Information Systems 73 (2018) 61–78

Contents lists available at ScienceDirect

Information Systems

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

Multi-party business process compliance monitoring through IoT-enabled artifacts

Giovanni Meroni[a,*], Luciano Baresi[a], Marco Montali[b], Pierluigi Plebani[a]

[a] Politecnico di Milano, Piazza Leonardo da Vinci, 32, Milano 20133, Italy
[b] Free University of Bozen-Bolzano, Piazza Università, 1, Bolzano 39100, Italy

ARTICLE INFO

Article history:
Received 21 February 2017
Revised 23 November 2017
Accepted 29 December 2017
Available online 30 December 2017

Keywords:
Business process compliance
Runtime compliance monitoring
Internet of things
Guard-Stage-Milestone
E-GSM
Declarative languages
Artifact-centric languages
Business process model transformation

ABSTRACT

Monitoring the compliance of the execution of multi-party business processes is a complex and challenging task: each actor only has the visibility of the portion of the process under its direct control, and the physical objects that belong to a party are often manipulated by other parties. Because of that, there is no guarantee that the process will be executed — and the objects be manipulated — as previously agreed by the parties.

The problem is usually addressed through a centralized monitoring entity that collects information, sent by the involved parties, on when activities are executed and the artifacts are altered. This paper aims to tackle the problem in a different and innovative way: it proposes a decentralized solution based on the switch from control- to artifact-based monitoring, where the physical objects can monitor their own conditions and the activities in which they participate.

To do so, the Internet of Things (IoT) paradigm is exploited by equipping physical objects with sensing hardware and software, turning them into smart objects. To instruct these smart objects, an approach to translate classical Business Process Model and Notation (BPMN) process models into a set of artifact-centric process models, rendered in Extended-GSM (E-GSM) (our extension of the Guard-Stage-Milestone (GSM) notation), is proposed.

The paper presents the approach, based on model-based transformation, demonstrates its soundness and correctness, and introduces a prototype monitoring platform to assess and experiment the proposed solution. A simple case study in the domain of advanced logistics is used throughout the paper to exemplify the different parts of the proposal.

1. Introduction

Modern organizations are more and more required to become open, reactive, and flexible entities able to satisfy the ever-changing needs of their customers. This is why they are redesigning their internal structures and business processes to increase dynamism and be open to cooperate with new organizations. Many business processes — once internal to single organizations — now cross the boundaries of single organizations and require the coordination among different, potentially changing actors. This transformation heavily impacts on how the process is executed. Organization no longer have full control on the whole process. Instead, they control only the portion of that process that is assigned to them. At the same time, the physical objects belonging to an organization can now be manipulated by the other actors, and the ownership of these objects can change while the process is performed.

To ensure that organizations coordinate properly, and that physical objects are correctly handled, the correctness and compliance of these distributed processes has to be monitored. In particular, the execution order and the successful execution of the activities composing the process have to be checked. To automate and keep track of business processes, organizations deploy Business Process Management Systems (BPMSs). In fact, today's BPMSs include a monitoring module to oversee the execution of fully automated business processes that can be confined within a single party. BPMSs also provide dashboards to inform the process owner of the current status, bottlenecks, and possible alerts.

Unfortunately, when moving to multi-party processes, the BPMS of each organization can only manage the activities under its control, but it has no jurisdiction on the activities carried out by the other parties. Consequently, it can only monitor the process portions carried out by the organization. This limitation is traditionally addressed by federating the BPMSs, or by deploying a centralized

* Corresponding author.
E-mail addresses: giovanni.meroni@polimi.it (G. Meroni), luciano.baresi@polimi.it (L. Baresi), montali@inf.unibz.it (M. Montali), pierluigi.plebani@polimi.it (P. Plebani).

https://doi.org/10.1016/j.is.2017.12.009
0306-4379/

one. However, these solutions lack flexibility, as whenever a new party is introduced, leaves, or the process changes, the underlying infrastructure must be heavily reconfigured.

When activities are automated, the BPMS is in charge of executing them. Therefore, it exactly knows when such activities start and when they finish, and which is their outcome. However, when dealing with non-automated activities, a BPMS relies on human operators to know about the outcome of such activities. As these operators could forget to notify the events of interest, they could make mistakes, or they could even intentionally postpone, fake, or alter provided inputs, monitoring manual activities can be unreliable. This has an impact not only to the party in charge of executing these activities, but also to the other connected parties.

To overcome these issues, this paper proposes a novel approach to autonomously and continuously monitor multi-party business processes in a distributed way. To this aim, we move the monitoring tasks directly onto the artifacts, i.e., the physical objects that participate in the process, which are equipped with sensors and computing devices, thus becoming "smart".

