An Adaptive Process Management System Implementation based on Situation Calculus, Indigolog and Classical Planning

Andrea Marrella and Massimo Mecella
Sapienza - Università di Roma, Italy
{marrella—mecella}@dis.uniroma1.it

Sebastian Sardina
RMIT University, Melbourne, Australia
sebastian.sardina@rmit.edu.au

Abstract

In this paper, we introduce an adaptive Process Management System implementation that combines business process execution monitoring, unanticipated exception detection and automated resolution strategies leveraging on well-established formalisms developed for reasoning about actions in Artificial Intelligence, including the situation calculus, IndiGolog and classical planning. Such formalisms provide a natural framework for the formal specification of explicit mechanisms to model world changes and responding to anomalous situations, exceptions, exogenous events in an automated way during process execution.

1 Introduction

In recent years, Business Process Management (BPM) approaches and technologies received considerable attention, as they are highly relevant from a practical point of view while offering many technical challenges for researchers. BPM focuses on overseeing how work is performed in a company by managing and optimising its business processes. Nowadays, the automation of business processes not only spans classical business domains (e.g., banks and governmental agencies), but also new settings such as healthcare, domotics or emergency management. More and more such processes are cyber-physical, as the information flowing through the process is often produced by human activities or is acquired by sensors and software services. Consequently, the execution context of these processes is more complex and not entirely predictable, as increasing amounts of data may influence unexpectedly the running of process instances.

Today there is an abundance of Process Management Systems (PMSs) driven by semi-formal activity-centric process modeling languages to enact and manage traditional business processes. However, such PMSs shy away from dealing with the inherent dynamic nature of processes enacted in cyber-physical domains, which must be robust and adaptable to unexpected conditions [Di Cicco et al., 2015].

In this paper, we tackle the above challenge by presenting SmartPM, a PMS implementation for automatically adapting processes enacted in cyber-physical domains in case of unanticipated exceptions and exogenous events. SmartPM aims at demonstrating that a targeted use of some well-established formalisms for reasoning about actions in Artificial Intelligence (AI), such as situation calculus [Reiter, 2001], IndiGolog [De Giacomo et al., 2009] and classical planning [Ghallab et al., 2001], provides a natural framework for the formal specification of explicit mechanisms to model world changes and responding to anomalous situations in an automated way during process execution. Specifically (the formal model underlying SmartPM is described in Marrella et al., 2014): (i) situation calculus theories are used to model the process data and to represent the set of tasks of the application domain of interest; (ii) the IndiGolog high-level agent language provides the formal executable semantics for processes in cyber-physical domains; (iii) classical planning techniques are used to synthesise resolution strategies in case of unanticipated exceptions and exogenous events.

2 The Approach and the System

SmartPM relies on an approach (cf. Figure 1) that builds on the dualism between an expected reality $\Psi$, the (idealized) model of reality used by the PMS to reason, and a physical reality $\Phi$, the real world with the actual values of conditions and outcomes. While $\Phi$ records what is concretely happening in the real environment during a process execution, $\Psi$ reflects what it is expected to happen. Process execution steps and exogenous events have an impact on $\Phi$ and any deviation from $\Psi$ results in the invocation of a state-of-the-art planner, which synthesises a recovery procedure to adapt the faulty process instance by removing the gap between the two realities.

To realize this approach, the implementation of SmartPM covers the modeling, execution and monitoring stages of the process life-cycle, by capturing the connection of implemented processes with the real-world objects of the cyber-physical domain of interest. To that end, the architecture of SmartPM relies on five architectural layers.

The Presentation Layer provides a graphical editor developed in Java that assists the process designer in the definition of the process model at design-time. Process knowledge is represented as a domain theory that includes all the contextual information of the domain of concern, such as the people/services that may be involved in performing the process, the tasks, the data and so forth. Data are represented through some atomic terms that range over a set of data objects, which depict entities of interest (e.g., capabilities, ser-
Specifically, if process adaptation is required, we translate (i) the domain theory defined at design-time into a planning domain, (ii) the physical reality into the initial state of the planning problem and (iii) the expected reality into the goal state of the planning problem. The planning domain and problem are the input for the planner component. If the planner is able to synthesize a recovery procedure \( \delta_x \), the Synchronization component combines \( \delta' \) (which is the remaining part of the faulty process instance \( \delta \) still to be executed), with the recovery plan \( \delta_x \), builds an adapted process \( \delta'' = (\delta_x; \delta') \) and converts it into an executable IndiGolog program so that it can be enacted by the IndiGolog engine. Otherwise, if no plan exists for the current planning problem, the process designer can try to manually adapt the faulty process instance.

The Cyber-Physical Layer is tightly coupled with the physical components available in the domain of interest. Since the IndiGolog engine can only work with defined discrete values, while data gathered from physical sensors have continuous values, the system provides several web tools that allow process designers to associate some of the data objects defined in the domain theory with the continuous data values collected from the environment. For example, we developed a web tool (as a Google Maps plugin) that allows to mark areas of interest from a real map (by selecting latitude/longitude values) and associate them to the discrete locations defined during the design stage of a process. The mapping rules generated are then saved into the Communication Manager and retrieved at run-time to allow the matching of the continuous data values collected by the specific sensor into discrete data objects.

SmartPM has been validated through several experiments that confirm the effectiveness of its AI-based approach for adapting processes in medium-sized cyber-physical domains (cf. Marrella et al., 2014). More information about the system can be found at: http://www.dis.uniroma1.it/~smartpm.

Acknowledgments. This work has been partly supported by the Italian projects SM&ST, RoMA and NEPTIS. The authors would like to thanks Pätris Halapuu for his valuable help in the SmartPM development.

References