Introducing Entity-Based Concepts to Business Process Modeling

Klaus Sperner, Sonja Meyer, and Carsten Magerkurth

SAP Research Switzerland
Kreuzplatz 20, 8008 Zürich, Switzerland
{klaus.sperner,sonj.meyer,carsten.magerkurth}@sap.com

Abstract. The so-called Internet of Things (IoT) that comprises interconnected physical devices such as sensor networks and its technologies like Radio Frequency Identification (RFID) is increasingly adopted in many industries and thus becomes highly relevant for process modeling and execution. As BPMN 2.0 does not yet consider the idiosyncrasies of real-world entities we suggest new modeling concepts for a physical entity as well as a sensing task and an actuation task to make BPMN IoT-aware.

Keywords: Internet of Things, IoT, Business Process Modeling, BPMN.

1 Introduction

Today, the gap between the sensors and actuators in the Internet of Things and the business systems at the higher layers of the enterprise world is still a reality. A unified reference architecture is therefore a key prerequisite for realizing interoperability within the IoT world and especially for integration with business processes, so that applications can be realized that are both IoT-aware and meet the requirements of enterprise systems. Currently, the IoT domain is being standardized based on a unified IoT domain model [4] that is discussed in the next section. We apply core concepts from the IoT domain model to the Business Process Model and Notation (BPMN) 2.0 [2]. As BPMN focuses on activities and the implicated flow of process steps while the IoT domain model stresses the relationship between entities and other constructs for which well defined processes might or might not exist, a mapping and integration problem between both domains becomes apparent.

In the remainder of this paper we will use the following typographical conventions: Concepts from the IoT domain model are typeset in sans serif, and concepts from the BPMN metamodel are typeset in `constant width`.

2 A Domain Model for the Internet of Things

Based on [1] and [3] a first version of the domain model for the Internet of Things has been developed in [4]. As this model is an extensive conceptual representation

R. Dijkman, J. Hofstetter, and J. Koehler (Eds.): BPMN 2011, LNBIP 95, pp. 166–171, 2011.

of the IoT domain, it serves as a basic fundament for the presented research work. In this section we shortly explain its core concepts, please confer Fig. 6 in [4].

As the term Internet of Things suggests, the *thing* is the most important concept in the domain model; it is called Physical Entity. The main purpose of the domain model is to show, in which way any kind of User can interact with a Physical Entity, so the association interacts with between the User and the Physical Entity is the key concept of the model.

To represent a Physical Entity in the digital world, a Virtual Entity is associated to it. While a Physical Entity exists only once, there can be multiple Virtual Entities representing it. Every combination of a Virtual Entity and its Physical Entity forms one Augmented Entity.

In order to make the Physical Entity accessible from the digital world some hardware, a so-called Device, is attached to the Physical Entity. To distinguish between Devices which observe a Physical Entity and Devices which control a Physical Entity, three subclasses of Device are introduced in the domain model: A Sensor can be attached to a Physical Entity to monitor it, an Actuator can be attached to a Physical Entity to act on it, and a Tag can be attached to a Physical Entity to identify it.

To close the gap between the Device and the digital world, it hosts several software components in its memory, which are called Resources. Depending on the type of the Device, such a Resource has information about the Physical Entity, or it can act on the Physical Entity. Virtual Entities can be associated with Resources via the Device and the Physical Entity.

As the Resources hosted on the Devices are expected to have different interfaces, Services are introduced as an abstraction concept; they access the Resources. These Services provide well-defined interfaces to the Users for invoking them. Since the Virtual Entities are associated with the Resources, they can be associated with the Services, which access these Resources, as well.

Summing up, the interaction of the User with the Physical Entity can be detailed as follows: The User invokes a Service, which accesses a Resource, which is hosted on a Device, which either monitors the Physical Entity or acts on the Physical Entity.

3 Mapping between IoT Concepts and BPMN Concepts

To bring the IoT domain and business process modeling together, we analyse, how the concepts from the domain model can be modeled in BPMN, and if the chosen BPMN concepts are adequate to reflect the specificities of the Internet of Things. Table 1 summarizes the proposed mapping of IoT and BPMN concepts.

