IoT for BPMers. Challenges, case studies and successful applications

Francesco Leotta, Andrea Marrella, and Massimo Mecella

Sapienza Università di Roma, Italy
{leotta,marrella,mecella}@diag.uniroma1.it

Abstract. The Internet-of-Things (IoT) refers to a network of connected and interacting devices (e.g., sensors, actuators) collecting and exchanging data over the Internet. In the last years, we have witnessed an increasing presence of IoT devices in scenarios of the Business Process Management (BPM) domain, which can strongly influence the coordination of the real-world entities (e.g., humans, robots) that execute specific tasks or entire business processes in such environments. While, on the one hand, the IoT can provide many opportunities for improving BPM initiatives, on the other hand, it poses challenges that require enhancements and extensions of the current state-of-the-art in BPM. This paper discusses how BPM can benefit from IoT, *(i)* showing which emerging challenges have to be tackled to integrate the IoT technology in a BPM project, and *(ii)* presenting concrete case studies on process adaptation and habit mining exploiting IoT and addressing the specific challenges posed by IoT itself.

Keywords: IoT · Business process · Habit mining · Process adaptation

1 Introduction

Business Process Management (BPM) is an active area of research based on the observation that each product and/or service that an organization offers, is the outcome of a number of performed activities. Business processes (BPs) are the key instrument for organizing such activities and improving the understanding of their interrelationships. Nowadays, BPs are enacted in many complex industrial (e.g., manufacturing, logistics, retail) and non-industrial (e.g., emergency management, healthcare, smart environments) domains [10]. In all these domains, we have witnessed an increasing presence of Internet-of-Things (IoT) devices (e.g., sensors, RFIDs, video cameras, actuators) that operate over the existing network infrastructure, including the Internet, to collect data from the physical environment, monitor in detail the evolution of several real-world objects of interest, and actuate concrete feedbacks (e.g., in the form of suggestions or alerts) in response to the observed information. From a BPM perspective, the knowledge extracted from the physical environment by IoT devices allows to depict the contingencies and the context in which BPs are carried out, providing a fine-grained monitoring, mining, and decision support for them.

The interplay of IoT devices with BPM can provide many opportunities for improving the enactment of BPs. For example, among the main benefits, the execution of BPs can be driven by event data detected at real-time, enabling BPs to become more *adaptive* and *reactive* to what is happening in the real world. However, on the other hand, there is a conflict between the stability and meaningfulness of the services at work in a BP as opposed to the dynamic and changing environment that IoT is able to offer. This poses several challenges to concretely interconnect the two worlds and make them interact, which require enhancements and extensions to the current state-of-the-art in BPM.

According to the BPM-Meet-IoT manifesto [4], sixteen challenges have been identified to make this vision a reality. In this contribution, we focus on two specific areas from the BPM literature where it is strongly required to tackle these challenges, namely *habit mining* and *process adaptation*.

In Section 2, we show how data collected by IoT devices (a.k.a. IoT data) should be properly abstracted and managed when willing to employ BP discovery techniques [1] to model human habits as "personal processes". Secondly, in Section 3, we present a reference conceptual architecture to build a BPM engine that is able to reason over the discrete counterpart of the "continuous" IoT data for achieving automated adaptation of running BPs in case of unanticipated exceptions. Finally, in Section 4, we conclude the paper.

2 Visual Process Maps for Habit Mining

A smart space (e.g., a smart house) represents a typical example of IoT environment. The aim of a smart space is providing people with automatic or semi-automatic services realizing the concept of ambient intelligence (AmI). To this aim, a set of both software and hardware networked artefacts, acting as sensors (e.g, presence, temperature sensors) or actuators (e.g., ovens, rolling shutters), are coordinated according to a previously acquired knowledge expressed in the form of models representing human preferences and environmental dynamics.

