Information Systems 49 (2015) 102–125

Contents lists available at ScienceDirect

Information Systems

journal homepage: www.elsevier.com/locate/infosys

Detecting approximate clones in business process model repositories

Marcello La Rosa [a,b,*], Marlon Dumas [c], Chathura C. Ekanayake [a], Luciano García-Bañuelos [c], Jan Recker [a], Arthur H.M. ter Hofstede [a]

[a] Queensland University of Technology, Australia
[b] NICTA Queensland Lab, Australia
[c] University of Tartu, Estonia

ARTICLE INFO

Article history:
Received 4 April 2014
Received in revised form
16 November 2014
Accepted 27 November 2014
Recommended by: M. Weske
Available online 6 December 2014

Keywords:
Business process model
Clone detection
Model collection
Repository
Standardization

ABSTRACT

Empirical evidence shows that repositories of business process models used in industrial practice contain significant amounts of duplication. This duplication arises for example when the repository covers multiple variants of the same processes or due to copy-pasting. Previous work has addressed the problem of efficiently retrieving exact clones that can be refactored into shared subprocess models. This paper studies the broader problem of approximate clone detection in process models. The paper proposes techniques for detecting clusters of approximate clones based on two well-known clustering algorithms: DBSCAN and Hierarchical Agglomerative Clustering (HAC). The paper also defines a measure of standardizability of an approximate clone cluster, meaning the potential benefit of replacing the approximate clones with a single standardized subprocess. Experiments show that both techniques, in conjunction with the proposed standardizability measure, accurately retrieve clusters of approximate clones that originate from copy-pasting followed by independent modifications to the copied fragments. Additional experiments show that both techniques produce clusters that match those produced by human subjects and that are perceived to be standardizable.

© 2014 Elsevier Ltd. All rights reserved.

1. Introduction

Ample evidence suggests that duplication is a widespread phenomenon in software and model repositories [1,2]. Not surprisingly, duplication is also found in repositories of business process models used in industrial practice [3]. Clones in process model repositories emerge for example as a result of copy-pasting, but also when multiple variants of a process co-exist and are described as separate models. For example, an insurance company typically runs multiple claims handling processes for different types of claims. Naturally, these process variants share commonalities, which manifest themselves in the form of clones.

Detecting clones in process models allows modelers to identify opportunities for standardization and refactoring. For example, consider the case of multiple variants of an insurance claims handling process, where each variant is captured as a separate process model. Given that disbursement of the insurance payout occurs in every variant (albeit differently depending on the type of claim), it is likely that these separate models will contain clones corresponding to disbursement activities. These clones can potentially be

* Corresponding author at: Queensland University of Technology, GPO Box 2434, Brisbane, Qld 4001, Australia.
E-mail addresses: m.larosa@qut.edu.au (M. La Rosa), marlon.dumas@ut.ee (M. Dumas), chathura.ekanayake@gmail.com (C.C. Ekanayake), luciano.garcia@ut.ee (L. García-Bañuelos), j.recker@qut.edu.au (J. Recker), a.terhofstede@qut.edu.au (A.H.M. ter Hofstede).

http://dx.doi.org/10.1016/j.is.2014.11.010
0306-4379/© 2014 Elsevier Ltd. All rights reserved.

standardized[1] and refactored as a shared subprocess. In this way, duplication is reduced and uniformity across process models is increased to the benefit of model maintainability.

Standardization of clones however is only possible if the clones to be standardized are either exact clones or they are sufficiently similar that they can be replaced by a standardized fragment with minor changes to each original clone. Indeed, while some changes to a clone may be lexical (e.g. uniformizing the nomenclature of tasks), other changes may entail alterations to the underlying process, such as adding or skipping a task, leading to similar fragments that may or may not be standardizable depending on the business implications of the change.

The problem of clone detection has been widely studied in the field of software engineering, primarily in the context of source code clone detection, but also in the context of model clone detection (e.g. clones in Simulink models) [2,5]. In this context, a distinction is made between four types of clones [2], which can be defined in the context of process models as follows:

- *Type*-1 (*also called exact clones*): Identical fragments except for layout variations and comments.
- *Type*-2: Syntactically identical fragments except for possible layout variations, comments and labeling variations (e.g. different task, event or data object labels with the same semantics).
- *Type*-3 (*also called approximate clones* [6] or *near-miss clones*): Copied fragments with further modifications such as changed, added or removed model elements in addition to variations allowed in Type-2 clones. Note that two Type-3 clones are not necessarily behaviorally equivalent.
- *Type*-4: Behaviorally equivalent fragments with syntactic differences (e.g. fragments with different combinations of gateways but same set of traces). Note that Type-4 clones are a superset of Type-2. While Type-2 clones only allow for one-on-one substitutions, Type-4 allow for any variation so long as behavior is preserved. On the other hand, Type-4 clones are not a superset of Type-3 clones or vice-versa. Rather, Type-3 and Type-4 clones are alternative ways of extending the notion of Type-2 clones.

In previous work, we proposed a technique for identifying Type-1 (exact) clones in process models [7]. This technique can also be adapted to detect Type-2 clones by pre-processing the labels of model elements and replacing semantically equivalent labels with a standard label. However, this technique cannot detect Type-3 (approximate) clones, which are arguably likely to emerge in process model repositories when modelers copy–paste fragments across models – thus creating exact clones – and later on these exact clones evolve separately.[2]

To address this gap, this paper presents and compares two techniques for identifying Type-3 (approximate) clones in repositories of process models for the purpose of standardizing and refactoring them as shared subprocesses. The paper also proposes and validates a measure of standardizability of a set of approximate clones, meaning a measure of the feasibility of replacing the clones with a single shared subprocess. This measure captures the tradeoff between the magnitude of changes required to achieve standardization and the simplification benefits that standardization yields.

The proposed techniques and standardizability measure are evaluated in a two-pronged manner. First, we evaluate the runtime performance and accuracy of the two techniques using a combination of real-life and synthetic datasets. Second, we report two experiments with human subjects in which we compare the proposed techniques in terms of (i) their ability to retrieve groups of clones that human subjects perceive to be standardizable (that is, replaceable and refactored as a single shared subprocess); and (ii) their ability to replicate clusters produced by human subjects.

This paper is an extended version of a previous conference paper on the subject [8]. The main extensions are the two empirical evaluations with human subjects (Section 6), as well as a more comprehensive discussion of related work, differences between the two techniques, limitations of the approach and threats to validity.

The paper is organized as follows. Section 2 defines and justifies the notion of approximate clone adopted in this paper and the proposed measure of standardizability. Next, Section 3 introduces techniques for process model parsing and exact clone detection, which are used as the basis for the proposed techniques. Section 4 presents the techniques. Next Sections 5 and 6 present the results of the evaluation while Section 7 discusses threats to the validity of the evaluation and limitations. Finally, Section 8 frames the contributions in relation to the literature while Section 9 concludes and discusses possible extensions of the research. The instruments used for the evaluation with human subjects are available as supplementary material attached to this paper.

2. Approximate clones and standardizability

This section defines the notion of similarity adopted in this paper and, on this basis, it defines a notion of approximate clone cluster and a measure of standardizability for approximate clone clusters.

2.1. Process model similarity

When designing an approximate clone detection method, a first step is to define what an approximate clone is. Generally, such a definition relies on a similarity or (equivalently) a distance metric.

The similarity of process models specified in a graph-based notation can be measured on the basis of three complementary aspects: (i) the labels attached to tasks, events and other model elements; (ii) their graph structure; and (iii) their execution semantics. In this paper, we adopt a measure that combines structural and label similarity (distance) and that has been shown to be correlated with perceived similarity [9]. We define this measure over an

[1] We use the term *standardization* to refer to the act of replacing discrepant but similar process fragments with a single unified fragment. Other authors use the term *harmonization* [4] instead to emphasize that the unified fragment is not necessarily a "standard".

[2] Type-4 clone detection in process models, while potentially relevant, deserves a separate treatment as it involves a very different set of techniques (behavioral equivalence checking).

Download English Version:

https://daneshyari.com/en/article/6858701

Download Persian Version:

https://daneshyari.com/article/6858701

Daneshyari.com