A Physical Entity can be modeled as a `TextAnnotation` to an `Activity`. Since the concept of a Physical Entity is not defined in BPMN, we use the `TextAnnotation` as the general BPMN concept for attaching further details to a modeling element. The usage of a `TextAnnotation` for the modeling of a Physical Entity is not sufficient, and leads to multiple problems: As specified in [2] a `TextAnnotation` only provides additional information for the reader of a

Table 1. Mapping of IoT concepts to current BPMN concepts and their sufficiency for modeling of IoT aware processes

IoT concept	BPMN concept	Sufficiency for Modeling
Physical Entity	TextAnnotation	not sufficient
Virtual Entity	DataObject	sufficient
Augmented Entity	—	not needed in BPMN
Sensor	Participant	sufficient
monitoring	ServiceTask	not sufficient
Actuator	Participant	sufficient
acting	ServiceTask	not sufficient
Tag	—	not needed in BPMN
Resource	—	not needed in BPMN
Service	ServiceTask	not sufficient
User	Participant, Event	sufficient

diagram, but does not affect the flow of the process. Accordingly, the specification of a dedicated Physical Entity in the model would not be obeyed during the execution of the process. Second, a TextAnnotation can be attached to only one object in the collaboration diagram, but one Physical Entity could for instance be monitored by multiple Sensors. Third, there is no definition of a lifecycle for a TextAnnotation, but a Physical Entity naturally persists between several process executions.

The Virtual Entities relating to the Physical Entities can be modeled as collection DataObjects in a BPMN collaboration diagram, because this is the concept for modeling data, which BPMN provides. As the concept of a Virtual Entity is not only specific to the IoT domain, but a general concept in informatics, this approach of modeling is considered to be sufficient, and fully serves our purposes.

As the Augmented Entity in the IoT domain model is only an abstract concept to combine a Virtual Entity and a Physical Entity, there is no corresponding concept in BPMN. Since such an abstract concept would not add any benefit for the practical modeling of business processes, the Augmented Entitiy is not needed in BPMN.

Sensors and Actuators can sufficiently be modeled as Participants in a collaboration diagram, because this concept is defined in BPMN to represent a PartnerEntity, which executes a process. If multiple instances of Sensors or Actuators are involved, the corresponding pools can be decorated with multi-instance markers.

As a Sensor is described in the domain model to monitor a Physical Entity and an Actuator is said to act on a Physical Entity, these monitoring and actuation associations can be expressed with ServiceTasks in a BPMN diagram. In BPMN Tasks are the central concept of execution during the process flow, so this modeling is appropriate. The particular Task can implicitly be identified as a Task for monitoring or actuation through the Participant executing it, the Physical Entity, which is represented in the attached TextAnnotation, and the name of the Task, which states the means of monitoring or actuation. So far it can not be explicitly marked as a specific Task for monitoring or actuation.

Even though the concept of attaching Tags to Physical Entities to make them identifiable is one of the most widely adopted applications of Internet of Things technologies, it is a low-level technology, and an introduction to BPMN is not needed. If the process of this identification is of particular interest, the Tag can be considered as a Physical Entity, which is monitored by a Sensor.

The Resources introduced in the domain model are the software components running on the Devices. Since BPMN collaboration diagrams focus on `Participants` executing `Tasks`, introducing the Resource to BPMN is not reasonable.

In the domain model the Services provide a consistent interface for the Resources, which are hosted by Sensors and Actuators. In BPMN these Services can be modeled as `ServiceTasks`, which are executed by the `Participants`, which represent Sensors and Actuators. As described above, the Physical Entities, on which these `ServiceTasks` operate, are referenced in the attached `TextAnnotations`. Although the modeling of the Services via `ServiceTasks` seems appropriate, it is not sufficient: In a collaboration diagram it is impossible to distinguish, if the Service performs a sensing or an actuating interaction with the Physical Entity.

For the User in the domain model, who invokes the Service, the different BPMN concepts for the invocation of `Tasks` can be considered. This could be a `Message` from another `Participant` or an `Event`. As there are many concepts for the invocation of `Tasks` in BPMN, no new concept for the User needs to be introduced.

The result of our analysis is summarized in Table 1: The Physical Entity and the monitoring and actuating Services can not be represented sufficiently in BPMN.

4 New Modeling Concepts

4.1 Modeling of Physical Entities

To facilitate the modeling of Physical Entities we propose a new BPMN element `PhysicalObject`, which is subclassed from `FlowElement` in the BPMN metamodel. In contrast to the `DataObject` the `PhysicalObject` is not subclassed from `ItemAwareElement`, because these are designed to store items, but a Physical Entity is an item itself. Accordingly, the BPMN elements `DataState` and `ItemDefinition` are not needed for the `PhysicalObject`. To enable the modeler to refer to the same Physical Entity in multiple places of a diagram, we introduce a `PhysicalObjectReference` analogous to the `DataObjectReference`.

