

PROtEUS++: A Self-managed IoT Workflow Engine with Dynamic Service Discovery

Ronny Seiger, Steffen Huber, and Peter Heisig

Institute of Software and Multimedia Technology,
Technische Universität Dresden, D-01062, Germany
{Ronny.Seiger, Steffen.Huber, Peter.Heisig}@tu-dresden.de

Abstract. Despite offering various advantages, the usage of Business Process Management technologies to orchestrate workflows in the Internet of Things (IoT) is still in its infancy. In this work, we demonstrate an extended version of our PROtEUS process execution system for IoT. Besides the processing of sensor streams and interacting with humans, PROtEUS++ is capable of dynamic service invocation as well as self-management to detect and repair errors that happen during process execution. We show the system executing various dynamic and error-prone processes in the Smart Home.

1 Introduction

The IoT and its associated technologies are currently transforming the world of purely digital information systems into Cyber-physical Systems (CPS). The interaction with the physical world requires feedback loops and flexible service composition to compensate errors, which makes selection of IoT services highly context dependant. The BPM and SOA communities provide promising technologies for increasing the automation within CPS. However, the convergence of these research areas has only just started. With PROtEUS++ we present a novel self-managed IoT/CPS workflow engine with dynamic service discovery, which combines BPM and SOA concepts applied in the context of CPS. We show new use cases for PROtEUS++ in the smart home providing an adaptive light control system and executing distributed processes on mobile robots.

2 PROtEUS Base System

The *PROtEUS* base system and its highlighted extensions are shown in Fig. 1. PROtEUS is a workflow system designed to execute processes in the IoT [4]. It features a Petri net based core engine. A complex event processing (CEP) engine is used to process sensor streams from digital and physical sources. The service invoker is able to call a variety of external Web services or services deployed on its local service platform and thereby invoke physical actuators. Users can interact with processes via the HumanTask Handler and the Process Manager to control the execution. The Distribution Manager allows for executing subprocesses on other remote PROtEUS instances. The extended *PROtEUS++* system adds the *Semantic Access Layer* and the *Feedback Service* to the base system.

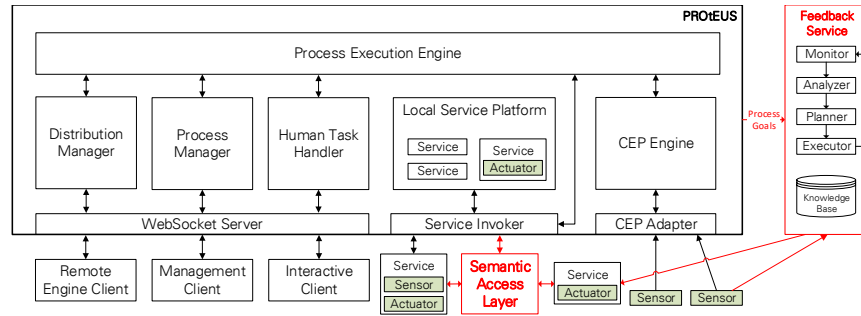


Fig. 1. The basic process execution system (PROtEUS) and its extensions.

3 Semantic Service Selection

PROtEUS is extended with a *Semantic Access Layer* (SAL) [2] as shown in Fig. 2. The core of this service is a knowledge base that contains information about all IoT sensors and actuators, their capabilities and contexts, as well as their associated IoT services and interfaces. User defined SPARQL queries can be sent to the SAL from a process activity to find and invoke IoT services in a specified context at runtime. If an IoT service can be found as a process resource, a call to our IoT middleware (here: *OpenHAB*) is issued to execute this service.

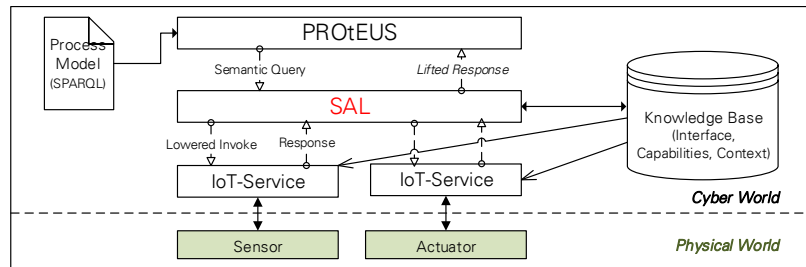


Fig. 2. Architecture of the semantic access layer (SAL).

4 Feedback Service for Self-management

The base system can be used in combination with the *Feedback Service* [3], which adds self-management (here: *self-healing*) in the form a cyber-physical feedback loop to PROtEUS (cf. Fig. 1). This service implements the *MAPE-K* loop for autonomous systems applied to the process execution [1]. This loop enables the linking of the execution of process activities to their physical effects: *Monitoring*

gathers relevant sensor data from the environment and execution system; *Analysis* analyzes the data regarding the fulfillment of the process goals; *Planning* searches for compensations (here: alternative actuators or services) in case of unexpected errors; and *Execution* executes these compensations. All components use the *Knowledge Base* to store and retrieve relevant information (e.g., for finding replacement services). This loop is executed until a process goal is reached or cancelled if errors cannot be compensated. These *Process Goals* are specified on the activity, subprocess or process level. They contain the paths to relevant sensor data, a condition defining the successful execution, and a condition defining the need for entering the Planning phase due to an error [4].

5 Demo

We present several real-world use cases of PROtEUS++ executing three example processes. 1) A process demonstrating the base system and the dynamic service selection via the SAL—a health monitoring process asking the user for its well-being in case of a detected emergency and calling an ambulance if the user is unresponsive. 2) A process demonstrating the base system interacting with the Feedback Service. A continuous process controls the light levels in a room to be within certain thresholds—selecting an alternative light source or notifying the user if the lights fail. 3) A process demonstrating the execution and feedback control of a distributed process on a service robot—repeating the subprocess on another robot if it fails during the driving to different locations. Along with these processes, we present the modelling tool [5] as well as a mobile control center app to interact with the workflows [6].

Acknowledgements

This research has received funding under the grant number 100268299 by the European Social Fund (ESF) and the German Federal State of Saxony.

References

1. Autonomic Computing: An architectural blueprint for autonomic computing. IBM Publication (2003)
2. Huber, S., Seiger, R., Schlegel, T.: Using semantic queries to enable dynamic service invocation for processes in the internet of things. In: ICSC. pp. 214–221 (Feb 2016)
3. Seiger, R., Huber, S., Heisig, P., Assmann, U.: Enabling Self-adaptive Workflows for Cyber-physical Systems, pp. 3–17. Springer International Publishing (2016)
4. Seiger, R., Huber, S., Schlegel, T.: Proteus: An integrated system for process execution in cyber-physical systems. In: Enterprise, Business-Process and Information Systems Modeling, LNBIP, vol. 214, pp. 265–280 (2015)
5. Seiger, R., Keller, C., Niebling, F., Schlegel, T.: Modelling complex and flexible processes for smart cyber-physical environments. JOCS 10, 137–148 (2015)
6. Seiger, R., Lemme, D., Struwe, S., Schlegel, T.: An interactive mobile control center for cyber-physical systems. In: UbiComp Adjunct Proceedings. pp. 193–196. UbiComp '16, ACM, New York, NY, USA (2016)