. 1-12, 2006.

[Gr03] Group, W.W.: Qos for web services: Requirements and possible approaches, 2003.

[HK81] Henry, S.; Kafura, D.: Software structure metrics based on information-flow. IEEE transactions on Software Engineering, 7(5); pp. 510–518, 1981.

[ISO98] ISO/FCD 9126-1.19, Software quality characteristics and metrics, Information Technology Part 1: Quality characteristics and sub characteristics, ISO/FCD 9126-1. 1998.

[Ja94] Jablonski, S.: Mobile: a modular workflow model and architecture, Noordwijkerhout, The Netherlands Digital Equipment GmbH, AIT Karlsruhe, September 1994.

[KAH+97] Khoshgoftaar, T.M.; Allen, E.B.; Halstead, R.; Trio, G.; Flass, R.: Process Measures for Predicting Software Quality. 1997 High-Assurance Systems Engineering Workshop; pp. 155-161, 1997.

[KB04] Kaner C.; Bond W. P.: Software Engineering Metrics: "What Do They Measure and How Do We Know?". 10th international software metrics symposium, metrics, 2004.

[KSW02] Kahn B.; Strong D.; Wang R.: Information quality benchmarks: Product and service performance. Communications of the ACM 45; pp. 184-192, 2002.

[LK06] List, B.; Korher, B.: An Evaluation of Conceptual Business Process Modeling Languages, SAC'06, April 23-27, Dijon, France, 2006.

[McC76] McCabe, T.J.: A Complexity Measure. IEEE Transactions on Software Engineering, 2(4); pp. 308-320, 1976.

[Me08] Mendling, J.: Metrics for Process Models: Empirical Foundations of Verification, Error Prediction, and Guidelines for Correctness, Vol. 6 of Lecture Notes in Business Information Processing, Springer, 2008.

[NR00] Naumann, F.; Rolker, C. In: Assessment Methods for Information Quality Criteria, Cambridge, MA, USA; pp. 148-162, 2000.

[OMG04] Object Management Group: Business Process Model and Notation (BPMN), 2004.

[OMG09] Object Management Group: Business Process Modeling Notation, V2.0Beta1, Available Specification OMG Document Number: formal/-08-14 Standard document, 2009.

[Pe07] Peter, L.: Perspectives on Reference Modeling. Reference Modeling for Business Systems Analysis, Idea; pp. 1-20, 2007.

[RRG+06] Rolón E., Ruiz F., García, F. and Piattini M. : Applying software process metrics in business process model. University Center Tampico-Madero, 89336 Tampico, Tamps. México. RPM-AEMES, VOL. 3, Nº 2. ISSN: 1698-2029, September , 2006.

[RV04] Reijers, H. A.; Vanderfeesten,I.: Cohesion and Coupling Metrics for Workflow Process Design. In J. Desel, B. Pernici, and M. Weske, editors, Proceedings of the 2nd International Conference on Business process Management, Lecture Notes in Computer Science, Springer-Verlag, Berlin, volume 3080; pp. 290-305, 2004.

[SS99] Samuel, K.M.; Samuel, K.H.: Operations and Quality Management, 1999.

[VCM+07] Vanderfeesten, I.; Cardoso, J.; Mendling, J.; Reijers, H.A.; Aalst, W.M.P. van der: Quality Metrics for Business Process Models. In: L. Fischer, ed.: Workflow Handbook 2007, Workflow Management Coalition, Lighthouse Point, Florida, USA. 2007.

[VCR07] Vanderfeesten, I.; Cardoso, J.; Reijers, H.A.: A weighted coupling metric for business process models, The 19th International Conference on Advanced Information Systems Engineering (CAiSE 2007), CAiSE Forum, 2007.

[VRM08] Vanderfeesten, I.; Reijers, H.A.; Mendling, J.; Wil, M.P.; Aalst, V.; Cardoso, J.: On a quest for good process models: the cross-connectivity metric. Advanced Information Systems Engineering, CAiSE'08, Montpellier, France, 2008.