Journal of Computational Science 10 (2015) 137–148

Contents lists available at ScienceDirect

Journal of Computational Science

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

Modelling complex and flexible processes for smart cyber-physical environments

Ronny Seiger*, Christine Keller, Florian Niebling, Thomas Schlegel

Software Engineering of Ubiquitous Systems, Technische Universität Dresden, Germany

ARTICLE INFO

Article history:
Received 21 February 2014
Received in revised form 12 June 2014
Accepted 23 July 2014
Available online 11 August 2014

Keywords:
Smart workflows
Process modelling
Cyber-physical systems
Smart factory
Flexible processes

ABSTRACT

Cyber-physical systems introduce new requirements for modelling and executing autonomous processes. Current workflow languages are not able to completely fulfil these requirements, as they lack expressiveness and flexibility. In this paper, we present an object-oriented workflow language for formalizing processes within heterogeneous and dynamic environments. Its component-based meta-model enables the hierarchical composition of processes and process variants. Domain-specific typing and polymorphism leverage the flexibility of processes by enabling the dynamic selection of process components. We present a modelling environment and a distributed execution engine for the meta-model. In addition, we discuss the use of semantic technologies for smart workflows.

1. Introduction

Business processes have gained an increasing importance in describing complex correlations between distributed systems and executing composite workflows. Especially in the field of online trading and manufacturing, modelling and execution languages for business processes, e.g., BPMN and BPEL, have proven to be well suited to formalize high-level sequences of tasks and activities involving web service invokes and human interaction.

However, the on-going integration and combination of embedded systems and distributed cloud-based services into cyber-physical systems (CPS) and smart environments, lead to a number of new requirements for process modelling and execution. Most current workflow languages lack structure, expressiveness, and flexibility to meet these requirements.

Some of the drawbacks of state of the art process modelling languages include: only weak means for typing of process components and data, mostly static calls to a fixed set of service types, and reduced flexibility considering runtime modelling and adaptation. Modelling tools often produce code, which is incompatible with execution environments, and only a subset of the model elements is supported.

In addition, many long-established workflow modelling languages have been extended and evolved over time, mostly by adding new components and modifying the respective meta-models in order to meet new requirements and provide new functionality. This has led to complex and ambiguous process modelling languages containing special solutions for specific problems and domains.

In this paper, we present a new meta-model for processes designed to meet the requirements of current and future ubiquitous systems. We believe that by using model-based approaches, we can create a modular and extensible workflow language. With the help of this language, we will then be able to model flexible and dynamic processes for the automation of workflows. Current semantic technologies will help us with developing a smart and context-adaptive process engine and modelling environment. We focus on adhering to simple structures for the core of the process meta-model and at the same time being able to easily extend this model by means of object-orientation. Nevertheless, we are able to map process models of other workflow languages to models compatible with our system.

The paper is structured as follows: Section 2 presents some basic terms and explanations. Section 3 lists requirements that are introduced with the emergence of ubiquitous systems. Section 4 gives a brief overview of related work and evaluates state-of-the-art workflow languages with respect to their suitability for cyber-physical systems. Section 5 describes our process model for complex and

* Corresponding author at: Technische Universität Dresden, Fakultät Informatik, Jun.-Prof. SEUS, D-01062 Dresden, Germany. Tel.: +49 0 35146338286; fax: +49 0 35146342663.

E-mail addresses: Ronny.Seiger@tu-dresden.de (R. Seiger), Christine.Keller@tu-dresden.de (C. Keller), Florian.Niebling@tu-dresden.de (F. Niebling), Thomas.Schlegel@tu-dresden.de (T. Schlegel).

http://dx.doi.org/10.1016/j.jocs.2014.07.001
1877-7503/

flexible business processes in detail, including an extended example and an evaluation of the meta-model. Section 6 demonstrates practical aspects with respect to implementing the model, a modelling tool, and a process execution engine. Section 7 discusses our approach and shows some aspects to further extend our research. Section 8 concludes the paper.

2. Basic concepts

We will start with clarifying basic terms and concepts that are used within the context of this paper. As our focus lies on the scope of ubiquitous computing and cyber-physical systems, we will introduce these concepts first, as well as, our understanding of processes. Second, the paradigms of model-driven architecture and meta-modelling will be presented, as our own approach is based on these concepts.