Models in smart spaces are usually classified as *specification-based*, which are hand-made by experts, or *learning-based*, which are instead obtained by applying machine learning and data mining. In the first case, models are usually based on logic formalisms, relatively easy to read and validate (once the formalism is known to the reader), but their creation requires a major cost in terms of expert time and effort. In the latter case, the model is automatically learned from a training set (whose labeling cost may vary according to the proposed solution) but employed formalisms are usually not "explanaible" due to the statistic techniques they are based on, making them less immediate to understand.

Authors in [5] suggested that applying methods originally taken from BPM to human habits may represent a compromise between specification-based and learning-based methods, provided that the gap between raw sensor measurements and human actions can be filled in by performing a log-preprocessing step. Such a step may consist of simple inferences on data, or complex machine learning algorithms. On the line of this argument, [2,6] propose the Visual Pro-

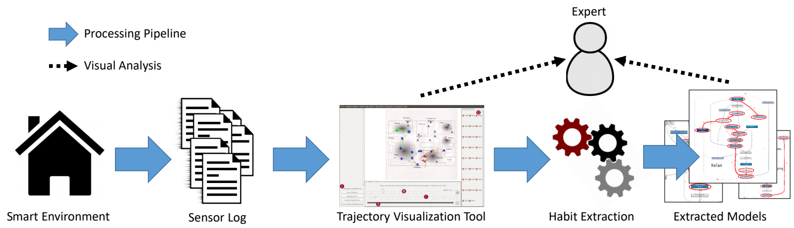

Fig. 1. The conceptual architecture of the Visual Process Maps (VPM) system.

cess Maps (VPM) system, consisting of a complete pipeline formed by *(i)* a tool for the visual analysis of sensor logs, *(ii)* a method to transform raw movement measurements into actions, and *(iii)* a method to identify and visually analyze precedence relationships between human actions through the employment of fuzzy mining [3].

Figure 1 shows the conceptual architecture of VPM. A smart space produces, during runtime, a *sensor log* containing raw measurements from available sensors. Measurements can be produced by a sensor on a periodic base (e.g., temperature) or whenever a particular event is detected (e.g., a door opening). The current version of VPM focuses on sensor logs produced by a grid of Passive InfraRed (PIR) sensors triggering upon the detection of an object entering their field of view and automatically reset after a fixed amount of time since the last detected movement. The detection area of a PIR can be usually tuned to cover different area sizes ranging from a tile on the floor to an entire room.

The first step of VPM consists in a visual analysis tool, named Trajectory Visualization Tool, able to "play" specific portions of a log, perform automatic analysis tasks and visualize the result. Here, playing means to animate the sensor log showing the trajectory followed by a person in the house. The tool also allows to produce an *event log* obtained from the sensor log by aggregating simple PIR sensor measurements into sub-trajectories representing movement actions belonging to the following categories: *(a)* moving between areas of the house, *(b)* staying still under a PIR, or *(c)* moving in a specific area of the house.

Such an event log can be used as input for fuzzy mining. The rationale here is that, if we know the location of devices that humans can interact with inside the space, we can associate to each of these movement actions, the physical actions performed by humans (e.g., using the oven). As an example, if the model contains a precedence relation between the action "moving inside the bathroom" and "stay under the PIR sensor corresponding to the bed", we have a clear idea of the human actions determined by the movement actions.

The process extracted by fuzzy mining depends on how the sensor log fed into VPM is labeled. If no label is available, the mined process model will represent the daily habit of a person. If instead, labels corresponding to the beginning and the end of daily routines are available, it is possible to obtain specialized process models for them.

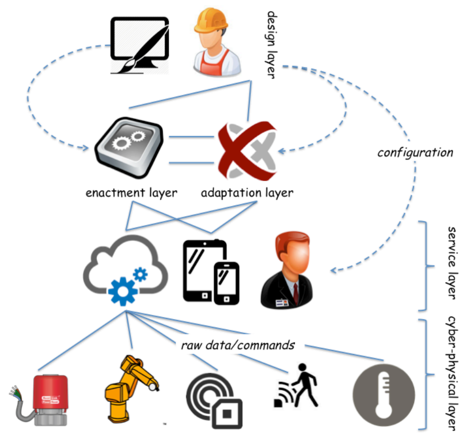

Fig. 2. A conceptual architecture for BPs enacted in IoT-based environments.