By doing so, these smart objects can autonomously keep track of all the activities in which they were involved, regardless of the organization performing them. Additionally, smart objects can track all the changes in their states, i.e., their conditions, throughout the execution of the process. This way, a smart object can autonomously monitor the compliance of the process it participates in, as well as its own lifecycle, that is, the transitions from one state to a new one that are expected to occur while the process is executed. On this basis, the characterizing contributions of the proposed solution are the following:

- We combine control-flow analysis, as defined using Business Process Model and Notation (BPMN), and artifact-centric analysis, as defined using Extended-GSM (E-GSM), our extended version of the Guard-Stage-Milestone (GSM) notation [1]. The user starts defining the multi-party process in BPMN, a widely known process modeling language. Then, for each artifact, E-GSM models suited to monitor the process and the lifecycle of the artifact are semi-automatically derived from the BPMN model. The combination of these two perspectives allows one to predicate on both executions and involved artifacts. If we say that an execution is *compliant* if it evolves through the foreseen control flow, and an artifact is *compliant* if it evolves according to its lifecycle, our solution can distinguish among *(i)* compliant executions that produce compliant artifacts, *(ii)* non-compliant executions that lead to compliant artifacts, and *(iii)* non compliant executions that lead to non-compliant artifacts.
- We adopt *smart* objects (a-la IoT) to transform artifacts into active entities that can both enact the E-GSM models and communicate with the others. The former capability means that each artifact (smart object) can: *(i)* infer its current state, *(ii)* know the admissible next states, and *(iii)* know the order in which the process' activities should manage it.
- We propose an innovative architecture for the distributed execution and monitoring of multi-party processes that embed the characteristics highlighted above. The proposed architecture is based on the use of simple Single-board Computer (SBC) boards, such as the Raspberry PI and the Intel Galileo, and exploits `Node.js` as implementation language.

All the key features of the proposed solution are exemplified through a (simplified without being trivial) real example process borrowed from the domain of advanced logistics. The same process is also used for accessing the solution.

With respect to our previously published articles, this article extends the E-GSM based process monitoring approach presented in [2] by proposing a structured approach to instruct the monitoring platform, and providing an implementation of the solution. It also extends the BPMN to E-GSM translation presented in [3] and [4] by formalizing how to extract the portion of the process relevant to each participating object, and to also represent in E-GSM the lifecycle of each object. Compared to the monitoring approach described in [5] and [6], which focuses mostly on runtime conformance checking, the main focus of this article is on monitoring how physical objects are impacted by the process execution. Therefore, it extends the approach to also monitor the lifecycle of these objects, and to detect compliance violations even if they do not explicitly violate the control flow. The monitoring architecture presented in this article differentiates from mArtifact [7] by running directly on top of the smart objects participating in the process.

The rest of the paper is organized as follows. Section 2 discusses the limitations of current monitoring approaches and presents the main elements of our solution by means of a concrete case study, then used consistently throughout the paper. Section 3 describes the proposed extensions to the E-GSM notation, then exploited in Section 4, which presents our approach. Section 5 argues about the correctness of the automated transformation of BPMN models into E-GSM ones, while Section 6 introduces the distributed architecture defined for supporting the presented process compliance monitoring solution. Section 7 analyzes the state of the art and Section 8 concludes the paper.

2. Motivations

Fig. 1 shows the BPMN representation of a real multi-party process, taken from the logistics domain. It describes the initial phase of a multimodal transport. At first, the *Carrier*, the entity responsible for the physical shipment of the goods, collects an empty shipping container from the warehouse of the *Multimodal Transport Operator (MTO)*, which is in charge of organizing the entire shipment, and ships it to the *Producer* of the goods. In parallel, the *Producer* prepares the goods and produces the documentation for the shipment. Once the *Carrier* reaches the *Producer*'s site, the *Producer* loads the goods onto the container, verifies that all the documents are correct and, if not, updates them. Finally, the *Carrier* starts the shipment. Both the *MTO* and the *Producer* verify the identity of the *Carrier* before granting it access to their sites.

This BPMN model is treated as an agreement between the various organizations. Process portions carried out by each organization (i.e., the ones inside the pools) are disclosed, and the other organizations agree on how the whole process is executed. Therefore, no privacy restriction holds on this process model, which is shared among the participating organizations.

To know if this process is correctly executed, organizations have to both monitor their internal activities (i.e., the ones under their control) and verify that their objects were correctly manipulated by the other organizations. Since each party can already monitor its own process portions, the focus of this paper is on monitoring the objects. To this aim, for each object (e.g., the goods, the container), it is necessary to monitor the activities involving that object. This way, it is possible to know the exact steps that caused an object to be in its current conditions. It is worth noting that, since an object may be manipulated by organizations other than the owner, monitoring only internal activities is not sufficient. For example, although it belongs to the *MTO*, the container is manipulated by both the *Carrier* and the *Producer*. Therefore, activities belonging to other organizations, as long as they interact with the objects, should be monitored as well.

Additionally, the conditions of the objects have to be also monitored, and anomalies have to be promptly notified. For example, in case of drugs, the producer may want to be sure that the temperature of the goods remains stable during the whole transportation. These objects (i.e., those that must be monitored) are rendered as