Information and Software Technology 74 (2016) 86–104

Contents lists available at ScienceDirect

Information and Software Technology

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

Variability management in process families through change patterns

Clara Ayora[a,*], Victoria Torres[a], Jose Luis de la Vara[b,1], Vicente Pelechano[a]

[a] Centro de Investigación en Métodos de Producción de Software, Universitat Politècnica de València, Camino de Vera s/n, 46022 València, Spain
[b] Computer Science Department, Carlos III University of Madrid, Avda. Universidad 30, 28911 Leganés (Madrid), Spain

ARTICLE INFO

Article history:
Received 19 February 2015
Revised 20 January 2016
Accepted 21 January 2016
Available online 18 February 2016

Keywords:
Business process modeling
Business process variability
Process-aware information system
Process family
Change patterns

ABSTRACT

Context: The increasing adoption of process-aware information systems together with the high variability in business processes has resulted in collections of process families. These families correspond to a business process model and its variants, which can comprise hundreds or thousands of different ways of realizing this process. Managing process variability in this context can be very challenging, labor-intensive, and error-prone, and new approaches for managing process families are necessary.

Objective: We aim to facilitate variability management in process families, ensure process family correctness, and reduce the effort needed for such purposes.

Method: We have derived a set of change patterns for process families from variability-specific language constructs identified in the literature. For validation, we have conducted a case study with a safety standard in which we have measured the number of operations needed to model and evolve the variability of the standard with and without the patterns.

Results: We present 10 change patterns for managing variability in process families and show how they can be implemented. The patterns support the modeling and evolution of process families and ensure process family correctness by automatically introducing and deleting modeling elements. The case study results show that the application of the defined change patterns can reduce the number of operations when modeling a process family by 34% and when evolving it by 40%.

Conclusions: The application of the change patterns can help in effectively modeling and evolving large and highly-variable process families. Their application can also considerably reduce variability management effort.

1. Introduction

Process-aware information systems (PAISs) manage, execute, and analyze the business processes of an enterprise (e.g., sales business processes) based on explicitly specified *process models* [17,52]. The increasing adoption of PAISs during the last decade has resulted in large process model repositories [13,51], which usually comprise collections of related *process model variants* (*process variants* for short) [37]. Process variants pursue the same or a similar business objective (e.g., product sale) and can have activities (and their ordering constraints) in common. Nevertheless, process variants differ in their *application context*, such as the regulations to comply with in different countries, and some activities may be relevant only for certain contexts [10,13,37]. All the context factors causing process variability are typically known at design time [15,37].

A collection of related process variants can be referred to as process family [37]. In practice, such a family may comprise hundreds or thousands of process variants. *Hallerbach et al.* describe a process family from automotive that comprises over 900 process variants [23] and *Li* reports on over 90 variants for medical examinations [28]. Finally, check-in procedures at airports are similar irrespective of the airport or airline, but variations exist depending on the type of check-in (e.g., online or at the counter) or passenger (e.g., unaccompanied minors) [10]. Example 1 describes the check-in process, discussing its different sources of variability. We will use this process as a running example throughout the paper.

Example 1 (Check-in process)**.** Numerous variations exist for this process depending on different factors. For example, variability is caused by the type of passenger (e.g., unaccompanied minors and handicapped people might require extra assistance and special seats). Another source of variability includes the flight destination

* Corresponding author. Tel.: +34 963877007; ext.83533.
E-mail addresses: cayora@pros.upv.es (C. Ayora), vtorres@pros.upv.es (V. Torres), jvara@inf.uc3m.es (J.L. de la Vara), pele@pros.upv.es (V. Pelechano).
[1] Tel.: +34 916 24 91 15.