3 A Conceptual Architecture for Process Adaptation

During the enactment of BPs in IoT-based environments, variations or divergence from structured reference models are common due to exceptional circumstances arising in form of *exogenous events*, thus requiring the ability to properly *adapt* the process behavior. *Process adaptation* can be seen as the ability of a BP to react to exceptional circumstances (that may or may not be anticipated) and to adapt/modify its structure accordingly.

Since in IoT-based environments the number of possible anticipated exceptions is often too large, manual implementation of exception handlers at design-time is not feasible, since it is required to anticipate all potential problems and ways to overcome them in advance directly in the BP [10]. Furthermore, in such environments, many unanticipated exogenous events may arise during the BP execution, and the needed knowledge to tackle such events at the outset is often missing. Finally, a BPM engine can only reason over a discrete knowledge of the world, thus requiring to convert the continuous raw data collected by the IoT technology into discrete information.

To tackle this issue, we summarize the main ideas discussed in [8,9] and we introduce our architectural solution to build a BPM engine that is able to automatically adapt BPs at run-time when *unanticipated exceptions* occur in IoT-based environments, thus requiring no specification of recovery policies at design-time. The general idea builds on the dualism between an *expected reality* and a *physical reality*: process execution steps and exogenous events have an impact on the physical reality and any deviation from the expected reality results in a mismatch to be removed to allow process progression. As shown in Figure 2, we identified 5 main architectural layers that we present in a bottom-up fashion.

The *cyber-physical layer* consists mainly of two classes of physical components: *(i)* sensors (such as GPS receivers, RFIDs, 3D scanners, cameras, etc.)

that collect data from the physical environment by monitoring real-world objects and *(ii)* actuators (robotic arms, 3D printers, electric pistons, etc.), whose effects affect the state of the physical environment. The cyber-physical layer is also in charge of providing a physical-to-digital interface, which is used to transform *raw* data collected by the sensors into machine-readable events, and to convert *high-level* commands sent by the upper layers into *raw* instructions readable by the actuators. The cyber-physical layer does not provide any intelligent mechanism neither to clean, analyse or correlate data, nor to compose high-level commands into more complex ones; such tasks are in charge of the uppers layer.

On top of the cyber-physical layer lies the *service layer*, which contains the set of services offered by the real-world entities (software, robots, agents, humans, etc.) to perform specific BP activities. In the service layer, available data can be aggregated and correlated, and high-level commands can be orchestrated to provide higher abstractions to the upper layers. For example, a smartphone equipped with an application allowing to sense the position and the posture of a user is at this layer, as it collects the raw GPS, accellerometer and motion sensor data and correlates them to provide discrete and meaningful information.

On top of the service layer, there are two further layers interacting with each other. The *enactment layer* is in charge of *(i)* enacting complex BPs by deciding which activities are enabled for execution, *(ii)* orchestrating the different available services to perform those activities and *(iii)* providing an execution monitor to detect the anomalous situations that can possibly prevent the correct execution of BP instances. The execution monitor is responsible for deciding if process adaptation is required. If this is the case, the *adaptation layer* will provide the required algorithms to *(i)* reason on the available BP activities and contextual data and to *(ii)* find a recovery procedure for adapting the BP instance under consideration, i.e., to re-align the BP to its expected behaviour. Once a recovery procedure has been synthesized, it is passed back to the enactment layer for being executed.

