

Improving Business Process Model after Reverse Engineering

María Fernández-Ropero, Ricardo Pérez-Castillo, and Mario Piattini

Instituto de Tecnologías y Sistemas de la Información, University of Castilla-La Mancha,
Paseo de la Universidad 4, 13071, Ciudad Real, Spain
{marias.fernandez,ricardo.pdelcastillo,mario.piattini}@uclm.es

Abstract. An appropriate business process management helps companies to quickly adapt to changes while their competitiveness is maintained or even improved. For this reason, business process models are recognized as being important assets for companies and companies are currently demanding mechanisms to ensure business processes with an appropriate quality degree. These business process models can be mined by reverse engineering from existing information systems. Regrettably, these reversed models usually have a lower quality degree and may not exactly reflect the actual business processes. This paper describes detected common problems in business processes retrieved by reverse engineering (e.g., missing, fine-grained or non-relevant elements, ambiguities, etc.). Refactoring is widely-used to fix quality problems in business process models. Despite this, this work suggests addressing the above problems by defining three stages: repairing, refactoring and expert-based improvement. Additionally, some preliminary results from refactoring stage are provided using real-life retrieved business process models.

Keywords: Business Process Model, Business Process Improvement, Business Process Challenges, Refactoring, Understandability, Modifiability, Reverse Engineering.

1 Introduction

Business process management allows organizations to be more efficient, more effective and more readily adaptable to changes than traditional management approaches. Business processes depict sequences of coordinated business activities as well as the involved roles and resources that organizations carry out to achieve their common business goal [1]. They are recognized as one of the most important assets in an organization due to the competitive advantages that they provide for organizations [2]. In order to supply the management of business processes they can be represented by models following standard notations such as BPMN (Business Process Modeling and Notation) [3].

However, organizations may not have their business process models explicitly or aligned with current behavior. In these cases, reverse engineering can be used to mine business process model from existing information system [4]. Nevertheless, the retrieved business process models by reverse engineering entail some problems that can

affect to their quality degree since every reverse engineering technique implies a semantic loss [5]. Despite the fact that much academic literature is devoted to identify challenges presented in business process model discovered by mining process (e.g., using event logs [6]) or by hand [7], there are no identified challenges to address in those business process model retrieved from existing information system, for example, from source code. This kind of business process models can be incomplete or can contain non-relevant information, or even may contain ambiguities or uncertainties that decrease their understandability and, therefore, their quality degree. In these cases, it is necessary to improve business process model with the aim to address these quality challenges while making it as similar as possible to the reality that they represent [8].

For this reason, this paper presents a set of challenges detected in business process models obtained from reverse engineering. These challenges are been collected after a literature review and practical experiences with business process models mined from several real-life information systems. With the purpose to have several retrieved business process models to analyze, MARBLE [4], a reverse engineering approach and tool, has been selected to mine them. Moreover, the paper introduces an approach to address the above challenges. The proposal combines reverse engineering with other analysis approaches in order to mitigate the semantic loss that reverse engineering techniques entails. This issue is due to some reverse engineering techniques are focused on source code and there are more knowledge sources from which to extract knowledge. Hence, the approach is divided in three stages: repairing, refactoring and expert-based improvement. Each stage uses additional knowledge (such as recorded event logs, guidelines, heuristics, and expert decision, among other) to improve the business process model. Despite refactoring techniques are the most widely-used solution to improve the quality degree of business process models [9, 10], this work also proposes other two additional stages in order to assist refactoring and enhance business process models. The work also presents some preliminary results achieved by using the proposed approach.

The remainder of the paper is organized as follows: Section 2 introduces the related works. Section 3 summarizes the challenges that retrieved business process models involve. After that, Section 4 introduces the proposed approach in an attempt to address these challenges along three stages. Afterwards, some results obtained by using the proposed approach will be shown in Section 5. Particularly, results obtained after refactoring stage are provided. Finally, conclusions and future works are discussed in Section 6.

2 Related Works

Improvement of business process models have been discussed by several authors in the last years with the aim to fix quality faults. In fact, several techniques such as merging, mining, refactoring or re-use are been provided by Dijkman et al., [10] in order to increase the quality degree of business process models. Particularly, refactoring is the most wide-used by authors in literature. For example, Weber et al., [11] proposes a catalogue of *bad smells* for the identification of refactoring opportunities. Similarly, Dijkman et al., [12] show a metrics-based technique for detecting

refactoring opportunities. La Rosa et al., [13], in turn, are devoted to identify patterns for reducing model complexity through, among other ways, compacting, compositing, and merging. Dumas et al., [14] and Ekanayake et al., [15], for their part, focus on detection of duplicate fragments. Pittke et al., [16] focus on labels and define a mechanism to identify synonym and homonym labels in models repositories. Other works like [17-20] focus on identifying coarse-grained activities by means of business process abstraction to exclude non-relevant information.

