World Academy of Science, Engineering and Technology
International Journal of Industrial and Systems Engineering
Vol:6, No:6, 2012

Quality-Driven Business Process Refactoring

María Fernández-Ropero, Ricardo Pérez-Castillo, Ismael Caballero, Mario Piattini

***Abstract*—**Appropriate description of business processes through standard notations has become one of the most important assets for organizations. Organizations must therefore deal with quality faults in business process models such as the lack of understandability and modifiability. These quality faults may be exacerbated if business process models are mined by reverse engineering, e.g., from existing information systems that support those business processes. Hence, business process refactoring is often used, which change the internal structure of business processes whilst its external behavior is preserved. This paper aims to choose the most appropriate set of refactoring operators through the quality assessment concerning understandability and modifiability. These quality features are assessed through well-proven measures proposed in the literature. Additionally, a set of measure thresholds are heuristically established for applying the most promising refactoring operators, i.e., those that achieve the highest quality improvement according to the selected measures in each case.

***Keywords*—**business process model, modifiability, refactoring, understandability

Open Science Index, Industrial and Systems Engineering Vol:6, No:6, 2012 publications.waset.org/610.pdf

I. Introduction

Business processes define a sequence of coordinated activities aimed at archived the common business goals defined by companies and organizations [1]. Business processes are nowadays considered as one of the most important intangible assets for organizations. The main business processes' advantage is that organizations are more agile for adapting their business goals according to the changes in uncertain environments. It allows organizations to keep, or even increase, their competitiveness. For this reason, companies and organizations recognizes as a critical requirement to manage business processes with the adequate quality degree, i.e., without faults that influence understandability or modifiability among others features. Indeed, understandability and modifiability proved to be two of the most important features to accomplish business processes with appropriate quality degrees [2, 3]. A potential and promising solution in order to improve understandability and modifiability of business process models is refactoring techniques. Business process refactoring are a set of techniques and algorithms devoted to change the internal structure of business process models without changing or altering its external behavior. The refactoring techniques make it possible to improve quality features to get business processes models that are more understandable and modifiable by business expert, which saves management costs [4].

María Fernández-Ropero is with the Instituto de Tecnologías y Sistemas de Información (ITSI) at the University of Castilla-La Mancha, Ciudad Real, Spain (phone: 0034926295300; fax: 0034926295300; e-mail: marias.fernandez@uclm.es).

Ricardo Pérez-Castillo is with the Instituto de Tecnologías y Sistemas de Información (ITSI) at the University of Castilla-La Mancha, Ciudad Real, Spain (e-mail: Ricardo.pdelcastillo@uclm.es).

Ismael Caballero is with the Instituto de Tecnologías y Sistemas de Información (ITSI) at the University of Castilla-La Mancha, Ciudad Real, Spain (e-mail: ismael.caballero@uclm.es).

Mario Piattini is with the Instituto de Tecnologías y Sistemas de Información (ITSI) at the University of Castilla-La Mancha, Ciudad Real, Spain (e-mail: mario.piattini@uclm.es).

The side effects of business process quality faults usually become worse when business processes were retrieved through reverse engineering from existing information systems. This fact is due to any reverse engineering technique, which increases the abstraction level, undergoing a loss of semantics. For example, business process models mined by reverse engineering would probably have a vast number of nested gateways, which may imply a higher business process complexity. In these cases, the retrieved business processes could be refactor toward alternative models less intricate, and therefore, more understandable by business experts. As a result, a higher understandability and modifiability of business process models contributes to maintain them with less effort.

Unfortunately, refactoring techniques entail some challenges. One of the most important challenges is to know when a particular refactoring technique should be applied. Business process refactoring applies different refactoring operators to business process models, which replaced existing fragments for equivalent ones improving quality of such models. Most authors often propose in the literature refactoring operators that are applied following one of the two following approaches. The first approach arbitrarily applies the whole set of refactoring operators that are available. On the opposite way, the second approach applies a sub-set of refactoring operators according to an expert decision. Both approaches do not ensure that the applied sub-set of refactoring operators is the most appropriate and consequently provides the greatest gain.