Finally, the *design layer* provides a GUI-based tool to define new BP specifications. A BP designer must be allowed not only to build the BP control flow, but also to explicitly formalize the data reflecting the contextual knowledge of the IoT-based environment under study. It is important to underline that data formalization must be performed without any knowledge of the internal working of the physical components that collect/affect data in the cyber-physical layer. To link activities to contextual data, which are the main driver for triggering process adaptation, the GUI-based tool must go beyond the classical "activity model" as known in the literature, by allowing the BP designer to explicitly state what data may constrain an activity execution or may be affected after an activity completion or an exogenous event. Finally, besides specifying the BP, configuration files should also be produced to properly configure the enactment, the services and the sensors/actuators in the bottom layers.

The SmartPM system presented in [7] is a concrete instantiation of the above reference architecture.

4 Concluding Remarks

This paper provides an introduction to the IoT with the eyes of a BPM researcher. The focus is on identifying and presenting those IoT features that directly impact BPM, i.e., *data*, *quality* and *granularity of such data*, *events*, *identification of process instances*, etc. In particular, we have focused on the issue of dealing with continuous and frequent data readings, and on the low level of abstraction provided by IoT measurements wrt. the traditional concept of "events" and "traces" in the BPM literature. Through two specific outcomes of our research activities, we have exemplified the above concepts in order to provide insights for further research.

Acknowledgments. *The multi-year research work has been partly supported by the "Dipartimento di Eccellenza" grant, the H2020 RISE project FIRST (grant #734599), the H2020 ERC project NOTAE (grant #786572), the Sapienza grants IT-SHIRT, ROCKET and METRICS, the Lazio regional initiative "Centro di eccellenza DTC Lazio" and the project ARCA.*

References

1. Augusto, A., Conforti, R., Dumas, M., Rosa, M.L., Maggi, F.M., Marrella, A., Mecella, M., Soo, A.: Automated discovery of process models from event logs: Review and benchmark. TKDE **31**(4) (2019).
2. Dimaggio, M., Leotta, F., Mecella, M., Sora, D.: Process-based habit mining: Experiments and techniques. In: 2016 Intl IEEE Conferences on Ubiquitous Intelligence & Computing (UIC). pp. 145–152, IEEE (2016).
3. Günther, C.W., Van Der Aalst, W.M.: Fuzzy mining–adaptive process simplification based on multi-perspective metrics. In: BPM 2007, pp. 328–343.
4. Janiesch, C., Koschmider, A., Mecella, M., Weber, B., Burattin, A., Di Ciccio, C., Gal, A., Kannengiesser, U., Mannhardt, F., Mendling, J., et al.: The Internet-of-Things meets Business Process Management: Mutual benefits and challenges. arXiv preprint arXiv:1709.03628 (2017).
5. Leotta, F., Mecella, M., Mendling, J.: Applying process mining to smart spaces: Perspectives and research challenges. In: CAiSE Workshops 2015, Lecture Notes in Business Information Processing vol. 215, Springer, pp. 298–304.
6. Leotta, F., Mecella, M., Sora, D.: Visual process maps: A visualization tool for discovering habits in smart homes. Journal of Ambient Intelligence and Humanized Computing pp. 1–29 (2019).
7. Marrella, A., Halapuu, P., Mecella, M., Sardiña, S.: SmartPM: An adaptive process management system for executing processes in cyber-physical domains. In: BPM Demo Track 2015, CEUR-WS.org (2015), http://ceur-ws.org/Vol-1418/paper24.pdf.
8. Marrella, A., Mecella, M., Sardina, S.: Intelligent Process Adaptation in the SmartPM System. ACM Trans. Intell. Syst. Technol. **8**(2) (2016).
9. Marrella, A., Mecella, M., Sardiña, S.: Supporting adaptiveness of cyber-physical processes through action-based formalisms. AI Communications **31**(1) (2018).
10. Reichert, M., Weber, B.: Enabling Flexibility in Process-Aware Information Systems - Challenges, Methods, Technologies. Springer Berlin Heidelberg (2012).