Information and Software Technology 62 (2015) 187–197

Contents lists available at ScienceDirect

Information and Software Technology

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

Automating correctness verification of artifact-centric business process models

Diana Borrego *, Rafael M. Gasca, María Teresa Gómez-López

University of Seville, Department of Computer Languages and Systems, Av Reina Mercedes S/N, 41012 Seville, Spain

ARTICLE INFO

Article history:
Received 25 July 2014
Received in revised form 19 February 2015
Accepted 21 February 2015
Available online 3 March 2015

Keywords:
Artifact-centric business process model
Verification
Constraint programming

ABSTRACT

Context: The artifact-centric methodology has emerged as a new paradigm to support business process management over the last few years. This way, business processes are described from the point of view of the artifacts that are manipulated during the process.
Objective: One of the research challenges in this area is the verification of the correctness of this kind of business process models where the model is formed of various artifacts that interact among them.
Method: In this paper, we propose a fully automated approach for verifying correctness of artifact-centric business process models, taking into account that the state (lifecycle) and the values of each artifact (numerical data described by pre and postconditions) influence in the values and the state of the others. The lifecycles of the artifacts and the numerical data managed are modeled by using the Constraint Programming paradigm, an Artificial Intelligence technique.
Results: Two correctness notions for artifact-centric business process models are distinguished (reachability and weak termination), and novel verification algorithms are developed to check them. The algorithms are complete: neither false positives nor false negatives are generated. Moreover, the algorithms offer precise diagnosis of the detected errors, indicating the execution causing the error where the lifecycle gets stuck.
Conclusion: To the best of our knowledge, this paper presents the first verification approach for artifact-centric business process models that integrates pre and postconditions, which define the behavior of the services, and numerical data verification when the model is formed of more than one artifact. The approach can detect errors not detectable with other approaches.

1. Introduction

Nowadays, organizations model their operations with business processes. Traditionally, business processes are modeled as activity-centric business process models [1] in which activities are focused on and data just serve as inputs and outputs of some services. They follow the imperative principles, implying that the workflow of the activities can be defined at design time. But for some types of problems, it is easier to represent how the data are modified during the process execution instead of the activities that execute the data evolution.

For this reason, the artifact-centric methodology (data-centric approach) has emerged as a new paradigm to support business process management, where business artifacts appeared for the necessity of enrich the business process model with information about data [2], providing a way for understanding the interplay between data and process. Artifacts are business-relevant objects that are created, evolved, and (typically) archived as they pass through a business, combining both data aspects and process aspects into a holistic unit [3].

Artifact-centric modeling establishes data objects (called artifacts) and their lifecycles as focus of the business process modeling. This type of modeling is inherently declarative: the control flow of the business process is not explicitly modeled, but follows from the lifecycles of the artifacts [4].

The lifecycle represents how the state of an artifact may evolve over the time. The different activities change the state of the artifact and the values of the data associated to each artifact; these may be manual (i.e. carried out by a human participant of the process) or automatic (i.e. by a web service). The evolution of the artifacts implies a change of the state and the values of the data, until a goal state of an artifact is reached. One of the reasons

* Corresponding author. Tel.: +34 954 556 234; fax: +34 954 557 139.
E-mail addresses: dianabn@us.es (D. Borrego), gasca@us.es (R.M. Gasca), maytegomez@us.es (M.T. Gómez-López).

http://dx.doi.org/10.1016/j.infsof.2015.02.010
0950-5849/

why the artifact-centric paradigm facilitates the process description is the capacity to model the relations between objects with different cardinalities, not only 1-to-1 relations. This modeling capabilities are not entirely supported in activity-centric scenarios. For instance, BPMN 2.0 [5] (currently wide accepted activity-centric notation) allows to easily represent multi-instance activities and pools (processes), but with some limitations: (i) relations between different processes can only be expressed as hierarchies, where a process can invoke multiple instances of its subprocess; (ii) the return value of an executed sub-process instance is only accessible when its execution finishes, not allowing the interaction of another process during the execution; and (iii) the definition of Data Objects, Data Inputs, Data Outputs, Sets, and Data Associations in BPMN 2.0 allows to specify collections of elements, but it does not permit data instance differentiation or to include the relation between the data objects between them. Regarding this last limitation, the proposal in [6] proposes an extension of BPMN data objects adding annotations to manage data dependency and instance differentiation. However, these annotations are very low level representations and no significant for business stakeholders. Based on this idea, the work in [7] uses more complex objects that can involve N-to-M relations, using all the advantages of ORM to incorporate the data objects in a more natural way into the activity-centric business processes.