The goal of this paper is to provide a mechanism to detect the most appropriate sub-set of refactoring operators to be applied to business process models attending to quality criteria. The quality-driven business process refactoring makes it possible to obtain a higher improvement in terms of the understandability and modifiability. The understandability and modifiability are evaluated in this paper by using well-proven measures proposed by *Rolon et al* [5, 6], *Cardoso* [7], and *Sanchez-Gonzalez et al.* [2]. This proposal then establishes a set of thresholds for each measure taking into account the theoretical values of these measures, which can be used as indicators for applying a particular set of refactoring operators in each case to improve the business process model.

The remaining of the paper is organized as follows: Section 2 summarizes related work. Section 3 introduces concepts related to quality of business process models (i.e., understandability and modifiability) and presents measures to be evaluated. Section 4 provides the whole set of refactoring operators together with the proposed indicators to be applied in each case in Section 5. Finally, Section 6 discusses conclusions and future work.

II. Related Work

There are many stakeholders in quality issues when modeling business process. This interest has grown in recent years and many researchers have investigated the impact of the characteristics of quality in business process models. All of them conclude that it is necessary to promote the understandability and modifiability in a business process model because they are a key artifact in the development information system. The main objective of all work is to determine which parameters influence the understandability and modifiability as in the work of [2, 3, 6, 8, 9]. These works propose several metrics to measure these quality characteristics as the number of tasks, the number of connectors, etc.

Other works like [4, 10, 11] propose to improve the quality of business process models through refactoring and, thus, making process models better understandable and easier to maintain. To do this, they propose various scenarios that would need to refactor to improve the model and what type of refactoring is necessary to apply.

However, none of the above proposed to use the quality measures of understandability and modifiability to discover those scenarios where it would be necessary to implement each of the refactorings proposed and, thus, be able to apply the most appropriate set of refactorings instead of applying all of them indiscriminately.

III. Background

The quality of product is one of the most important issues currently in the development of Software. A recently published ISO/IEC 25000 provides a guide for the use of the new series of international standards, called Software Quality Requirements and Evaluation (SQuaRE) [12]. SQuaRE is the second generation of standards for the quality of software product, is an evolution of ISO/IEC 9126 and ISO/IEC 14598. Its main objective is to guide the development so software product with the specification and evaluation of quality requirements, establishing criteria for the specification of quality requirements of software product, measurements and evaluation. SQuaRE is comprised of the following division (see **Fig. 1**).

This paper focuses on the quality model division. ISO/IEC [13] classifies the quality attributes of eight characteristic (see Fig. 2). Each characteristic is composed of a set of related subcharacteristics. Among these are understandability and modifiability, both are subcharacteristics of usability and maintainability, respectively.

The definition of these subcharacteristics are: understandability (appropriateness recognisability) is the degree to which users can recognize whether the product is appropriate for their needs; and modifiability is the degree to which a product can be effectively and efficiently modified without introducing defects or degrading performance.

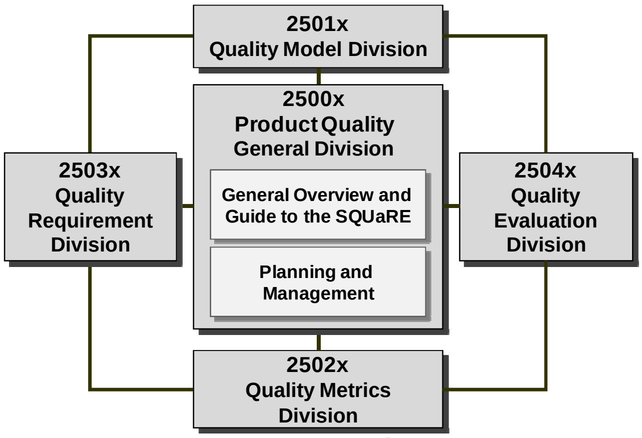

Fig. 1 ISO/IEC 25000 divisions.