All the above approaches identify challenges of business process model but they are focus on business process model obtained by mining process (e.g., using event logs [6]) or by hand [7]. Therefore, none of them attempts to identify and address quality challenges in retrieved business process models by means reverse engineering.

To address this very issue, this paper identifies challenges in retrieved business process models and proposes an approach for addressing these challenges.

3 Challenges in Retrieved Business Process Models

This section presents the challenges to address the most common problems identified in the business process models obtained through reverse engineering. These challenges are been collected after a literature review and practical experiences with business process models mined from several real-life information systems. The selected tool to mine business process model was MARBLE. This tool is an adaptive framework to recover the underlying business process models from legacy information system using source code [4]. MARBLE has been applied to several industrial case studies to recover business processes from a wide variety of legacy information systems. The conduction of these industrial case studies has enabled the tool to be improved and the MARBLE technique to be refined. So far, MARBLE has been used with six legacy systems in all: (i) a system managing a Spanish author organization; (ii) an open source CRM (Customer Relationship Management) system; (iii) an enterprise information system from the water and waste industry; (iv) an e-government system used in a Spanish local e-administration; (v) a high school LMS (Learning Management System); and finally (vi) an oncological evaluation system used in Austrian hospitals [21]. All business process models obtained in each case study (from each of the six systems) were analyzed by experts in order to figure out common errors that frequently occur. Challenges that retrieved business process models entail are collected in the following subsections.

3.1 Completeness

Business process models mined by reverse engineering models may not be fully complete due to the data can be distributed in several sources, not just at the source code itself and therefore they cannot be obtained solely through a static analysis. Business process models may have missed nodes such as business tasks, gateways, events and data objects, as well as missed connections such as sequence flows (between tasks) and association flows (between tasks and data objects). This loss affects the semantic completeness of the model [22]. All these missing elements may not have been instantiated at design time and, for that reason, may not be appear in the business

process model. As a consequence, one of the biggest challenges is to rediscover elements that were not recovered in the reverse engineering phase, as well as the order among different business activities. The order between activities is a very issue since complete sequence flows between activities may not be provided through reverse engineering due to the fact that not all information can be automatically derived from source code. The start points and end may not have been defined in the model because there is not enough information to determine which activities are the beginning or ending of a model or what task is executed before another [23]. An example of this scenario is shown in Fig. 1, where the fragment (a) is related to the real retrieved business process model while the fragment (b) is related to the expected business process model, i.e., with all elements connected.

3.2 Granularity

According to the approach proposed by Zou et al., [24], each callable unit in an information system is considered a candidate business task to be discovered by reverse engineering. However, existing information systems typically contain thousands of callable units of different sizes that are usually considered business tasks that can have different levels of granularity [6, 23] such as (1) large callable units that support the main business functionalities of the system (e.g., methods and functions of the domain and controller layer), (2) small callable units as getter and setter methods in object-oriented programming that only read and write program variables but perform no real business task, (3) a set of small callable units that have similar behavior and perform a business task jointly, or (4) a set of small callable units that can together support another. In that case, the main task may be considered as father task while small tasks may be considered as children tasks. This last scenario is shown graphically in Fig. 2, where Task t1, Task tn, among others tasks, support the Task T.

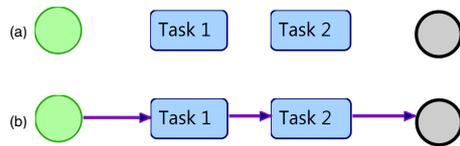


Fig. 1. Incomplete retrieved business process model (a) and its expected version (b)

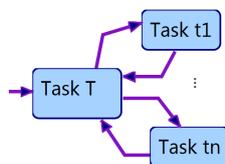


Fig. 2. Retrieved business process model with small tasks that support another task

With the purpose to address this challenge, a solution could be taking into account only coarse-grained callable units as candidate to be business tasks while fine-grained ones are discarded since fine-grained granularity makes models closer to source code perspective. Nevertheless, the dividing line between coarse- and fine-grained callable

units is unknown. Authors like *Polyvyanyy et al.* [19] propose to abstract these business process models to reduce unwanted details and to represent only the relevant information. These authors address the issue of different types of granularity by proposing two techniques: (1) eliminating those small tasks that are considered as irrelevant and (2) grouping certain tasks into one while the information is preserved.

3.3 Relevance

In contrast with completeness which is related to missing elements, relevance is related to business elements that have been retrieved erroneously. The information is considered as non-relevant when it can be removed without losing information, preserving the behavior. This information (such as activities, events, etc.) may have been created in compilation time but is not used in execution time, i.e., these elements do not carry out any business logic in the organization. Fig. 3 shows a sample fragment of a retrieved business process model with an unnecessary gateway. In that case, the gateway connects one input and one output so that the connection between Task1 and Task2 can be directly in that case due to gateways are designed for use in branches with two or more elements.

The relevance of a business process model is an important aspect since it ensures the model contains enough elements to convey their information [22]. The challenge must be addressed by identifying and removing all non-relevant elements in the business process model while preserving semantics of the relevant parts.

3.4 Uncertainty

The enhancement of the understandability of a business process model is a challenge given that poor understandability of the model can lead to a wrong conclusion. Understandability is usually worse in those models that have been obtained by reverse engineering from existing information systems since identifiers and names of elements may not be enough descriptive. This is because many identifiers are inherited from the elements in source code. For example, task labels usually consist of the concatenation of various capitalized words according to naming conventions present in most programming approaches (see Fig. 4). This kind of names is uncertain but can provide a clue to find more representative task names.

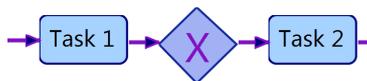


Fig. 3. Retrieved business process model with unnecessary gateway



Fig. 4. Retrieved business task labeled using the CamelCase [25] naming convention

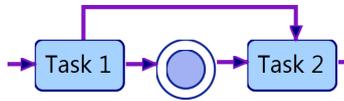


Fig. 5. Ambiguous retrieved business process model

This issue is focused on the interpretability from a language-usage perspective, i.e. how intuitive is the language used to define the elements of the model. For this reason, the labeling of elements can negatively affect the model interpretation when it does not follow an appropriate convention [22]. This challenge should be addressed by renaming elements of business process models in order to they faithfully represent the semantics performed actually.

3.5 Ambiguity

Another challenge to be taken into account is ambiguities that may be present in some business process elements. For example, redundancy faults sometimes occur during reverse engineering owing to different source code pieces (e.g., two callable units) lead to build two redundant business tasks that actually are part of a more complex task (e.g., a business task supported by both callable units). Fig. 5 gives an example where there are two different paths are obtained between Task 1 and Task 2 (direct path and conditional path) from different source code pieces. One of the paths contradicts the logic of the other path since the second is more restrictive.

Ambiguity is important to determinate the quality of business process models since it affects the understandability and modifiability of the model, i.e., how far elements in the model are intuitively formulated [22]. The ambiguity, therefore, negatively affects the ability to communicate efficiently the behavior of the business process. A model is considerate unambiguous when it is free of redundancies and it contains no elements that contradict the logic of other element. The ambiguity must be addressed by detecting and removing redundancies and inconsistencies in a business process model.

4 Business Process Model Improvement Approach

In order to address the challenges outlined above, this paper presents an approach for improving business process models obtained from information systems with the aim that they reflect as faithfully as possible the business reality with optimal levels of quality. This approach proposes three stages: repairing, refactoring and expert-based improvement. Each stage addresses some challenges above mentioned and uses some knowledge sources to carry out its purpose. Fig. 6 symbolizes the horseshoe model that characterizes the reengineering, where the upper stages (refactoring and expert-based improvement) represent a higher abstraction level than the bottom stage (repairing).

Repairing stage is considerate in reverse engineering level since it uses knowledge sources such as recorded event logs to address the completeness challenge. The aim of this stage is to ensure that business process models reflect the real execution of the information system. Preliminary results concerning this stage are given in [26].

That work shows a set of steps that are carried out taking as input a business process model and event logs and returning as output an enhanced business process model with additional sequence flows retrieved from event logs. The technique detects unrecovered sequence flows as regards the event log and tidily adds these sequence flows to the target business process model. After the conduction of a case study to demonstrate the feasibility of the technique, the results show that the fitness of the process model increases, i.e., repairing business process model leads to a more faithful representation of the observed behavior.