http://dx.doi.org/10.1016/j.infsof.2016.01.007
0950-5849/

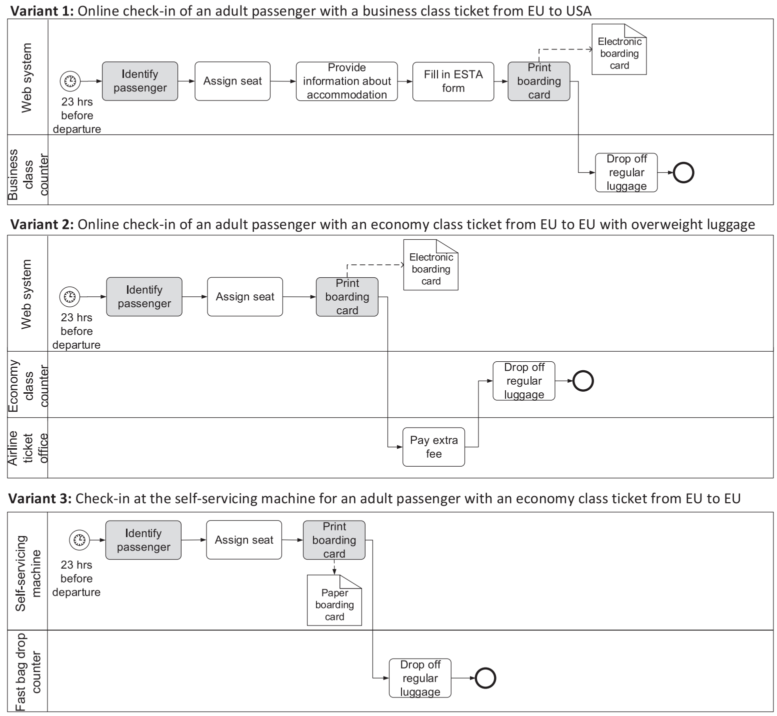

Fig. 1. Variants of the check-in process (1).

(e.g., accommodation information is required when traveling to the USA). Finally, depending on the type of luggage (e.g., bulk or overweight luggage), the process may differ because an extra fee has to be paid.

Figs. 1 and 2 show six simplified variants of the check-in process represented in BPMN (*Business Process Modeling Notation*) [12]. The variants have been modeled in collaboration with domain experts. While these process variants share commonalities (colored in grey), they also show differences. *Variants 1* and *2* (cf. Fig. 1) presume that the check-in is done online by the passenger, who is identified and assigned in a seat. *Variant 1* describes the case of a passenger flying from Europe to the USA, which requires accommodation information as well as filling in the electronic form for travel authorization (i.e., ESTA form). An electronic boarding card is printed and the passenger drops off the luggage at the business class counter. The payment of an extra fee at the ticket office is required in *Variant 2* due to luggage overweight. For *Variant 3*, the check-in is done at the self-servicing machine and the luggage is dropped off at the fast bag-drop counter. Check-in for these three process variants becomes available 23h before departure.

In contrast, *Variants 4–6* (cf. Fig. 2) represent the check-in process accomplished at the counter at the airport. *Variant 4* describes the check-in for an unaccompanied minor (UM). A special seat is assigned, an extra form is filled in, and a copy of the boarding card is required for the relative accompanying the minor to the gate. *Variant 5* refers to a handicapped passenger requiring extra assistance to accompanying him, whereas *Variant 6* corresponds to the check-in process of a passenger carrying bulk luggage. In these three process variants, check-in may only be performed at maximum 3h before departure, once the counters are opened. The boarding card is printed in paper format.

1.1. Problem statement

Modeling and evolving process families and ensuring their correctness can be very challenging mainly due to their size and heterogeneous application context (e.g., type of passenger, flight destination, and type of luggage) [37]. This has resulted in the development of approaches that support process variability along the process lifecycle, such as C-EPC (Configurable *Event-driven Process Chain*) [20]. These approaches enable the analysis, design, configuration, enactment, diagnosis, and evolution of process families, and specify process variants by means of configurable process models that represent a complete process family [10]. By treating variability as a first class citizen, configurable process models contribute to avoiding model redundancies, fostering model reusability, and reducing modeling efforts [37]. Fig. 3 shows a configurable process model for the check-in process family, created with C-EPC (introduced in Section 2.2).

When using process variability approaches, PAIS engineers need assistance because they have to manually model and manage all the elements and dependencies of a configurable process model individually [23]. In the model of Fig. 3, a PAIS engineer needs to represent that the activity *Print duplicated boarding card for the relative* for the relative is allowed only if a seat for UM has been assigned before. Modeling such constraints manually with the primitives currently offered by the existing process variability approaches can be tedious and error-prone, especially when a process family comprises a high number of variants and with many dependencies. There is a lack of approaches to deal with this variability in an explicit manner, especially at a level of abstraction higher than the one provided by the existing process variability approaches [10].

The use of modeling patterns (reusable solutions to a commonly occurring problem [50]) is a promising way to address these issues. For example, *adaptation patterns* have been proposed for creating and managing (individual) process models [50]. These patterns allow creating and modifying process models and ensure *correctness by construction*. They also provide systematic means for realizing change operations and aim to reduce modeling efforts [14]. However, these patterns do not deal with process variability in an explicit manner [7] and hence are not accurate for process family management [9].