Chapter 3

IBUPROFEN: Business Process Model Refactoring

María Fernández-Ropero
Instituto de Tecnologías y Sistemas de la Información, University of Castilla-La Mancha, Spain

Ricardo Pérez-Castillo
University of Castilla-La Mancha, Spain

Mario Piattini
Instituto de Tecnologías y Sistemas de la Información, University of Castilla-La Mancha, Spain

ABSTRACT

Organizations are increasingly choosing process-oriented organizational designs as a source to achieve competitive advantages. Business process models represent the sequence of tasks that an organization carries out. However, organizations must cope with quality problems of business process models (e.g., lack of understandability, maintainability, reusability, etc.). These problems are compounded when business process models are mined by reverse engineering (e.g., from information systems that support them), owing to the semantics loss that it involves. Refactoring techniques are commonly used to reduce these problems through changing their internal structure without altering their external behavior. Although several refactoring operators exist in the literature, there are no refactoring techniques especially developed for models obtained by reverse engineering and their special features. For this reason, this chapter presents IBUPROFEN, a refactoring technique (and supporting tool) for business process models obtained by reverse engineering. Moreover, a case study is conducted to determine how the refactoring operator’s order influences the understanding and modification of business process models. The case study reveals there is a clear influence in these quality features in terms of the size and separability of the models under study, and therefore, refactoring operators do not satisfy the commutative property among them.
INTRODUCTION

Due to an optimal business process management enables organizations to be more efficient, more effective and more readily adaptable to changes than traditional, functional management approaches; organizations are devoting more and more efforts to ensure that their business process representations have adequate quality levels. Moreover, in order to supply their analysis and management, business processes can be represented by models following standard notations such as BPMN (Business Process Modeling and Notation) (OMG, 2011).

Nevertheless, organizations sometimes do not explicitly have their business processes models since they have never modeled their business processes before. Besides, even when the organization has business process models, such models can be outdated or misaligned regarding the actual processes supported by enterprise information systems. In both cases, reverse engineering techniques is usually used to retrieve business process models, for example from existing information systems (Pérez-Castillo, García-Rodríguez de Guzmán, & Piattini, 2011). Unfortunately, the retrieved business process models are prone to have a lower quality degree since every reverse engineering technique is characterized by a semantic loss as a result of the progressive increase of the abstraction level (in this case from existing information systems to business processes) (María Fernández-Ropero, Pérez-Castillo, & Piattini, 2012c). For example, retrieved business process models can contain several nested branches that increase their complexity. In those cases an improvement of these business process models is required in order to increase the abstraction level and therefore they can represent the business reality as faithful as possible (Fahland & Aalst, 2012), while their understandability and modifiability is enhanced.

The aim is therefore to improve business process models, particularly those that have been obtained by reverse engineering. In the last years, refactoring techniques have emerged as a solution to improve business process models, increasing their quality degree (Dijkman, Gfeller, Küster, & Völzer, 2011; Dijkman, Rosa, & Reijers, 2012; La Rosa et al., 2011; Weber, Reichert, Mendling, & Reijers, 2011). These techniques change the internal structure of business process models without altering or modifying their external behavior, and therefore, improving the understandability and modifiability. Refactoring operators replace some fragments for equivalent ones. Despite in the literature there are several proposed refactoring operators to be applied to business process models, there are no refactoring techniques specially developed for those models obtained through reverse engineering and their peculiarities (e.g., missing elements, mining of non-relevant elements, etc.). Moreover, most business process model refactoring techniques consist of recognizing refactoring opportunities and then applying them. However, refactoring operators may be sequentially applied, i.e., the output business process model of a refactoring operator is the input model for the next refactoring operator. Thus, a great deal is to determine the best combination of refactoring operators in order to ensure the great gain of understandability and modifiability.

In order to size up the gain provided for each refactoring operators it is possible to use the different measures that literature offers concerning to assess the understandability and modifiability. These measures are regarding to the number of elements in the business process models, the connectivity between all the elements, among others. Hence, the obtained gain after the application of each refactoring operator is calculate based on these measures (i.e., the difference between before and after refactoring).

The objective of this chapter is to introduce the concept of business process model and their importance for organization as well as the concept of refactoring as a means to improve business process models. This chapter additionally discusses the problems that entail those business process models that have been retrieved by reverse engineering.
Related Content

Autonomous Human Flow Counting Service With Deep Neural Network

Multi-Modeling, Meta-Modeling, and Workflow Languages
[www.igi-global.com/chapter/multi-modeling-meta-modeling-and-workflow-languages/91941?camid=4v1a](www.igi-global.com/chapter/multi-modeling-meta-modeling-and-workflow-languages/91941?camid=4v1a)

Implications of Markup on the Description of Software Patterns
[www.igi-global.com/chapter/implications-markup-description-software-patterns/52881?camid=4v1a](www.igi-global.com/chapter/implications-markup-description-software-patterns/52881?camid=4v1a)

A Case Study of Dynamic Analysis to Locate Unexpected Side Effects Inside of Frameworks
[www.igi-global.com/article/a-case-study-of-dynamic-analysis-to-locate-unexpected-side-effects-inside-of-frameworks/126614?camid=4v1a](www.igi-global.com/article/a-case-study-of-dynamic-analysis-to-locate-unexpected-side-effects-inside-of-frameworks/126614?camid=4v1a)