Refactoring stage is concerning to modify the internal structure of business process models without changing or altering the external behavior. This stage maintains the abstraction level while maintaining the semantic. Refactoring techniques therefore improve the quality of business processes, so that they become more understandable, maintainable and reusable [12]. This stage addresses some challenges as relevancy, granularity, uncertainty and completeness. Guidelines, literature, heuristics and experience are additional resources used in this stage. Some refactoring operators are introduced in [27], especially designed for use with reversed business process models. For example, some refactoring operators address the relevancy by means of the elimination of isolated nodes, unnecessary nesting, among other. Other refactoring operators address the granularity by grouping elements. Other refactoring operators address the completeness following good practices in business process modeling and incorporating additional needed elements. Some results obtained after applying these refactoring operators are shown in next section. Section 5 gives some results of this stage. There, each refactoring operator is applied in isolation in order to visualize the change that each operator provides to the business process model.

Finally, expert-based improvement stage addresses ambiguity and relevancy by means of expert decision. This stage is because not all challenges can be addressed automatically by the previous two stages, it is necessary also the opinion and feedback of an expert in certain situations to improve the business process model.

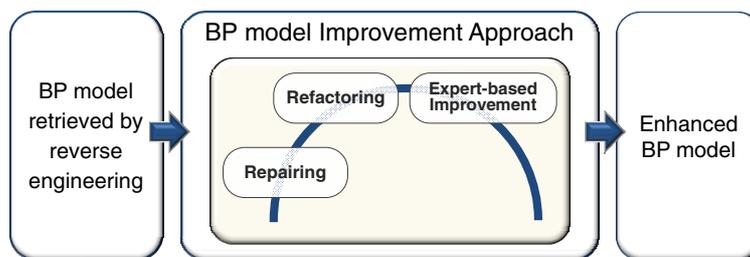


Fig. 6. Proposed improvement approach by means of three stages: repairing, refactoring and expert-based improvement

5 Refactoring Results

This section shows some results obtained in the second stage considered in the approach. In order to illustrate the effect of refactoring operators on business process models some aspects are defined:

Business process models taken as *independent variables* have been mined from the source code using MARBLE, the business process archeology tool used to figure out the above challenges (cf. Section 2). The selected information system was *Tabula*, a web application of 33.3 thousands of lines of code devoted to create, manage and simulate decision tables for associating conditions with domain-specific actions. From this information system was retrieved 15 business process models.

Measures used to assess the understandability and modifiability of a business process model [28] are considered as *dependent variables*: the *size* (number of elements such as tasks, events, gateways and data objects), *density* (ratio between the total number of flows in a business process model and the theoretical maximum number of possible flows regarding the number of elements), and *separability* (the ratio between the number of nodes that serve as bridges between otherwise strongly-connected components and the total number of nodes) of the model.

With the aim to illustrate briefly the result obtained by refactoring stage some refactoring operators are used from [27]: **R1** removes nodes (i.e., tasks, gateways or events) in the business process model that are not connected with any other node in order to contribute to the removing of non-relevant elements; Similarly, **R2** removes elements in the business process model that are considered sheet nodes; **R6** creates compounds tasks grouping several small tasks that support another main task. The goal is to remove the fine-grained granularity; **R7** combines data objects that are used for the same task in order to remove the fine-grained granularity; **R8** joins the start and end event to the starting and ending tasks, respectively, to complete the model; **R9** adds join and split gateways that are not present in branches in an effort to complete the model.

After the application of each refactoring operator on each business process model in isolation, values for each dependent variable are collected in Table 1, as well as the gain obtained with respect to the original value. The gain is defined as the ratio between the difference of measure values and the original measure value. Hence, a positive gain means that the refactoring affects the measure positively while a negative gain means that the refactoring affects the measure negatively. A zero gain means that the value for a certain measure did not change after refactoring.

Table 1. Effect of each refactoring operator on the size, density and separability

	Size		Density		Separability	
	Mean	Gain	Mean	Gain	Mean	Gain
Original	35.200	0.000	0.110	0.000	15.533	0.000
R1	30.667	0.395	0.196	-0.607	11.000	0.460
R2	34.400	0.011	0.113	-0.023	14.733	0.019
R6	33.267	0.059	0.106	0.031	15.667	-0.006
R7	33.667	0.026	0.106	0.009	14.600	-0.003
R8	37.600	-0.410	0.207	-0.338	17.933	-0.447
R9	59.400	-0.142	0.105	0.076	15.600	-0.002

Table 1 reveals that removing isolated nodes decreases the size and separability while the density is increased. Despite the density is higher after R1, the relevance of the model has been increased since non-relevant elements have been removed. Similarly, R2 causes an increase of density when the size is decreased. Separability is decreased slightly. R6 creates compound tasks in several business process models.