When more than one artifact is involved in the process, it is possible that a combination of services and data values violate the policies of the business. In order to avoid this situation at runtime, it is possible to detect some of these possible errors even at design time. Specifically, the errors derived from an incorrect design of the model. In spite of the unknown runtime data in the design time phase, our proposal is able to perform a data verification of the models by means of the use of mandatory domains of values, which can be obtained from previous executions and/or knowledge from experts. Making use of this information, it is possible to determine the existence of certain errors in the structural and data perspectives of the model before it is deployed.

The goal of this paper is to develop an approach for verifying the correctness of artifact-centric business process models at design time, including the state relation between the artifacts of the model, and the data values that define the relations between them. To develop the automatic verification, we model the services formally using pre and postconditions over the data associated to the artifacts' states. To analyze the correctness of the model, it is necessary to study when the services can be executed. A service can be executed if the evolution of the lifecycle of the artifact arrives at the service and its precondition is satisfied. Upon completion, the service delivers data that satisfies its postconditions. The no satisfiability of a pre or postcondition can cause that the lifecycle gets stuck at a service and fails.

The automatic verification is performed using Artificial Intelligence techniques, both to compute the possible evolutions on the lifecycles and to model the pre and postconditions of the services as numerical constraints.

This paper is organized as follows. Section 2 presents a motivating example to illustrate the concepts of reachability and weak-termination. Section 3 introduces artifact graphs and artifact union graphs as a formal model for artifact-centric business process models, and defines reachability and weak-termination on artifact union graphs. Section 4 defines the CSP formulation of an artifact union graph. The process of verification is explained, and two algorithms are presented. The verification of the motivating example is performed, and their tractability is discussed. Section 5 presents an overview of related work found in the literature. And finally, conclusions are drawn and future work is proposed in Section 6.

2. A motivating example

In this paper, the example presented in [8] has been enriched, including characteristics that cannot be described using the activity-centric paradigm. The original example describes the handling of a conference by an organizing committee. At the beginning of the process, the establishment of the conference rate is performed, even before the submission period is open. Then, the external services that are needed during the conference are booked (e.g. gala dinner, coffee breaks and proceedings), at the same time that the sponsorship money collection is carried out and the origin of the guest speaker is decided.

Likewise, the authors submit the papers which are received by the organizing committee, and are reviewed by members of the scientific committee (reviewers) in order to select the papers accepted for the conference. The decision about the approval or rejection of the papers is notified to the authors. Meanwhile, the conference registration period is open, which will remain open until the conference ends, so that the authors can register when they already know if their papers will be presented at the conference. Finally, the number of conference attendees is known, and the payment for the booked services is performed.

Since the relations between the execution instances of the different processes are not all 1-to-1, this presented scenario is not modellable by means of the activity-centric paradigm due to the limitations explained in Section 1. That is, while the tasks performed by the organizing committee only requires an instance of execution, several instances of the different reviewers and submitted papers are running simultaneously.

The described process can be represented with five artifacts: (1) the Finances artifact, involving the tasks regarding the economic decisions, performed by the organizing committee; (2) the Organization artifact, entailing the tasks concerning the publication of papers and registration of attendees, performed by the program committee; (3) the Paper artifact, including the tasks performed by the authors of the submitted papers; (4) the Reviewer artifact, containing the tasks executed by the reviewers of the papers; and (5), the Registration artifact, allowing the registration of attendees to the conference. The associations between these five artifacts present different cardinalities, existing relations 1-to-1, 1-to-N and N-to-M between the artifact instances, as it is shown in Fig. 1.

As the execution of the tasks changing the states of the artifacts takes place, some data are consumed and produced by reading and writing the attributes of the mentioned artifacts. Those attributes are listed in Table 1 with their corresponding meaning. As mentioned, the behaviors of the tasks are defined by means of pre and postconditions over the artifacts and attributes.

Semantically ordered tasks of different and independently modeled artifacts may be executed in any order. But other combinations are not desirable, for instance, it makes no sense to perform the payment of the gala dinner at the restaurant before we know the number of conference attendees. Therefore, the executions have to be constrained using policies and goal states.

A policy tells us the constraints that must comply the state changes within one artifact, or between different artifacts. These changes can be related to the values of the attributes, to the number of artifact instances, or because a service has been executed over the artifact. Finally, goal states restrict final states by reducing those combinations of artifacts' final states that should be considered successful.

For our motivating example, there are ten policies restricting the inter-artifact behavior: