

# Internet of Things Architecture

## IoT-A

### Project Deliverable D2.2 – Concepts for Modelling IoT-Aware Processes

Project acronym: IOT-A  
 Project full title: The Internet of Things Architecture  
 Grant agreement no.: 257521

|                            |  |  |
|----------------------------|--|--|
| Doc. Ref.:                 | D2.2   |  |
| Responsible Beneficiary :  | SAP AG   |  |
| Editor(s):                 | Sonja Meyer  |  |
| List of contributors:      | Sonja Meyer (SAP), Klaus Sperner (SAP), Carsten Magerkurth (SAP), Stefan Debortoli (SAP), Matthias Thoma (SAP) |  |
| Reviewers:                 | Gregorio Martin Romero (TID)   |  |
| Contractual Delivery Date: | 28.02.2012   |  |
| Actual Delivery Date:      | 12.03.2012   |  |
| Status:                    | Final  |  |
| Version and date           | Changes  | Reviewers / Editors  |
| V01- 21.04.2011            | Table of Contents  | Sonja Meyer (SAP), Klaus Sperner (SAP), Carsten Magerkurth (SAP) |
| V02- 01.10.2011            | Input all chapters   | All  |
| V03- 01.02.2012            | Refinement all chapters  | All  |
| V04- 02.03.2012            | Input 3.3 and 3.1.3  | Klaus Sperner (SAP)  |
| V05- 02.03.2012            | Completion all chapters  | Sonja Meyer (SAP)  |
| V06- 03.03.2012            | Proof reading  | Sonja Meyer (SAP)  |
| V07- 12.03.2012            | Review   | Gregorio Martin Romero (TID)                                     |

Project co-funded by the European Commission within the Seventh Framework Programme (2007-2013)

|                      |   |    |
|----------------------|---|----|
| PU<br>PP<br>RE<br>CO | Dissemination Level<br>Public<br><del>Restricted to other programme participants (including the Commission Services)</del><br><del>Restricted to a group specified by the Consortium (including the Commission Services)</del><br><del>Confidential, only for members of the Consortium (including the Commission Services)</del> | PU |
|----------------------|---|----|

## Executive Summary

Existing graphical and executable process modelling languages are not designed for representing IoT-aware business processes. In IoT-A, we therefore introduce the IoT-aware Process Modelling Concept (IAPMC) seeking to lower the barrier for applying IoT technology like sensors and actuators to current and new business processes. Professional process modellers without having a deep programmatic understanding get an advanced guidance during the modelling activities, so that they can easily work in the IoT world. The main objective of this deliverable is to provide a specification of the IoT-aware modelling methodology and the process modelling concept. We will describe in detail the identified requirements, design principles and concept components consisting of a first language independent concept proposition and a concrete BPMN 2.0 component implementation.

IoT characteristics in terms of process modelling requirements will be defined and evaluated against current modelling notations. In the context of IoT-aware process modelling, a new modelling concept will be required, since evaluated existing modelling approaches lack to completely cover the IoT domain. The design principles will comprise a phase model from the process design view on the three process phases design, resolution and execution and their interfaces, as well as the BPMN 2.0 standard conventions that need to be considered for any BPMN extensions. In order to realize these design principles, we will identify eight weak points of current process modelling notations and create seven new elements for the IAPMC. Current process modelling concepts can be thus complemented by the new IAPMC elements. We will present a coherent metamodel of these elements, properties and relationships. Based on this metamodel we apply the IAPMC elements to BPMN 2.0 and come up with a first IAPMC BPMN 2.0 integration.

The BPMN 2.0 extension will be used for the IoT-A WP2 tool implementation. The abstract process models will be created by end-users using a process editor and will be stored in files based on the BPMN 2.0 XML format extended by the IAPMC. This abstract process models will be constitutively completed by the IoT-A process resolution environment that will create an executable IoT-aware process model. Finally, the process models presented in the IAPMC BPMN 2.0 integration can be executed by the IAPMC BPMN 2.0 compliant process execution engine, which will be also developed as part of the IoT-A WP2 tool implementation due to M30. As part of IoT-A this deliverable aims to increase the application of IoT technologies in the world of enterprise systems by bridging the gap on business process modelling level.

## Table of Acronyms

| Acronym | Definition   |
|---------|--|
| ARIS    | Architecture of Integrated Information Systems                       |
| BPEL    | see WS-BPEL  |
| BPEL4WS | Business Process Execution Language for Web-Services                 |
| BPM     | Business Process Management  |
| BPML    | Business Process Management Language                                 |
| BPMN    | Business Process Model and Notation                                  |
| D       | Deliverable  |
| DS      | Design Science   |
| ebBP    | eBusiness Business Process   |
| ebXML   | eBusiness Extensible Markup Language                                 |
| EPC     | Event-Driven Process Chain   |
| eEPC    | Extended Event-Driven Process Chain                                  |
| ERP     | Enterprise Resource Planning   |
| IAPMC   | IoT-aware Process Modelling Concept                                  |
| IoT     | Internet of Things   |
| IQM     | IoT Quality Metric   |
| OASIS   | Organization for the Advancement of Structured Information Standards |
| OMG     | Object Management Group  |
| PPP     | Public-Private Partnership Programs                                  |
| QoI     | Quality of Information   |
| RFID    | Radio-Frequency Identification                                       |
| SOA     | Service-Oriented Architecture  |
| UML     | Unified Modelling Language   |
| W3C     | World Wide Web Consortium  |
| WfMC    | Workflow Management Coalition  |



|         |  |
|---------|--|
| WP      | Work Package                                     |
| WPDL    | Workflow Process Definition Language             |
| WS-BPEL | Web Services Business Process Execution Language |
| WS-CDL  | Web Services Choreography Description Language   |
| WSDL    | Web Service Description Language                 |
| XML     | Extensible Markup Language                       |
| XPDL    | XML Process Definition Language                  |
| YAWL    | Yet another Workflow Language                    |

## Table of Content

|           |  |               |
|-----------|--|---------------|
| <b>1.</b> | <b>Introduction.....</b>                               | <b>- 8 -</b>  |
| 1.1       | Purpose of Deliverable .....                           | - 8 -         |
| 1.2       | Methodology .....                                      | - 9 -         |
| 1.3       | Document Structure .....                               | - 9 -         |
| <b>2.</b> | <b>Existing Modelling Approaches.....</b>              | <b>- 11 -</b> |
| 2.1       | Main Terminology Summarized .....                      | - 11 -        |
| 2.2       | History of Business Process Modelling .....            | - 12 -        |
| 2.3       | Main Concepts .....                                    | - 13 -        |
| 2.3.1     | Business Process Modelling and Notation .....          | - 13 -        |
| 2.3.2     | Unified Modelling Language .....                       | - 15 -        |
| 2.3.3     | Event-driven Process Chain .....                       | - 16 -        |
| 2.3.4     | Web Services Business Process Execution Language ..... | - 17 -        |
| 2.3.5     | XML Process Definition Language .....                  | - 18 -        |
| 2.3.6     | eBusiness Business Process .....                       | - 19 -        |
| 2.4       | Further Approaches Summarized .....                    | - 19 -        |
| 2.5       | Related Research Initiatives.....                      | - 20 -        |
| <b>3.</b> | <b>Requirements Analysis for the IAPMC .....</b>       | <b>- 21 -</b> |
| 3.1       | IoT-Specific Characteristics .....                     | - 21 -        |
| 3.1.1     | General Process Notation Aspects .....                 | - 21 -        |
| 3.1.2     | IoT specific Aspects .....                             | - 22 -        |
| 3.1.3     | Process Editor Aspects .....                           | - 25 -        |
| 3.2       | Evaluation of Existing Modelling Concepts .....        | - 27 -        |
| 3.3       | Evaluation of Existing Modelling Tools .....           | - 29 -        |
| 3.3.1     | Activiti .....   | - 30 -        |
| 3.3.2     | Bonita Open Solution .....                             | - 32 -        |
| 3.3.3     | jBPM.....  | - 33 -        |
| 3.3.4     | Conclusions .....                                      | - 36 -        |
| <b>4.</b> | <b>General Design Principles.....</b>                  | <b>- 37 -</b> |
| 4.1       | The Three Phases Model .....                           | - 37 -        |
| 4.2       | BPMN 2.0 Conventions.....                              | - 38 -        |
| <b>5.</b> | <b>IoT-Aware Process Modelling (IAPM) .....</b>        | <b>- 39 -</b> |
| 5.1       | Main Conceptual Components.....                        | - 39 -        |
| 5.2       | Formalisation .....                                    | - 40 -        |
| 5.2.1     | IoT Activity .....                                     | - 41 -        |
| 5.2.1.1   | Actuation Activity .....                               | - 41 -        |
| 5.2.1.2   | Sensing Activity .....                                 | - 45 -        |
| 5.2.2     | IoT Device .....                                       | - 50 -        |
| 5.2.3     | Process Resources .....                                | - 54 -        |



|                        |   |               |
|------------------------|---|---------------|
| 5.2.4                  | Physical Entity .....                                   | - 58 -        |
| 5.2.5                  | Real World Data Object / Store .....                    | - 64 -        |
| 5.2.6                  | Mobility Aspect.....                                    | - 68 -        |
| 5.2.7                  | IoT Process Ratios.....                                 | - 76 -        |
| <b>5.3</b>             | <b>From IAPMC to BPMN 2.0 Model Instantiation .....</b> | <b>- 77 -</b> |
| 5.3.1                  | Complete IAPMC Overview.....                            | - 77 -        |
| 5.3.2                  | Complete IAPMC BPMN 2.0 Integration .....               | - 78 -        |
| <b>5.4</b>             | <b>Process Examples .....</b>                           | <b>- 80 -</b> |
| 5.4.1                  | Business Process “Sensor Based Quality Control” .....   | - 80 -        |
| <b>5.5</b>             | <b>Conclusions.....</b>                                 | <b>- 81 -</b> |
| <b>6.</b>              | <b>Conclusions and Outlook.....</b>                     | <b>- 83 -</b> |
| 6.1                    | Summary .....   | - 83 -        |
| 6.2                    | Outlook.....  | - 83 -        |
| <b>References.....</b> |   | <b>- 85 -</b> |
| <b>Glossary .....</b>  |   | <b>- 88 -</b> |
| <b>Annex .....</b>     |   | <b>- 94 -</b> |

## Index of Figures

|  |        |
|--|--------|
| Figure 1: Applied DS Research Methodology Process Model of [37].....                     | - 9 -  |
| Figure 2: Development of the most popular process modelling notation [10] .....          | - 12 - |
| Figure 3: Activiti Modeler.....  | - 30 - |
| Figure 4: Activiti Designer .....  | - 31 - |
| Figure 5: Bonita Open Solution .....   | - 33 - |
| Figure 6: jBPM Eclipse Plugin .....  | - 34 - |
| Figure 7: BPMN 2.0 Eclipse Editor .....  | - 35 - |
| Figure 8: Business Process Phases Model.....   | - 37 - |
| Figure 9: Actuation Activity Metamodel.....  | - 43 - |
| Figure 10: Actuation Task BPMN Integration .....   | - 44 - |
| Figure 11: Actuation Task Object .....   | - 44 - |
| Figure 12: Sensing Activity Metamodel.....   | - 48 - |
| Figure 13: Sensing Task BPMN Integration.....  | - 49 - |
| Figure 14: Sensing Task Object .....   | - 49 - |
| Figure 15: IoT Device Metamodel.....   | - 52 - |
| Figure 16: IoT Device BPMN Integration .....   | - 53 - |
| Figure 17: Collapsed and Extended IoT Device Object.....                                 | - 53 - |
| Figure 18: BPMN Resource Metamodel .....   | - 55 - |
| Figure 19: IoT Resource Metamodel Integration .....                                      | - 56 - |
| Figure 20: Physical Entity Metamodel.....  | - 60 - |
| Figure 21: Physical Entity BPMN Integration .....  | - 61 - |
| Figure 22: Collapsed and Extended Physical Entity Object.....                            | - 62 - |
| Figure 23: Real-World Data Metamodel.....  | - 66 - |
| Figure 24: Real-World Data Object / Store Integration .....                              | - 67 - |
| Figure 25: Natural, Collapsed and Extended Real-World Data Object.....                   | - 67 - |
| Figure 26: Natural, Collapsed and Extended Real-World Data Store.....                    | - 67 - |
| Figure 27: Mobility Class Diagram .....  | - 71 - |
| Figure 28: Mobile Property (a), Location-based Property (b).....                         | - 71 - |
| Figure 29: BPMN Class Diagram Mobile Process .....                                       | - 71 - |
| Figure 30: Mobile BPMN Example Process .....   | - 72 - |
| Figure 31: BPMN 2.0: Participant extension .....   | - 73 - |
| Figure 32: Mobile Participant Collaboration Diagram .....                                | - 74 - |
| Figure 33: Activity Class Diagram.....   | - 75 - |
| Figure 34: Location-based Activity (a) and Collapsed location-based Sub-Process (b)..... | - 75 - |
| Figure 35: Ecore Model of IQM .....  | - 77 - |
| Figure 36: IAPMC Metamodel Components.....   | - 78 - |
| Figure 37: IAPMC BPMN 2.0 Integration .....  | - 79 - |
| Figure 38: Sensor Based Quality Control.....   | - 81 - |

## Index of Tables

|  |        |
|--|--------|
| Table 1: Overview of BPM efforts .....                                     | - 13 - |
| Table 2: Basic Elements of BPMN [14] .....                                 | - 14 - |
| Table 3: Basic Elements of UML Activity Diagram [18] .....                 | - 16 - |
| Table 4: EPC Notation Main Elements .....                                  | - 17 - |
| Table 5: WS-BPEL Language Constructs .....                                 | - 18 - |
| Table 6: Requirements List of General Process Notation Aspects .....       | - 22 - |
| Table 7: Requirements List of IoT specific Aspects .....                   | - 25 - |
| Table 8: Requirements List of Process Editor Aspects .....                 | - 27 - |
| Table 9: Coverage of IoT Characteristics by BPM Notations .....            | - 29 - |
| Table 10: Coverage of IoT Domain Model Concepts by BPMN 2.0 .....          | - 40 - |
| Table 11: ActuationTask Attributes and Associations .....                  | - 44 - |
| Table 12: TaskRequirement Attributes and Associations .....                | - 45 - |
| Table 13: TaskRequirementParameter Attributes and Associations .....       | - 45 - |
| Table 14: IoTTask Model Attributes and Associations .....                  | - 50 - |
| Table 15: IoTDeviceRole Attributes and Associations .....                  | - 54 - |
| Table 16: IoTDeviceRoleParameter Attributes and Associations .....         | - 54 - |
| Table 17: IoTPerformer Attributes and Associations .....                   | - 56 - |
| Table 18: IoTDeviceRole Attributes and Associations .....                  | - 57 - |
| Table 19: IoTDeviceRoleParameter Attributes and Associations .....         | - 57 - |
| Table 20: PhysicalEntityRole Attributes and Associations .....             | - 63 - |
| Table 21: PhysicalEntityRoleParameter Attributes and Associations .....    | - 63 - |
| Table 22: PhysicalEntity Attributes and Associations .....                 | - 63 - |
| Table 23: IoTTask Attributes and Associations .....                        | - 63 - |
| Table 24: Real-World Data Object / Store Attributes and Associations ..... | - 68 - |
| Table 25: Process Attributes and Associations .....                        | - 72 - |
| Table 26: IoTDeviceRole and PhysicalEntity Role Attribute .....            | - 74 - |
| Table 27: Activity Attribute .....   | - 76 - |
| Table 28: IAPMC Elements vs. Weak Points .....                             | - 82 - |



## 1. Introduction

### 1.1 Purpose of Deliverable

In this section we answer four key questions in order to describe the benefits and focus of this deliverable.

#### **The motivation - why do we need an IoT-aware process modelling concept (IAPMC)?**

Standard business process modelling notations provide graphical symbols that enable the specification and documentation of business processes and workflows. Some of the graphical notations offer execution semantics, as well for supporting the direct execution of specified process models through a process execution engine. So far, modelling notations are used for purely software-based environments such as specialized ERP systems. *The lowest level*, which represents the actual information in form of data and native interfaces, *is not considered* - since this is usually one single resource component (e.g. a SQL database). Thus, features brought through the individual devices and device types of this lowest level are neither included in current business process modelling approaches, nor in the business process execution process.

A major area of research is the Internet of Things (IoT). It envisions a world that is connected with each other by clearly-identifiable devices in a web-like structure. With the help of services the physical objects can interact with each other, so that they can become active participants of future business processes. The direct integration of real-world technology promises entirely new and changing business processes that currently cannot be completely represented and executed using existing process modelling techniques and notations. For the area of business process modelling, it comes to new requirements, which shall be covered by introducing the IAPMC within this deliverable.

Existing process modelling notations are not designed to specify aspects of real-world technologies. There is a clear demand from the industry side, to assist these new technologies in process modelling environments in order to ensure the seamless IoT integration into the enterprise software world. The new IAPMC facilitates to consider a variety of IoT specifics already during the design phase of a business process. Performing a standard modelling process with a modelling language including execution semantics does require more effort than the collection and documentation of processes in a simple textual, tabular or graphical form, but it is an initial condition and urgently needed for the automatic and semi-automated business process support within the IoT.

#### **The target group - who will use the IoT-aware process modelling concept?**

Basically, the developed concept can be transferred to different process modelling languages that are standardized. From this perspective it can be used in particular by developers of modelling tools aiming to provide the IAPMC in one of the many modelling language standards in a more comprehensive and client oriented Business Process Management (BPM) solution.

After performing a comprehensive evaluation we decided for this deliverable to realize the IAPMC in close conjunction with the promising standard Business Process Modelling and Notation 2.0 (BPMN 2.0). Developing the IAPMC in form of BPMN 2.0 extension, we focus on the typical BPMN business user as a target group. Following [20], the target group comprises:

- The process analyst, who produces an initial professional IoT-aware business process design.
- The process engineer, who implements the technically detailed IoT-aware business process.
- The process owner, who takes over the strategic responsibility for the IoT-aware business process.
- The process manager, who is operationally responsible for the IoT-aware business process.
- The process participant, who is involved in the execution responsibilities of the IoT-aware business process.

#### **The tool support - how can the IoT-aware process modelling concept be used?**

To demonstrate and evaluate the benefits of the IAPMC approach it is directly transferred to BPMN 2.0. As part of IoT-A Task 2.3 and D2.4 due to M30 (February 2013), the proposed IAPMC will be

implemented in BPMN 2.0 of a dedicated process modelling tool. Through the provided implementation the IAPMC can be used as a regular extension to BPMN 2.0.

### The IoT-A project - how does this deliverable contribute to IoT-A and the research world?

As part of the project IoT-A, this deliverable aims to increase the use of IoT technologies in the enterprise world. To lock the gap between the traditional world of business process systems and the imperfect IoT devices the fundamental IoT-aware concepts are integrated in the service-based business process modelling approach of the enterprise world. In particular, this deliverable relates to IoT-A research activities of WP1, WP6 and WP7: Based on the reference architecture of D1.2 [58] in this report the domain model, the information model and the communication model are transferred to the concepts of the standard business process modelling language BPMN 2.0. Achievements of WP6 allow applying the IoT-A requirements gathering and refinement methodology to this deliverable. The numerous use case and scenario definitions of IoT WP7 enable to explore typical IoT-aware business processes. Multiple parts of this deliverable were successfully published and contributed to scientific conferences of the IoT and BPM research world [c.f. References].

## 1.2 Methodology

In order to develop concepts for modelling IoT-aware processes, we follow the Design Science (DS) Research Methodology for Information Systems Research of [37]. DS has been successfully developed to produce artefacts in Information Systems. The research methodology offers a proven methodology to operate the DS bringing principles, practices and procedures. Thus, a process model for the actual procedure is offered that we apply in the context of this deliverable. The DS process consists of 6 general iterative steps: problem identification and motivation, definition of the solution properties, design and development, demonstration, evaluation and communication. While this deliverable is mainly concerned with 5 of 6 stages, D2.4 will focus on the coding and demonstration by an editor prototype of the developed artefacts due to M30. However, as it is shown in Figure 1, this is not strictly sequential, but an iterative process. The defined requirements, design principles, concepts and evaluations of this deliverable have not reached a final mature state, so that we will continue with the research work in the two closely interlinked projects IoT-A and FI-Ware<sup>1</sup>.

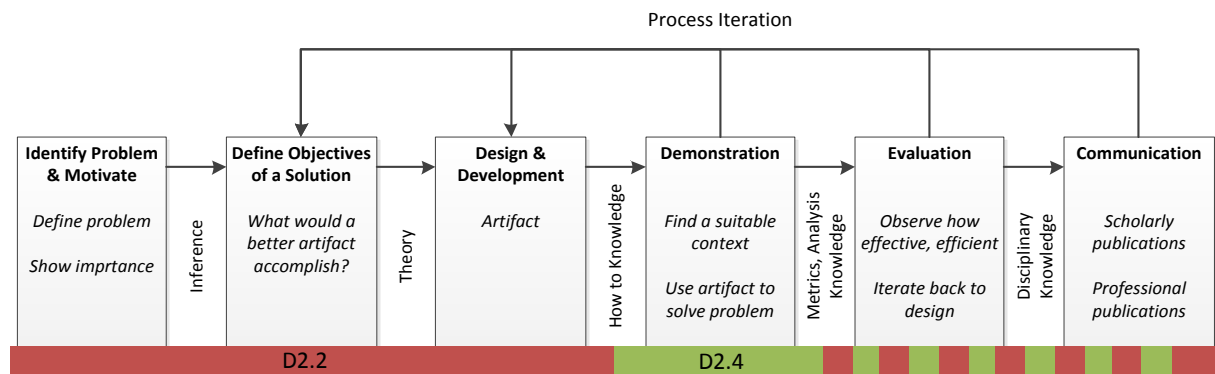


Figure 1: Applied DS Research Methodology Process Model of [37]

## 1.3 Document Structure

This deliverable is based on the IoT-A state-of-the-art report on existing integration frameworks / architectures for WSN, RFID and other emerging IoT related technologies, wherein we already presented shortly the main industry context of business process modelling. It is organized as follows: section 1 gives an introduction to the scope, general methodology and content of this deliverable. Section 2 covers a more detailed summary of current graphical and technical modelling approaches as well as related work. The requirements for the IAPMC and a primary evaluation of the modelling approaches against the requirements are covered by section 3. The design principles of the IAPMC are addressed by section 4 comprising a phase model and the BPMN 2.0 standard conventions. Section 5 presents the IAPMC by summarizing the eight main weak points for using current notations

<sup>1</sup> <http://www.fi-ware.eu/>

in the IoT domain, formalizing and designing seven IAPM elements and its BPMN 2.0 implementation and presenting the main ideas of the envisioned IAPMC evaluation. Finally, section 6 concludes the document and provides an outlook on future steps.

## 2. Existing Modelling Approaches

This chapter discusses the basic concepts of business process modelling and presents the first step of the applied DS research methodology (c.f. section 0). Therefore, the terms business process, business process management, business process modelling and execution are defined. A brief summary addressing the progress of the evolutionary development of business process modelling notations in recent years shows the development of various standards. The last section aims to present selected concepts of process modelling, which were applied in the context of the research work carried out for this IoT-A deliverable. Reviewing the literature represents an “essential first step and foundation when undertaking a research project” [11]. By an effective use of the existing knowledge base of the topic Business Process Modelling, reinvestigation can be avoided and the relevance of the research contribution is improved.

### 2.1 Main Terminology Summarized

**Business Process:** There are various definitions of the term business process that have evolved over time. The most popular definitions are from Davenport [16] and Hammer and Champy [23] of the year 1993. In this research report, we stick to the more recent definition by Weske of 2007, who defines the term business process in [59] building on several known characterizations as follows: „A business process consists of a set of activities that are performed in coordination in an organizational and technical environment. These activities jointly realize a business goal. Each business process is enacted by a single organization, but it may interact with business processes performed by other organizations.“

**Business Process Management (BPM):** Depending on the selected BPM methodology it is differentiated between different BPM activity phases in the BPM lifecycle, but within this work we will build on the definition provided by the European Association of BPM. They define Business Process Management (BPM) in [7] as „a systemic approach geared to capture, design, execute, document, measure, monitor and control automated as well as non-automated processes in order to meet the objectives that are aligned with the business strategy of a company. BPM embraces the conscious, comprehensive and increasingly technology-enabled definition, improvement, innovation and maintenance of end-to-end processes. Through this systemic and conscious management of processes, companies achieve better results in a faster and more flexible way.“ In Task 2.3 of WP2 we mostly focus on the lifecycle phases design, documentation and execution because they are considered to be a prerequisite for most of the further phases.

**Business Process Modelling:** The result of capturing a business process is gathered by a process model. This design activity of representing a business process by abstracting the business process in a model is termed Business Process Modelling. The process model can comprise a graphical or a non-graphical representation and is an initial step for progressing with one of the other BPM lifecycle phases. [59] defines a business process model as “a set of activity models and execution constraints between them”. In operational use, [20] distinguishes between two modes of modelling a business processes:

- **Business level:** The goal of the business level is the result-oriented and operational representation of the process. Initially, a rough understanding of the process flow is mediated, in which the process is considered in a few steps without using a process modelling language from bird's eye view. This first model is fundamental for the integration of operational details into the process. From the organizational process model a technical model is developed. To develop such a model a specific proceeding is needed, which depends on the used process modelling language.
- **Technical level:** The objective of the technical level is the implementation of the technical described details. To execute a technical process, a so-called process engine can be used. Some process modelling languages currently do not support the direct execution of the process model on a process engine. Therefore, it is often necessary to refine the technical process model. If no process engine is used, the process logic is transferred and implemented in a certain programming language.

The transformation between the business and the technical level brings several problems; nevertheless to generate the technical process model automatically from the business model is seen

as the ideal solution. Within this IoT-A deliverable we focus on extending the business and technical level of business process modelling approaches by IoT-specific characteristics.

**Business Process Execution:** Business Process Execution consists of the manual, semi-manual and automated execution of business processes by actors. An important precondition for the automated execution of business processes is the technical specification of the business process model by technical analysts. The relevant industry standards for describing an executable business process model are WS-BPEL and BPMN 2.0. Common IT-Systems supporting the execution of a huge number of automated processes often use a process execution engine as a central controlling component. This IoT-A report does not address the execution itself of the proposed IoT-specific characteristics in the business and technical process model, but provides a generally executable IoT-aware process representation. As a central component of the process execution, we share the view of [1] and assume a process engine that deploys and executes the technical process model. The precondition for a successful execution is that the process engine understands the introduced IoT specifics of the technical process models. It results that expansions in the technical process model implicate enhancements on process engine side.

## 2.2 History of Business Process Modelling

In the 90s, companies reinforced the data-oriented perspective by a process-oriented perspective, in order to record, streamline and improve their working processes. Analogously, the development of business process management systems to automate some of today's BPM phases progressed. Thus, a veritable uncontrolled growth of different process modelling notations started. Figure 2 of [10] presents an overview of the development of the five main strands of the last 10 years, in which most of the widely used notations can be grouped in.

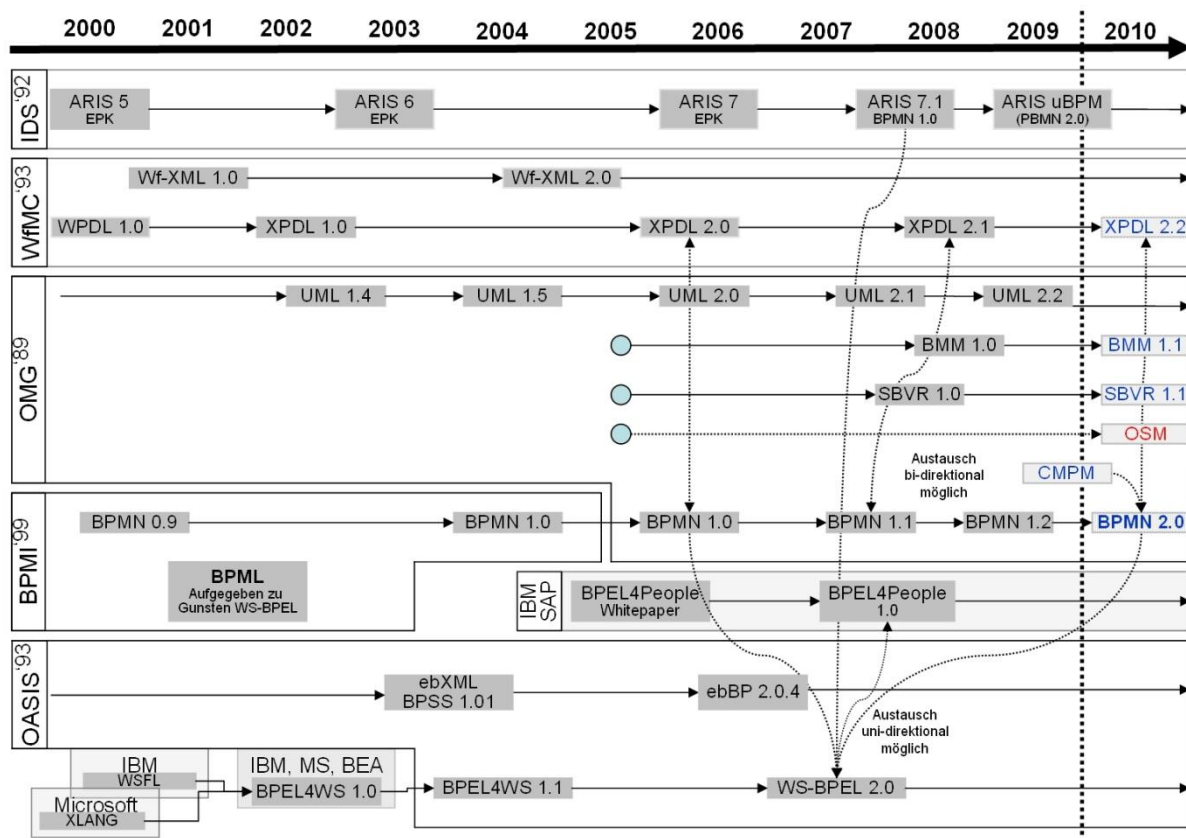


Figure 2: Development of the most popular process modelling notation [10]

Business processes can be modelled using different modelling languages. Many of these notations are driven and specified by large companies such as SAP, Software AG and IBM, or standardization organizations including OMG, WfMC and OASIS. Figure 2 highlights that there is no single standard for modelling business processes. But in their origination, all process modelling notations concentrate either on the graphical or on the execution level. Since both levels are linked more and more to one

another within recent years, the different performed intentions to improve the exchange between both levels can be grouped into three strands:

- Definition of mapping schemes between graphical and execution level (e.g. BPMN 1.2 / 2.0 and WS-BPEL 2.0)
- Addition of human interaction to machine readable notation (e.g. BPEL4People)
- Definition of execution semantics for graphical notation (e.g. BPMN 2.0).

## 2.3 Main Concepts

The notations presented in section 2.2 are grown in about 20 years and tailored to the modelling of business processes within companies. The direct integration of IoT technology into business processes can indeed result in entirely new or modified processes, but for the next years it is expected that most of the business processes of an enterprise can be preserved. This assumption leads to the approach taking a well-established business process modelling notation for expanding it with the ability to be IoT-aware. The following subsections provide an overview of the business process modelling notations with an industry focus, which are evaluated in the requirements analysis of section 3 of this deliverable. Table 1 presents a summary on available BPM notation efforts with industry relevance.

| Effort  | Latest Version | Graphical Representation | Machine-Readable Representation | Execution Semantics | Organization | Relevance |
|---------|----------------|--------------------------|---------------------------------|---------------------|--------------|-----------|
| BPMN    | 2.0            | Yes                      | Yes                             | Yes                 | OMG          | Industry  |
| EPC     | eEPC           | Yes                      | Yes                             | No                  | No, ARIS     | Industry  |
| UML     | 2.4            | Yes                      | Yes                             | Yes                 | OMG          | Industry  |
| WS-BPEL | 2.0            | No                       | Yes                             | Yes                 | OASIS        | Industry  |
| XPDL    | 2.1            | No                       | Yes                             | No                  | WfMC         | Industry  |
| ebBP    | 2.0.4          | No                       | Yes                             | No                  | OASIS        | Industry  |



Table 1: Overview of BPM efforts

### 2.3.1 Business Process Modelling and Notation


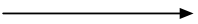
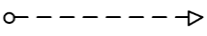

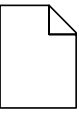


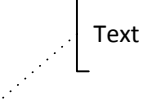
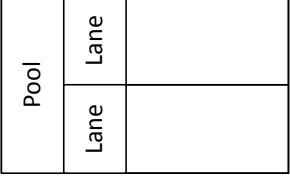
The Business Process Modelling Notation (BPMN) was developed by the Business Process Modelling Initiative (BPMI), which aimed to develop a graphical process notation for the Business Process Modelling Language (BPML). The first specification was published by Stephen White of IBM in 2004, while BPML was discontinued in favour of BPEL. Meanwhile, the Object Management Group (OMG) has adopted BPMN as a standard. By the standardization process BPMN became also interesting for companies. [9]

Until recently, the available version was 1.2 and the new version 2.0, which is based on version 1.1, has been long awaited. Finally, the renamed Business Model and Notation (BPMN) 2.0 was released in early 2011 [14].

Process diagrams that are created by using BPMN are called Business Process Diagrams (BPD). A BPD consists of BPMN basic modelling elements and extending modelling elements, with which many different diagram types can be expressed. Table 2 shows the BPMN basic elements of the five element categories Flow Object, Data, Connecting Object, Swim lanes and Artefacts. For a detailed BPMN overview we refer to [14].

| Element   | Name     | Category    | Explanation   |
|---|----------|-------------|---|
|  | Activity | Flow object | A work unit that a company performs in a process.       |
|  | Event    | Flow object | Something that happens during the course of a progress. |



|   |               |                   |   |
|---|---------------|-------------------|---|
|    | Gateway       | Flow object       | Points to control the divergence or convergence of a sequence flow.   |
|    | Sequence Flow | Connecting object | Flow to show order that activities will perform in a process.   |
|    | Message Flow  | Connecting object | Flow to show the message exchange between participants.   |
|    | Association   | Connecting object | Association used to link information and artefacts with BPMN elements. Arrow indicates direction of flow.   |
|    | Data object   | Artefacts         | Information provision about what a single activity requires to be performed or what they produce  |
|    | Message       | Artefacts         | Notation used to depict the contents of a communication between two participants  |
|   | Group         | Artefacts         | Tool to visually summarize elements of a category independently from the sequence flow  |
|  | Annotation    | Artefacts         | Mechanism for providing additional text information   |
|  | Pool/Lane     | Participant       | A pool is the graphical representation of a participant in collaboration. A Lane is a sub-portion in process to organize and categorize activities. |

**Table 2: Basic Elements of BPMN [14]**

Since BPMN 1.2 was a purely graphical process notation, many meta-models have been developed to convert it into an executable notation. However, the conversions are still limited by several problems, so that current transformations (i.e. BPMN to BPEL) could not be considered as a sufficient end result. The new version BPMN 2.0 promises to bridge the gap between the business and the technical level of process modelling by including defined execution semantics and an XML serialization format. Therewith, BPMN 2.0 is the first notation combining a technical specification of an executable process model with a graphical and end-user oriented process model [57]. In particular, the integrated execution semantics promise progress, since the efforts of recent years to translate BPMN to BPEL automatically can be omitted.

For the proposed research work possibly interesting features of BPMN 2.0 include the following points:

- Default extensibility mechanism.
- Detailed mapping schema of BPMN to BPEL allows a subset of processes described in BPMN to be directly executed by BPEL engines.
- Two new events escalation and parallel event could support IoT relevant aspects.
- Definition of global tasks that can be referenced through an activity call.
- Event-sub process allowing to run an entire sub-process in dependency on an event. If an event is passive, it is possible to define process-interrupting and non-process-interrupting events [20].

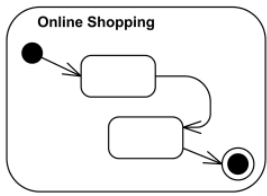
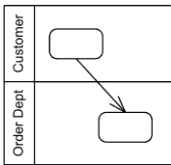
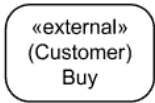

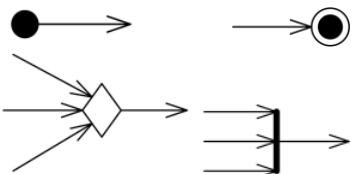
By now, there are several large IT companies such as SAP which support the new BPMN 2.0 standard on their modelling tools as well as the XML files for serialization and execution support [50].

BPMN is currently a very promising approach and a broad support from manufacture side is expected. Nonetheless, many execution engine manufactures are still at an early stage to ensure full BPMN 2.0 support. Beside the internal project efforts, no IoT related BPMN research work could be identified.

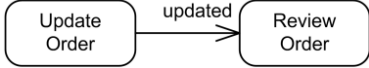
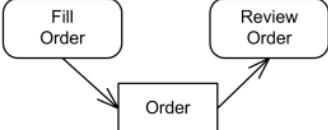
### 2.3.2 Unified Modelling Language

Unified Modelling Language (UML) is a graphical language visualizing, specifying, constructing, and documenting the artefacts of distributed object systems [51]. It is the most widely used standard for software architects to specify business applications. UML is primarily used for object oriented software development in the area of software engineering.

The UML was created during the 90s as a modelling language and methodology to support object-oriented programming. In 1997 it was taken over as a standard by the OMG. The first versions 1.X have been replaced in 2005 by the newly-revised versions 2.X. The current version is 2.4 and defines thirteen types of diagrams, divided into three categories: Six diagram types represent static application structures, three represent general types of behaviour and four represent different aspects of interactions. To illustrate the dynamic flow of business processes it is suitable to use the activity diagram of the behaviour diagrams. This deliverable focuses on the activity diagram which allows to model control and object flows of business processes. In the following, an overview of the available basic activity diagram elements is presented [18]:

| Element   | Name  | Explanation   |
|---|---|---|
|   | Activity                                    | An activity diagram is graphically modelled by one or more activities and is a parameterized behaviour represented by a coordinated flow of actions. Within the activity actions, objects and control nodes are connected with one another. |
|   | Activity Partition                          | An activity partition is an activity group for actions that have some common characteristic. In general two types of activity partitions are available.   |
|  | Activity Partition: Swim lanes              | An Activity partition may be shown by a swim lane notation using parallel lines, either horizontal or vertical, and a name labelling the partition in a box at one end.   |
|  | Activity Partition: Alternate text notation | If swim lanes can't be used to show partitions, alternate text notations with qualified action names could be used instead.   |
|  | Action                                      | Action is a named element which represents a single atomic step within an activity. The action types Object Action, Variable Action, Invocation Action, Structural Feature Action, Link Action and Event Action are distinguished.          |
|  | Control node                                | A Control node is an activity node used to coordinate the flows between other nodes. It includes initial node, flow final node, activity final node, decision node, merge node, fork node and join node.                                    |



|   |               |   |
|---|---------------|---|
|  | Activity Edge | An activity edge is an abstract class for the directed connections between activity nodes. It includes control edges and object flow edges. |
|  | Object        | An object node is an activity node that is part of defining object flow in an activity.   |

**Table 3: Basic Elements of UML Activity Diagram [18]**

Since 2002, there are first tools available offering mapping solutions to automatically generate BPEL, WSDL and XSD artefacts from the visual UML notation. Therefore, a so called UML profile describing the UML modelling context is used, in order to customize UML for this particular benefit. [21]

While several mapping proposals have been published for mapping older versions of the purely graphical BPMN diagrams to UML, no efforts could be found for transforming the activity diagrams of UML to BPMN 2.0 process models. The reason for this is probably that both notations are standards of the OMG and a future consolidation is likely.

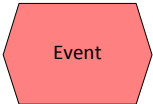
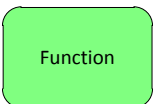

[15] presents a mapping approach between a physical object and a web service which completely describes the physical object in terms of WSDL metamodel extension. Therefore, the process consists of a transformation between UML models and XML syntax. This approach will be considered for the Physical Entity representation in the process model.


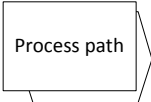



### 2.3.3 Event-driven Process Chain

The Event-driven Process Chain (EPC) is a modelling language for the graphical representation of business processes of organizations. As part of the Architecture of Integrated Information Systems (ARIS) concept, this method was developed as a holistic approach for view-oriented modelling of business processes. The EPC provides business processes in a semi-formal modelling language with syntax rules from an event-centric perspective. In principle, it is possible to express the control and information flow of a business process.

EPC was developed by August-Wilhelm-Scheer at the University of Saarland in line with a research project with SAP in 1992 for describing semi-formal business processes. The latest version extended Event-driven Process Chain (eEPC) extends EPC by introducing elements for the organization, data and performance modelling. Thus, associations to functions can be modelled as well. eEPC competes with other established process notations.

Since modelling languages mostly offer a service-centric view, EPC distinguishes in focusing the event as the fundamental concept of the notation and by its proximity to standard software systems (e.g. ERP-systems). It comes to problems with EPC with the identification of organisational changes and system breaks, the mapping of complex activities, as well as in the modelling of monitoring and control activities. The following table presents the main EPC elements:

| Element   | Name      | Explanation   |
|---|-----------|---|
|  | Event     | Events describe under what circumstances a function or a process works or which state a function or a process results in. A diagram starts with an event and ends with an event.  |
|  | Function  | Functions model the tasks or activities within the company. Functions describe transformations from an initial state to a resulting state.  |
|  | Connector | Logical operations within the control flow are described by logical connectors. It is possible to split the control flow from one flow to two or more flows and to synchronize the control flow from two or more flows to one flow. |

|   |                    |  |
|---|--------------------|--|
|  | Control Flow       | A control flow connects events with functions, process paths or logical connectors creating chronological sequence and logical interdependencies between them. |
|  | Process Path       | Process paths serve as navigation aid. They show the connection from one to another process.   |
|  | Information Object | Information Objects represent documents or other data storage.   |
|  | Organisation Unit  | Organization units determine which person or organization within the structure of an enterprise is responsible for a specific function.                        |
|  | Information Flow   | Information flows show the connection between functions and input or output data, upon which the function reads changes or writes.                             |

**Table 4: EPC Notation Main Elements**

The technical nature of business processes cannot be described by using EPC. To close this gap, a transformation into a more technical orientated view from the business users' requirements is necessary. For this purpose several mappings to WS-BPEL have been developed. [63], [32]

[40] presents a meta-model for complementing EPCs to present RFID based business processes. This work is highly EPC dependent. EPC varies from modelling approaches regarding the event centric view, so that an application on a general BPM level will be difficult.

### 2.3.4 Web Services Business Process Execution Language

The Web Services Business Process Execution Language (WS-BPEL, short form BPEL), is an XML-based language to describe business processes that are composed out of web services to be more powerful services. These services are usually orchestrated to processes.

BPEL was introduced in 2002 by a consortium of the IT companies BEA Systems, IBM and Microsoft. In 2003, a new version of BPEL was published and SAP and Siebel joined as well. Due to version 1.1 the standard was named Business Process Execution Language for Web-Services (BPEL4WS). Since the year 2008, BPEL is managed by OASIS and an established standard in the area of process execution languages. The current and most popular version 2.0 [26] defines basic activities and structured activities. There are also further versions available such as Oracle BPEL. Many major vendors of business solutions joined the technical committee of BPEL. A standardized graphical representation of BPEL is not available and differs from tool to tool, but the IBM and SAP initiative BPEL4People [31] aimed to provide human interaction for BPEL.

BPEL behaves like a programming language that is processed block-oriented and brings a rigid control structure. This is an advantage for the process engine and supports various aspects such as error or deadlock detection. For composing web services, BPEL builds directly on top of Web Service Description Language (WSDL) and can use one or more WSDL services. A BPEL web service composition glues composed services together into one process model and provides the means to specify such a process model.

BPEL 2.0 comprises different elements out of three groups, listed in the table below. Thereby, basic and structured activities are distinguished. While basic activities remain atomic activities not build out of other activities, structured activities contain other activities.

| Element       | Group               | Explanation  |
|---------------|---------------------|--|
| Assign        | Basic Activity      | Change of the content of a variable                                  |
| Invoke        | Basic Activity      | Synchrony or asynchrony call of a web service                        |
| Receive/Reply | Basic Activity      | Offering of a synchrony or asynchrony web service interface          |
| Throw         | Basic Activity      | Explicit signal of an error  |
| Wait          | Basic Activity      | Wait until a point of time or for a time period                      |
| Empty         | Basic Activity      | Doing nothing  |
| Sequence      | Structured Activity | Sequentially execution of activities                                 |
| While         | Structured Activity | Execution of an activity as long as a Boolean condition is fulfilled |
| Switch        | Structured Activity | Conditional execution of an activity                                 |
| Flow          | Structured Activity | Parallel or sequence of any order                                    |
| Pick          | Structured Activity | Non-deterministic choice of external events                          |
| Scope         | Scope               | Group of activities to a transactional unit                          |

**Table 5: WS-BPEL Language Constructs**

Nevertheless, the block structure of BPEL heavily differs from the graph based BPMN. Thus, the past years have produced many research activities such as [60], [36] that aim to automate the mapping of the standardized professional business process models (BPMN) into the standardized technical models (BPEL). The forthcoming BPMN version 2.0 promises to bridge the gap between the notation and the real execution of a business process by defining own execution semantics and by specifying BPEL execution conformance, therefore rendering WS-BPEL is obsolete.

[17], [22] present approaches for applying BPEL to execute IoT-aware logistics business processes, which all focus on transferring business process parts to sensors or sensor networks. Therefore, [19] proposes to extend the BPEL activity design, which more focuses in the direction of the targets of this deliverable.

### 2.3.5 XML Process Definition Language

XPDL stands for XML Process Definition Language and is an XML based graphical orientated, machine readable process notation that is also executable. The today's main aim of XPDL is to provide a file format that supports every aspect of the BPMN process definition notation. This includes the graphical descriptions of the diagram, as well as the executable properties used at run time. With XPDL, a product can write out a process definition with full reliability, and another product can read it and reproduce the same diagram that was sent. Until 2011, XPDL was used by 80 different products in order to exchange process definitions. [62] Many organizations don't take one tool by one vendor, but instead different process tools to accomplish specific process oriented tasks, such as simulation and optimization. While BPMN is a visual process notation standard, XPDL focuses on storing and interchanging process definitions.

The history of the Workflow Management Coalition (WfMC) started already in 1993 when defining a Workflow Reference Model. After coming up with a first Workflow Process Definition Language (WPD), in 1998 the first XML based and updated process definition expression language known as XPDL 1.0 appeared. XPDL was improved over the years and got more and more to be an industry standard for process interchanges. Since 2005 the third revision known as XPDL 2.0 supports completely the version BPMN 1.2. In 2008, the WfMC Membership Committee approved the latest

version 2.1 of XPD. This release includes new functionality to update the BPMN to version 1.1 where the BPMN version 2.0 is based on. Before the introduction of BPMN 2.0, XPD was used as a standard exchange for BPMN 1.2 diagrams. Due to the introduction of an own XML-based exchange format we assume that the importance of XPD will decrease continually. BPEL and XPD are entirely different complimentary standards, which can be merged partially through BPMN 2.0.

XPD is generally extensible. It allows each different tool to store implementation specific information within the XPD. This could comprise IoT specific process information like the reference XML schema for process-centred definitions of RFID solutions introduced by [30], but should be more seen as a possible exchange enabler for the IoT specific process information.

### 2.3.6 eBusiness Business Process

eBusiness Business Process (ebBP), or its formal title eBusiness Extensible Markup Language (ebXML) Business Process Specification Schema, is a technical specification for business processes managed by OASIS and part of the ebXML framework specification. It defines a choreography language targeting the support of business systems in executing collaborating parties from a global point of view and to monitor business expectations of the business quality of service contract. In particular it provides:

- Standard and extensible business transaction patterns
- Support for modular definitions to complex nested activities
- Support for use of web service, hybrid and ebXML assets
- Semantic tailoring for business processes and business document

The latest and approved standard is version 2.0.4. ebBP can be used in a complementary way with other process standards such as WS-BPEL or BPMN. The benefit in association with BPMN lies in particularly in the introduction of so-called business signals that are not included in the recent BPMN 2.0 specification. This is a business runtime transaction to ensure state alignment between two parties. Since in the IoT signals are mainly a part of a lower level between performing devices, but this aspect is not covered by ebBP, we desist of a use at this point of time.

## 2.4 Further Approaches Summarized

Beside the most commonly used industry notation, further methods have acquired reputations, which are summarized below:

One of the earliest examples are Petri nets, which are mostly used in the academic area. Petri Nets were originally developed based on finite automation for modelling discrete, typically distributed systems. The benefits for business process modelling comprise, that Petri nets are formal, have associated analysis techniques, and are state-based rather than event-based. [55] Many modelling techniques such as activity diagrams of UML 2 and BPMN have adopted principles of Petri nets. One very promising approach are coloured Petri nets being highly recognized by research activities, but still without major industrial importance. [54] Since Petri Nets had some deficiencies in terms of control flow dependencies, Yet Another Workflow Language (YAWL) was developed on the basis of coloured Petri Nets specifying the formal semantics as a transition system. [53] In BPM practice, the minimalistic approach of YAWL comes with weaknesses to deal with cancelation, synchronization of only active branches and the concurrent execution of multiple instances. [24]

A further known standard is Web Services Choreography Description Language (WS-CDL), an XML-based non-executable language by the World Wide Web Consortium which describes peer-to-peer collaborations of parties by defining, from a global viewpoint, their common and complementary observable behaviour. [29] In an industry context, it is mainly used in addition to WS-BPEL as it is based on a block structure-based approach. It is expected that most of the features are subsumed by the beta version of BPMN 2.0, but however WS-CDL couldn't be seen as an all-embracing notation. It is an additional fragment on execution side for expressing peer-to-peer collaborations between the participating services across the trusted organization domains.

## 2.5 Related Research Initiatives

This section focuses on related approaches from the general research world and European projects. There are two related active research activities and one related research activity that has been completed. Between the involved project participants the following SAP internal exchange could be achieved until M18:

- **MakeSense:** Joint workshop and coordination meeting in Karlsruhe during first quarter of 2011. Considerations whether the project objectives can be combined. As a result, MakeSense focuses on wireless sensor networks, while IoT-A works on general IoT domain-wide concepts, so that the IoT-aware business process modelling is brought forward on different BPM levels.
- **ADiWa:** Joint calls with the research group from Dresden to integrate existing solutions in the conceptual proposal of this deliverable. As a result it was argued that no specific IoT concepts have been explored, but block-based extensions for dynamic IoT based eventing. The results don't satisfy the requirements of a European project.
- **CoBIs:** IoT-A is a follow up project to CoBIs, so that it can be built on the obtained results.

### 3. Requirements Analysis for the IAPMC

This chapter discusses the requirements analysis by listing all requirements and the results of the previous section in suggesting a most IoT-aware Business Process Notation as well as an editor which fits most for our extension purposes. The proceeding of this chapter represents the second step of the applied DS research methodology (c.f. section 0).

The goal of WP2 T2.3 is to provide design principles, a language and tool specifications, which allow modelling IoT-aware business processes in an enterprise software environment. This results in the goal of this deliverable to design and implement the IAPMC using a selected business process notation. The task internal requirements definition is integrated into the IoT-A cross-WP requirement gathering process. As part of WP2 it has a close relationship to the IoT-A project activities of WP1, WP6 and WP7.

During the first project year WP6 approached WP2 in order to specify the WP2 internal requirements. A requirements guide was provided including a methodical recommendation. As part of this task, T 2.3 defined three different categories of requirements. First, general requirements for process notations were created, independently of the IoT area, but including aspects that ensure the future integration of the process modelling concept in the enterprise world. Second, IoT specific aspects of business process modelling requirements were defined to elaborate the differences between IoT and non-IoT aware business processes. Building on the first two categories, a third group of requirements was defined. It does not contain requirements for the process modelling approach, but the requirements for the project task that continues to develop a process modelling tool. This task will implement the prototype of the IoT-specific modelling concept due to M30. After defining the requirements, they were passed in a first version to WP6. In the course of an iterative quality assurance process they several times revised. A later version was revised after feedback from WP1.

As part of the implementation of the use case prototype WP2/WP7 “Quality-based pricing of sensitive goods”, the considered business process was modelled in different ways. Analysing this process models lead to the creation of the early set of IoT specific requirements. Based on this initial WP2 results, BPMN 2.0 was identified to be the most IoT-aware business process modelling notation and was applied in D7.1 by all project partners as a modelling notation to describe the processes of their IoT-A use cases. Using these numerous IoT business processes WP2 validated and revised the early set of IoT specific requirements.

#### 3.1 IoT-Specific Characteristics

The subsections of this chapter provide the requirements of the three defined categories:

- General process notation aspects
- IoT specific aspects
- Process editor aspects

##### 3.1.1 General Process Notation Aspects

As presented in Chapter 2, there is a wide variety of process modelling notations available. Not all available notations are basically usable for our purposes neither they allow the expansion intentions in this research project. Independently of the IoT area, in this category we define requirements for limiting the amount of suitable notations and to deal with a restricted pre-selection. In the requirements definition we focus on the integration of the modelling approach into the existing business community, so that the new IoT specific concept smoothly comes as an additional complement to the conventional and established business process modelling approaches. Since these requirements can be isolated from the requirements for the actual IoT-specific extension and the tool-specific implementations, they are listed in a separate and initial category.



| ID     | Type                       | Priority | Description   | Rationale  | Owner | Fit Criterion   |
|--------|----------------------------|----------|---|--|-------|---|
| WP2.11 | Functional Requirement     | High     | The process modelling notation has to be extensible in terms of the definition of new stencils, the specification of new syntax, the definition of serialisation and execution semantics. | The reuse of an existing process modelling notation allows focusing the effort on the IoT-extension. | SAP   | The notation allows extensions by default or the notation was already extended in the past.   |
| WP2.12 | Functional Requirement     | Medium   | The process modelling notation has to be executable.  | The project tasks 2.2 and 2.3 should closely work together and represent a hand in hand solution.    | SAP   | Execution Semantics for the artefacts of the modelling notation are defined.                  |
| WP2.13 | Non-Functional Requirement | High     | The process modelling notation has to be IoT-aware.   | Due to the DOW the project focuses on IoT processes.   | SAP   | See aspects defined in [3].   |
| WP2.14 | Functional Requirement     | Medium   | The process modelling notation has to offer a graphical representation.   | A graphical process notation offers a symbolism to easily model and document business processes.     | SAP   | A symbolism is available for the notation.  |
| WP2.15 | Non-Functional Requirement | High     | The process modelling notation has to be a standard.  | A common standard maximizes the potential application of industrial stakeholders.                    | SAP   | The standard implementation is published and administrated by the corresponding organisation. |

**Table 6: Requirements List of General Process Notation Aspects**

### 3.1.2 IoT specific Aspects

Current approaches focus on modelling and executing planned processes in a constant enterprise environment. In contrast, standard interfaces of IoT technology enable flexibly implementing business processes and quickly reacting with adapted processes to newly appearing requirements. This includes the fast integration of real-world technologies into the existing business environment. One

requisite for integrating smart items like sensors and actuators into business processes is to represent the IoT specific properties of this novel technology in the graphical and technical process notation. In order to highlight clearly the difference between current and future business processes and to analyse the IoT-awareness of standard business process notations, a number of IoT specific process properties is presented in Table 7. These IoT properties were obtained by implementing a survey with IoT and BPM experts [35] as part of the IoT-A WP2 Workshop in M5.

| ID     | Type                   | Priority | Description   | Rationale  | Owner | Fit Criterion   |
|--------|------------------------|----------|---|--|-------|---|
| WP2.16 | Functional Requirement | High     | The BPMN extension must support an entity based approach defined by the domain model of WP1.                    | The domain model is one key result by WP1 and should fit to the business modelling approach of WP2.  | SAP   | All relevant domain model concepts are reflected by the process modelling approach.                 |
| WP2.17 | Functional Requirement | High     | The BPMN extension must support the process execution distributed over several devices.                         | In the IoT the execution of process steps can be distributed over several devices.   | SAP   | An example process can be executed over more than one agent based system including several devices. |
| WP2.18 | Functional Requirement | High     | The BPMN extension must support the modelling of different IoT specific interaction types.                      | The interaction between different devices, the integration of EoI information, and the interaction between Services characterizes the IoT. | SAP   | IoT specific interaction types are definable.   |
| WP2.19 | Functional Requirement | Low      | The BPMN extension must support to arrange data distribution over several data storages (resources) of devices. | Business Processes in the IoT distribute data objects in resources of many devices.  | SAP   | For each data object and data storage the resource is definable.                                    |



|        |                        |        |  |   |     |  |
|--------|------------------------|--------|--|---|-----|--|
| WP2.20 | Functional Requirement | Low    | The BPMN extension must provide means to scalable model and execute processes independently of the number of involvement process components. | In IoT processes multiple Eols, devices, resources and services can appear which could negatively affect the performance of the execution.  | SAP | For each process model indicators are available, that allow to predict the scalability of the process. |
| WP2.21 | Functional Requirement | Low    | The BPMN extension must support the abstraction of individual process components.  | In the IoT multiple devices, resources and services can appear. The accuracy and availability of accumulated data can be of much higher importance for the process than the data of individual components. The extension shall provide abstractive individual process components. | SAP | Individual process components are abstrahable.   |
| WP2.22 | Functional Requirement | Medium | The BPMN extension must support means to express the availability of a process component.  | Eols, devices and its services and data often have a mobile nature. Due to that, a business process can have a different availability depending on its involved components.   | SAP | An indicator of the availability of individual process components is available.                        |

|        |                        |        |   |  |     |   |
|--------|------------------------|--------|---|--|-----|---|
| WP2.23 | Functional Requirement | Medium | The BPMN extension must provide means to express the tolerable error rate of a process.     | Depending on the process, a process result is still acceptable as far as it stays under a tolerable error rate.  | SAP | Defective business processes can be modelled and executed (not exceeding a certain error threshold) |
| WP2.24 | Functional Requirement | Medium | The BPMN extension must provide means for designing context-aware business processes.       | Depending on occurring events, the IoT processes need to be highly flexible.                                     | SAP | Several event types are representable using the BPMN extension.                                     |
| WP2.25 | Functional Requirement | Low    | The BPMN extension must provide means for expressing the uncertainty of process components. | The uncertainty of individual process components can influence the process creation on model and execution time. | SAP | The uncertainty of different process components can be indicated.                                   |
| WP2.26 | Functional Requirement | High   | The BPMN extension must provide means for expressing real-time constraints.                 | As the process interacts with Eols of the real world, real-time constraints apply to these processes.            | SAP | Different real-time constraints can be expressed.   |

**Table 7: Requirements List of IoT specific Aspects**

### 3.1.3 Process Editor Aspects

In the further course of action in Task 2.3, the IoT-aware process modelling concept will be implemented within a modelling environment and an execution environment for business processes. As a foundation for this implementation, an existing business process modelling tool and an existing execution can be used in order to simplify the development effort. Therefore the existing tool, which is supposed to serve as a basis for this implementation, must meet certain requirements. The requirements are listed in a separate table shown below, because they don't belong to the requirements for the process notation or to the requirements for the IoT-aware process modelling concepts presented in the previous tables. The requirements list contains aspects from the user perspective as well as aspects from the development perspective.

| ID    | Type                       | Priority | Description  | Rationale  | Owner | Fit Criterion  |
|-------|----------------------------|----------|--|--|-------|--|
| WP2.1 | Functional Requirement     | High     | The process editor must be able to create BPMN 2.0.                        | BPMN 2.0 was evaluated to be the most IoT-aware process notation.                              | SAP   | A BPMN 2.0 file is created by the editor.  |
| WP2.2 | Functional Requirement     | High     | The process editor must be extendable.                                     | The reuse of a comprehensive tool allows focusing the effort.                                  | SAP   | New capabilities can be added to the process editor.   |
| WP2.3 | Functional Requirement     | Medium   | The process editor must provide facilities to model on business level.     | A business user is not able to specify an executable process model.                            | SAP   | The editor provides a special business view on the process, which excludes some execution details.           |
| WP2.4 | Functional Requirement     | Medium   | The process editor must provide facilities to model on technical level.    | A technical user is not able to specify the business frame of processes.                       | SAP   | The editor provides a special technical view on the process, which enables to specify all execution details. |
| WP2.5 | Non-Functional Requirement | Low      | The process editor has to be end-user-friendly.                            | A business user needs to be able to model a process.   | SAP   | A person with no process execution background is able to model a process.                                    |
| WP2.6 | Functional Requirement     | Low      | The process editor must be able to verify the syntax of the process model. | The technical user needs information about the correctness of the syntax before the execution. | SAP   | The editor provides a syntax checking functionality.   |
| WP2.7 | Non-Functional Requirement | Medium   | The process editor must be "easily and fast" extendable.                   | First project results should be presentable in a small time frame.                             | SAP   | Small effort needed for the implementation of a new stencil.   |

|        |                            |        |   |   |     |   |
|--------|----------------------------|--------|---|---|-----|---|
| WP2.8  | Non-Functional Requirement | Medium | The process editor has to provide an attractive graphical user interface.               | The project results need to be representable in a research review.          | SAP | A user looking at the editor for the first time must say: "Wow, that's cool!"                 |
| WP2.9  | Functional Requirement     | Medium | The process editor must be interoperable with developments of other WPs and Tasks.      | The project results should be combinable to reach the common project goals. | SAP | The process editor uses the interfaces, commonly defined with the other WPs, where necessary. |
| WP2.10 | Design Constraint          | Medium | The process editor must support BPMN 2.0 completely (in particular the IoT-aware parts) | The development effort should focus on the BPMN IoT extension.              | SAP | An IoT-aware sample process is completely representable.                                      |

**Table 8: Requirements List of Process Editor Aspects**

### 3.2 Evaluation of Existing Modelling Concepts

This section aims to analyse and evaluate which of the previously defined IoT aspects can be modelled by using existing business process modelling approaches. Therefore, we examine stepwise how the latest versions of the three standard notations BPMN 2.0, eEPC 2.0 and the activity diagram of UML 2.3 cover each of the introduced IoT requirements of section 3.1.2. In the following we will leave out the version information of the notation in order to improve the readability of this section. Finally, a table indicating the coverage of IoT aspects by the standard business process modelling notations provides an overview of the obtained results.

**Entity-based concept:** The terminology of the IoT-world is highly focused on modelling processes in an enterprise context. In this environment the physical entity where a particular activity in form of a service performs on is so far less important. All considered notations allow to model actors or roles, but except BPMN and UML where the physical entity can be annotated to an activity or modelled using lanes, there is no possibility to specify a physical entity within a business process. In none of the notations the physical entity, device and resource are distinguished so far in the sense of IoT.

**Distributed execution:** BPMN and UML offer the ability to annotate an activity with the name or identification of a physical entity or a device, while this option is entirely missing in eEPC. Since further information would be needed for the distributed execution, such as the activity shall be executed on any available physical entity of a certain entity type (e.g. all fridges of one manufacturer), the classic capabilities provided through the modelling notation are considered insufficient.

**Interactions:** BPMN, UML and eEPC provide the possibility to specify the process flow and the data flow of a business process. So far, different types of interaction are less considered on process modelling level by all standard notations. The aspects response time, uncertainty of information or error handling of different types of interactions is not supported in current notations. (E.g. distinguishing between interacting with a database or a RFID gate)

**Distributed data:** All considered notations allow the modelling of data objects. It is possible to specify a separate data flow from/to a data object to/from a specific activity. Additionally in UML and BPMN, it is possible to annotate the data object with the name of the resource containing the data object in the appropriate lane. A business process lane could contain in this sense all activities corresponding to a certain resource. eEPC does not include this information.

**Scalability:** To assess the scalability of a business process, information about the involved physical entities and its devices, resources and services is needed. Since eEPC exclusively offers to provide the information of an organization unit to a certain activity or data object, it is unclear how many physical entities, devices, resources and services are involved in a business process. Thus, the scalability cannot be evaluated. BPMN and UML offer the opportunity to symbolize different physical entities and devices using swim lanes as separated actors. Using this symbolism in a structured way, different entities and devices could be represented. In order to evaluate the scalability of a business process, further information about the involved entity or device needs to be provided during the modelling phase. Otherwise, by abstracting devices and resources on a higher level, it is uncertain how many instances will be involved during the final execution of the business process during execution time.

**Abstraction:** The notations BPMN and UML offer concepts to abstract activities to higher-value activities. BPMN provides two concepts: First, depending on the process flow parts of the process can be grouped to sub processes. Second, a selection of activities independently from the process flow can be grouped content wise. UML permits grouping process elements not depending on process or data flow as well as indicating an activity as a sub process. eEPC contains a process path symbol in order to link to other processes.

**Availability / Mobility:** All notations consider the modelling of a business process before the real execution. Modelled activities can unexpectedly become unavailable by the mobility of the devices at execution time. Conversely, devices that were not available at modelling time might become available at run time. The first of these two cases can be modelled with all notations except eEPC, but there is so far no option to mark the activity that it could become unavailable during a short period of time. To react to this particular mobile behaviour at run time, additional information about the mobile entity and devices would be helpful at modelling time such as marking a device as mobile. This mobility and availability aspects are currently not considered by any of the present notations.

**Fault tolerance:** The aspect fault tolerance lies in contradiction to the targets of current business process modelling notations, since a business process needs to be highly reliable. In BPMN, there is the possibility to combine process activities by using the process flow or the message flow. A process that extends over several entities, devices and resources can be represented in the BPMN process model by using one pool and several lanes (process flow between activities of different devices) or by using several pools (message flow between activities of different devices). The message flow allows a more fault tolerant flow than the process flow, as the message transfer will not affect activities of the previous device. UML offers to model bulk processing, which could also be considered to improve fault information. (E.g. to represent the repeated reading of a RFID-Gates) The eEPC concept expects that an activity follows sequent to an event. This results that a previously carried out activity is directly verified. This success mechanism complicates modelling a fault tolerant process.

**Flexibility / Event-based:** All notations are designed to completely model all variants of the process flow before the process is executed. None of the notations offers tools for flexibly changing the process during run time. A process may vary depending on activities and events occurring during the execution and thus follow a certain flow of the possible modelled process and data flow. Using BPMN, different types of events can be represented (e.g. message and time events). Depending on an event, a process flow can be started or influenced and a process can trigger a new event, whereby further processes can be started. UML does not distinguish between different event types, but also offers the ability to model events. eEPC allows a changing sequence of events and activities. Processes can also be triggered by events, but events are also used for the verification after a branch. Thereby, it is not possible to distinguish between technical events that were triggered by a sensor and functional events that serve to validate the executed activity.

**Uncertainty of information:** All considered notations provide the opportunity to model branches, and thus to verify the correctness of individual information by using error detection mechanisms or to start an error handling. However, the information accuracy strongly differs according to the respective device or resource. For example, the probability of obtaining error-free information is much higher if the information comes from a secure billing system, as if the information comes from a mobile sensor, which is influenced by frequent signal failures. Depending on the description of the information source, the business process could be modelled differently. Currently, none of the considered notations offers to model a more accurate description of the technical source of services and data objects.

**Real-time:** The languages UML and eEPC do not foresee to model time-based restrictions. With the help of different time-based events (start event, intermediate event, end event) BPMN enables taking into account already in the modelling some real-time restrictions of IoT-aware business processes.

Wrapping up, the graphical representation is quite similar for all approaches if we consider the process models of all notations - except BPEL, which is XML-based and focused on the technical process execution using a process engine. The most extensive graphical notations are BPMN and the activity diagram of UML that provide a variety of graphical elements to describe processes. Compared to eEPC, BPMN and UML can already represent parts of the entity-based approach and the distributed execution. None of the notations consider different types of interactions on the modelling level (besides differentiating between process and data flow). The IoT-aspects distributed data and scalability can be partly considered using BPMN or UML, while eEPC does not support the representation of these properties. In comparison to the other approaches, BPMN offers several instruments for representing abstraction aspects focused on environments designed by service-oriented architecture principles. The aspects mobility and availability are not taken into account by the investigated modelling languages. Fault tolerant behaviour of a business process is partially and exclusively covered by BPMN and UML by supporting to distinguish between data and process flows. Compared with the other approaches, BPMN already offers a comprehensive set of many events types and thus provides the most flexible notation for IoT-aware business processes. Currently, the aspect of uncertain information is not covered by any language. Of the considered approaches, only BPMN covers the modelling of real-time restrictions.

| IoT characteristics        | BPMN 2.0 | eEPC   | UML 2.3 |
|----------------------------|----------|--------|---------|
| Entity-based concept       | Partly   | No     | Partly  |
| Distributed execution      | Partly   | No     | Partly  |
| Interactions               | No       | No     | No      |
| Distributed data           | Partly   | No     | Partly  |
| Scalability                | Partly   | No     | Partly  |
| Abstraction                | Yes      | No     | Partly  |
| Availability / Mobility    | No       | No     | No      |
| Fault tolerance            | Partly   | No     | Partly  |
| Flexibility / Event based  | Yes      | Partly | Partly  |
| Uncertainty of information | No       | No     | No      |
| Real-time                  | Yes      | No     | No      |

**Table 9: Coverage of IoT Characteristics by BPM Notations**

Concluding, in order to describe business processes that include IoT-technology and to match the IoT domain model, BPMN 2.0 is the most suitable state of the art approach and provides coverage for more IoT specific properties than other approaches. Table 9 provides a final overview of the stepwise analysis of this section. In addition to the listed IoT-focused analysis, the newest BPMN version offers the following further advantages, which are significant for the upcoming research work in the IoT-A WP2:

- First end-user friendly modelling notation that is executable by defining execution semantics.
- General extensibility for defining new artefacts, which are based on the IoT-aware modelling conventions.

### 3.3 Evaluation of Existing Modelling Tools

BPM tools for process automation are increasingly spreading. In general, a trend can be seen that widely used simple graphically oriented tools (e.g. Visio) are replaced by extensive business process modelling tools. In Europe this trend concerning companies is larger than in America. [61] In contrast



to that trend, we highly focus on selecting an open source modelling tool without taking into account comprehensive commercial tools. In order to facilitate the development of a holistic tool support for IoT-aware business process modelling and execution, we propose to use a fitting pair of modelling tool and execution engine as a foundation for the further course of work in WP2. In D2.3 [34] a first pre-selection of business process execution engines for BPMN2.0 is performed. This pre-selection follows a blog post from Bruce Silver on executable BPMN 2.0, in which he states that the open source tools Activiti, Bonita Open Solution, and to some extent jBPM are ahead in implementing the execution of BPMN 2.0 process models [44]. Therefore D2.3 concentrates on these tools and in chapter 3.1.2 the history, general structure, and, to some extent, the internal functioning of the execution engines of the tools is presented. In the following we describe their modelling capabilities.

### 3.3.1 Activiti

The Activiti software package [8] provides two different components for the modelling of business processes. The Activiti Modeler is a web application, while the Activiti Designer is a dedicated plugin for the Eclipse IDE [47].

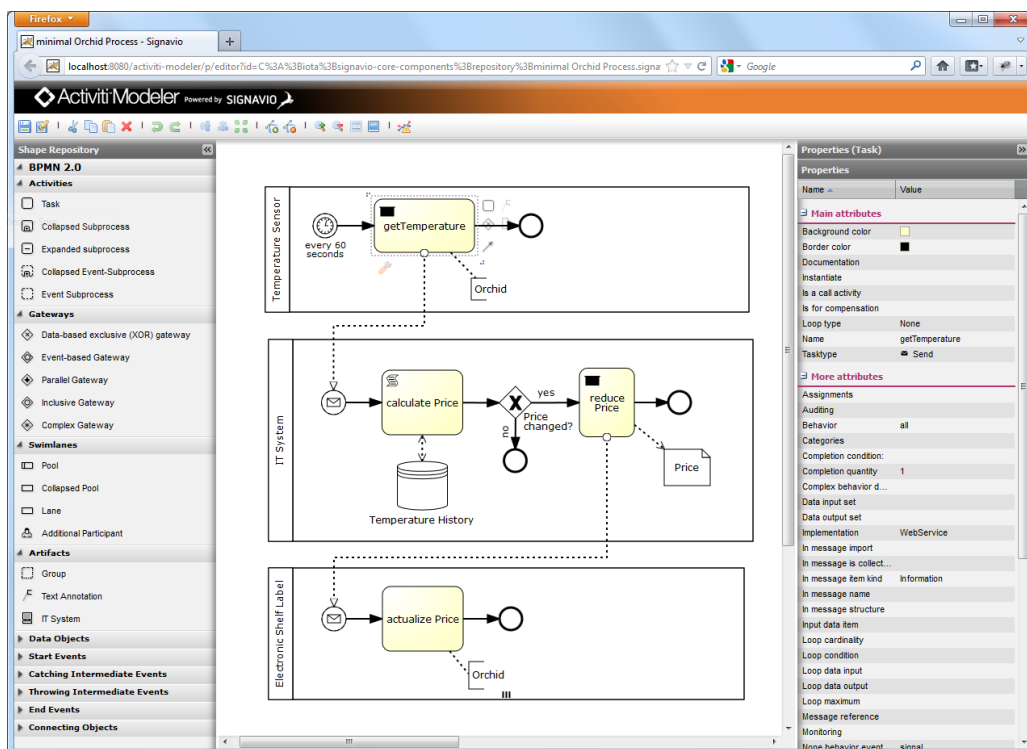


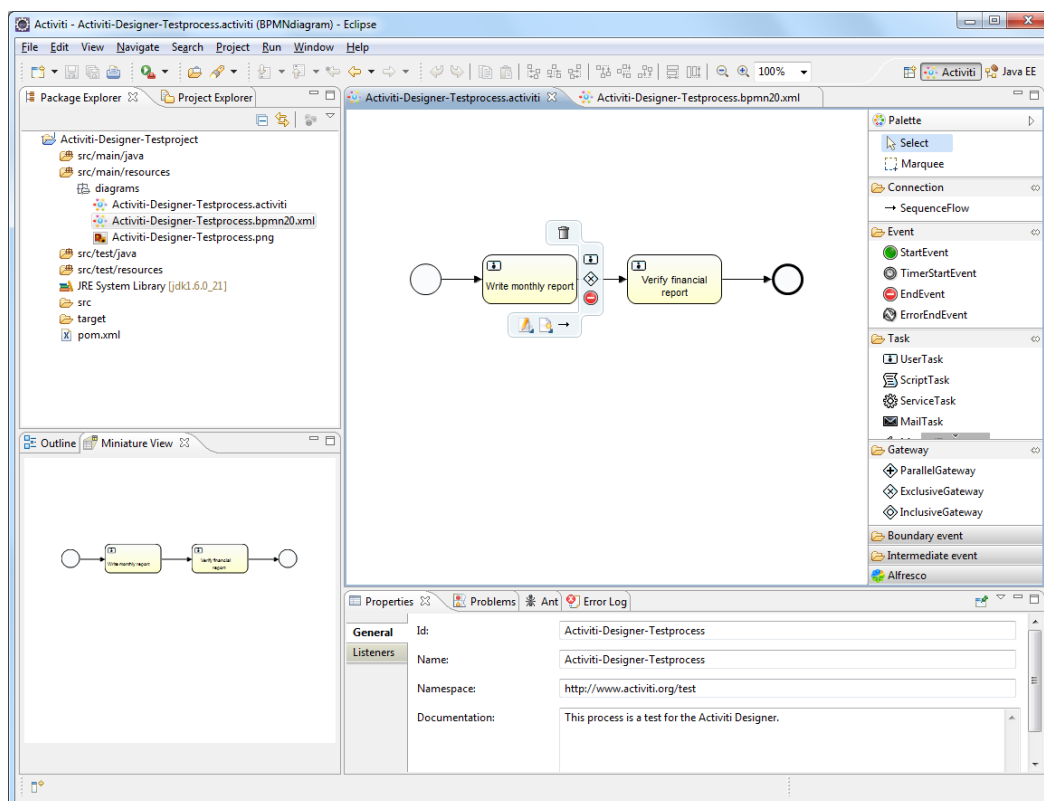
Figure 3: Activiti Modeler

The Activiti Modeler is a customised version of the Signavio Process Editor [42], and since Activiti 5.7 it is moved to a separate project, which is called Signavio Core Components [41]. The web-based user interface for the modelling of Collaboration diagrams is shown in Figure 3. The Modeler does not support Conversation diagrams or Choreography diagrams. On the left side of the UI the palette with the complete BPMN 2.0 stencil set for Collaboration diagrams is displayed. The canvas, on which the user can draw the process model, is arranged right beside to the palette. The properties of the selected model element are presented on the right side of the UI. These properties are subdivided into the main attributes in the upper part and more detailed attributes in the lower part. If a model element is selected in the model, some context sensitive buttons are displayed right next to it and below it. The buttons to the right show the model elements, which could be connected to the selected element, and therefore facilitate a fast way of process modelling without selecting every stencil newly from the palette. The button below the selected element provides the possibility to change the type of the selected element without using the properties section on the right. The bar above the three described areas provides general editing functions and syntax checking. Functionality to export the serialisation of the model to the client is not provided.

On the server-side the Activiti Modeler consists of a web application, which is packaged into a WAR file and can be deployed to an Apache Tomcat Server [45] or a JBoss Application Server [27]. This

web application stores the actual process models in a directory on the server's file system, which is specified as build parameter. For every model, which is stored in this repository, two files are generated: The first one contains the serialisation of the model in the BPMN 2.0 XML format including the Diagram Interchange information. Beyond the elements, which are defined in the BPMN 2.0 XSD, this serialisation contains additional metadata defined by Signavio, e.g. the background colour of diagram elements like events and tasks. The second file is the serialisation of a proprietary oryxmodel, which consist of a JSON representation and an SVG representation of the model. These are stored as monolithic CDATA sections within that file. The communication between client and server is performed via the exchange of objects, which are serialised with JSON.

The source code of the Activiti Modeler is partitioned into the following sub-projects: The core components Platform, Editor, and Libs, and the platform extensions BPMN 2.0 XML Basic, Core Diagram API, and syntax checker. The Platform core component contains all server-side java code and the jar files needed for inclusion into the web application. Besides the handling of requests, which come from the client, this java code also stores the models into the file repository. The Libs core component contains general JavaScript files and common resources like Cascading Style Sheets and images, which are needed to display and run the editor in the user's web browser. The Editor core component contains the BPMN 2.0 stencil set and images, as well as further JavaScript files, Cascading Style Sheets, and images, which are more specific to business process modelling. The BPMN 2.0 XML Basic platform extension contains amongst others the java classes, which instances represent the BPMN 2.0 models internally. These classes are generated from the XML Schema definition of BPMN 2.0 and are used for marshalling java objects to BPMN 2.0 XML files and unmarshalling such files to java objects with the JAXB framework. The Core Diagram API platform extension serves as a foundation for the BPMN 2.0 XML Basic platform extension and contains java classes for the generic representation of a graph model like bounds, points, edges, nodes, shapes, etc. The syntaxchecker Platform Extension contains the java code for the built-in syntax checker for BPMN 2.0 models.



**Figure 4: Activiti Designer**

The Activiti Designer is a plugin for the Eclipse IDE, which can be easily installed into the IDE via a dedicated Update Site. The plugin adds an Activiti perspective to the Eclipse workbench, which is shown in Figure 4. This perspective contains a graphical editor for BPMN 2.0 process diagrams in the upper right area, a customised properties view in the lower right area, and a miniature view for the diagram in the left lower area. Additionally, the plugin defines an Activiti project type and two file types for the BPMN 2.0 diagram and the BPMN 2.0 XML serialisation. The graphical editor displays the

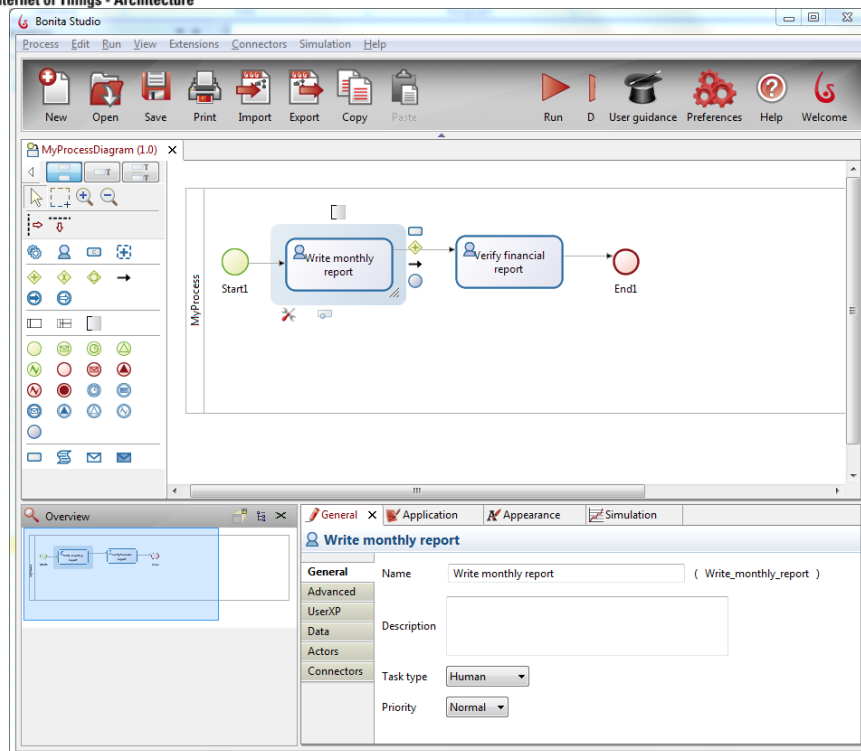


BPMN 2.0 palette right to its canvas, but this palette is incomplete: It does not allow the modelling of Pools, Lanes, and Message Flows, and some Event and Gateway types defined in the BPMN Specification are missing. Analogous to the Activiti Modeler a context sensitive editing palette is displayed around the selected model element to facilitate faster modelling with common diagram elements and for changing the current element quickly. The text editor for BPMN 2.0 XML files also provides a context sensitive help in an Eclipse standard conform way. The properties view shows the characteristics of the modelling element, which is selected in the canvas, and allows modifying them. For example the email header fields for a SendTask can be specified here. The project shown in the Package Explorer in Figure 4 is an Activiti project and contains the BPMN files in a diagrams folder under src/main/resources. For every BPMN model three files are generated: The .activiti file contains a proprietary XML serialisation of the process model, which is manipulated via the graphical editor. The .bpmn20.xml file contains a serialisation of the same model according to the BPMN 2.0 XML Schema Definition with the diagram interchange information. The third file contains a picture of the process model in the Portable Network Graphics (PNG) format. The latter two formats are generated from the first one automatically during editing. The Designer includes a syntax checker for the model, which is running when the modeller saves the model. If the syntax check fails, the .bpmn20.xml file is not (re)generated. To allow a holistic workflow with the Activiti Modeler and the Activiti business process execution engine, the Designer offers two additional functionalities: It enables the modeller to import a process model, which was created with the Modeler component, and allows the deployment of the modelled process to the execution engine. Unfortunately, the import of a model with Pools and Lanes fails, because the Modeler is not capable of modelling them.

The architecture of the Activiti Modeler is based on the Eclipse Modeling Framework EMF [46], the Graphiti Graphical Tooling Infrastructure [48], and the BPMN2 component of the Eclipse Model Development Tools MDT [49]. The source code is subdivided into the following sub-projects: The project org.activiti.designer.model contains the BPMN 2.0 meta-model from which the java code for the internal representation of BPMN process models is generated via the Eclipse Modeling Framework. The project org.activiti.designer.eclipse contains the core java code for the plugin. The project org.activiti.designer.gui contains the UI code and the project org.activiti.designer.util contains general utility classes for the plugin. The projects org.activiti.designer.export.bpmn20 and org.activiti.designer.export.image contain the exporters for the generation of BPMN 2.0 XML serialisations and png images from the modelled process diagrams. The project org.activiti.designer.validation.bpmn20 contains the syntax checker for the process models. The project org.activiti.designer.help contains the help files for the plugin. The project org.activiti.designer.integration provides an API for extending the Modeler with custom Task types. The exact capabilities of this extension mechanism are described in the Activiti documentation. The projects org.activiti.designer.feature and org.activiti.designer.updatesite are supplementary projects for the distribution of the eclipse plugin, and the project org.activiti.designer.parent contains the parent build file for building the plugin with maven.

### 3.3.2 Bonita Open Solution

The Bonita Open Solution software package [13] is built as rich client application on the Eclipse platform, which includes both, a business process modelling component and a business process execution engine.



**Figure 5: Bonita Open Solution**

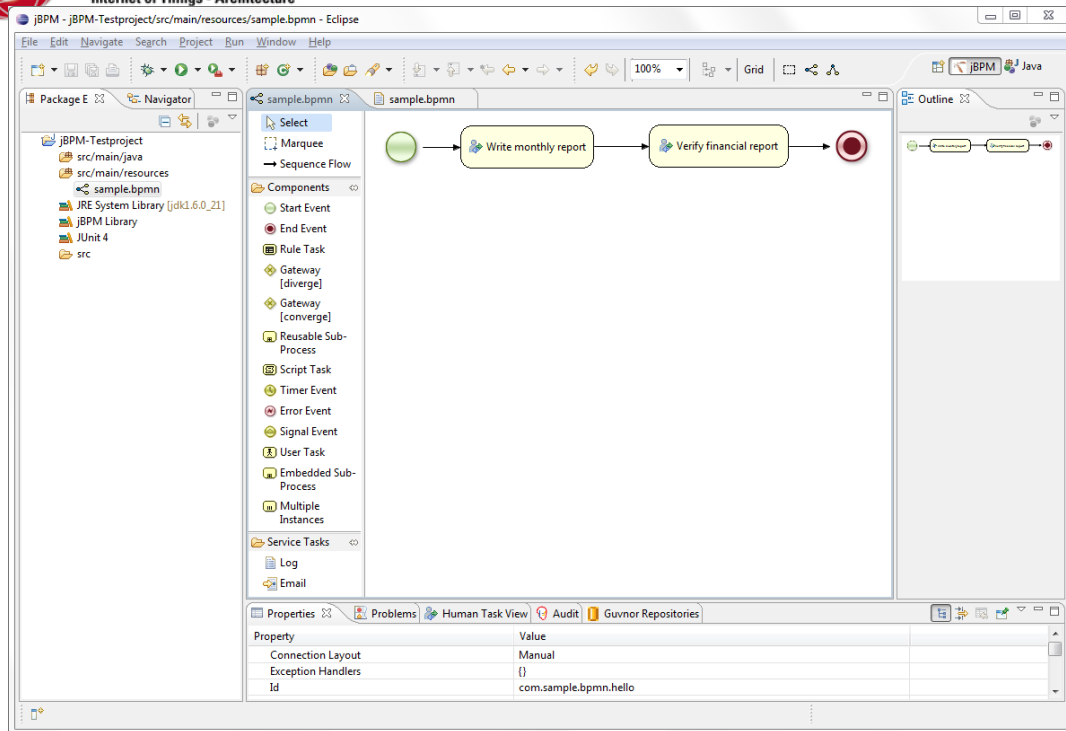
The graphical user interface for modelling business processes with Bonita Open Solution is shown in Figure 5. The palette with the stencils is displayed left to the canvas. It contains most of the stencils, which are defined in BPMN 2.0 – few Events and Gateways are missing. An overview of the process model is shown in the lower left part of the user interface. Right beside to this overview the properties of the selected process element or of the whole process are displayed and can be modified. Additionally, if a model element is selected in the canvas, some context sensitive buttons are displayed around it to facilitate fast process modelling. A validation of the modelled process is performed constantly during modelling. For the implementation of Service Tasks Bonita Open Solution offers an extensive list of connectors, which allow the specification of connections to different software systems. This includes amongst others interactions with different database systems, content management systems, enterprise resource planning systems, customer relationship management systems, web services, the execution of scripts written in several scripting languages or java code, and sending mails. Bonita Open Solution stores the modelled processes in its workspace in an XML serialisation, which does not use any BPMN 2.0 schemata. The user can export the process models to BPMN 2.0 serialisations, to bitmaps in the Portable Network Graphics (png) format, or as a Bonita application.

As Bonita Open Solution is an integrated software package for modelling and execution of business processes, the source code for these two components cannot be clearly separated from each other. Therefore a detailed discussion is omitted here.

### 3.3.3 jBPM

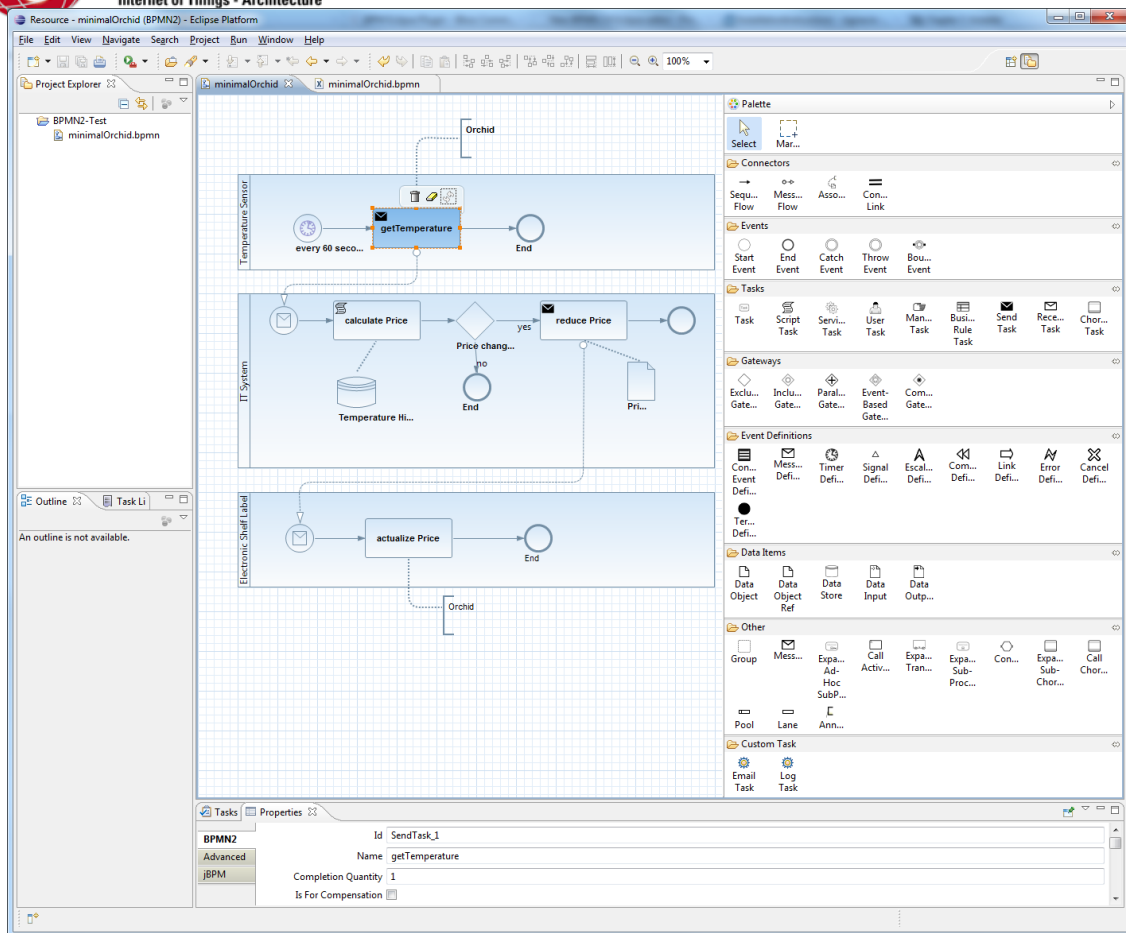
Like the Activiti Modeler, the jBPM software package [28] also provides different components for business process modelling. The jBPM Designer is a web application and the jBPM Eclipse Plugin and the BPMN 2.0 Eclipse Editor are plugins for the Eclipse IDE.

The jBPM Designer uses the Signavio Core Components as a technical foundation like the Activiti Modeler does. Therefore, we don't present it in detail here, but refer to chapter 3.3.1 for the discussion. As jBPM is a JBoss project, the Designer is usually deployed to a JBoss Application Server instead of an Apache Tomcat Server.



**Figure 6: jBPM Eclipse Plugin**

The jBPM Eclipse Plugin is a plugin for the Eclipse IDE, which adds a jBPM perspective and a BPMN editor to the IDE. The perspective and the editor are shown in Figure 6. The palette of the editor is displayed left to the canvas. The stencils are looking similar to the BPMN 2.0 stencils, but the stencil set is not the complete one of BPMN 2.0. Besides missing Gateway types and Event types, the concepts of Pools and Lanes cannot be modelled. Despite this, the Plugin serialises the models according to the BPMN 2.0 XML Schema Definition. The editor is supplemented with an outline view which provides an overview of the graphical model. For displaying the properties of the selected model element the standard Eclipse Properties view is used. Additionally, the jBPM Eclipse Plugin provides views, which can be used during the runtime of business processes: The Human Task view displays the human tasks, which can be performed by the user who is currently using the eclipse plugin. Here the user can amongst other things claim a task, start it, stop it, and set it completed. Furthermore several views for the interaction with the JBoss components Drools and Guvnor are provided.



**Figure 7: BPMN 2.0 Eclipse Editor**

The BPMN 2.0 Eclipse Editor [25] is also a plugin to the Eclipse IDE, which is developed for the jBPM project; it is shown in Figure 7. The plugin adds the shown editor and a dedicated properties view to the IDE. The palette is displayed right to the canvas and contains the full BPMN 2.0 stencil set. A context sensitive button area is displayed above the selected model element and provides functions to remove the element from the model or to refresh it. The properties view displays allows editing the details of the selected model element. For storing the model in the workspace, it is serialised according the BPMN 2.0 XML Schema Definition. Additionally the model can be exported to several bitmap formats.

The source code for the new BPMN 2.0 Eclipse Editor is subdivided as follows: The projects `org.eclipse.bpmn2`, `org.eclipse.bpmn2.edit`, and `org.eclipse.bpmn2.editor` are exactly taken from the BPMN2 component of the Eclipse Model Development Tools MDT. They provide the java classes for the BPMN 2.0 meta-model based on the Eclipse Modeling Framework EMF. The source code, which implements the actual editor on top of this meta-model and the Graphiti Graphical Tooling Infrastructure, is divided into the projects `org.eclipse.bpmn2.modeler.core` and `org.eclipse.bpmn2.modeler.ui`. The former project defines an extension point, which allows the definition of additional custom tasks. The projects `org.eclipse.bpmn2.modeler-feature` and `org.eclipse.bpmn2.modeler.updatesite-feature` are used for packaging and distributing the plugin.

Following the presentation of different business process modelling tools, the selection of the tool, which should serves as a basis for the IoT-aware process modelling environment, can be performed by applying a method of elimination. Because the Activiti Designer and the jBPM Eclipse Plugin don't contain Lanes and Pools in their palettes, they are not sufficient to serve as a basis for the IoT-aware process modelling environment. As Bonita Open Solution is an integrated tool for the modelling and execution of business processes, its code basis is much larger and not so clearly arranged in comparison to the other tools presented. Therefore Bonita Open Solution is not the best foundation to facilitate an easy development. Consequently, two tools remain as possible candidates as basis for the development: The web-based Signavio Core Components (which can be branded as Activiti Modeler or jBPM Designer) and the BPMN 2.0 Eclipse Editor. The former one has the nicer graphical user interface, but its code basis is contains heterogeneous components for client and server side.

The latter one has a small and therefore comprehensible code basis, but the user interface is less elaborated. So finally, to ease the implementation we suggest using the Activiti Modeler as basis for the development.

### 3.3.4 Conclusions

After the requirements analysis for the IAPMC and an evaluation of different existing modelling concepts we have selected BPMN 2.0 as an extendable modelling approach and the Activiti Modeler as the most suitable tool component for realizing the IAPMC.

## 4. General Design Principles

### 4.1 The Three Phases Model

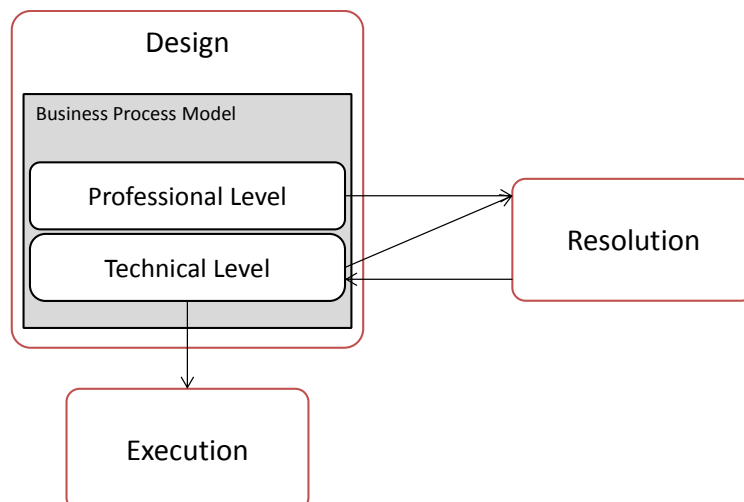
IoT-A WP2 distinguishes between different main phases [34] of a business process. This phase model and the associations between the phases need to be considered as general design principles for designing the IAPMC. In particular, the process notation BPMN 2.0 is used in practice as a professional and executable modelling notation [38]. In IoT-A, BPMN 2.0 will also serve as a uniform notation covering the three process phases design, resolution and execution.

The first phase, **business process design**, consists of two main areas: the professional and the technical modelling. Both areas are supported by BPMN 2.0. The central component of the process design is the business process model. To use a process notation for the professional modelling of IoT processes, as part of this deliverable it has to be decided which new constructs will be used to represent typical and all possible IoT situations in the process model. It therefore makes sense to define such constructs and the decision for them as modelling conventions. If the modelled processes shall be executed automatically by a process execution engine further constructs are needed on the technical level of the process model. These additional technical constructs may contain supplementations, restructuring and details, which are also defined in terms of modelling conventions.

This deliverable provides a modelling shell for the WP2 business process design in IoT-A allowing to cover and contain professional and technical process models. To get from the professional and already partly specified technical model to the complete executable model, IoT-A envisions the usage of a supportive and automated phase that is commonly named resolution. During this phase, the modeller may attempt to influence the results of the resolution phase. This results in an *interface* between the two phases. That means the specified process model contains conditions for some constructs that might be understood during the resolution phase.

In the design phase we support an end-user-friendly approach to create these conditions having the professional process model understanding in the foreground with no required programming knowledge by the end-user. The schema of the available conditions in the process model might be based on efforts presented in D2.1 such as device, entity and service descriptions. The *second interface* is given by the results of the process resolution phase accessing the technically specified process model. At this point, a technically skilled modeller may check and if required change the resolved process model. From a purely theoretical process modelling perspective the work performed during the process resolution is not obligatory, since a technical versed modeller envisioned by [9] might be able to directly create an executable process model knowing the availability and accessing details of the involved IoT capacities without any resolution phase. If the technical process model is ready for execution purposes, as a *third interface*, the execution engine accesses the complete model during the execution phase.

Figure 8 shows the three phases and their interfaces to one another within the phase model.



**Figure 8: Business Process Phases Model**



Second, the **process resolution phase** is concerned with the automated resolution of the so far specified professional and technical business process model in order to produce a complete technical model in terms of IoT capacities that is executable by a process execution engine. Within the resolution phase, SOA is a central idea, which abstracts from software to service notations. This includes many detailed different aspects such as searching for fitting services in an IoT, based on process conditions for the constructs such as physical entity, IoT device or IoT service. In addition, the binding between services and process tasks, the orchestration respectively composition of services are important aspects. To get a detailed overview of the concepts proposed by WP2 for the resolution phase, we refer to D2.3.

Third, detailed, technically specified process models can be performed automatically by a process execution engine during the **process execution phase**. As explained in section 2.3.1, for BPMN 2.0 execution semantics are defined as part of the standard. By introducing the IoT construct extensions to the process model, a gap occurs on execution site between conventional technical models and the advanced technical models containing additionally IoT constructs. This means that the process execution engine doesn't know without any additional execution semantics how to deal with the execution of the additional IoT constructs in the technical process model. Also for the conversion of BPMN models into other IoT executable process descriptions, such as BPEL, a description is helpful on how to execute the IoT constructs. The BPMN specification itself contains a proposal in the annex about mapping between BPMN and BPEL. BPMN models can also be run directly by novel BPMN compliant process execution engines. To get a detailed overview of the planned process execution in WP2, we refer to D2.3.

## 4.2 BPMN 2.0 Conventions

This section clarifies the BPMN 2.0 conventions which have to be considered as general design principles for the BPMN extension implementation of section 5.

All BPMN constructs can actually be changed and adjusted, if the appropriate rules are followed. It must be guaranteed especially with graphical elements, that the original BPMN shape remains, so that it cannot be confound. Constructs can be adjusted by the use of fonts and colours. Critical has to be evaluated a change of lines of activities, gateways, events, flows and associations. In the same way it has to be ensured that all types of lines remain. Modelling elements may be extended by graphical symbols. It is important to ensure that all basic BPMN symbols remain visible. For example, if a new activity type is introduced, a new icon can be used, but the typical activity rectangle has always to be used.

Moreover, in BPMN, new attributes can be defined, which have to be defined using the provided extension mechanism provided. It is also possible to define additional artefacts and to connect by associations with existing BPMN elements.

In this research work the extensibility mechanism can be used to add non-standard elements or artefacts to satisfy the defined IoT requirements. Such the BPMN core provides together with the IoT specific BPMN extension an executable business process notation for the IoT domain. According to [14] IoT extensions must follow the following general conventions:

- BPMN 2.0 IoT extension attributes **MUST** not contradict the semantics of any BPMN 2.0 element.
- The BPMN 2.0 IoT diagram **MUST** still have the look and feel of a BPMN 2.0 diagram.
- The footprint of the basic BPMN 2.0 elements event, activity and gateway **MUST** not be altered.

The BPMN 2.0 specification distinguishes further between mandatory and optional extensions. A mandatory extension means that a compliant implementation **MUST** understand the extension while an optional extension means, that a compliant implementation **MAY** ignore the extension. Building upon this definition, we plan to provide a mandatory IoT extension. [14] defines a concrete syntax for BPMN compliant extensions. Through this approach more interchangeable models can be guaranteed, where only additional attributes and elements may be lost during interchange. The supported IoT specific BPMN extension consists of the four different elements: Extension, ExtensionDefinition, ExtensionAttributeDefinition and ExtensionAttributeValue.

All BPMN extensions focus on the fact that an extended BPMN diagram can be read by anyone who knows the BPMN standard. Only the individual IoT extensions need additional explanations that are provided by section 5.

## 5. IoT-Aware Process Modelling (IAPM)

### 5.1 Main Conceptual Components

To define the key conceptual components, this section builds on the results gained by section 3.2 where existing modelling concepts were evaluated. It contains the IoT awareness of the most important modelling notations and classifies them according to topic areas in the categories: covered, partly covered and not covered. Summing up, the currently most appropriate notation for modelling IoT-based business processes is BPMN 2.0. Nonetheless, the usage of BPMN 2.0 in the IoT domain has some weaknesses. As weaknesses we consider all fields of the BPMN column that are not marked with a "yes" for covered. The evaluation for the points abstraction, flexibility, event-based and real-time has shown that primarily there is no need for rectification.

All weaknesses are analysed based on [3]:

- **Entity based concept:** While the service is the central concept of business processes in enterprise systems like ERP following nowadays a SOA based approach, the Physical Entity, and its associated devices having resources and providing services using a web like structure, are a key concept in the IoT domain [58]. Physical Entities relate to the real-world "things" that a business process is interested in. Devices are hardware that is associated to a Physical Entity, e.g. a tachometer measuring the velocity of a car. Resources are computational elements, e.g. sensor software hosted on the sensor device. An IAPMC shall support the entity based concept, but in the same way it shall be still integrable with current industry modelling standards.
- **Distributed execution:** In an enterprise environment the automated and semi-automated execution of a predefined and modelled business process is a huge benefit amongst many other leading to time and cost efficiency. In contrast to having one central process engine in one system, IoT-enabled processes would bring the possibility to distribute the process execution over many devices. An IAPMC shall support these distributed execution activities, while still supporting current business process engines following a one-device approach.
- **Interactions:** Business processes in the IoT introduce two additional forms of interactions: First, the device interacts or associates to Physical Entities. Second, the services that are known by enterprise systems processes interact with the software components of devices. An IAPMC shall provide means for expressing those types of interactions during process design.
- **Distributed data:** When business processes are realized in a nowadays enterprise system, often one central data storage is used. One potential of the IoT is the possibility to distribute data over several data storages. An IAPMC shall support arranging this distribution of data.
- **Scalability:** In enterprise systems business processes mainly use one central service repository having a relative static number of services. The IoT characterizes that multiple Physical Entities and its associated Devices and therewith its services can appear constantly. Even the associations between Physical Entities and Devices may vary. The complexity of an IoT-aware business process shall be independent from the number of Physical Entities, Devices and services. The IAPMC shall support means to cover this complexity to modeller. Additionally, the growing number of Devices shall not influence the performance of the three process phases. Therefore, IAPMC shall support the definition of ratios.
- **Availability / Mobility:** The availability of services in an enterprise system can be often considered as static. Due to the mobile nature of some Devices and Physical Entities participating in a defined business process, the availability of its related services might not be guaranteed throughout. This can affect the execution of a resolved business process in which an involved mobile Device or Physical Entity has become unavailable. The IAPMC shall support the disappearance and re-emergence of mobile process Devices or Physical Entities.
- **Fault tolerance:** The availability of Devices in the IoT is uncertain. A business process relies on the presence and communication technology of a Device and its related services. Faults may result from the absence of Devices or its communication technology, while a business process still needs to be highly reliable. The IAPMC must support the fault tolerance of a business process.
- **Quality of Information:** Within enterprise systems information can be mostly considered to be accurate. As Devices and its services providing, accessing and updating data with different levels of information accuracy may be involved in business process, the quality of an



accessed data object may vary widely and its entropy can get uncertain. In order to deal with this difficulty, the IAPMC shall provide means to express the quality of information.

By far not all the weaknesses are fully solved by the extension of the process model, but an IAPMC can contribute to solving problems at other stages. To close these weaknesses, the IoT domain model as part of the IoT-A project results from D1.2 has been mapped to the area of process modelling.

As a basic step [6] takes up the main concepts of the domain model and applies it to BPMN. Therefore, it focuses on process activities of sensors and actuators as well as the Physical Entity. A mapping and integration problem between the two strong domains could be identified. As a first result it comes up with the two IoT specific process activities ActuationTask und SensingTask and proposes to introduce the Physical Entity as a new object with associations to the two tasks.

Further and iterative work on the IoT domain model mapping could revise and refine these results for this deliverable. Table 10 shows a selection of concepts of IoT domain model and its mapping capabilities in BPMN 2.0, which are particularly relevant for the meta-model of the following sections.

| IoT Domain Concept | BPMN 2.0 Concept                     | Coverage by BPMN 2.0             |
|--------------------|--------------------------------------|----------------------------------|
| User               | Participant, Resource                | Sufficient                       |
| Human User         | Participant, Human Task, Manual Task | Sufficient                       |
| Physical Entity    | Participant                          | Not sufficient                   |
| Virtual Entity     | -                                    | Not sufficient                   |
| Device             | Participant, Resource                | Not sufficient                   |
| Sensor             | Participant, Resource                | Not sufficient                   |
| Tag                | Participant, Resource                | Not sufficient                   |
| Actuator           | Participant, Resource                | Not sufficient                   |
| Resource           | -                                    | Deviating definition, not needed |
| Service            | ServiceTask                          | Sufficient                       |

**Table 10: Coverage of IoT Domain Model Concepts by BPMN 2.0**

In addition, there are aspects such as quality of information, or mobility, which are not affected by a mapping of the domain model. However, the further above mentioned analysis aspects are included for the design of the IAPMC.

## 5.2 Formalisation

This section proposes an initial version of the IAPMC and its BPMN 2.0 implementation, which will be further improved and developed by reworking and feedback during the IoT-A project. By a planned subsequent tool implementation, an evaluation of the presented concept elements will be performed. The results of the evaluation will serve as a feedback for the second version of the IAPMC. At this point of time the IAPMC comprises seven main pillars:

- IoT Activity
- Sensing Activity
- Process Resources
- Physical Entity
- Real World Data Object / Store
- Mobility Aspect
- IoT Process Ratios

## 5.2.1 IoT Activity

In this section we consider a basic conceptual introduction of an IoT specific activity to the process model. This work is divided into two major sections of content: First, the introduction of an Actuation Activity is considered such as the air cooling through a refrigerator and second, the introduction of a Sensing Activity is considered such as the temperature measuring by a temperature sensor. The two new types of activities are considered independently of BPMN and expressed in a metamodel. As a second step, the activities are transmitted on the BPMN specific metamodel, a symbolism is proposed, new attributes and associations are defined and finally established in the XML schema file as a BPMN standard extension.

### 5.2.1.1 Actuation Activity

An actuator is a physical construction element which can implement electronic signals into mechanical motion, or other physical quantities. Thus, an actuator regulates a control system or concrete sizes are pretend to it to be achieved. After the successful completion of the Actuation Activity the physical state of a Physical Entity may has changed. The actual process of transformation is called actuation. Within the IoT the interaction with software components of such actuators is provided by services within concrete processes that dispose of well-defined service interfaces.

#### What does process modelling already offer to reflect Actuation Activities?

In process modelling, there can be distinguished between service activities, human activities and send activities, which have similar properties like Actuation Activities. In the following they are examined for their potential usefulness.

Service activities are used for web services and automated applications. Automatically means that the activity will start automatically once the sequence flow arrives, completely without human intervention. Some process modelling approaches know beside service activities, human activity. Human activities allow reflecting typical workflow tasks that are performed semi-automatically by the help of people or fully manual tasks that do not require any help of a process execution engine. In addition to service activities and human activities, some process modelling notations know send activities. They are used to send messages to external process partners. Upon reaching the process flow, a send activity has to realize just one task: to send a message, after which the process flow continues immediately.

#### Why are the existing concepts not sufficient to reflect Actuation Activities?

Service, human and send activities are all similar to the Actuation Activity, but none of these three types allow to fully capture the Actuation Activity properties.

Among the service activities there is no way to distinguish whether a specific value is calculated or whether software components of actuators are accessed that take over the task execution for them in the field of a dedicated Physical Entity. While a service activity has a maximum amount of one output set, an Actuation Activity provides no date output set (out-only). It remains unclear for the processes execution engine without verification of the Actuation Activity by an additional Sensing Activity, whether the physical state of the physical entity is achieved.

Although human activities get along without Process Execution Engine like Actuation Activities, but only refer to activities that are performed with the help of a person. An actuator, however, can edit tasks without human intervention completely automatically.

Send activities may send messages to external process participants, but actuators are not external process participants. In the process flow, Actuation Activities are directly accessed as well as other service activities via its interface; consequently no message flows occur across process boundaries.

#### What are the reasons for the introduction of additional concepts for modelling Actuation Activities in a business process?

- The process modeller needs to be explicitly advised that he adds an Activity Actuation to the process model. So he can adjust the business process, if this is required through the specific characteristics of the real world component.
- The process model has to express the new relationship “acts on” between the process participant actuator device and the Physical Entity. [6]
- The process model must encapsulate both services consisting of the web service and the native software components of the actuator as a unit. The web service, which has well-defined interfaces, knows to access the native actuator components.



- All related activity types such as service, human and send activity do not match the properties of an Actuation Activity.

### What are the functional properties of an Actuation Activity?

#### An Actuation Activity

- has an Input Data Set. (If RESTful, it would have a PUT)
- has no Output Data Set.
- is not executed by Business Process Execution Engine.
- is not managed by Business Process Execution Engine.
- has a well-defined service interface.
- has a connection from a data object or data store.
- provides a service to act on a Physical Entity.

### What are the requirements of an Actuation Activity in the process model?

It is presented a short list of properties that can be formulated as requirements with values for an Actuation Activity. In the process model, these requirements are used to select an appropriate Actuation Activity out of the stock. The full list of these properties will be based on the service descriptions with which the formulated requirements can be reconciled. At this point we refer to D2.1, which deals in detail with such service descriptions.

The following aspects are considered to be beneficial in the process context:

- Temporal properties
  - o Start time: When should the Actuation Activity be started?
  - o End time: Until when should the Actuation Activity be terminated?
  - o Execution Time: How long should the Actuation Activity be performed?
  - o Delay: Which average delay may occur when executing the Actuation Activity?
- Spatial properties
  - o Location: Where should the Actuation Activity be performed?
  - o Area of influence: Which area is intended to be affected by the Actuation Activity?
  - o Local tolerance: What is the average local tolerance that the Actuation Activity should have?
- Reliability properties
  - o Error: Which error rate should the Actuation Activity have?
  - o Accuracy: Which accuracy should the Actuation Activity have?
  - o Restrictions: What restrictions should the Actuation Activity have?
- Optional, since it could be also specified on an overall IoT device level (see 5.2.2):
  - o Actuator Features: By which device / device type should the Actuation Activity be executed?
    - Device ID
    - Device type
    - Manufacturer
    - Place of manufacture
    - ...
    - Sensors: Which sensors / sensor types shall measure the current physical state of the actuator?

### BPMN independent implementation

A process contains many process elements. These process elements can include activities that carry out a working unit within a process. A particular type of activity is the Actuation Activity that is introduced in this section as a subclass to Activity. An Actuation Activity is a fully automatic activity without any human interaction that starts after the process flow reaches the activity. It is an atomic activity, which cannot be further subdivided and is designed to pass actuation information to process internal actuators, which cannot be accessed by the process execution engine. If this information is transferred, the activity is performed. In addition to the associations of the class activity, it references to exactly one service description (e.g. USDL file). The description specifies a maximum of exactly one message output that can be filled by a data input for the one-sided communication. A service description can contain any number of parameters that describe the service interface. To even refer to a meaningful service description for the process, the process modeller can create a requirement catalogue for an Actuation Activity. This activity specific requirement set contains a set of parameters with assigned values. The condition catalogue of the activity can thus be compared in the context of the process resolution with potential service descriptions and be bound to the Actuation Activity.

Figure 9 shows the metamodel of the Actuation Activity. Furthermore, the Actuation activity has an important relationship to the Physical Entity whose physical state can be changed by successful execution. This relationship is introduced in section 5.2.4.

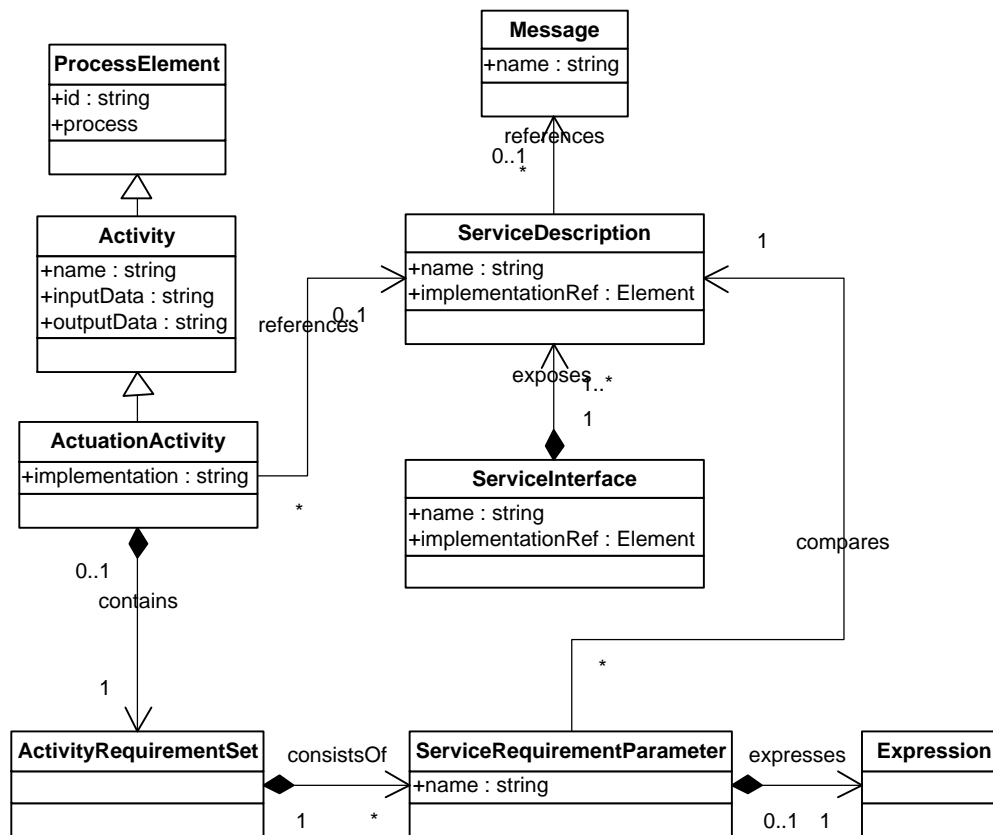


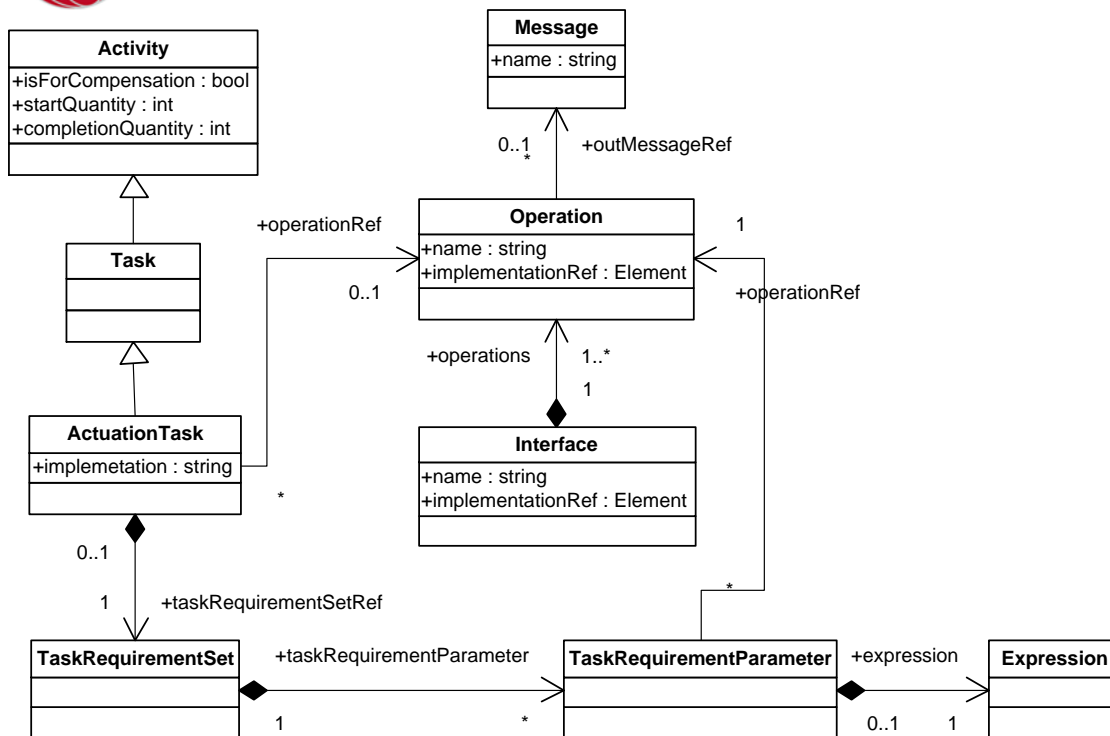
Figure 9: Actuation Activity Metamodel

### Technical implementation of the enhancements in BPMN

In the following the metamodel for the Actuation Activity is transferred to the modelling notation BPMN 2.0. Although BPMN covers different types of tasks, yet the Actuation is not captured perfectly by any of the tasks.

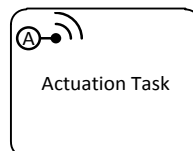
In BPMN 2.0, an atomic activity is called task if it cannot be decomposed further into a process flow. Similarly, there is in consequence an *ActuationTask*, which is a subclass of *Task*. The Actuation Task has the same attributes and model compounds such as an *Activity*. In addition, for the Actuation Task introduces the following restrictions: It is an abstract task in the business process (which access is mostly based on REST principles) that uses a software service, which in turn transfers data to the software components of the IoT Device of the type actuator [58]. The Actuation Task has exactly one data input with an *ItemDefinition* equivalent to those referenced in the attribute *outMessageRef* of the class *message* by the associated class *operation*. Since the actuator doesn't receive back a message, in contrast to the BPMN service task there is no *outputSet*. The implementation of the Actuation Task specifies the technology to send the messages. In our case the Actuation Task references to a service interface. A service interface may include multiple operations.

These operations can be specified by the optional attribute *ImplementationRef*, which can refer to a concrete implementation like USDL, in which all functional and non-functional properties of the service operations are described. This description of an Actuation Task defines the message element for the actuation request. In order to enable the resolution infrastructure to generate a suitable reference, as described above requirements can be described. Thus, the automatic resolution is influenced by the process model.



**Figure 10: Actuation Task BPMN Integration**

To mark that an Actuation Task differ from other tasks in the process model the existing BPMN task objects are extended by an supplementary object type. Figure 11 shows the Actuation Task object "Actuation Task" indicated by an actuation marker in the upper left corner of the figure. For this purpose, the shape of a rectangle with rounded corners is kept. A plus symbol in the upper right corner of the folded version indicates that the Actuation Task can be further specified. In the extended version, the modeller can specify the values for the parameters.



**Figure 11: Actuation Task Object**

The following tables present all classes and the newly added associations. For a complete attribute and association overview of Activity, please reference [14].

| Attribute Name                                | Description / Usage  |
|---|--|
| <b>implementation:</b> string = ##WebService  | This attribute specifies the technology that will be used to send the Messages. Valid values are "##unspecified" for leaving the technology open and "##WebService" or an URI identifying any other technology. A Web Service is the default technology. |
| <b>operationRef:</b> Operation[0..1]          | References the operation of a service interface of the pool that is invoked by the Actuation Task.   |
| <b>taskRequirementSet:</b> TaskRequirementSet | References the Task Requirement Set that is specified for the Actuation Task.  |

**Table 11: ActuationTask Attributes and Associations**

| Attribute Name  | Description / Usage   |
|---|---|
| <b>taskRequirementParameter:</b><br>TaskRequirementParameter [0..*] | Reference to the Parameters used to specify constraints to customize the automatic assignment of a suitable Actuation Service by the resolution infrastructure. |

**Table 12: TaskRequirement Attributes and Associations**

| Attribute Name                 | Description / Usage   |
|--------------------------------|---|
| <b>operationRef:</b> Operation | Reference to the parameter defined by the service description.  |
| <b>expression:</b> Expression  | The expression that evaluates the value used to specify constraints to customize the automatic assignment of a suitable actuation service by the resolution infrastructure. |

**Table 13: TaskRequirementParameter Attributes and Associations**

Below, the XML schema of the Actuation Task within the context of the BPMN 2.0 standard is offered.

```

<xsd:element name="actuationTask" type="tActuationTask" substitutionGroup="flowElement"/>
<xsd:complexType name="tActuationTask">
  <xsd:complexContent>
    <xsd:extension base="tTask">
      <xsd:sequence>
        <xsd:element ref="taskRequirementSet" minOccurs="1" maxOccurs="1"/>
      </xsd:sequence>
      <xsd:attribute name="implementation" type="tImplementation" default="##IoTService"/>
      <xsd:attribute name="operationRef" type="xsd:QName" use="optional"/>
    </xsd:extension>
  </xsd:complexContent>
</xsd:complexType>

<xsd:element name="taskRequirementSet" type="tTaskRequirementSet"/>
<xsd:complexType name="tTaskRequirementSet">
  <xsd:complexContent>
    <xsd:extension base="tBaseElement">
      <xsd:sequence>
        <xsd:element ref="taskRequirementParameter" minOccurs="0" maxOccurs="unbounded"/>
      </xsd:sequence>
    </xsd:extension>
  </xsd:complexContent>
</xsd:complexType>

<xsd:element name="taskRequirementParameter" type="tTaskRequirementParameter"/>
<xsd:complexType name="tTaskRequirementParameter">
  <xsd:complexContent>
    <xsd:extension base="tBaseElement">
      <xsd:sequence>
        <xsd:element ref="expression" minOccurs="1" maxOccurs="1"/>
      </xsd:sequence>
      <xsd:attribute name="operationRef" type="xsd:QName" use="optional"/>
    </xsd:extension>
  </xsd:complexContent>
</xsd:complexType>

```

### 5.2.1.2 Sensing Activity

A sensor is a physical construction element, which converts mechanical movement or other physical quantities into electronic signals. Thus, a sensor can intervene measuring in a control system – viz. a sensor can detect the physical state of a Physical Entity. The actual transformation process is called Sensing. Within the IoT the interaction with the software components of such sensors is provided by services that dispose of well-defined interfaces. The control engineering defines sensors as the signal



converter-based counterpart to actuators. An actuator changes the state of a Physical Entity, while the sensor measures the state so that a state change can be detected.

### **What does process modelling already offer to reflect Sensing Activities?**

In process modelling, there can be distinguished between service activities, human activities and receive activities, which have all similar properties like Sensing Activities. We will examine below on their potential usefulness.

As already described in section 5.2.1.1, service activities are typically used in process modelling for web services without any human interaction. In contrast, human activities allow semi-automated activities without having access to the process execution engine e.g. typical workflow tasks. In addition to these activities, some familiar notations know so-called receive activities, analogously to the send activities. They are used to receive messages from external process partners. A receive activity exclusively realizes one task: waiting for a message. After having received the message the process flow continues immediately.

### **Why are the existing concepts not sufficient to reflect Sensing Activities?**

Service, human and receive activities are all similar to the Sensing Activity, but none of these three types allow to fully capture the Sensing Activity properties.

Sensing Activities automate a process task and contain like service activities exactly one output set. While service activities also have exactly one input set, Sensing Activities don't provide any input amount of information. It remains unclear to the process execution engine, if the state of the Physical Entity provided by the service was actually measured by a sensor. The Sensing Activity only receives the sensed current state of the entity by the sensor (in-only). Thereupon the activity makes this information accessible to the process in form of a data object or store.

Human activities get along as Sensing Activities without any Process Execution Engine, but relate solely to activities with human intervention. A sensor can perform tasks without any human intervention.

A receive activity may receive messages from external process participants, but sensors are not external process participants. Sensing Activities are directly accessed as well as other service activities via its interface; consequently no message flow occurs across process boundaries.

### **What are the reasons for the introduction of additional concepts for modelling Sensing Activities in a business process?**

- The process modeller needs to be explicitly advised that he adds a Sensing Activity to the process model. So he can adjust the business process, if this is required through the specific characteristics of the real world component.
- The process model has to express the new relationship „monitors“ between the process participant sensor device and the Physical Entity. [6]
- The process model must encapsulate both services consisting of the web service and the native software components of the device unit. The native sensor components know how to access the web service, which has well-defined interfaces.
- All related activity types such as service, human and receive activity don't match the properties of a Sensing Activity.

### **What are the functional properties of a Sensing Activity?**

A Sensing Activity

- has no Input Data Set.
- has an Output Data Set. (If RESTful, it would be a GET)
- is not executed by Business Process Execution Engine.
- is not managed by Business Process Execution Engine.
- has a well-defined service interface.
- has a connection to a data object / store.
- provides a service to monitor a Physical Entity.

### **What are the requirements of a Sensing Activity in the process model?**

It is presented a short list of properties that can be formulated as requirements with values for a Sensing Activity. In the process model, these requirements are used to select an appropriate Sensing Activity out of the stock. The full list of these properties will be based on the service descriptions with



which the formulated requirements can be reconciled. At this point we refer to D2.1 dealing in detail with such service descriptions.

The following aspects are considered to be beneficial in the process context:

- Temporal properties
  - o Start time: When should the Sensing Activity be started?
  - o End time: Until when should the Sensing Activity be completed?
  - o Execution time: How long should the Sensing Activity be executed?
  - o Delay: Which average delay may have been occurred when executing a Sensing Activity?
- Spatial properties
  - o Location: Where should the Sensing Activity be executed?
  - o Area of sensing: Which area is intended to be measured by the Sensing Activity?
  - o Local tolerance: What is the average local tolerance that the Sensing Activity may have?
- Reliability properties
  - o Error: Which error rate should the Sensing Activity?
  - o Accuracy: Which accuracy should the Sensing Activity have?
  - o Restrictions: What restrictions should the Sensing Activity have?
- Optional, since it could be also specified on an overall IoT device level (see 5.2.2):
  - o Sensor properties: By which device / device type should the Sensing Activity be executed?
  - o Device ID
  - o Device type
  - o Manufacturer
  - o Place of manufacture
  - o ...
  - o Actuators: Which actors / actuator types shall have influenced the physical state measured by the sensor?

### **BPMN independent implementation**

In addition to the Actuation Activity introduced in section 5.2.1.1 a particular subclass `SensingActivity` is presented as a further IoT specific activity type. A Sensing Activity is a fully automatic activity that waits for the receipt of the sensor information after the reaching of the sequence flow. It immediately terminates when the data is received. The sensing activity is atomic, because it cannot be further subdivided. It is designed to receive sensor information of in-process sensors, which are not controlled by the process execution engine. The class `SensingActivity` inherits all attributes and associations of the class `Activity` and furthermore references to a service description. The description specifies exactly one input message and therewith fills a data output. Like an Actuation Activity the description can contain any number of parameters describing the service interface. The condition catalogue can then be contrived for the process resolution in order to bind a suitable service to the process model. Figure 12 shows the metamodel of the Sensing Activity. For the relationship between Physical Entity and activity it is referred to section 5.2.4.

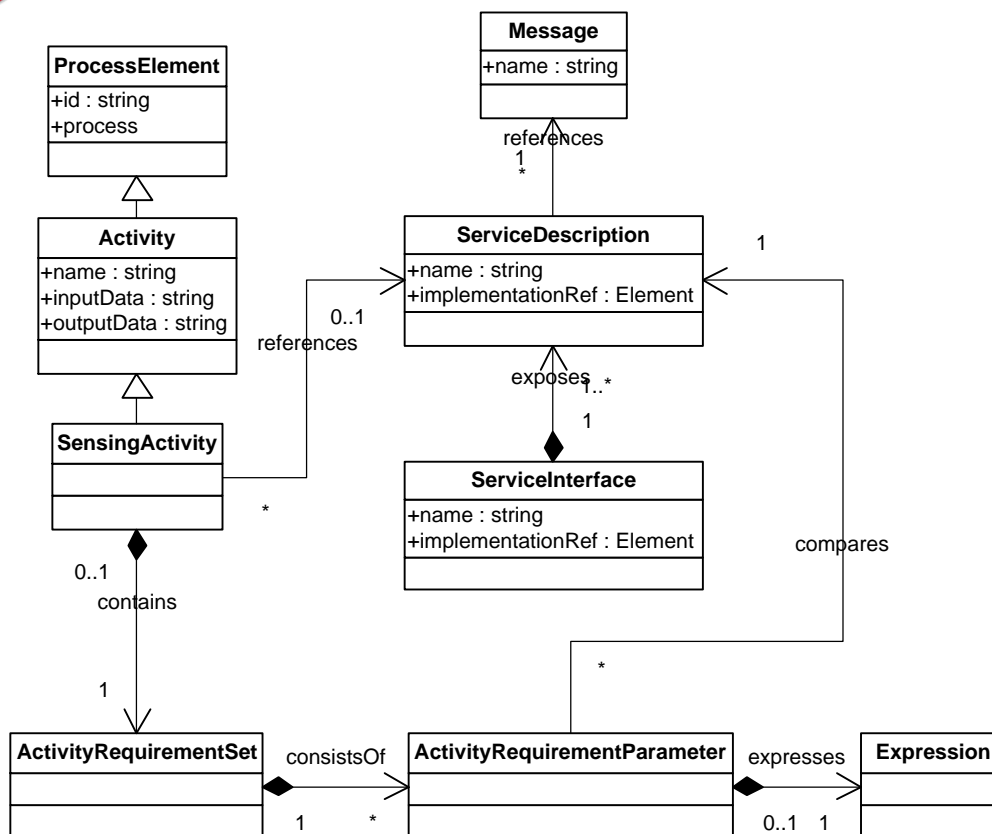


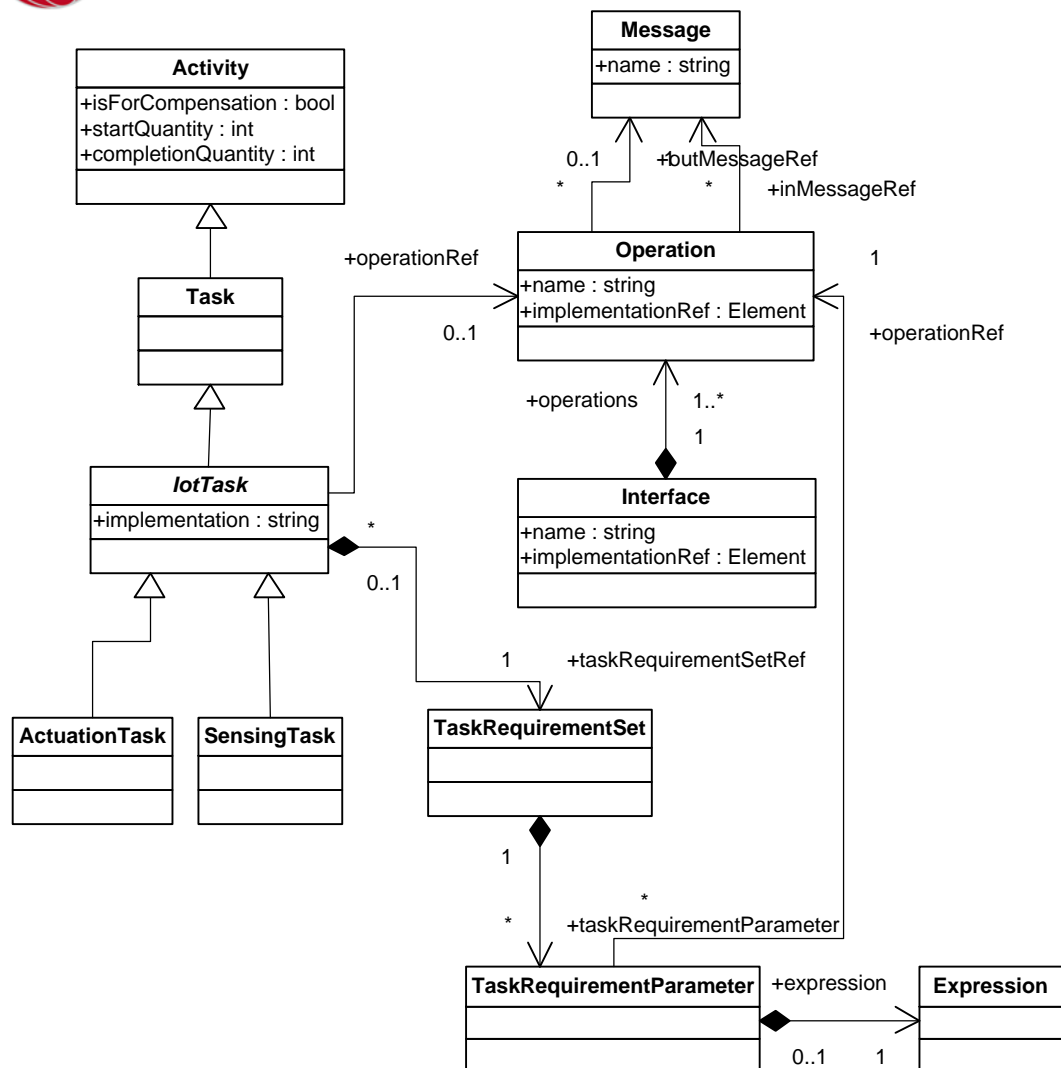
Figure 12: Sensing Activity Metamodel

### Technical implementation of the enhancements in BPMN

In the following the metamodel for the Sensing Activity is transferred to BPMN and combined with the BPMN Model of the Actuation Activity of the previous section. Since both meta-models contain the same associations, but the BPMN class `Task` must remain unchanged, we present the new abstract class `IoTTask` as a subclass of `Task`.

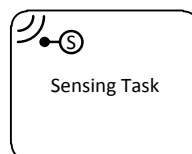
An IoT Task can either be of type actuation or sensing and inherits all the attributes and associations of `Activity`. A Sensing Task is an abstract task within a business process that uses a software service (which is mostly based on REST principles) to receive data from one software component of the sensor device.

In contrast to an Actuation Task, the Sensing Task comes with exactly one data output with an `itemDefinition` equivalent to those referenced in the attribute `inMessageRef` of the class `message` by the associated class `operation`. Since the service does not send any information, there is no `inputSet`. The other concepts of the Sensing Task act analogously to those already defined for the Actuation Task.



**Figure 13: Sensing Task BPMN Integration**

To distinguish a Sensing Task from other tasks in the process model the existing BPMN task objects are extended by a supplementary object type. Figure 14 shows the Sensing Task object "Sensing Task", indicated by a sensing marker in the upper left corner of the figure. A plus symbol in the upper right corner indicates that the Sensing Task can be further specified. In the extended version, the modeller can specify the required values.



**Figure 14: Sensing Task Object**

The following tables present all classes and newly added associations. For a complete attribute and association overview of Activity, please reference [14].

| Attribute Name                               | Description / Usage  |
|--|--|
| <b>implementation:</b> string = ##WebService | This attribute specifies the technology that will be used to send the Messages. Valid values are "##unspecified" for leaving the technology open and "##WebService" or an URI identifying any other technology. A Web Service is the default |

|   |  |
|---|--|
|   | technology.  |
| <b>operationRef:</b> Operation[0..1]          | References the operation of a service interface of the pool that is invoked by the Actuation Task. |
| <b>taskRequirementSet:</b> TaskRequirementSet | References the Task Requirement Set that is specified for the Actuation Task.                      |

**Table 14: IoTTask Model Attributes and Associations**

Below, the XML schema of the Sensing Task within the context of the BPMN 2.0 standard is offered.

```

<xsd:element name="iotTask" type="tIotTask" substitutionGroup="flowElement"/>
<xsd:complexType name="tIotTask">
  <xsd:complexContent>
    <xsd:extension base="tTask">
      <xsd:sequence>
        <xsd:element ref="taskRequirementSet" minOccurs="1" maxOccurs="1"/>
      </xsd:sequence>
      <xsd:attribute name="implementation" type="tImplementation" default="##IoTService"/>
      <xsd:attribute name="operationRef" type="xsd:QName" use="optional"/>
    </xsd:extension>
  </xsd:complexContent>
</xsd:complexType>

<xsd:element name="sensingTask" type="tSensingTask" substitutionGroup="flowElement"/>
<xsd:complexType name="tSensingTask">
  <xsd:complexContent>
    <xsd:extension base="tIotTask"/>
  </xsd:complexContent>
</xsd:complexType>

```

## 5.2.2 IoT Device

In this section, a basic conceptual introduction of the IoT Device is presented, and the translation to process modelling is considered. Excluded from this section is the relation between the device and activities and the process repository, which is a detailed issue of section 5.2.3.

### What is an IoT Device?

An IoT device is a physical construction element or component with communication capabilities to other IT systems. It can either belong to a physical unit, or can be directly integrated into it. The physical entity itself or its neighbours can be monitored or changed by the IoT Device. A mobile telephone, a sensor node, a single sensor or an actuator is for example an IoT device. Within the IoT, the interaction with the software components of IoT Devices is provided by services that have well-defined interfaces.

### What does process modelling already offer to reflect IoT Devices?

From the perspective of business process modelling a physical device is usually not considered and is therefore also not mentioned in most specifications. The reason for this is that often one ERP system is assumed that automates all business processes without including further devices. In our view, an IoT device in the process modelling language behaves as a so-called process participant.

A process participant is in turn defined as a business entity or as a business role controlling a business process or having any responsibility in it [14]. If the process notation implies pools and lanes, a participant is usually represented by using a lane [14]. An IoT device operates in a business process and depending on its type the individual IoT device is not important but its business role taken over.

It is therefore obvious one to consider an IoT device as a technical process participant. It is modelled using the pool or swim lane notation. Even though this representing possibility was intuitively used by many modellers for the WP7 retail use cases in order to represent IoT based processes. Also [33] considers this new situation in relation to mobile phones in business processes and recommends "extracting the mobile participant out of the pool of the organization to model its process on its own pool / lane." However, an IoT device in a business process doesn't only overtake the role of a participant, but in the same time the role of a performer, which will be discussed in section 5.2.3.

### Why are the existing concepts not sufficient to reflect IoT Devices?

The process modeller needs clear rules for the modelling of IoT Devices in IoT aware processes

- to ensure a standardized implementation of the modelled processes.
- to distinguish an IoT from an organisation presented by other lanes.
- to represent the interaction between IoT Devices and organizations as one common process flow.

So far, there is no way to formulate requirements by direct annotations for an individual participant as an IoT Device. These requirements could be used for the automated election of possible IoT Devices when resolving the process by the resolution infrastructure.

### **What are the reasons for the introduction of additional concepts for modelling IoT Devices in a business process?**

- The process modeller must be enabled to point out explicitly that he adds an IoT device to the business model. So he can adjust the business process, if this is required by the particular device properties.
- The process modeller needs for readability and standardization issues of the process models clear rules for the representation of IoT Device using pools or lanes.
- The process modeller must be able to express the relationships between process participants like IoT devices and Physical Entities in the model.
- The process modeller needs a way to affect the device selection by the resolution infrastructure by formulated requirements in form of annotations in the process model.
- The available process modelling concepts, such as process participants don't yet completely fit to the characteristics of IoT Devices.

### **What are the functional properties of an IoT Device?**

An IoT Device

- is a technical and external process participant.
- is a process internal artefact that is not executed by the Business Process Execution Engine.
- is a process internal artefact that is not managed by the Business Process Execution Engine.
- performs its activities.
- has an interface to the Business Process Execution Engine.
- may have functional and non-functional requirements which need to be considerable for the resolution infrastructure and Execution Engine.

### **What are the requirements of IoT Devices in the process model?**

It is presented a short list of properties that could be formulated as requirements with values to select a suitable IoT Device executing its assigned activities. The full list of these properties will be based on the IoT device properties with which the requirements are matched. At this point we refer to IoT-A D 2.1 that deals in detail with such resource descriptions.

The following non-exhaustive aspects could be considered in the process context to be useful:

- Device ID: Which device should be selected?
- Device type: Of which type shall the IoT Device be? (e.g. sensor, actuator)
  - Device type == sensor
    - By what actuator shall the IoT Device measure a physical state?
  - Device type == actuator
    - By what sensor shall the IoT Device affect a physical state?
- Manufacturer
- Location of manufacture

### **BPMN independent implementation**

A participant in a process is often responsible for the process execution of the assigned activities to the participants' pool / lanes. A participant may represent a division of his organization and within this division he may take over a certain business role. Transferred to the IoT device a typical participant is a mobile phone with the business role "mobile phone" in the division "Future Retail Centre, SAP Research Switzerland." In IoT-aware process modelling it makes sense to indicate the participant to dispose the choice of a suitable IoT Device to the resolution infrastructure. Is the participant of the subclass `IoTDeviceRole`, the indication of the participant role is required, otherwise optionally. It is supposed that in the given division are offered various mobile phones to execute the assigned tasks of the mobile phones pool / lane. To influence the selection of the desired device, it is possible to specify

various `IoTDeviceRoleParameter`. For each parameter using class `Expression` values can be entered, which constrains the device selection. So for each participant having the role `IoTDeviceRole` a set of parameters results with optional values. This set is used during the resolution process as a condition catalogue to find a suitable `Participant` (mobile phone) and to bind the IoT Device to the `Participant`. The detailed binding abilities are described in section 5.2.3.

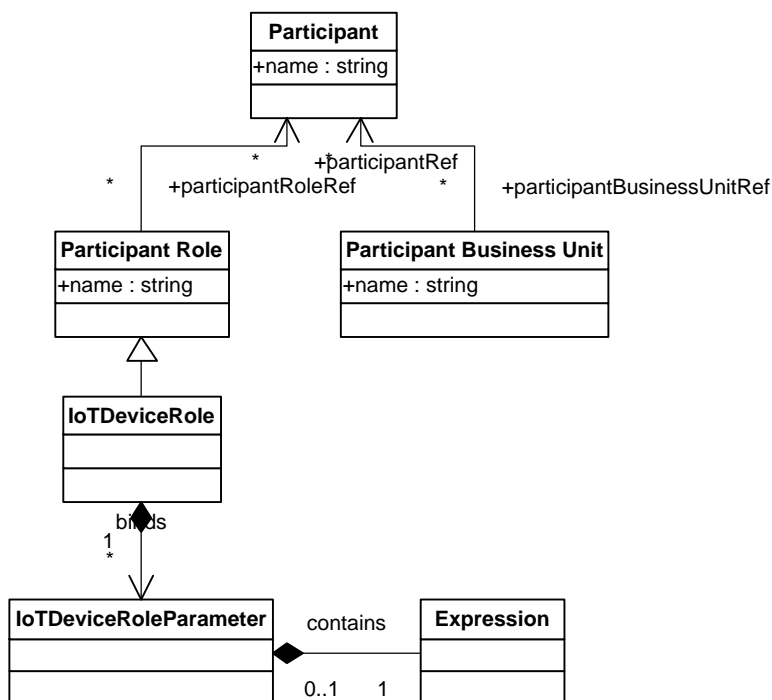


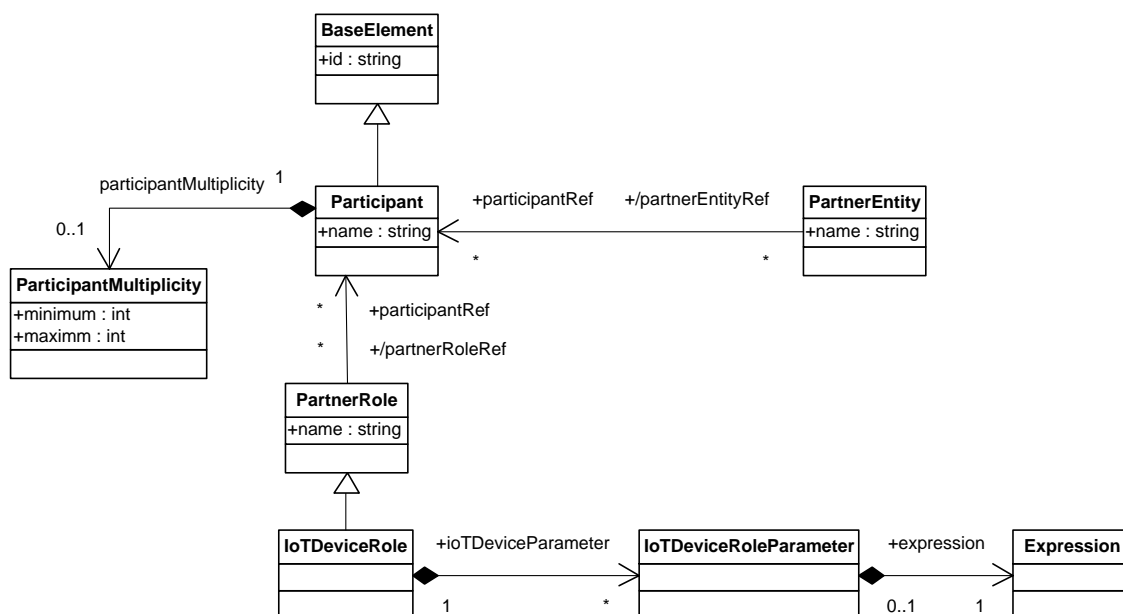
Figure 15: IoT Device Metamodel

### Technical implementation of the enhancements in BPMN

In the following the class diagram of the IoT Device metamodel is adapted to the modelling notation BPMN 2.0.

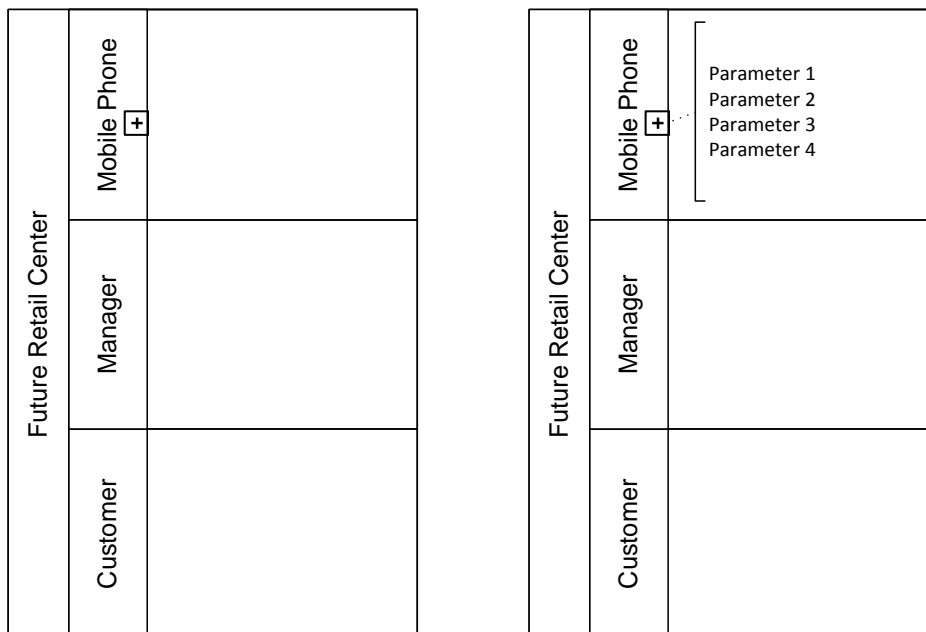
In a collaboration diagram of BPMN, a `Participant` characterises a specific `PartnerEntity` (e.g. Future Retail Centre, SAP Research Switzerland) and or a more general `PartnerRole` (e.g. Mobile Phone). Although a `Participant` is usually responsible for the process execution encircled with a `Pool/Lane`. A `Pool/Lane` might be also defined without any artefacts, but is not considered in this case. Since an IoT Device always acts in the same time as a performer in the process, an IoT Device participant without any artefacts would not be considered as a performer and would not be needed any longer for the process.

Figure 16 presents the BPMN participant model with the IoT Device extension. The `Participant` element is a subclass of `BaseElement` and thereby inherits the attributes and model associations. Integrating the IoT Device metamodel, we extend the `PartnerRole` class introducing the new subclass `IoTDeviceRole`. The `IoTDeviceRole` class provides additional information about the execution nature of the process `Participant`. An IoT Device `Participant` shows a new element in the process pool of the `Collaboration` which enables to specify the semantic description of the IoT Device for resolution purposes. By default, the `Participant` is not of the Subclass `IoTDeviceRole`. For supplementing the `IoTDeviceRole` class by the set of parameters we add accordingly to the IoT Device Metamodel the two classes `IoTDeviceRoleParameter` and `Expression`.



**Figure 16: IoT Device BPMN Integration**

To enter the selection of properties directly to the process model the existing BPMN elements pool / lane need to be also graphically expanded. Figure 17 shows a process pool called "Future Retail Centre", which contains three process participants. The process participants are the regular participants "Customer" and "Manager" and the IoT Device "Mobile Phone". A plus symbol in the collapsed version in the corresponding lane indicates that the Mobile Phone operating in the process can be specified further. In the extended version the modeller can specify the required values.



**Figure 17: Collapsed and Extended IoT Device Object**

Table 15 and Table 16 present all classes and the newly added associations. For a complete attribute and association overview of Participant, please reference [14].

| Attribute Name                 | Description / Usage                            |
|--------------------------------|--|
| <b>IoTDeviceRoleParameter:</b> | Defines the Parameter bindings used to specify |



|                               |  |
|-------------------------------|--|
| IoTDeviceRoleParameter [0..*] | constraints to customize the automatic assignment of a suitable IoT Device by the resolution infrastructure. |
|-------------------------------|--|

**Table 15: IoTDeviceRole Attributes and Associations**

| Attribute Name                       | Description / Usage  |
|--------------------------------------|--|
| <b>expression:</b> Expression [1..1] | The Expression that evaluates the value used to specify constraints to customize the automatic assignment of a suitable IoT Device by the resolution infrastructure. |

**Table 16: IoTDeviceRoleParameter Attributes and Associations**

Below, the XML schema of the IoT Device extension within the context of the BPMN 2.0 standard is offered.

```

<xsd:element name="IoTDeviceRole" type="tIoTDeviceRole" substitutionGroup="partnerRole"/>
<xsd:complexType name="tIoTDeviceRole">
  <xsd:complexContent>
    <xsd:extension base="tPartnerRole">
      <xsd:sequence>
        <xsd:element ref="IoTDeviceRoleParameter" minOccurs="0"
          maxOccurs="unbounded"/>
      </xsd:sequence>
    </xsd:extension>
  </xsd:complexContent>
</xsd:complexType>

<xsd:element name="IoTDeviceRoleParameter" type="tIoTDeviceRoleParameter"/>
<xsd:complexType name="tIoTDeviceRoleParameter">
  <xsd:complexContent>
    <xsd:extension base="tBaseElement">
      <xsd:sequence>
        <xsd:element ref="expression" minOccurs="1" maxOccurs="1"/>
      </xsd:sequence>
    </xsd:extension>
  </xsd:complexContent>
</xsd:complexType>
  
```

### 5.2.3 Process Resources

In this section we consider a basic conceptual introduction to connect the concepts IoT devices with resources and activities in the process modelling. The resource view of this chapter is heavily dependent on BPMN 2.0. This section bases fundamentally on the already established IoT Device extensions of section 5.2.2.

#### What is a resource in process modelling?

The term of resource in business process modelling is different from the resource IoT definition of IoT-A [58]. In business process modelling, a resource is defined as an abstract concept. It is used to classify technical or human operational capacity during the process resolution for the execution. A resource can execute multiple activities and vice versa multiple activities may use the same resource. An example of a resource in process modelling is an IoT device such as a temperature sensor that can provide measurement capabilities in form of process activities for the process execution. Only a few modelling notations support this resource-specific view in detail. Contrary to the conventional procedures of this deliverable, we adopt the BPMN view for this aspect. Since the BPMN resource assessment is already developed and tested in detail we will overtake this knowledge to the BPMN independent and general IoT metamodel.

#### What is a resource in BPMN?

In BPMN, an arbitrarily large set of parameters can be defined to each applied resource, which describes the resource properties. So it would be possible that all available processing resources are kept in a universal directory for execution reasons. A resource is then assigned to one or more activities for executing the process. On side of the process activity requirements can be formulated in

order to influence the allocation of the resources. For the time resolution of the process can be a resource request with specified resource requirements, which have been entered for a process activity can be resolved. A process activity is then referenced the assigned resource. At resolution time a resource request can be resolved according the resource requirements which have been entered for the process activity. A process activity then references the assigned resource.

### Why are the existing concepts not sufficient to reflect IoT Devices as resources of activities?

In the previous section, the resource allocation is specified on participant-level and not on activity-level of the process. This is due to the separate and reasonable viewing of IoT devices as process participants. To merge the two models the existing BPMN model lacks of the following two aspects:

- There is no possibility to match specified requirements for an IoT Device with the resource stock to resolve inquiries.
- There is no possibility to assign the resulting IoT Device amount to the activities of an IoT Device.

### Technical implementation of the enhancements in BPMN

To map the missing aspects in BPMN, it is essential to analyse the resource-metamodel on the relationship between activities and resources: In BPMN there is the class `Resource` of which are all resources. The characteristics of the resource are assigned to the resource via the association `resourceParameter`. The allocation between the process activity and the resource is realized by the reference `ResourceRef`. To influence the allocation between resource and activity requirements to the resource allocation can be formulated by using the class `ResourceRole`. BPMN distinguishes between a resource allocation using expressions and parameters. While expressions imply resources-related data types such as users and groups, parameters are used to formulate queries that serve for allocations at runtime. Both types of resource allocation may have multiple resource references as a result, which are assigned to subclass of `ResourceRole`. The class `Performer` is a subclass of `ResourceRole` and thus inherits all the attributes and model associations. It defines the resource that performs an activity. Figure 18 shows the current BPMN metamodel for the resource allocation without the proposed IoT-specific extensions.

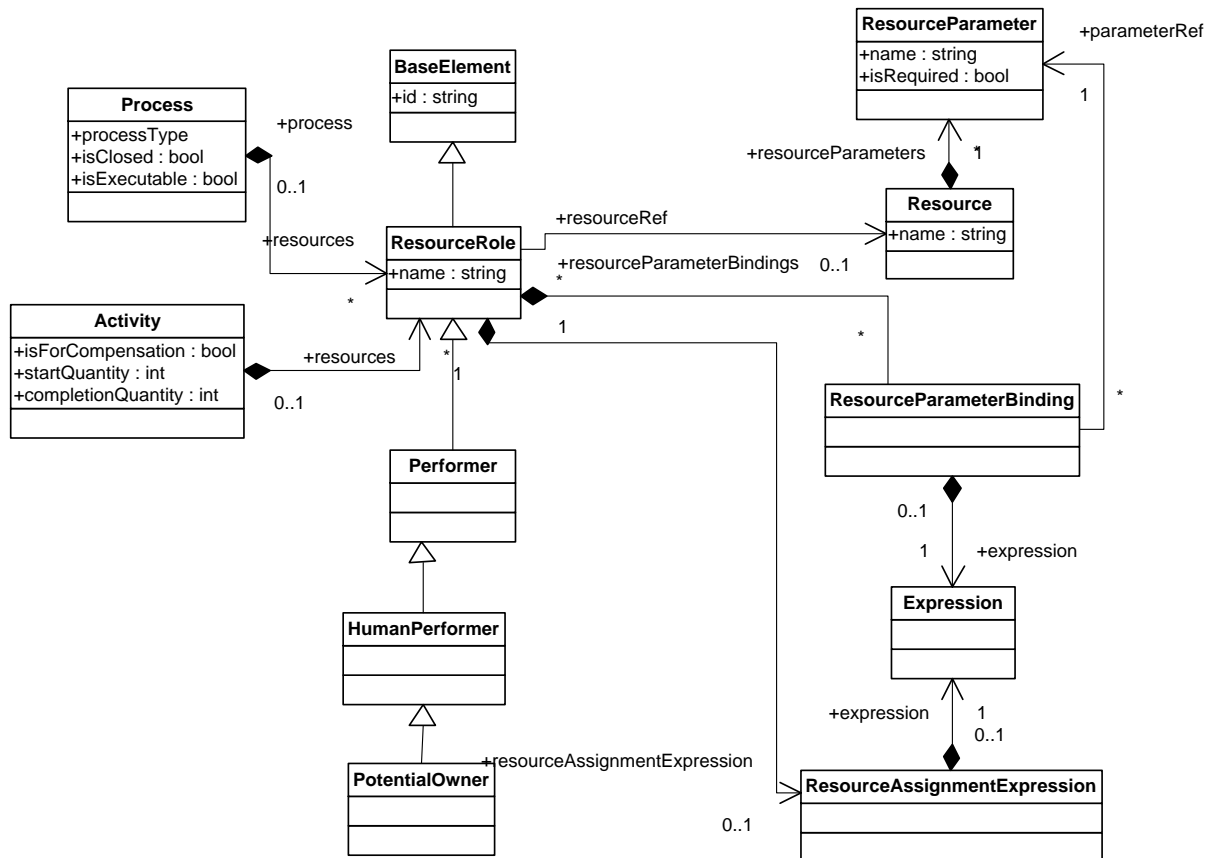
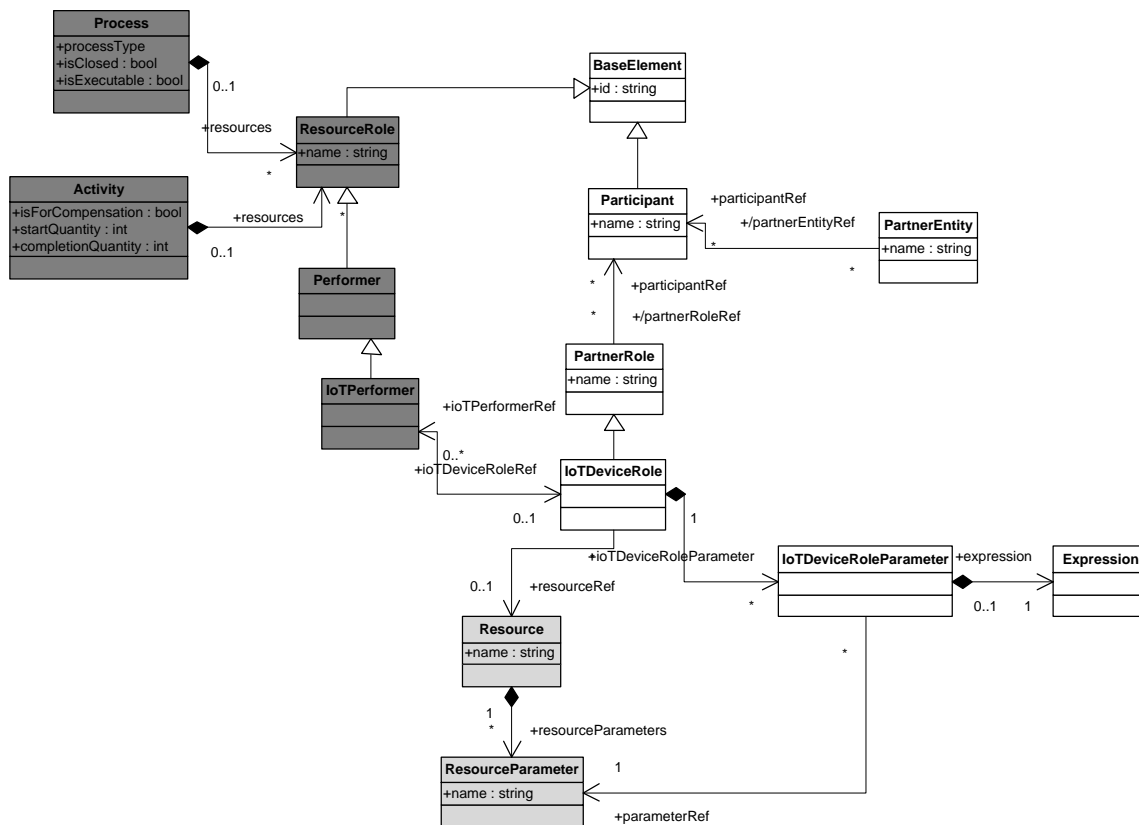


Figure 18: BPMN Resource Metamodel

First, to express the relation between activities and participants, the new subclass `IoTPerformer` is introduced to `Performer`. It has an association to the class `IoTDeviceRole` and thus indirectly references to a resource such as a temperature sensor. Second, an association between the class `IoTDeviceRole` and the `Resource` is created to refer to a resource of the resource stock. Third, an association between `IoTDeviceRoleParameter` and `ResourceParameter` is generated to reference between the real parameter property. Consequently, the resource allocation can be influenced by the in the previous section introduced requirement classes `IoTDeviceRoleParameter` and `Expression`. Figure 19 shows the current BPMN metamodel for resource allocation with the integration of the proposed IoT-specific extensions. In the diagram, the dark grey areas represent concepts that are part of the activity-centric resource allocation, while the light grey shaded areas represent the resources of the resource stock and its properties. This expansion proposal leaves open whether the IoT resources of a process should be separated from the other company resources (e.g. staff), so that this can be decided at a later project stage as part of the tool implementation.



**Figure 19: IoT Resource Metamodel Integration**

There is no icon for the resource assignment is yet. However, with the expanded lanes for the `IoTDevice` in section 5.2.2 we have created an abstraction system. This allows combining the assignment of IoT Devices and multiple activities, while the values for the parameters can be entered directly into the process model.

The following tables complete the model associations to the three classes `IoTPerformer`, `IoTDeviceRole` and `IoTDeviceRoleParameter` of the last section.

| Attribute Name                                | Description / Usage  |
|---|--|
| <b>IoTDeviceRoleRef:</b> IoTDeviceRole [0..1] | Reference to the Resource that is associated with the IoT Device and the Activities of the IoT Device. |

**Table 17: IoTPerformer Attributes and Associations**

| Attribute Name | Description / Usage |
|----------------|---------------------|
|----------------|---------------------|

|   |  |
|---|--|
| <b>ioTDeviceRoleParameter:</b><br>ioTDeviceRoleParameter [0..*] | Reference to the Parameters used to specify constraints to customize the automatic assignment of a suitable IoT Device by the resolution infrastructure. |
| <b>resourceRef:</b> Resource [0..1]                             | Reference to the Resource that is associated with IoT Device.  |
| <b>ioTPerformerRef:</b> IoTPerformer [0..*]                     | Reference to the Activities that are associated with the Resource associated with the IoT Device   |

**Table 18: IoTDeviceRole Attributes and Associations**

| Attribute Name                         | Description / Usage  |
|--|--|
| <b>parameterRef:</b> ResourceParameter | Reference to the parameter defined by the Resource.  |
| <b>expression:</b> Expression [1..1]   | The Expression that evaluates the value used to specify constraints to customize the automatic assignment of a suitable IoT Device by the resolution infrastructure. |

**Table 19: IoTDeviceRoleParameter Attributes and Associations**

Below, the XML schema of the IoT Resource extension within the context of the BPMN 2.0 standard is offered.

```

<xsd:element name="iotDeviceRole" type="tIoTDeviceRole" substitutionGroup="partnerRole"/>
<xsd:complexType name="tIoTDeviceRole">
  <xsd:complexContent>
    <xsd:extension base="tPartnerRole">
      <xsd:sequence>
        <xsd:element ref="iotDeviceRoleParameter" minOccurs="0"
          maxOccurs="unbounded"/>
        <xsd:element name="resourceRef" type="xsd:QName"
          minOccurs="0" maxOccurs="1"/>
        <xsd:element name="iotPerformerRef" type="xsd:QName"
          minOccurs="0" maxOccurs="unbounded"/>
      </xsd:sequence>
    </xsd:extension>
  </xsd:complexContent>
</xsd:complexType>

<xsd:element name="iotDeviceRoleParameter" type="tIoTDeviceRoleParameter"/>
<xsd:complexType name="tIoTDeviceRoleParameter">
  <xsd:complexContent>
    <xsd:extension base="tBaseElement">
      <xsd:sequence>
        <xsd:element ref="expression" minOccurs="1" maxOccurs="1"/>
        <xsd:attribute name="parameterRef" type="xsd:QName" use="required"/>
      </xsd:sequence>
    </xsd:extension>
  </xsd:complexContent>
</xsd:complexType>

<xsd:element name="iotPerformer" type="tIoTPerformer" substitutionGroup="performer"/>
<xsd:complexType name="tIoTPerformer">
  <xsd:complexContent>
    <xsd:extension base="tPerformer">
      <xsd:element name="iotDeviceRoleRef" type="xsd:QName" minOccurs="0"
        maxOccurs="1"/>
    </xsd:extension>
  </xsd:complexContent>
</xsd:complexType>

```

## 5.2.4 Physical Entity

In this section we consider a basic conceptual introduction of the Physical Entity into process modelling. This section assumes knowledge of the already introduced extensions IoT Activity (cf. 5.2.1) and IoT Device (cf. 5.2.2).

### What is a Physical Entity?

The IoT A reference model [58] generally defines a Physical Entity as any physical object that is relevant to a user or an application. In the following, this idea is applied to a business process. A business process describes a series of individual activities that are defined by a sequence flow and are executed accordingly. Physical Entities in a process can be relevant for one or more activities. Business processes often go across departmental and operational borders, which can be represented in a process by the help of pools and lanes. Physical entities can exist within these departmental and operational borders, but might extend beyond it. In an IoT-aware business process, one or more IoT devices are involved. Concerned IoT devices as well as organizational units can use pools and lanes all representing process participants. Physical entities within one process can be relevant for activities of one or more IoT Devices.

### What does process modelling already offer to reflect Physical Entities?

The modelling concepts participant, data object and resource have generally similarities to the Physical Entity. In the following they are examined regarding their potential usefulness.

A Physical Entity can act as a parallel process participant to other process participants such as organizational units, users, roles and IoT Devices. This process participant has the peculiarity that he is not responsible for executing process activities – meaning that he behaves somehow like a passive data object. Process participants can be modelled using pools/lanes. Pools or lanes that don't contain any process can be displayed in a collapsed view, because their content is unknown or not of interest. For example, an organizational unit (e.g. SAP Future Retail centre Switzerland) with a Physical Entity (e.g. Chinese orchid) as process participant could be presented through a pool with a lane without any content. In this case, the orchid would be a passive process participant, but without direct execution responsibility. However, this mapping of a Physical Entity brings one disadvantage: By most modelling standards two types of flows can be distinguished: the sequence flow and the messages flow. Actually, between the entity and the activity neither messages are exchanged nor a sequence flow for the process execution order needs to be defined. Thus, an association between the entity that is represented as a pool or lane and the activity of the IoT device, which is also shown as a pool or lane, would have to be created. Some notations offer general associations in order to link process artefacts with graphical elements. An association may also have a flow direction.

A data object on the other hand provides information for an activity or is information which generates an activity. Initially, a Physical Entity could be seen as an object having one or more properties. These properties can be measured or influenced by suitable IoT Devices. By measuring a property of the Physical Entity information arises and can be offered in form of a data object. The intervention in the physical state of the Physical Entity properties can happen on the basis of information which is also offered by a data object. Nevertheless, the data object itself is not used to symbolize the actual Physical Entity in the process. Rather an activity can create a data object by measuring a Physical Entity.

A Physical Entity cannot be seen as a physical resource of the process. Resources are in the process modelling solely considered to assign them activities for the process execution. As Physical Entities are indeed participants of the actual process, but do not assume any responsibility for the, there is no way to apply the concept of resources.

### Why are the existing concepts not sufficient to reflect Physical Entities?

In order to achieve automatic allocation of the IoT Devices (process participants) to Physical Entities (passive process participants) involved in the process model, a connection between the activity of the IoT device and the appropriate entity has to be created (between two process-participants). This connection must be distinguishable from other compounds (sequence flow, message flow) in the process model. There should be the possibility to execute a business process across multiple entities or entity groups. Thus, the multiple process instantiation due to entity groups would be necessary. To other restrictions on the allocation of an entity is needed that allows the resolution process entities on the requirement description reconcile and allocate. Thereby, the allocation of restrictions for entities is needed allowing to reconcile and allocate Physical Entities on their requirement description during process resolution.

For the physical entity is a physical object. It is convenient to represent the physical object through an optional object icon in the process model.

### **What are the reasons for the introduction of additional concepts for modelling Physical Entities?**

- The process modeller needs clear rules to represent Physical Entities standardized in the process model, so that an automatic resolution and execution can be guaranteed.
- The process modeller needs to be point out explicitly that a Physical Entity is added to the process model. So he can adjust the process, if this is required through the specific characteristic of the real world component.
- The relationship "acts on" / "monitors" between the Physical Entity and the IoT specific services have to be expressible in the process model.
- There must be a new way of modelling because the concept of process participant doesn't yet fully fit to the properties of a Physical Entity.
- The allocation between IoT Devices and Physical Entities and/ or their properties need to be expressible in the process model in order to use this information for process resolution and execution.
- The allocation between Physical Entities and IoT Devices to and/ or their properties need to be expressible in the process model in order to use this information for process resolution and execution.
- The allocation between IoT Devices and activities and/ or their properties need to be expressible in the process model in order to use this information for process resolution and execution.

### **What are the functional properties of a Physical Entity?**

A physical entity

- is a special process participant of the type physical entity.
- is not executable.
- does not contain any executable process activities.
- is not managed by the Business Process Execution Engine.
- can have an input association to a Sensing Activity and / or an output association from an Actuation Activity.
- can be described semantically to represent properties

### **What are the requirements to Physical Entities within the process model?**

It is presented a short list of properties that can be formulated as requirements with values to select an appropriate Physical Entity matching its assigned activities of the IoT Devices. The full list of these properties will be based on the descriptions which the formulated requirements are reconciled. At this point we refer to IoT-A D 2.1, which deals in detail with such descriptions.

The following non-exhaustive aspects are considered to be beneficial in the process context:

- Entity ID
- Entity Type
- Manufacturer
- Location of manufacture
- Position of Entity
- Quality of Entity
- Entity States

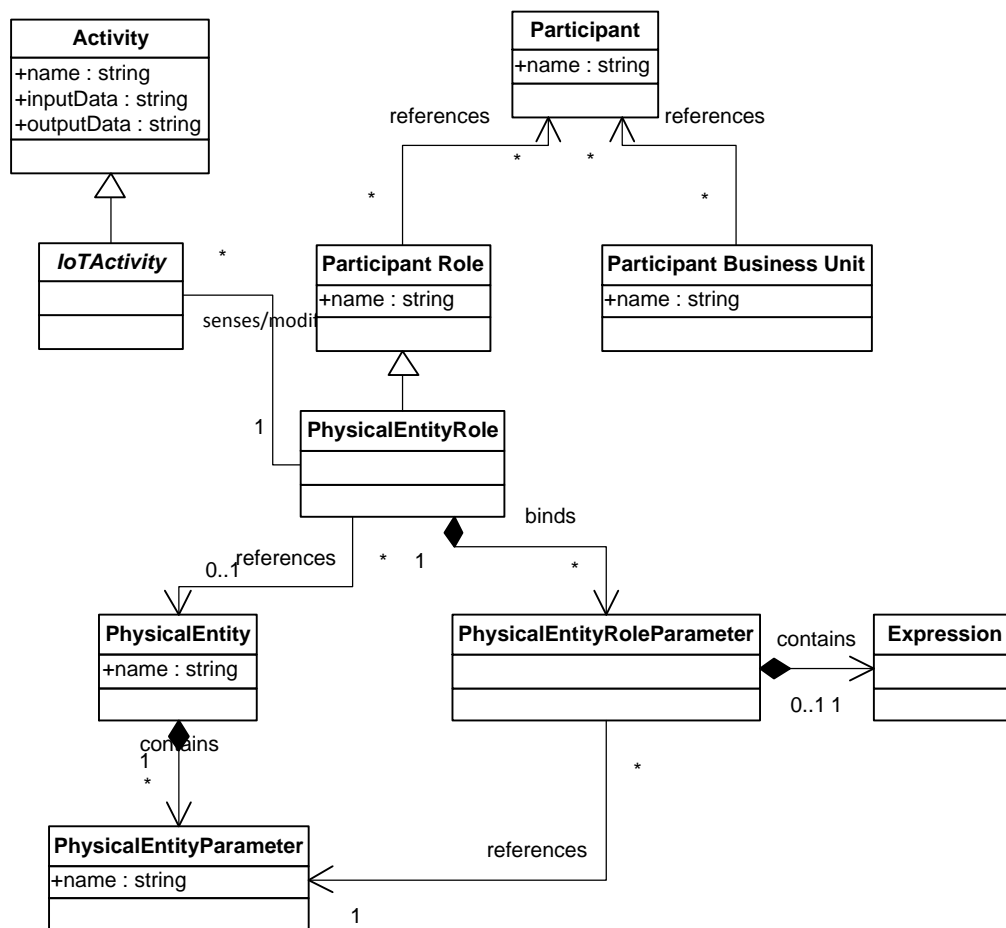
### **BPMN independent implementation**

A process participant is usually responsible for executing the assigned activities. A Physical Entity is a special process participant who doesn't perform any activities and doesn't include process flow elements in his pool or lane. A participant of type Physical Entity would be, for example, an orchid, whose state temperature is measured by a Sensing Activity of a temperature sensor.

To find a suitable entity, a sensor or an activity during process resolution, the automatic selection can be influenced by formulating requirements to a Physical Entity within the process model. For introducing the Physical Entity to BPMN the subclass `PhysicalEntityRole` is added as a participant role to the metamodel. When modelling a business process, different entities might be available in stock, which can be bound to the entity of the model during the resolution phase. To influence the choice of the desired entity, `PhysicalEntityRoleParameter` can be specified. By the class `Expression` for each parameter



a value can be entered, which restricts the entity selection. This results a set of parameters with optional values for each process participant of type Physical Entity. To reference between the requirements and the actual properties of the class `PhysicalEntity` an association between `PhysicalEntityRoleParameter` and `PhysicalEntityParameter` is introduced to the metamodel. In addition, the `PhysicalEntityRole` references to a `PhysicalEntity` of the entity stock, wherein this reference is a result of the resolution infrastructure. In order to further express the relation senses or modifies between the IoT Activity and the Physical Entity, `PhysicalEntityRole` references subclass `IoTActivity`.

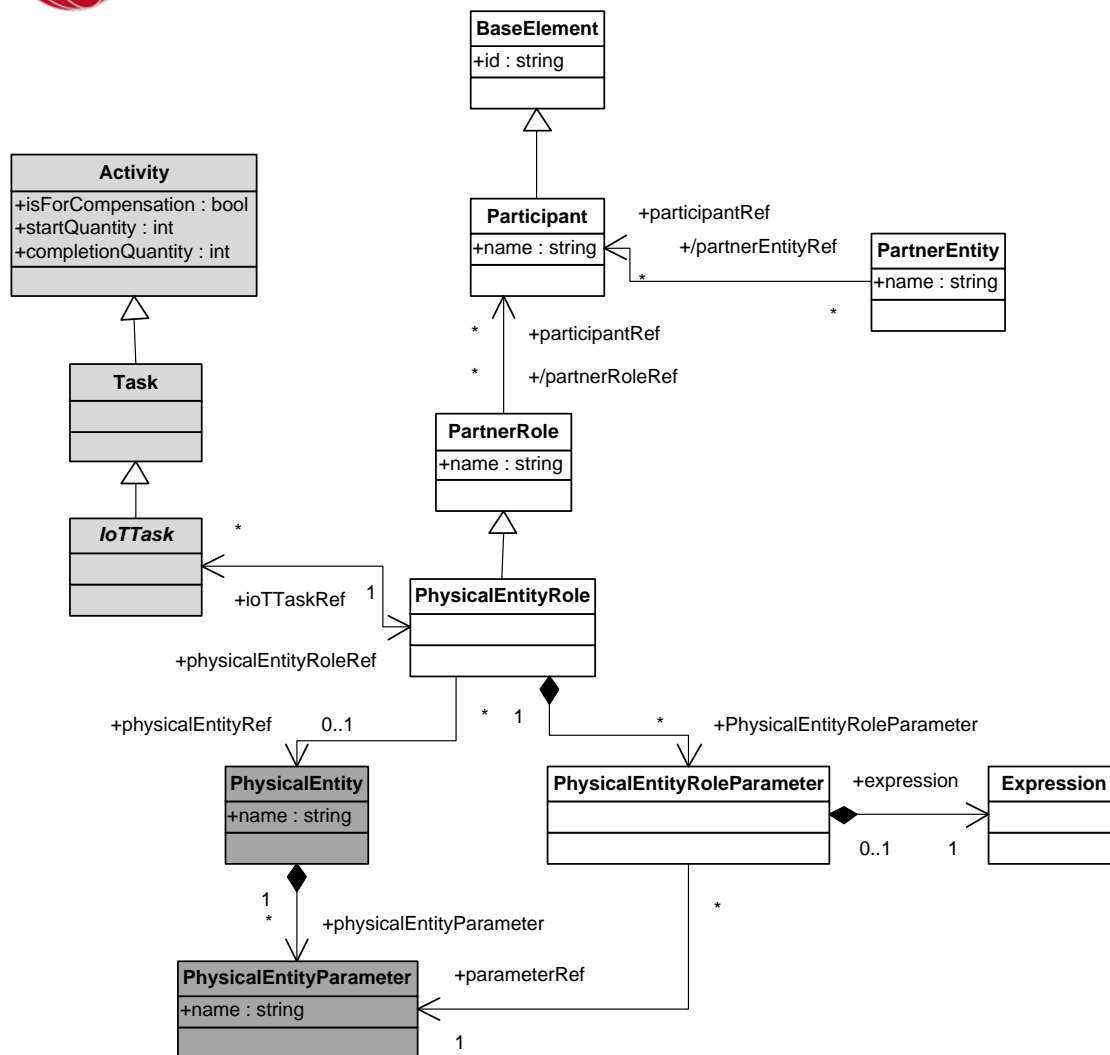


**Figure 20: Physical Entity Metamodel**

### Technical implementation of the enhancements in BPMN

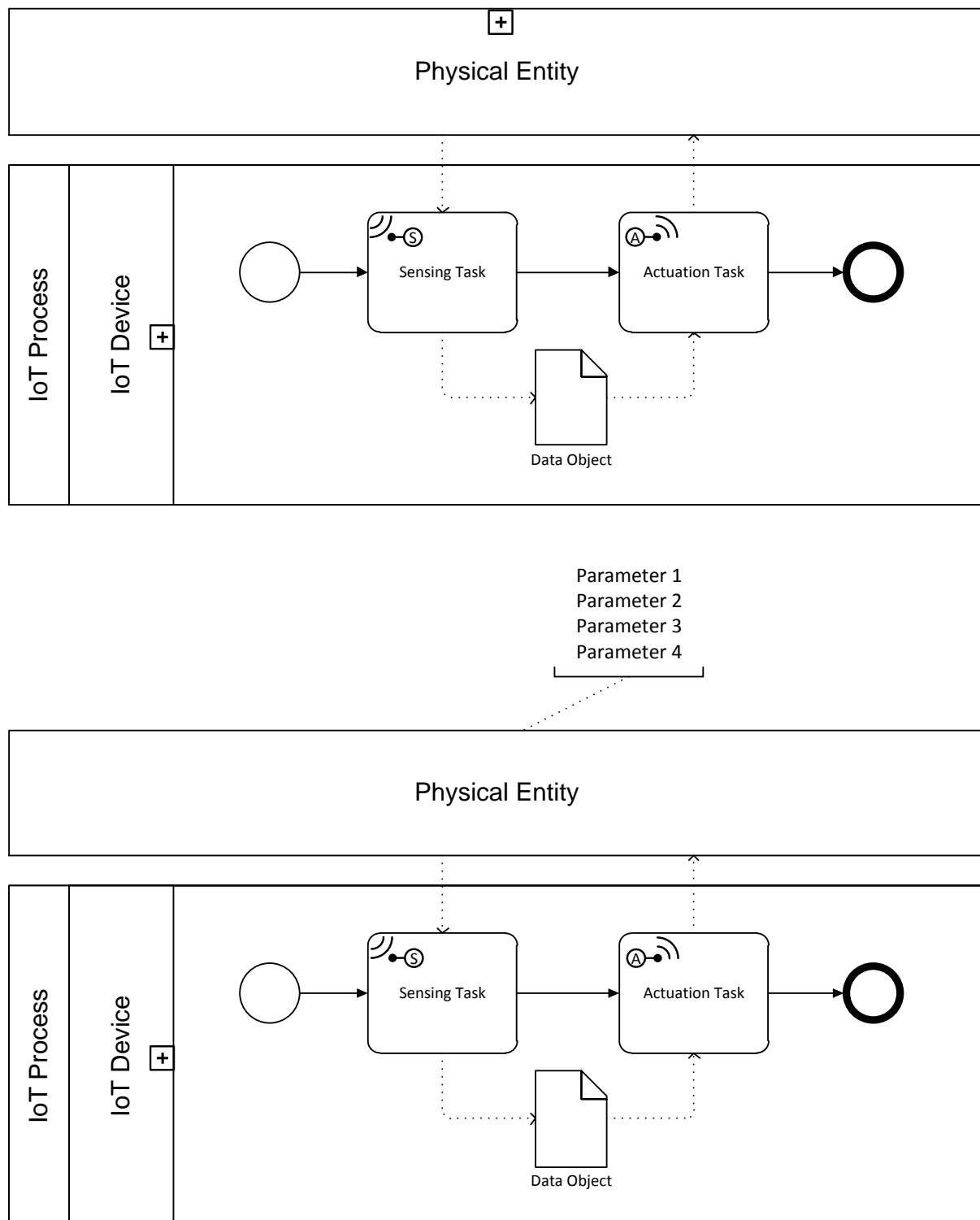
Figure 21 shows the BPMN metamodel with the integration of the proposed extension for the Physical Entity. In the diagram, the dark grey areas represent concepts that represent entities of the entity-stock and their properties, while the light grey areas represent concepts belonging to the IoT-specific activities. The bright areas represent all classes that are part of the passive participant Physical Entity. The suggested extensions are directly integrated in the BPMN Metamodel. As in the BPMN independent extension the class `PartnerRole` is completed by introducing the subclass `PhysicalEntityRole` with its relation to the abstract class `IoTTask`. Further extensions are analogously introduced to the BPMN independent extensions. This expansion proposal leaves open how the stock of Physical Entities is technically implemented, or how the resolution of the process model is realized. These issues can be decided if not yet included in D2.3, at a later project stage as part of the tool implementation.





**Figure 21: Physical Entity BPMN Integration**

To enter the requirements directly into the process model the existing BPMN elements Pool / Lane are graphically extended. Figure 22 shows a process pool called “IoT Process” containing two process participants. The process participants are the participants “IoT Device” of type IoT Device and the participant “Physical Entity” of Type Physical Entity. A plus symbol in the collapsed version of the lane indicates that the Physical Entity can be further specified. In the extended version, the modeller can specify the required values of the Entities’ requirements. Since the Physical Entity is a passive process participant, which does not have any flow elements, it is presented by a lane with horizontal lettering. The lane itself cannot be extended to its elements and from a modellers’ view it acts like a black box. The model association to/from the activity of the IoT Device has a direction showing orientation of the information flow. While an Actuation Activity with a connected Physical Entity acts as information sink, a Sensing Activity with a connected Physical Entity acts as information source. An icon within the lane of the Physical Entity is optional and can be displayed behind the lettering on the entity.



**Figure 22: Collapsed and Extended Physical Entity Object**

The tables below present all attributes and the newly added associations for the Physical Entity. For a complete association overview of *Participant*, please reference [14].

| Attribute Name  | Description / Usage   |
|---|---|
| <b>iotTaskRef:</b> IoTTask [0..*]   | Reference to the IoT Tasks that are associated with the Physical Entity.  |
| <b>physicalEntityRef:</b> PhysicalEntity [0..1]                           | Reference to the Physical Entity of the entity pool.  |
| <b>physicalEntityRoleParameter:</b><br>PhysicalEntityRoleParameter [0..*] | Reference to the Parameters used to specify constraints to customize the automatic assignment of a suitable Physical Entity by the resolution infrastructure. |

**Table 20: PhysicalEntityRole Attributes and Associations**

| Attribute Name                             | Description / Usage  |
|--|--|
| <b>entityParameterRef:</b> EntityParameter | Reference to the parameter defined by the Entity.  |
| <b>expression:</b> Expression              | The Expression that evaluates the value used to specify constraints to customize the automatic assignment of a suitable Entity by the resolution infrastructure. |

**Table 21: PhysicalEntityRoleParameter Attributes and Associations**

| Attribute Name  | Description / Usage  |
|---|--|
| <b>physicalEntityParameter:</b><br>PhysicalEntityParameter [0..*] | Reference to the parameter defined by the Physical Entity. |

**Table 22: PhysicalEntity Attributes and Associations**

| Attribute Name                                   | Description / Usage  |
|--|--|
| <b>physicalEntityRoleRef:</b> PhysicalEntityRole | Reference to the Physical Entity that is associated with the IoT Task. |

**Table 23: IoTTask Attributes and Associations**

Below, the XML schema of the Physical Entity within the context of the BPMN 2.0 standard is offered.

```

<xsd:element name="physicalEntityRole" type="tPhysicalEntityRole" substitutionGroup="partnerRole"/>
<xsd:complexType name="tPhysicalEntityRole">
  <xsd:complexContent>
    <xsd:extension base="tPartnerRole">
      <xsd:sequence>
        <xsd:element ref="physicalEntityRoleParameter" minOccurs="0"
          maxOccurs="unbounded"/>
        <xsd:element name="physicalEntityRef" type="xsd:QName"
          minOccurs="0" maxOccurs="1"/>
        <xsd:element name="iotTaskRef" type="xsd:QName"
          minOccurs="0" maxOccurs="unbounded"/>
      </xsd:sequence>
    </xsd:extension>
  </xsd:complexContent>
</xsd:complexType>

<xsd:element name="physicalEntityRoleParameter" type="tPhysicalEntityRoleParameter"/>
<xsd:complexType name="tPhysicalEntityRoleParameter">
  <xsd:complexContent>
    <xsd:extension base="tBaseElement">
      <xsd:sequence>
        <xsd:element ref="expression" minOccurs="1" maxOccurs="1"/>
      </xsd:sequence>
      <xsd:attribute name="parameterRef" type="xsd:QName" use="required"/>
    </xsd:extension>
  </xsd:complexContent>
</xsd:complexType>

```

```

</xsd:complexContent>
</xsd:complexType>

<xsd:element name="physicalEntity" type="tPhysicalEntity" substitutionGroup="rootElement"/>
<xsd:complexType name="tPhysicalEntity">
  <xsd:complexContent>
    <xsd:extension base="tRootElement">
      <xsd:sequence>
        <xsd:element ref="physicalEntityParameter" minOccurs="0"
          maxOccurs="unbounded"/>
      </xsd:sequence>
      <xsd:attribute name="name" type="xsd:string" use="required"/>
    </xsd:extension>
  </xsd:complexContent>
</xsd:complexType>

<xsd:element name="physicalEntityParameter" type="tPhysicalEntityParameter"/>
<xsd:complexType name="tPhysicalEntityParameter">
  <xsd:complexContent>
    <xsd:extension base="tBaseElement">
      <xsd:attribute name="name" type="xsd:string"/>
    </xsd:extension>
  </xsd:complexContent>
</xsd:complexType>

<xsd:element name="iotTask" type="tIotTask" substitutionGroup="flowElement"/>
<xsd:complexType name="tIotTask">
  <xsd:complexContent>
    <xsd:extension base="tTask">
      <xsd:sequence>
        <xsd:element name="physicalEntityRoleRef" type="xsd:QName"
          minOccurs="1" maxOccurs="1"/>
      </xsd:sequence>
    </xsd:extension>
  </xsd:complexContent>
</xsd:complexType>

```

### 5.2.5 Real World Data Object / Store

In this section we consider a basic conceptual introduction of the Real-World Data Object and Data Store into process modelling. This section is partly based on the previously introduced enhancements of the sections 5.2.1, 5.2.2 and 5.2.4.

#### What is a Real-World Data Object /Store?

A Real-World Data Object represents a temporarily stored data object of a running process instance, which were generated by an IoT Device measuring entity properties (cf. 5.2.1 and 5.2.2). The values of the data element are visible to other process participants and do not exist beyond the lifetime of a process. The Real-World Data Object has IoT specific properties such as the varying quality of information. Besides a Real-World Data Object, a business process can contain a Real-World Data Store. In contrast to the Real-World Data Object, the Real-World Data Store represents persistent data and exists beyond the lifetime of a process instance. The Real-World Data Store may be queried or updated by participants of the process, as well as participants from outside of the process.

#### What does process modelling already offer to reflect Real-World Data Objects / Stores?

So far, most process modelling notations know both data objects and data stores that can be linked via associations with specific data activities, sequence flows or events. In so doing a special data flow is created in the process model. Typically, the data flow can be of non-directional and directional nature. A data association, which is directed to the data object / store, symbolizes an update operation, while a data association, which is directed away from the data object / store, symbolizes a query operation.

#### Why are the existing concepts not sufficient to reflect Real-World Data Objects / Stores?

Some process modelling notations allow to document data objects in specially designated elements of an XML structure. In this field a lot of information could be integrated. The field may either contain a value like “real-world”, or it could be filled with links to descriptive external documents. Nevertheless, this documentation possibility doesn’t offer associated graphical elements that are part of the process diagram. Furthermore, some modelling notation offer a free text annotation in the chart itself such as an annotation text “real-world” to a data object / store. This annotation is typically represented by a

non-directed association between the annotated text and the graphical element. The main disadvantage of using this representation is that no external information documents can be bound to the annotation. Furthermore, there is no single form of expression for “real-world” that can be recognized and preserved for a particular purpose from the process modellers, the resolution infrastructure or the execution engine.

### **What are the reasons for the introduction of additional concepts for modelling Real-World Data Objects / Stores in a business process?**

- The modeller needs a way to express that a temporary data set has an unknown and variable “real-world” quality of the information. Thus, the process can be adjusted if necessary.
- The modeller needs a way to express that a persistent set has an unknown and variable “real-world” quality of the information. Thus, further processes that access to such data can be adjusted if necessary.
- The modeller needs a way to recognize that referenced data or stores in a process are of the type “real-world”.
- The process modelling environment needs a way to generate process indicators of the process quality, in which the quality of real-world information could be a characteristic value.
- The resolution infrastructure needs a way to detect if referenced data is of type “real-world” to guarantee the resolution with for example a minimal level of quality of the complete process.
- The execution engine must have a way to detect if referenced data of an IoT Device is of type “real-world” to start in case specific automatic error and exception handling.

### **What are the functional properties of a Real-World Data Object / Store?**

- The process modeller needs a standardized possibility to graphically mark data in a process model as “real-world”.
- The process modeller needs a standardized possibility to optionally link detailed description of the data by a reference document.
- The process modeller needs to directly access and if necessary to change the data documentation out of the process modelling environment.

### **What are the requirements to a Real-World Data Object / Store within the process model?**

In [12] the metrical concept Quality of Information (QoI) is introduced defined as “Any metadata that characterizes sensor or context information in such a way that it can be used to infer the reliability of the received information”. In [5] we discuss how different metrical models like the QoI on different process levels influence one another.