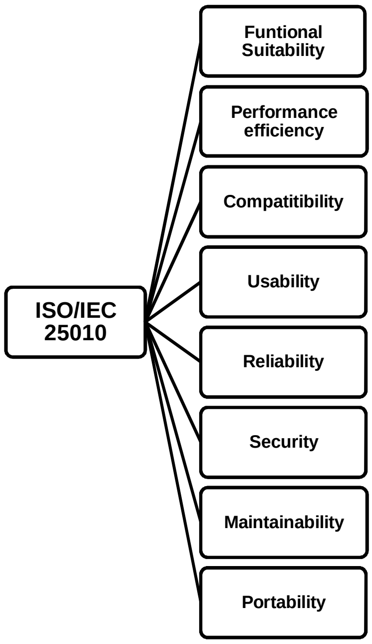

Fig. 2 ISO/IEC 25010 characteristics

To measure these two characteristics can be used measures. Currently there are many measures to measure the understandability and modifiability, that shown in following section, which have been empirically validated through several experiments.

A. Measures to assess the understandability and modifiability

This section shows a set of measures defined to measure the understandability and modifiability of a business process model. The ten first measures have been proposed by *Rolon et al.* in [5, 6] , the eleventh have been proposed by *Cardoso* in [7] and the latter have been proposed by *Sánchez-González et al.* in [2]. Below are their definition and their abbreviations that are used throughout the document.

- *Total Number of sequence flows* (TNSF): This variable is related of number of sequence flows in a business process model.
- *Total Number of events* (TNE): This variable is related of total number of events in a business process model.

- *Total Number of Start Event* (TNSE): This variable is related of total number of start events in a business process model.
- *Total Number of Intermediate Event* (TNIE): This variable is related of total number of intermediate events in a business process model.
- *Total Number of End Event* (TNEE): This variable is related of total number of end events in a business process model.
- *Total Number of gateways* (TNG): This variable is related of total number of gateways in a business process model.
- *Number of sequence flows from event* (NSFE): This variable is related of total number of sequence flows in a business process model whose origin is an event.
- *Number of association flows* (NAF): It is the number of association flows in a business process model.
- *Number of sequence flows from gateways* (NSFG): It is the number of message flows in a business process model whose origin is a gateway.
- *Connectivity level between pools* (CLP): Is the ratio between the number of messages flows between pools and the number of pools.
- *Number of data objects which are outputs of activities* (NDOOut): This variable is related to the number of data object which are output of activities.
- *Number of data objects which are inputs of activities* (NDOIn): This variable is related to the number of data object which are input of activities.
- *Connectivity level between activities* (CLA): This is the ratio between the total number of activities and the number of sequence flows between activities.
- *Control flow complexity* (CFC): This variable is the sum over all gateways weighted by their potential combinations of states after the split.
- *Number of Nodes* (NN): This variable is related to the number of activities and routing elements in a business process model.
- *Diameter* (Di): The length of the longest path from a start node to an end node in the process model.
- *Density* (Den): Relates to the ratio of the total number of arcs in a process model to the theoretically maximum number of arcs.
- *Coefficient of Connectivity* (CC): relates to the ratio of the total number of arcs in a process model to its total number of nodes.
- *Average Gateway Degree* (AGD): expresses the average of the number of both incoming and outgoing arcs of the gateway nodes in the process model.
- *Maximum Gateway Degree* (MGD): captures the maximum sum of incoming and outgoing arcs of these gateway nodes.
- *Separability* (Sep): is the ratio of the number of cut-vertices on the one hand, i.e. nodes that serve as bridges between otherwise disconnected components, to the total number of nodes in the process model on the other.
- *Sequentiality* (Seq): is the degree to which the model is constructed out of pure sequences of tasks.
- *Depth* (Dep): defines maximum nesting of structured blocks in a process model.
- *Gateway Mismatch* (GM): is the sum of gateway pairs that do not match with each other, e.g. when an AND-split is followed by an OR-join.
- *Gateway Heterogeneity* (GH): is the extent to which different types of gateways are used in a process model.