2.1. Ubiquitous computing

In his article "The Computer for the 21st Century", published in 1991, Mark Weiser introduced his vision of "the age of calm technology, when technology recedes into the background of our lives" and thereby coined the term "ubiquitous computing" [34]. Ubiquitous computing can be found at the intersection of pervasive computing, mobile computing, and ambient intelligence, and stands for systems that are unobtrusively integrated into everyday objects and activities.

2.2. Cyber-physical systems

Cyber-physical systems (CPS) can be regarded as a major step towards Weiser's vision. CPS comprise networks of embedded, heterogeneous sensors and actuators into complex distributed systems, that are often linked to cloud-based services and cross-boundary systems. A closed loop between local sensing, remote processing, and local controlling can often be found within cyber-physical systems. Real-world objects are represented digitally and taken into consideration when planning and executing processes in a cyber-physical system. In addition, CPS are highly dynamic with respect to their components, i.e. devices and services can be added and removed at any time. By constantly collecting context information [1], cyber-physical systems are able to adapt themselves to the current users and environment, thus evolving into so-called "smart spaces", e.g. smart homes, smart offices, and smart factories. CPS intend to create a strong link between the physical world and the cyber world, and to support their users with performing their daily tasks.

2.3. Processes

Processes (workflows) have been used to describe complex sequences of tasks and function calls in order to model the high-level behaviour of so-called systems of systems. Due to the large increase of distributed and loosely coupled systems over the last decades, the need for an additional layer describing workflows between multiple entities has been generated. With traditional approaches, it is not possible any more to implement all algorithms and cross-boundary interactions within the software application shipped with one product. The usage of processes helps with creating autonomous environment and the automation of repeating tasks.

We therefore define a process for the scope of our work as follows: Processes represent a set of actions (process steps), which are connected with each other by a unidirectional order relation describing the order of execution of the steps [27].

2.4. Model-driven architecture and meta-modelling

Using models throughout the development process of a software system incorporates several advantages with respect to modularization, reusability, extensibility, automatic code generation, and maintenance. The process layer on top of software products and systems should also be highly model-based and described by a platform independent model (PIM).

With the Meta-Object Facility (MOF) [22], the OMG introduced a de facto standard for model-driven engineering [2], describing several (meta-) levels of abstraction for modelling various kinds of systems. As we will also be dealing with models and meta-models throughout this paper, we want to clarify our understanding of these terms and their use within the context of process modelling at this point.

- *Process meta-meta-model*: A process meta-meta-model (MOF-M2) defines the semantic and syntactic elements and structures used in the process meta-model.
- *Process meta-model*: A process meta-model defines all elements, types, and relations that can be used for modelling processes as well as their structural combinations. The process meta-model (MOF-M1) is an instance of the process meta-meta-model.
- *Process model*: A process model is the abstract description of an actual process, which can be instantiated and executed at runtime. The process model (MOF-M0) is an instance of the process meta-model.
- *Process instance*: A process instance represents a concrete process at execution time, having a runtime state. The process instance is an instance of the process model.

In the main part of this work, we will put our focus on presenting a new process meta-model, but we will also briefly describe the underlying process meta-meta-model.

3. Requirements for modelling ubiquitous processes

In order to evaluate current workflow languages with respect to their suitability for being used within ubiquitous systems (UbiSys), we will first outline some special requirements that come along with developing ubiquitous systems. Some of the following requirements are already predominant within current system architectures. However, UbiSys combine them to a large degree.

- *Dynamics*: Ubiquitous systems, as well as cyber-physical systems (CPS), are characterized as being highly dynamic with respect to the number and availability of its components, devices, and services. Therefore, modelling service invocations within processes on the instance level, i.e. the invocation of a concrete service, may not be suitable due to its possible unavailability. Hence, we also need to be able to model process steps and service calls on the type level, i.e. a certain type of service should be invoked. This way, we do not necessarily need to know at modelling time, which concrete service or device will be executing the process step.
- *Heterogeneity*: In a CPS there are usually numerous heterogeneous services and devices integrated into a so-called system of systems. However, when modelling workflows, a unified view on these components would be helpful. In addition, we would like to support a wide range of different services types and be able to easily extend this set. Complementary to the aforementioned requirement, there should also be a way of assigning an activity to a certain handling entity (resource) on the instance or the type level.