We propose the illustration of a brick shown in Fig. 1 as stencil for the `PhysicalObject` to make this new concept usable in a BPMN collaboration

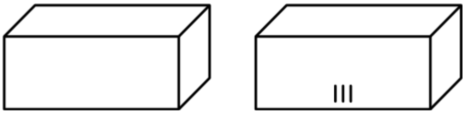

Fig. 1. Stencils for a single `PhysicalObject` (left) and a collection of `PhysicalObjects` (right)

diagram. As it is possible that not only one instance of a PhysicalObject is used in a process, the stencil can be decorated with a multiple instance marker, like it is defined in [2] for DataObjects. The name of the PhysicalObject will be placed above the base line of the stencil.

As PhysicalObjects are obviosly physical, their lifecycle is not limited to the lifecycle of the modeled process; they persist between process instantiations. This differentiates them from DataObjects, which are defined in [2] to not persist between process instantiations, but is similar to DataStores, which are also persistent according to [2].

4.2 Modeling of monitors and acts on Associations

To empower the modeler to express that a Task reflects a monitors or an acts on relationship between the Participant and the Physical Entity represented in the TextAnnotation, we introduce dedicated SensingTasks and ActuatingTasks as new subclasses of the Task class.

Since a Sensor produces data about a Physical Entity by monitoring it, the SensingTask must output this data and provide it for the remainder of the process. Hence, we can derive the following constraint for the SensingTask: The InputOutputSpecification, which is associated with the Activity superclass of the SensingTask, must reference at least one DataOutput, which must also be referenced by at least one OutputSet of the InputOutputSpecification. Because an Actuator needs an actuating value, an analogous constraint applies to the ActuationTask: The InputOutputSpecification associated to the ActuationTask must reference at least one DataInput, which must also be referenced by at least one InputSet.

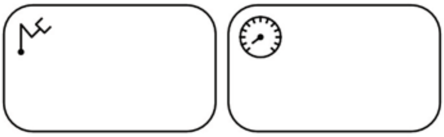

Fig. 2. Stencils for an ActuationTask (left) and a SensingTask (right)

For the new Tasks we propose the icons shown in Fig. 2 to decorate the stencil for the Task with: The ActuationTask is decorated with an illustration of a robot arm and the SensingTask is depicted with a gauge.

4.3 Connecting PhysicalObjects with ActuationTasks and SensingTasks

Analogous to the DataAssociation defined in [2], we define a new abstract class PhysicalAssociation, derived from BaseElement, and two concrete subclasses

ActuationAssociation and SensingAssociation. The former is a directed connection from an ActuationTask to the PhysicalObject, on which the represented Actuator acts on; it can be considered as a *flow of physical interaction*. The latter is directed from a PhysicalObject to a SensingTask; this can be considered as a *flow of physical information* from a Physical Entity to a Sensor.

To depict such associations in a BPMN diagram, we propose to reuse the same stencil as it is defined for a DataAssociation (cf. Fig. 10.65 in [2]).

5 Conclusions and Outlook

With this paper we have demonstrated first concrete steps towards bringing together the Internet of Things and Business Process Management. We have come up with first suggestions for augmentations to the BPMN 2.0 standard to reflect the most important aspects of the IoT domain model.

Our future work will deal with the further elaboration of the concepts presented in this paper including serializations of the new modeling concepts in the BPMN CMOF and the BPMN XML Schema and the serializations of the diagram elements in the BPMNDI XML Schema.

Acknowledgments. The authors would like to thankfully acknowledge the support for this work provided by the European Commission within the FP7 project IoT-A, contract number 257521. In this context we thank Stephan Haller, Alessandro Serbanati, Martin Bauer, Ralf Kernchen, Joachim Walewski, and Alessandro Bassi for their work on the Internet of Things domain model, and all other contributors to the model.

References

1. Haller, S.: The Things in the Internet of Things: Poster at the Internet of Things 2010 (2010), http://iot-a.eu/public/news/resources/TheThingsintheInternetofThings_SH.pdf (accessed June 30, 2011)
2. Object Management Group (OMG), Business Process Model and Notation (BPMN) Version 2.0 (2011), http://www.omg.org/spec/BPMN/2.0/ (accessed June 30, 2011)
3. Serbanati, A., Madaglia, C.M., Ceipidor, U.B.: Building Blocks of the Internet of Things: State of the Art and Beyond. In: Turcu, C. (ed.) RFID/Book 3. InTech, Rijeka (in press, 2011)
4. Walewski, J.W. et al.: Project Deliverable D1.2 - Initial Architectural Reference Model for IoT (2011), http://www.iot-a.eu/public/public-documents/project-deliverables/1/1/D1%202_Initial_architectural_reference_model_for_IoT.pdf/at_download/file (accessed June 30, 2011)