Table I shows each measure and their association with the characteristics of understandability and modifiability, according the authors.

TABLE I
MEASURES OF QUALITY CHARACTERISTICS

Measure	Understandability	Modifiability	Proposed by
TNSF	•		[5, 6]
TNE	•		[5, 6]
TNG	•		[5, 6]
NSFE	•		[5, 6]
NAF	•		[5, 6]
NSFG	•	•	[5, 6]
CLP	•		[5, 6]
NDOOut	•		[5, 6]
NDOIn	•		[5, 6]
CLA		•	[5, 6]
CFC	•	•	[7]
NN	•		[2]
Di	•		[2]
Den	•	•	[2]
CC	•	•	[2]
AGD	•		[2]
MGD		•	[2]
Sep		•	[2]
Seq	•	•	[2]
Dep	•		[2]
GM	•	•	[2]
GH	•	•	[2]

IV. REFACTORING OPERATORS

Below are the most relevant scenarios in which it is necessary to apply a refactoring operator, which have been collected from the literature [4, 10, 14, 15]. Along with these scenarios shows a identifier, for further identification, a brief description that shows an example of application, a brief discussion about it, the measures used for detection of the mentioned in previous section, the refactoring operator that would be necessary to apply and the improvement offered by the application of this refactoring operator in the understandability and/or modifiability.

A. RS1: Names of activities or processes not defined in natural language

1) Description

The activities and processes should have a distinctive name that reveals its purpose. By obtaining the business process model from reverse engineering and considering that the source code was developed following appropriate nomenclature (classes that start in uppercase, lowercase starting methods, no spaces between names, etc), activities and processes have a proper name without spaces. It will need to separate the words and place the first character uppercase as it shown in Fig. 3.

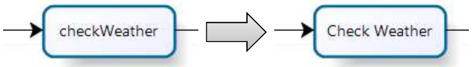

Fig. 3 Refactoring Operator 1

2) Discussion

Studies suggest using a nomenclature based on a verb-object format. In this case, this nomenclature is satisfied and it is only necessary separate the words to fit the natural language.

3) Refactoring Operator

RO1: Separate words considering that an uppercase character indicates the start of a new word and several consecutive uppercase characters indicate a single word (usually short).

4) Measure for its detection

There is no measure for detection. It applies to all activities and processes

5) Improvements after refactoring

Model improves the understandability.

B. RS2: Nesting unnecessary

1) Description

By obtaining the business processes model from reverse engineering increases the possibility of nested gateways that increase the complexity. In these cases, should be substituted for equivalent alternatives more easily understood by users. Thus, increasing the understandability of the model and makes maintenance less expensive. An example is shown in Fig. 4.

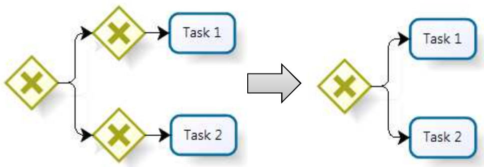

Fig. 4 Refactoring Operator 2

2) Discussion

Several studies show that this complexity in the model leads to a lack of understandability. There are also studies that propose measures to compare the similarity of various structures.

3) Refactoring Operator

RO2: If there are two or more nested gateways of the same type replace with a single gateway.

4) Measure for its detection

Den, NN, TNG, NSFG, Dep, Sep, AGD.

5) Improvements after refactoring

Increase the understandability and modifiability of the model.

C. RS3: Nesting unnecessary 2

1) Description

By obtaining the business processes model from reverse engineering increases the possibility of nested gateways that increases the complexity. In these cases, should be substituted for equivalent alternatives more easily understood by users. Thus, increasing the understandability of the model and make maintenance less expensive. An example is shown in Fig. 5.

