Computers in Industry 82 (2016) 104–118

Contents lists available at ScienceDirect

Computers in Industry

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

An automated approach for merging business process fragments

Mohamed Anis Zemni [a,*], Amel Mammar [b], Nejib Ben Hadj-Alouane [c]

[a] Ecole Nationale des Sciences de l'Informatique, ENSI, UR/OASIS, Tunisia
[b] Institut Mines-Télécom/Télécom SudParis, CNRS UMR 5157 SAMOVAR, France
[c] Ecole Nationale d'Ingénieurs de Tunis, ENIT, UR/OASIS, Tunisia

ARTICLE INFO

Article history:
Received 20 April 2016
Accepted 3 May 2016
Available online

Keywords:
Business processes
Fragments
Merge approach
Gateway paths
Gateway paths matrix
Process behaviors

ABSTRACT

In the field of business process management, adopting efficient building strategies can improve the quality of companies' business processes. The reuse of existing business processes or even fragments of them is a practical approach to build complete business processes or coarser-grained process fragments. In the present paper, we deal with the merge of a set of business process fragments for the construction of new complete processes. Our merge mechanism relies on a particular path matrix, that we call gateway path matrix. We use gateway path matrices to represent business process fragments to systematically compose shared components with individual ones. Moreover, our approach ensures that the resulting business process fragments subsume the behavior of initial ones and allows for adding new execution scenarios while controlling undesirable ones. In fact, we detect newly generated behaviors, and alert process designers of undesirable ones through behavioral constraints. We provide extensive experimental results derived from an implementation of our approach applied on a well-known industrial library of business process fragments.

1. Introduction

Optimizing the development periods of service-oriented software can improve the efficiency of various actors in the software industry. Moreover, modern software applications rely heavily on *business process* technologies [19]. Therefore, adding a degree of automation to the design of business processes can extensively improve the development period and quality of the underlying applications.

A business process development approach, based solely on constructing business processes completely from scratch, is costly, time consuming and lacks flexibility. In practice, designers need to implement, optimize and test new software applications and thereby business processes in short periods of time. To overcome this hurdle, business process practitioners and architects are now interested in *reusing* and integrating existing business processes into new ones, rather than completely building them from scratch. It is common knowledge that such practices can increase both the efficiency and the productivity of the development process [13].

In line with business process reuse strategy, some researchers [1,25] propose the use of what are known as configurable business processes [16]. Variation of business processes are obtained by modifying the underlying configuration. Such techniques have a limited efficiency as they are typically used to adapt business processes within restrained perimeters. Adding new activities to such processes generally implying going through the implementation, optimization and test phases.

A more practical emergent approach consists in reusing only particular portions of existing business processes, called *business process fragments* (*BPF*). These fragments are generally constrained with fewer business rules than their complete parent processes and are, hence, easier to integrate into new business processes. Moreover, the controlled reuse of well-established, tested and stable fragments, can improve significantly the quality of resulting new business processes [27,30]. Several approaches have been proposed to identify reusable fragments [35,31–33]. These fragments are, afterwards, stored in libraries for future use by business process building tools. In this paper, we focus our interest on the development of an automated approach for merging selected business process fragments from existing fragment library for the purpose of constructing complete viable business processes.

It is well noted in the literature that existing fragments may contain similarities or even overlapping knowledge and structures [7] as they may be retrieved from independent business processes offering similar services or even from the same business process. In the approach developed in this paper, we strive to take advantage

* Corresponding author.