This fact entails a decrease in the size and density while separability increases slightly. The same happens with R7, the number of nodes and the density is lower but separability is higher. Nevertheless, all measures after R8 are higher due to business process models were incomplete. R9, in turn, cause a significant increase in the size because there were several incoming and outgoing branches without gateways in the original business process models. The same occur with the separability after R9 while density decreases slightly.

6 Conclusions

Reverse engineering has become in a suitable solution to mine business process model from existing information system. Unfortunately, these retrieved business process models entail some challenges that are necessary to address in order to increase their quality degree.

Completeness is an important challenge to deal with in retrieved business process model since data are distributed in several sources. Different types of granularity are also a challenge to address because fine-granularity causes the degree of quality is lower. Moreover, non-relevant information causes a low degree quality since the model should not contain additional elements that do not carry out any business logic in the organization. The uncertain labeling of elements may negatively affect the understandability and therefore an appropriate convention should be followed. In addition, ambiguity is another challenge because a model should be free of redundancies and inconsistencies.

It is with all the above in mind that this paper presents an approach for improving business process models obtained from information systems in an effort to deal with above challenges. The approach defines three stages: repairing, refactoring and expert-based improvement. These stages address challenges above mentioned by using additional knowledge sources to perform its goal. Moreover, in order to illustrate one of the stages, this work presents some results of refactoring stage. The result shows that the measures selected for assessing the quality of business process models -in terms of their understandability and modifiability, are improved in the most of cases by removing non-relevant and fine-grained elements as well as by completing models. Despite the fact that this work applies refactoring operator in isolation, studies reveal that refactoring operators do not satisfy commutative property among them, making necessary to figure out the best execution order [27].

After the completion of this work a set of future works has been identified: (1) Refining the repairing stage in order to obtain more valuable information from event logs in order to repair retrieved business process models; (2) Refining the refactoring stage by defining new refactoring operators to address more challenges. In addition, the use of more measures for assessing the understandability and modifiability is required; (3) Definition of expert-based improvement stage by means of the use of expert decision to remove ambiguities in the business process model.

Acknowledgements. This work was supported by the FPU Spanish Program and the R&D projects MAGO /PEGASO (Ministerio de Ciencia e Innovación [TIN2009-13718-C02-01]) and GEODAS-BC (Ministerio de Economía y Competitividad & Fondos FEDER [TIN2012-37493-C03-01]).

References

1. Weske, M.: *Business Process Management: Concepts, Languages, Architectures*, Leipzig, Germany, p. 368. Springer, Heidelberg (2007)
2. Jeston, J., Nelis, J., Davenport, T.: *Business Process Management: Practical Guidelines to Successful Implementations*, 2nd edn. Butterworth-Heinemann (Elsevier Ltd.), NV (2008)
3. OMG. *Business Process Modeling Notation Specification 2.0* (2011), <http://www.omg.org/spec/BPMN/2.0/PDF/>
4. Pérez-Castillo, R., et al.: MARBLE. A Business Process Archeology Tool. In: 27th IEEE International Conference on Software Maintenance (ICSM 2011), Williamsburg, VI, pp. 578–581 (2011)
5. Fernández-Ropero, M., Pérez-Castillo, R., Piattini, M.: Refactoring Business Process Models: A Systematic Review. In: Filipe, J., Maciaszek, L. (eds.) 7th International Conference on Evaluation of Novel Approaches to Software Engineering (ENASE), pp. 140–145. SciTePress, Wrocław (2012)
6. van der Aalst, W.: Process Mining: Overview and Opportunities. *ACM Transactions on Management Information Systems (TMIS)* 3(2), 7 (2012)
7. Indulska, M., Recker, J., Rosemann, M., Green, P.: Business process modeling: Current issues and future challenges. In: van Eck, P., Gordijn, J., Wieringa, R. (eds.) CAiSE 2009. LNCS, vol. 5565, pp. 501–514. Springer, Heidelberg (2009)
8. Fahland, D., van der Aalst, W.M.P.: Repairing Process Models to Reflect Reality (2012)
9. Weber, B., Reichert, M.: Refactoring Process Models in Large Process Repositories. In: Bellahsene, Z., Léonard, M. (eds.) CAiSE 2008. LNCS, vol. 5074, pp. 124–139. Springer, Heidelberg (2008)
10. Dijkman, R., Rosa, M.L., Reijers, H.A.: Managing large collections of business process models—Current techniques and challenges. *Computers in Industry* 63(2), 91 (2012)
11. Weber, B., et al.: Survey paper: Refactoring large process model repositories. *Comput. Ind.* 62(5), 467–486 (2011)
12. Dijkman, R., et al.: Identifying refactoring opportunities in process model repositories. *Information and Software Technology* (2011)
13. La Rosa, M., et al.: Managing process model complexity via abstract syntax modifications. *IEEE Transactions on Industrial Informatics* 7(4), 614–629 (2011)
14. Uba, R., Dumas, M., García-Bañuelos, L., La Rosa, M.: Clone detection in repositories of business process models. In: Rinderle-Ma, S., Toumani, F., Wolf, K. (eds.) BPM 2011. LNCS, vol. 6896, pp. 248–264. Springer, Heidelberg (2011)
15. Ekanayake, C.C., Dumas, M., García-Bañuelos, L., La Rosa, M., ter Hofstede, A.H.M.: Approximate clone detection in repositories of business process models. In: Barros, A., Gal, A., Kindler, E. (eds.) BPM 2012. LNCS, vol. 7481, pp. 302–318. Springer, Heidelberg (2012)
16. Pittke, F., Leopold, H., Mendling, J.: Spotting Terminology Deficiencies in Process Model Repositories. In: Nurcan, S., Proper, H.A., Soffer, P., Krogstie, J., Schmidt, R., Halpin, T., Bider, I. (eds.) BPMDS 2013 and EMMSAD 2013. LNBIP, vol. 147, pp. 292–307. Springer, Heidelberg (2013)

17. Smirnov, S., Weidlich, M., Mendling, J.: Business process model abstraction based on synthesis from well-structured behavioral profiles. *International Journal of Cooperative Information Systems* 21(1), 55–83 (2012)
18. Smirnov, S., Reijers, H.A., Weske, M.: A semantic approach for business process model abstraction. In: Mouratidis, H., Rolland, C. (eds.) *CAiSE 2011*. LNCS, vol. 6741, pp. 497–511. Springer, Heidelberg (2011)
19. Polyvyanyy, A., Smirnov, S., Weske, M.: Business process model abstraction. In: *Handbook on Business Process Management*, vol. 1, pp. 149–166 (2010)
20. Smirnov, S.: *Business process model abstraction*, Universitätsbibliothek (2012)
21. Pérez-Castillo, R., et al.: A family of case studies on business process mining using MARBLE. *Journal of Systems and Software* 85(6), 1370–1385 (2012)
22. Overhage, S., Birkmeier, D.Q., Schlauderer, S.: Quality Marks, Metrics, and Measurement Procedures for Business Process Models. *Business & Information Systems Engineering*, 1–18 (2012)
23. Pérez-Castillo, R., et al.: Generating Event Logs from Non-Process-Aware Systems Enabling Business Process Mining. *Enterprise Information System Journal* 5(3), 301–335 (2011)
24. Zou, Y., Hung, M.: An Approach for Extracting Workflows from E-Commerce Applications. In: *Proceedings of the Fourteenth International Conference on Program Comprehension 2006*, pp. 127–136. IEEE Computer Society (2006)
25. Binkley, D., et al.: *To camelcase or under_score*. IEEE (2009)
26. Fernández-Ropero, M., Reijers, H.A., Pérez-Castillo, R., Piattini, M.: Repairing Business Process Models as Retrieved from Source Code. In: Nurcan, S., Proper, H.A., Soffer, P., Krogstie, J., Schmidt, R., Halpin, T., Bider, I. (eds.) *BPMDs 2013 and EMMSAD 2013*. LNBP, vol. 147, pp. 94–108. Springer, Heidelberg (2013)
27. Fernández-Ropero, M., et al.: Assessing the Best-Order for Business Process Model Refactoring. In: *28th Symposium On Applied Computing (SAC)*, Coimbra, Portugal, pp. 1400–1406 (2013)
28. Fernández-Ropero, M., et al.: Quality-Driven Business Process Refactoring. In: *International Conference on Business Information Systems (ICBIS)*, Paris, France, pp. 960–966 (2012)