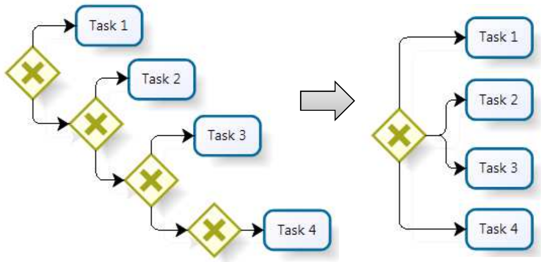

Fig. 5 Refactoring Operator 3

2) Discussion

Several studies show that this complexity in the model leads to a lack of understandability. There are also studies that propose measures to compare the similarity of various structures.

3) Refactoring Operator

RO3: If there are two or more nested gateways of the same type replace with a single gateway.

4) Measure for its detection

Den, NN, TNG, Dep, Sep, AGD.

5) Improvements after refactoring

Increase the understandability and modifiability of the model.

D. RS4. Redundant fragments

1) Description

In a business process model may exist fragment containing the same flow control logic. This makes maintenance more expensive since a change in the model must be propagated to all occurrences of that fragment manually and this may impact matching errors. An example is shown in Fig. 6.

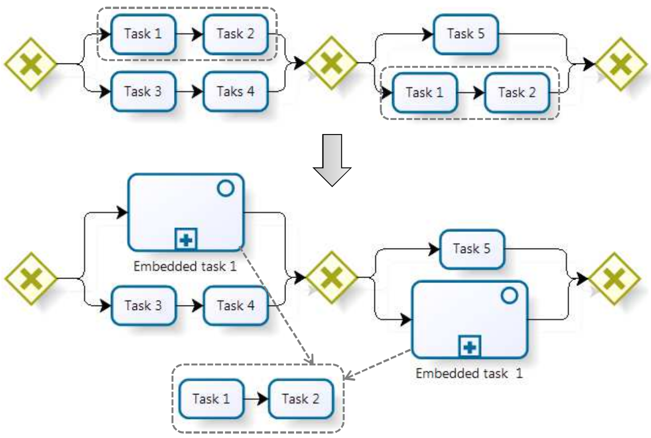

Fig. 6 Refactoring Operator 4

2) Discussion

One of the most common reasons for this redundancy in the models is due to the tendency to use copy-paste. This causes that a simple change in one of them must be reissued manually in each of its occurrences. Numerous articles are recommended to define this fragment only one globally to prevent errors and improve understandability.

3) Refactoring Operator

RO4: These duplicated fragments become a single complex activity that is referenced in the model.

4) Measure for its detection

For their achievement metrics can be used to determine the similarity between two fragments of the model as proposed by *Dijkman* in [4].

5) Improvements after refactoring

Increase the understandability and modifiability of the model.

E. RS5. Consecutive small activities

1) Description

By obtaining the business process model from reverse engineering is possible to obtain many small tasks that not represent large amount of business logic. If there are two or more consecutive activities that access (read and/or write) only to the same data object and they are executed by the same role this can be a symptom that can be grouped into a single activity that represents the same behavior. An example is shown in Fig. 7.

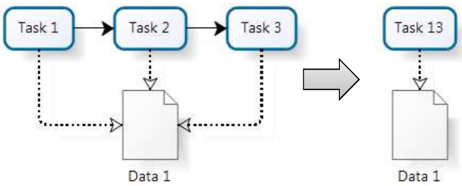

Fig. 7 Refactoring Operator 5

2) Discussion

In the business process models is possible that some activities have similar semantic [15].

3) Refactoring Operator

RO5: These small activities become a single activity with the same behavior.

In this refactoring should be given a name to the new activity that is representative. One option is to put the name of the first and last activity.

4) Measure for its detection

Metric proposed by [15], NAF, NDOOut, NDOIn, Seq.

5) Improvements after refactoring

Increase the understandability and modifiability of the model.

F. RS6. Bad practices in business process modeling

1) Description

By obtaining the business processes from reverse engineering is possible that the models don't follow good practices in the BPMN modeling. In these cases it is recommended to add an exclusive gateway as is shown in Fig. 8.

2) Discussion

In many articles is recommended as good practice to use joints and splitters. The application of these gateways should coincide.

3) Refactoring Operator

RO6: Exclusive gateway is added between these activities. This refactoring should be done before RO4 and RO5.

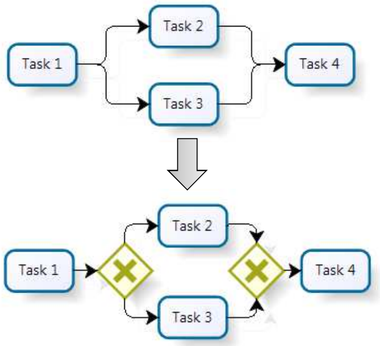

Fig. 8 Refactoring Operator 6

4) Measure for its detection

CLA

5) Improvements after refactoring

Increase the understandability of the model.

G.RS7. Start and/or end events not connected

1) Description

By obtaining the business processes models from reverse engineering is possible that the models are not connected with the start and/or end event as is shown in Fig. 9.

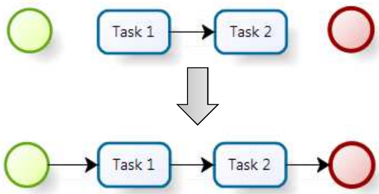

Fig. 9 Refactoring Operator 7

2) Discussion

All business processes models must begin with start event and finish with an end event.

3) Refactoring Operator

RO7: Making the connection between the first activity and the start event. The same way with the last activity and the end event.

4) Measure for its detection

TNE, NSFE, CC.

5) Improvements after refactoring

Increase the understandability of the model.

H.RS8. Several end events

1) Description

By obtaining the business processes models from reverse engineering is possible that there is a more of one end event. This is shown in Fig. 10.

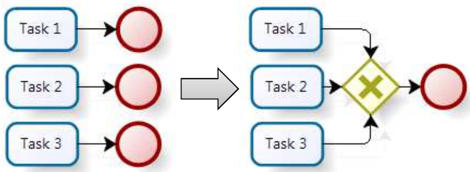

Fig. 10 Refactoring Operator 8

2) Discussion

When there are more than one end event the understandability decrease. For this reason, it is necessary to use similar fragments to provide more understandability.

3) Refactoring Operator

RO8: All end events will be grouped with an exclusive gateway, i.e., the flow finishes when one of the possible paths activates the gateway. Apply after RO7.

4) Measure for its detection

TNEE.

5) Improvements after refactoring

Increase the understandability of the model.

I. RS9. Several consecutive activities

1) Description

By obtaining the business process models from reverse engineering is possible that many activities are obtained in succession. This is because each method is represented as an activity and may be a method invokes another method and so on. An example is shown in Fig. 11.

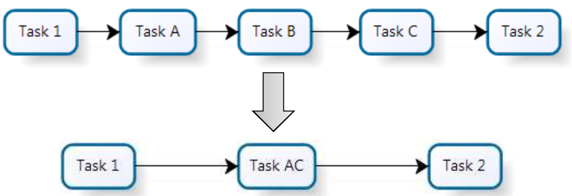

Fig. 11 Refactoring Operator 9

2) Discussion

A sequence of activities that are not relevant can cause lack of understandability in the model. For this reason it is preferable to group activities between the second and the penultimate in one activity.

3) Refactoring Operator

RO9: These activities become a single activity with the same behavior as earlier in succession.

After this refactoring is necessary give a name to the new activity. One option is used the name of the second join penultimate.

In this refactoring should be given a name to the new activity that is representative. One option is to put the name of the first and last activity.

4) Measure for its detection

CLA, Seq.

5) Improvements after refactoring

Increase the understandability and modifiability of the model.

A summary is presented as a matrix of refactoring operators and measures which specify what measures are used for the application of a refactoring operator in a business process model (see Table III).

V. Intervals for the Activation of Refactoring

After describing the refactoring operators with the measures for its detection in Table III is shown which values of these measures have to activate the refactoring. It also shows the improvements achieved in these measures.

TABLE II
RELATIONS BETWEEN MEASURES AND REFACTORING OPERATORS

Measure	RO1	RO2	RO3	RO4	RO5	RO6	RO7	RO8	RO9
TNSF									
TNE							♦	♦	
TNG		♦	♦						
NSFE							♦		
NAF					♦				
NSFG		♦							
CLP									
NDOOut					♦				
NDOIn					♦				
CLA						♦			♦
CFC									
NN		♦	♦						
Di									
Den		♦	♦						
CC							♦		
AGD		♦	♦						
MGD									
Sep		♦	♦						
Seq					♦				♦
Dep		♦	♦						
GM									
GH									
Dijkman				♦					

TABLE III
INTERVALS FOR REFACTORING ACTIVATION

Ref. Op.	Measure for its detection	Intervals for its activation	Improvement in the measures
RO2	Den, NN, TNG, NSFG, Dep, Sep, AGD.	If NSFG = 1 If AGD = 2 If $\frac{TNG}{NN} > \frac{1}{2}$ If Den = 2 & Dep = 2 → *Expert*	$NN_f < NN_i$ $NSFG_f = 1$ $TNG_f < TNG_i$ $Den_f < Den_i$ $Dep_f < Dep_i$ $AGD_f > AGD_i$
RO3	Den, NN, TNG, Dep, Sep, AGD.	If $\frac{TNG}{NN} > \frac{1}{2}$ If AGD = 3 → *Expert* If Den = 2 & Dep = 2 → *Expert*	$NN_f < NN_i$ $NSFG_f > NSFG_i$ $TNG_f < TNG_i$ $Den_f < Den_i$ $Dep_f < Dep_i$ $AGD_f > AGD_i$
RO4	Metric proposed by *Dijkman* [4]	Rules proposed by *Dijkman* [4]	$NN_f < NN_i$ $Den_f < Den_i$
RO5	NAF, NDOOut, NDOIn, Seq.	If NAF > NDOOut + NDOIn & Seq ≥ 3	$NAF_f < NAF_i$ $NN_f < NN_i$
RO6	CLA	If CLA < 1	$CLA_f > CLA_i$ $TNSF_f > TNSF_f$ $TNG_f > TNG_i$
RO7	TNE, NSFE, CC	If TNE > NSFE If CC < 1	$NSFE_f > NSFE_i$ $CC_f > CC_i$
RO8	TNEE	If TNEE > 1	$TNEE_f = 1$ NNf <NNi $CC_f > CC_i$
RO9	CLA, Seq	If CLA = 1 & Seq ≥ 5	$Seq_f < Seq_i$ $CLA_f = CLA_i$ $NN_f < NN_i$

The first column corresponds with the identifier of the refactoring operator, the second column contains the measures that are necessary to measure to promote its implementation, the third column shows the intervals that should belong measurement values to apply this refactoring, and de last column shows the new values of the measures following the implementation of refactoring, based on the initial values. These indicators of the intervals have been established heuristically after observation of numerous models of real business processes. The subindex "f" represents the final value of the measure and de subindex "i" represents the initial value of the measure.

In some cases the values of the measurements may be insufficient to apply the refactoring so it would take an expert opinion. These cases are symbolized in the table by the label "*Expert*".

VI. Conclusion and Future Works

The paper presents the use of the quality characteristics of understandability and modifiability, both collected from the ISO/IEC 25010, in order to choose the set of refactoring operators that provide a higher level of quality to the model, to improve the understandability and modifiability. For this purpose a set of measures related to theses quality characteristics are taken from literature. The aim of this study was to establish intervals, taking into account the measurement values of the measures proposed to measure the understandability and modifiability, which indicate which refactoring operator is advisable to apply to improve the quality levels of business process model concerning its modifiability and understandability. This idea deviate from conventional idea of apply refactoring operator indiscriminately, the idea is to guide the refactoring using quality measures so as to ensure a better end result.