http://dx.doi.org/10.1016/j.compind.2016.05.002
0166-3615/

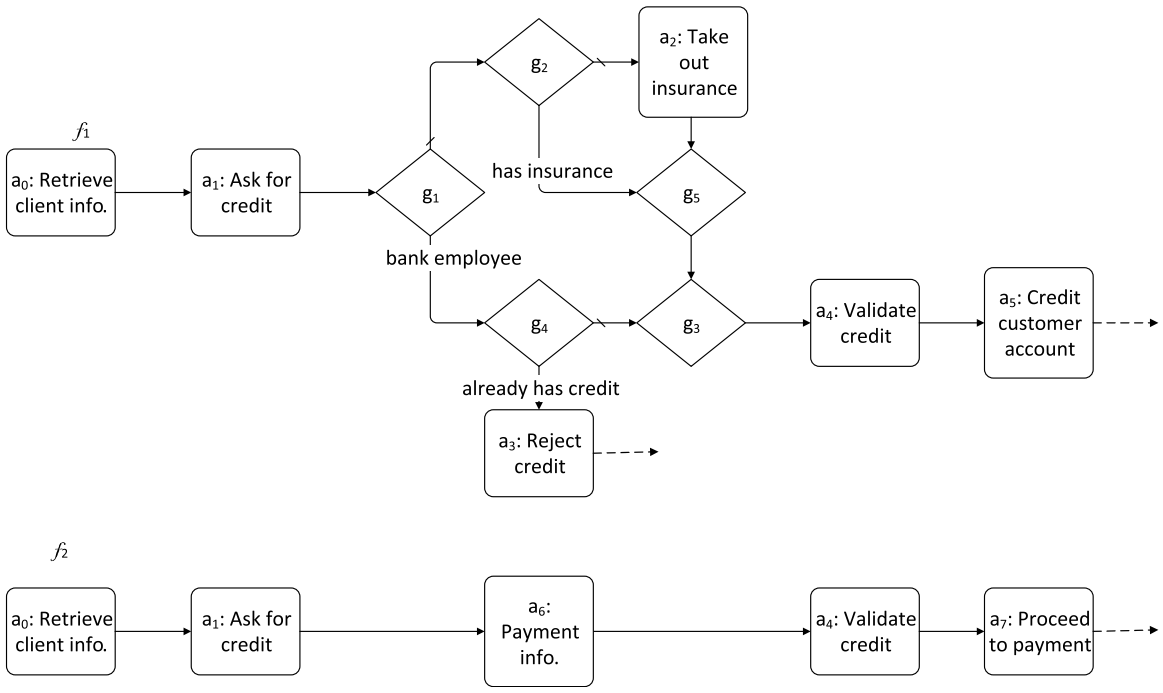

Fig. 1. Business process fragments represented in BPMN.

of such similarities as a way of interconnecting business process fragments.

The interconnection approach we defined ensures that the resulting interconnected business process fragments subsume the behavior of initial ones, add new execution scenarios, while controlling undesirable ones.

The main contributions of this paper reside in providing a flexible merge mechanism for business process fragments. This mechanism is based on the notion of path matrices [34] used as a way to represent node-based graphs capturing business processes. More specifically, path matrices are used to configure paths between adjacent nodes in node-based graphs. Our merge mechanism relies on path matrices properties to provide correct merged paths between pairs of activities belonging to interconnected business process fragments.

In order to properly handle the undesirable behaviors that may rise during the merge process, we present a mechanism that permits to detect and control them. As a matter of fact, we integrated into resulting merged fragments behavioral constraints that the designer can use and configure as a way of allowing the desirable behaviors and inhibiting undesirable ones.

To thoroughly evaluate our approach from a practicality, efficiency, and qualitatively point of view, we provide extensive experimental results derived from an implementation of our approach applied to a well known industrial library of business process fragments. We show that our approach provides good quality coarse-grained fragments or business process. The quality of the resulting fragment or business process is evaluated by comparing them to the manually constructed fragments.

The remainder of this paper is organized as follows: Section 2 presents a real-life motivating scenario to show in details the issues we face when merging a pair of fragments. Section 3, illustrates our business process artifacts we use throughout this paper. In Section 4, we introduce the path matrices that serve as a foundation for the merge mechanism used by our approach. In Section 5, we detail the merge mechanism based on path matrices. In Section 6, we give the behavioral constraints that are used to control the behavior of the resulting fragments. The presentation of the experimental results obtained from the software tool is given in Section 7. In Section 8, we review important and relevant literature and finally we conclude in Section 9. We note also that since our approach is well grounded in mathematically formal concepts, we provide among others in Appendix the necessary proofs for the properties stated along this paper.

2. Motivating scenario

Our objective is to compose a set of already selected *BPF*s into a single coarse-grained *BPF*. For this aim, each *BPF* must share at least an activity with another *BPF*. Moreover, we choose to use the merge principle for the composition, typically used to consolidate several versions of a *BP* [8,26]. In this section, we present the main issues we face while performing the merge through a real-life scenario.

Let us consider the pair of *BPF*s illustrated in Fig. 1 and owned by a software development company specialized in supplying applications for bank management software. The *BPF*s f_1 and f_2 were initially developed for two distinct services: "local banking" and "foreign banking". Inspired from the BPMN representation [17], activities are represented with rounded boxes and gateways with diamonds. Complete control flows are represented with solid arrows and dangling[1] ones with dashed arrows. Each control flow involves a passing condition that should be evaluated to *true* to allow passing to the target object. In Fig. 1, control flows with empty passing conditions means that the latter are fixed to *true*. The first *BPF*, f_1, depicts the main activities for credit application within local banking. The service is intended for normal customers and bank employees. Approved amounts are credited on the customer's account. The second *BPF*, f_2, is intended for professional customers carrying out foreign financial transactions. The *BPF* performs credit application for them. Unlike the first *BPF*, approved amounts are directly transferred to the foreign party with whom the professional customer realizes the transaction. Let us suppose that the company desires to provide a service, called

[1] Control flows with either no source or target objects specified.