This paper collected twenty-two (22) measures from the literature to measure the understandability and modifiability and nine (9) refactoring operators of business process models. Furthermore, it has shown what value would have these measures to enable each refactoring operator in order that the obtained benefit is greater. Thus, it can affirm that it is possible to guide the application of refactoring by the values of measurements of quality characteristics.

After the completion of this work a set of future lines has been identified in order to be addressed, following the idea of applying refactoring operators based on values of quality measurements. These future lines are the following:

- Validate the indicators shown in the intervals established for the application of refactoring operators. This can be done through studies case in order to refine the indicators have been established a priori heuristically.
- Implement the measures and the intervals, next to the application of refactoring operation, as an Eclipse plug-in in such that is compatible with other reverse engineering tools implemented.
- Determine the improvement obtained with each refactoring operators to establish not only the set that would be necessary, but the order in which to apply for higher levels of quality.
- Incorporate new measures on other quality characteristics of ISO/IEC 25010 in order to improve the model quality. One of the characteristics to consider may be the testability, which belongs to maintenance. This ensures that the model can then be easier to prove.

Acknowledgment

This work was supported by the FPU Spanish Program and the R&D projects ALTAMIRA (PII2I09-0106-2463), PEGASO/MAGO (TIN2009-13718-C02-01) and MOTERO (JCCM and FEDER, PEII11-0366-9449).

References

[1] Weske, M., Business Process Management: Concepts, Languages, Architectures. 2007, Leipzig, Alemania: Springer-Verlag Berlin Heidelberg. 368.

[2] Sánchez-González, L., et al., Quality assessment of business process models based on thresholds. On the Move to Meaningful Internet Systems: OTM 2010, 2010: p. 78-95.

[3] Reijers, H.A. and J. Mendling, A study into the factors that influence the understandability of business process models. Systems, Man and Cybernetics, Part A: Systems and Humans, IEEE Transactions on, 2011(99): p. 1-14.

[4] Dijkman, R., et al., Identifying refactoring opportunities in process model repositories. Information and Software Technology, 2011.

[5] Rolon, E., et al. Evaluation measures for business process models. 2006: ACM.

[6] Rolon, E., et al. Prediction models for BPMN usability and maintainability. 2009: IEEE.

[7] Cardoso, J. Process control-flow complexity metric: An empirical validation. 2006: IEEE.

[8] Mendling, J., H. Reijers, and J. Cardoso, What makes process models understandable? Business Process Management, 2007: p. 48-63.

[9] Mendling, J. and M. Strembeck. Influence factors of understanding business process models. 2008: Springer.

[10] Weber, B., et al., Survey paper: Refactoring large process model repositories. Comput. Ind., 2011. **62**(5): p. 467-486.

[11] Leopold, H., S. Smirnov, and J. Mendling, Refactoring of process model activity labels, in Proceedings of the Natural language processing and information systems, and 15th international conference on Applications of natural language to information systems. 2010, Springer-Verlag: Cardiff, UK. p. 268-276.

[12] ISO/IEC, ISO/IEC 25000:2005 in Software and system engineering - Software product Quality Requirements and Evaluation (SQuaRE)– Guide to SQuaRE. 2005.

[13] ISO/IEC, ISO/IEC 25010:2011, in Systems and software engineering – System and software product Quality Requirements and Evaluation (SQuaRE) – System and software quality models. 2011.

[14] Koehler, J., et al., Combining Quality Assurance and Model Transformations in Business-Driven Development, in Applications of Graph Transformations with Industrial Relevance, S. Andy, et al., Editors. 2008, Springer-Verlag. p. 1-16.

[15] Smirnov, S., H. Reijers, and M. Weske. A semantic approach for business process model abstraction. 2011: Springer.