Information in business processes can be transmitted in the form of data objects and data stores. Real-world Data Objects / Stores have the characteristic property of “real-world” information that is important for the reasons stated above. These qualitative characteristics include according to [12] adjusted to the in IoT -A terminology, for example:

- Traceability properties
  - o IoT Device / IoT Device Type: Which IoT Device has generated the data information?
  - o Physical Entity / Entity Type: About which Physical Entity does the data provide information?
  - o Owner: Who is the legal entity of the data?
  - o Network scope: From which domain has the data originated?
- Reliability properties: How reliable is the data? (Confidence, Confusion Matrix, Resolution etc.)
- Spatial properties
  - o Origin location: From which location does the data come from?
  - o Physical Entity scope: About what location range does the data provide information on?
- Temporal properties
  - o Temporal scope: From which time is the data?
  - o Delay time: Which time delay does the information have?
  - o Measurement: In what unit of time, are the time variables measured?

### **BPMN independent implementation**

Data of the subclass Data are process elements of the class ProcessElement, which are used for representing information which flows through the process. The class Data can consist of a DocumentReference and a DataReference and has the attributes name and isRealWorld. While the DocumentReference accesses the DataDescriptionItem, the Data Reference accesses the actual data record

DataItem. The class data inherits the properties of their two children DataObject und DataStore presenting process information at run-time of a process and beyond. An association is made up of two processing elements, which binds them together, that can be a non-directional, a forward or a backward directed association.

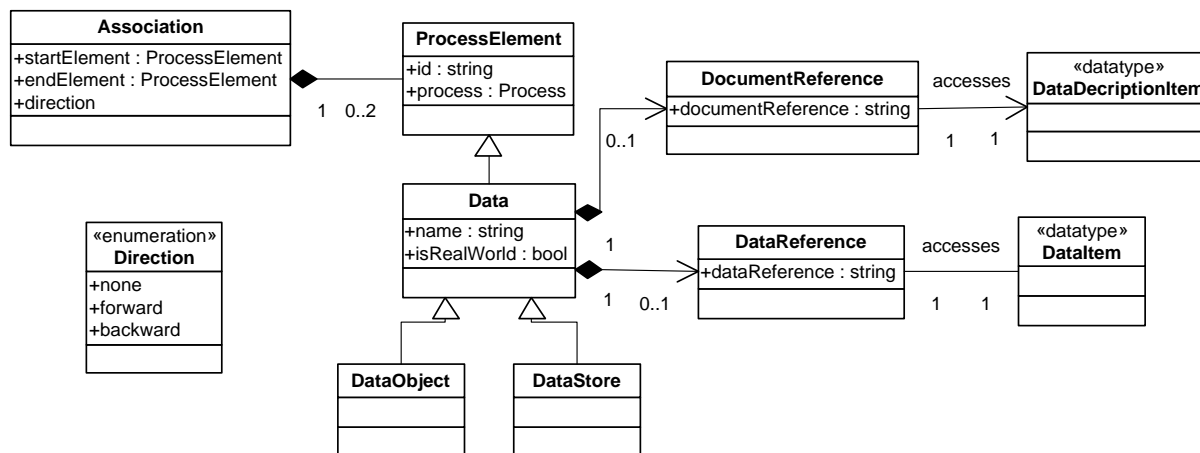
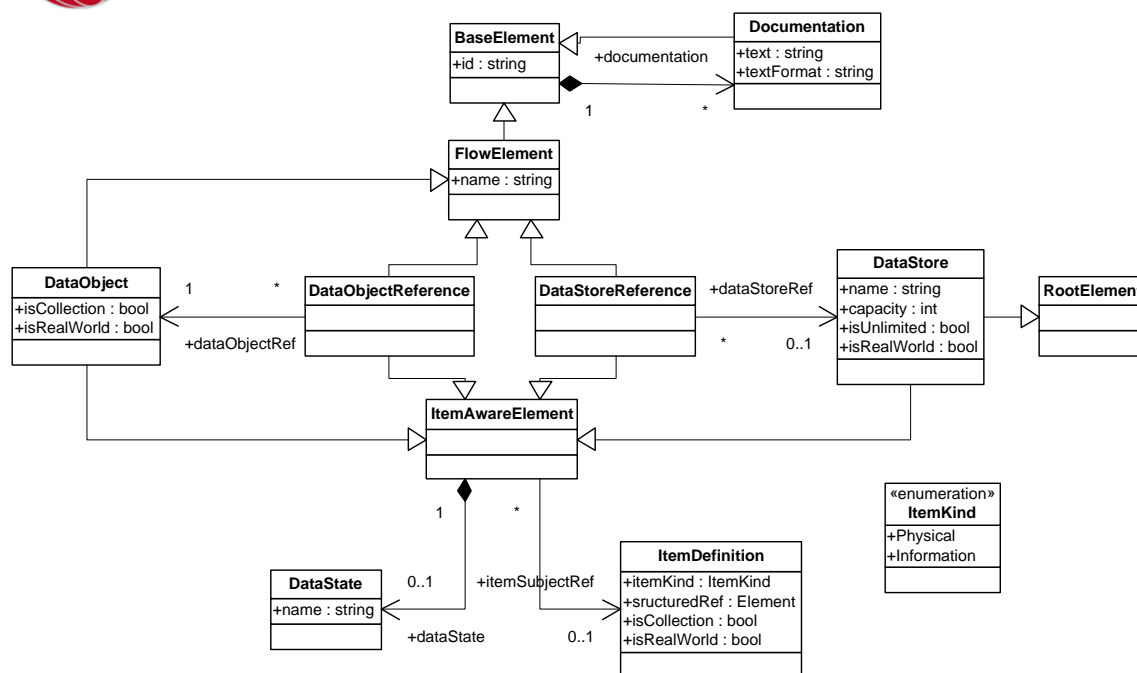


Figure 23: Real-World Data Metamodel

### Technical implementation of the enhancements in BPMN

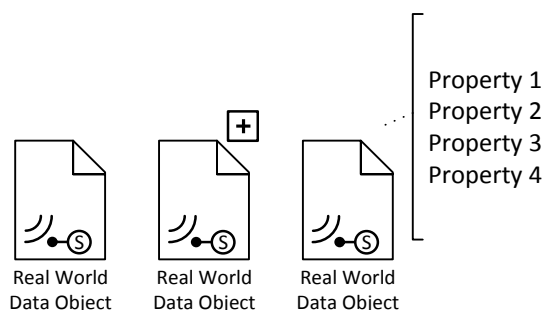
Below, the metamodel of the Real-World Data Object / Store is transferred to the modelling notation BPMN 2.0. In BPMN 2.0, the two most commonly used constructs for modelling data are the Data Object and the Data Store, which we similarly have already introduced above. They come with well-defined life cycles with restricted access mechanism. Both data elements are an item-aware element, a so-called subject to save various BPMN elements during process execution. As data objects are always part of a process or sub-process data stores can exist beyond a process instance. Data object / store references can be specified in various data states to different process points in time, so that the same object / store can be used multiple times in the diagram. Through the class DataObject / StoreReference the same data object / store can be referenced at different process points. To establish the "real-world" property of data objects and stores in BPMN, the two classes DataObject and DataStore are extended by the Boolean attribute `isRealWorld`. Analogously, the class `itemDefinition` is expanded by the attribute `isRealWorld` in order to use this property for referenced and not referenced definitions. The classes DataObject and DataStore can contain a documentation element like many BPMN elements, which can be filled with links to external documents. This DocumentationElement belongs is part of the BPMN descriptive subclass and thus is supported by any compliant tool. For the semantic description of "real-world" information, we propose to fill this item with a link to the external descriptive document.



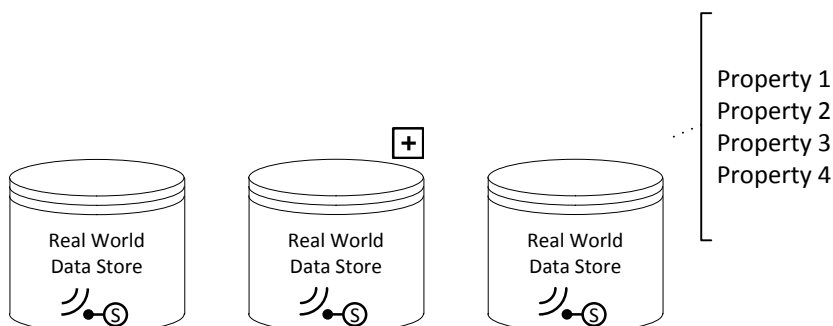


**Figure 24: Real-World Data Object / Store Integration**

The introduced “real-world” attribute and the documentation possibility don’t have any graphical elements associated in BPMN. In Figure 25 and Figure 26 we propose a symbolism for the Real-World Data Object and the Real-World Data Store. It can be used as an optional ability to link to an external describing document. If it is linked to such a semantic description, it is distinguished between collapsed and extended Real-World Data Object / Store. The extended view supports the direct modification of the information properties in the process model.



**Figure 25: Natural, Collapsed and Extended Real-World Data Object**



**Figure 26: Natural, Collapsed and Extended Real-World Data Store**

Table 24 presents the new attribute for the classes Data Object, Data Store and Item Definition. For a complete association overview of these classes, please reference [14].

| Attribute Name                       | Description / Usage  |
|--------------------------------------|--|
| <b>isRealWorld</b> : boolean = false | Defines if the <b>Data Object</b> / <b>Store</b> represent a real world element. It is needed when no itemDefinition is referenced. If an itemDefinition is referenced, then this attribute <b>MUST</b> have the same value as the isRealWorld attribute of the reference itemDefinition. The default value for this attribute is false. |

**Table 24: Real-World Data Object / Store Attributes and Associations**

Below, the XML schema of the real-world Data Object and Data Store extension within the context of the BPMN 2.0 standard is offered.

```

<xsd:element name="dataObject" type="tDataObject" substitutionGroup="flowElement"/>
<xsd:complexType name="tDataObject">
  <xsd:complexContent>
    <xsd:extension base="tFlowElement">
      ...
      <xsd:attribute name="itemSubjectRef" type="xsd:QName"/>
      <xsd:attribute name="isCollection" type="xsd:boolean" default="false"/>
      <xsd:attribute name="isRealWorld" type="xsd:boolean" default="false"/>
    </xsd:extension>
  </xsd:complexContent>
</xsd:complexType>

<xsd:element name="dataStore" type="tDataStore" substitutionGroup="rootElement"/>
<xsd:complexType name="tDataStore">
  <xsd:complexContent>
    <xsd:extension base="tRootElement">
      ...
      <xsd:attribute name="name" type="xsd:string"/>
      <xsd:attribute name="capacity" type="xsd:integer"/>
      <xsd:attribute name="isUnlimited" type="xsd:boolean" default="true"/>
      <xsd:attribute name="itemSubjectRef" type="xsd:QName"/>
      <xsd:attribute name="isRealWorld" type="xsd:boolean" default="false"/>
    </xsd:extension>
  </xsd:complexContent>
</xsd:complexType>

<xsd:element name="itemDefinition" type="tItemDefinition" substitutionGroup="rootElement"/>
<xsd:complexType name="tItemDefinition">
  <xsd:complexContent>
    <xsd:extension base="tRootElement">
      <xsd:attribute name="structureRef" type="xsd:QName"/>
      <xsd:attribute name="isCollection" type="xsd:boolean" default="false"/>
      <xsd:attribute name="itemKind" type="tItemKind" default="Information"/>
      <xsd:attribute name="isRealWorld" type="xsd:boolean" default="false"/>
    </xsd:extension>
  </xsd:complexContent>
</xsd:complexType>

```

### 5.2.6 Mobility Aspect

In this section we consider a basic conceptual introduction of the mobile aspect into process modelling. This section is based on the sections IoT Activity (cf. 5.2.1), Physical Entity (cf. 5.2.4) and IoT Device (cf. 5.2.2).

#### What does mobility mean in the IoT?

There are a growing number of mobile devices being involved in business processes. Mobility may concern process participants such as the IoT Device or the Physical Entity, which thereby influence the activities for which they are responsible for. This aspect relates to IoT activities as well as normal activities. As a consequence, processes or sub-processes are affected by the mobile behaviour of its participant's (i.e. their change of location). First, it requires a general definition of the terminology mobility in business processes.

After Valiente Haiden [52] a mobile process is characterized as follows:

1. Position of some participant change in time and so it cannot be accessed in the same way as participant whose position is stable. In addition to it the position of such participants is not possible to find without communication.
2. Decision, coordination and management depend on location of process participant – some decision may for example depend on participant's position and other location dependent context.

While [52] and [33] so far concentrate on suggesting extensions for the mobility aspect of participants and events, we come up with suggestions for a supplemental integration for mobile IoT Physical Entities and location-dependent activities. Below we consider the mobility aspect at different levels of a business process:

- Mobile process: A business process is called mobile in case process decisions or activities depend on the location of one or more process participant or of involved physical entities and these participants might change their location during process time.
- Mobile participant: A mobile user can change its location over time and must therefore be treated differently than process participants whose location is stable during modelling resolution and execution phase. This relates in particular to the participants types IoT Device and IoT Physical Entity. .
- Process decisions, depending on the location or other location based context of the participant or the Physical Entity:
  - o Location based Events: A location-based event expresses that something location-based occurs. It marks a concrete point of time and in contrast to an activity it does not express a time frame. Considering a location-based event the aspects cause and activator can be distinguished. The condition "change location of a process participant" becomes true. There are usually three types of conditions: start event, intermediate event and end event. Following [33] these types of event can be distinguished in 3 different positional events. :
    - Location achieved event: Occurs if mobile participant (i.e. the Physical Entity Orchid) has reached a requested location.
    - Location update event: Occurs if there is for the process there is a significant change of participant's position.
    - Conditional location event: Occurs if there is a general positional change, i.e. a Physical Entity enters in some region.

At this point we refer to [33] and the results of WP 2's Complex event processing.

- Activities in the context of mobile processes:
  - o Location based Activity: A location-activity is an activity that depends on the location of a further process-participant who is not already responsible for the execution of the activity. This process can be for example a participant such as a IoT device involved in the process or a Physical Entity, that have no execution responsibility of the designated activity.
  - o Mobile Activity: An activity is and mobile if a mobile device is responsible for its execution. An example of a mobile service is the interface to a temperature-sensing activity of a temperature sensor's cell phones.
  - o If this activity, as proposed in section 5.2.2, is modelled as an activity of the mobile Cell Phone modelled, the activity doesn't specifically need to be labelled as "mobile". The IoT Device already has the mobile property and therefore all activities for which it is responsible for.

### **What does process modelling already offer to reflect mobility in the IoT?**

In the standard of some process notations such as BPMN, UML or EPC, it is possible to describe the affected processes and process artefacts with annotations. Using this ability it would possible to annotate a key word like "mobile". The same approach would be possible to mark location-based activities, events or decisions of a process.

In [33] some mobility concepts are suggested to support the development of genuine mobile IT and business solutions.

- 1.) Location based events: In order to support mobility inside business process management it is suggested to enrich the process notation with three location based event types.
- 2.) Mobile participant: For marking or accentuating the mobility of a participant it is suggested to use an own pool with naming the type of the participants mobility.

### **What are the reasons for the introduction of additional concepts reflecting mobility in the IoT in business processes?**

Mobility is an important feature in the IoT. A growing number of mobile devices are equipped with various sensors (e.g. magnetic sensor, position sensor, light sensor, GPS receiver, and accelerometer). These mobile devices allow users to run business processes on wireless mobile communication technology. When modelling business processes the mobility aspect is so far not considered. For modellers, process execution engine or resolution infrastructure, it is important to mark explicitly the mobile elements of a business process. Thus, in all process phases the mobile nature of some elements can be handled. In particular, it is conceivable to introduce a dedicated automatic operation handling the frequent registration of a mobile participant to the resolution infrastructure. Mobility and accessibility are two closely related properties. In addition, it is important to note location-based decisions and activities as such. To deal with these typical features of mobile business processes in a lightweight and standardized way within the three process phases, we consider the introduction of the two solid IoT properties "mobile" and "location-based" as relevant. Consequently, we combine the approaches of [33] and [52] and integrate them in a mobility concept for process modelling.

### **What are the functional properties of the concept components reflecting IoT in business processes?**

- Mobile process: The process model needs a standardized possibility to graphically and technically express a process as mobile for all three process phases, in case one or more process participants such as IoT Device or Physical Entity are mobile.
- Mobile participant: The process model needs a standardized possibility to graphically and technically express a mobile participant such as an IoT Device or a Physical Entity as mobile for all three process phases.
- Process decisions:
  - o Location based events: The process model needs a standardized possibility to graphically and technically express that an event that is not part of a mobile participant can occur if the location of a process participant such as an IoT Device or a Physical Entity changes.
- Activity:
  - o Location based activity: The process model needs to present activities in graphically and technically way that are dependent on the location of a dedicated process participant such as an IoT Device or a Physical Entity.
  - o Mobile activity: The process model needs to present activities in exclusively in a technically way that are mobile because a mobile participant such as an IoT Device is responsible for their execution.

### **BPMN independent implementation**

To introduce mobility and location-based markers to the process model, some classes of the metamodel need to be extended. While extending the classes Process, Entity and Participant with a Boolean variable `isMobile`, the classes Activity, Gateway and Event are extended by a Boolean variable `isLocationBased`. The class SubProcess does not need supplements because it inherits all attributes of the class Activity.

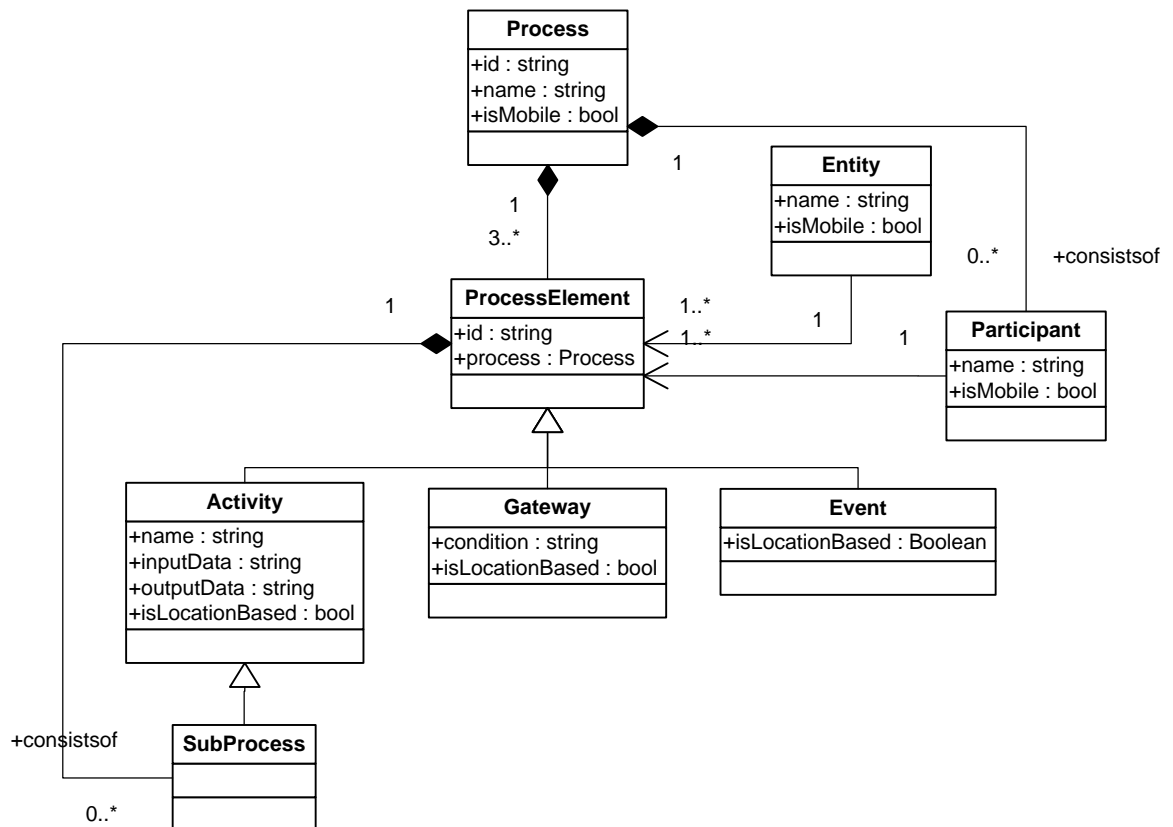


Figure 27: Mobility Class Diagram

### Technical implementation of the enhancements in BPMN

First, a general symbolism is proposed for the properties mobile and location-based. The marker for a mobile artefact or process (cf. Figure 28) must be a divided right arrow, standing for the mobility from one to another place of the affected component. The marker for a location-based artefact (Figure 28 b) must be a on the head rotated drop, which in the Internet generally is applied for marking places.

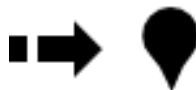


Figure 28: Mobile Property (a), Location-based Property (b)

### Mobile process

In BPMN, a Process is a callable Element behaving as a container for flow elements and allowing to be referenced by other processes. Transferring the metamodel to BPMN, in Figure 29 the class Process is extended accordingly.

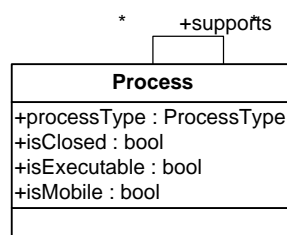
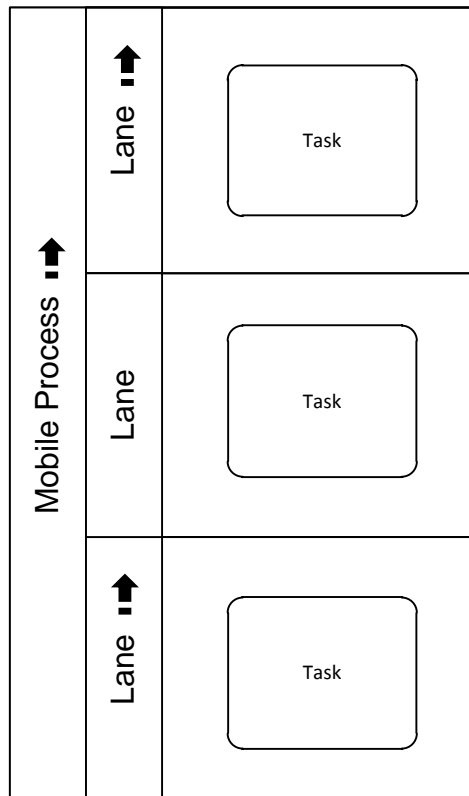


Figure 29: BPMN Class Diagram Mobile Process

To distinguish a mobile process from a none-mobile process, the associated graphical BPMN element “Pool” is extended by an additional marker. In Figure 30 we propose a symbolism for a mobile process “Mobile Process” and two mobile Lanes.



**Figure 30: Mobile BPMN Example Process**

The following table presents the new attribute for the class *Process*. For a complete association overview of *Process* please reference [14].

| Attribute Name                         | Description / Usage  |
|--|--|
| <b>isMobile:</b> boolean [0..1]= false | An optional Boolean specifying whether the process is mobile. A mobile Process is a private Process that has been modelled for the purpose of being executed over one or more mobile process participants to the execution semantics, which will be defined in D2.4. A non-mobile process is a private Process that has static participants that don't need a specific accessing mechanism. The default value for this attribute is false. |

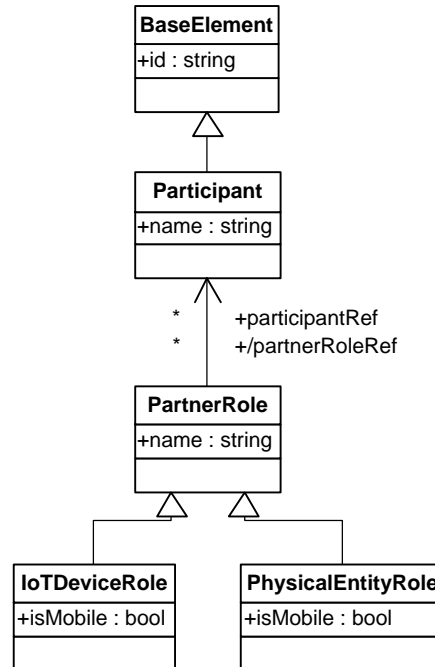
**Table 25: Process Attributes and Associations**

Below, the XML schema of the mobile components within the context of the BPMN 2.0 standard is offered.

```

<xsd:element name="process" type="tProcess" substitutionGroup="rootElement"/>
  <xsd:complexType name="tProcess">
    <xsd:complexContent>
      <xsd:extension base="tCallableElement">
        <xsd:sequence>
          ...
        </xsd:sequence>
        <xsd:attribute name="processType" type="tProcessType" default="None"/>
        <xsd:attribute name="isClosed" type="xsd:boolean" default="false"/>
        <xsd:attribute name="isExecutable" type="xsd:boolean"/>
        <xsd:attribute name="isMobile" type="xsd:boolean"/>
        <xsd:attribute name="definitionalCollaborationRef" type="xsd:QName" use="optional"/>
      </xsd:extension>
    </xsd:complexContent>
  </xsd:complexType>
  
```

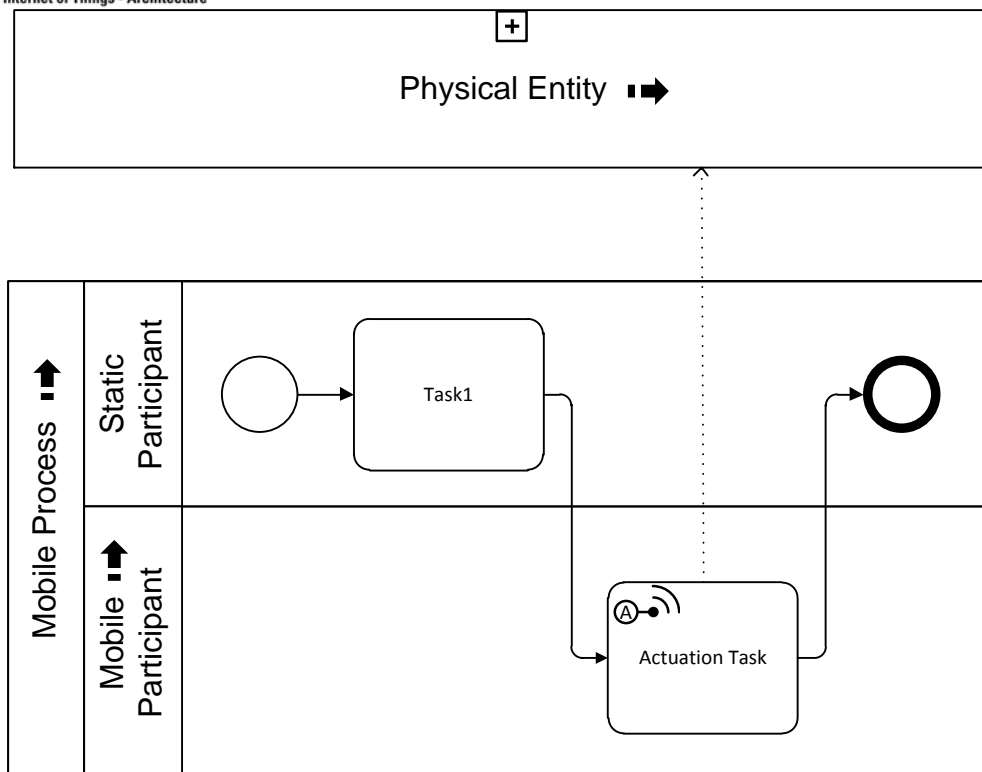
A Participant represents a specific PartnerEntry or a more general PartnerRole and is often responsible for the execution of a process enclosed in a pool [14]. In the IoT, we introduced in section 5.2.4 and 5.2.2 the new participant role types: IoT Device and Physical Entity. Since both could be mobile entities, their classes need to be extended by an additional property `isMobile`. As not all BPMN PartnerRole might have the property `isMobile` the mobile property is kept to the individual sub-classes.



**Figure 31: BPMN 2.0: Participant extension**

In a process lane the roll of the IoT Participant is entered. Since mobility is a property of the participant itself and lanes are the closest graphical representation form of IoT participants in BPMN models, these elements are extended by mobile property marker. In order to keep an analogy to the mobile Process, the same divided right arrow symbol is used again to mark the corresponding lane, if the `isMobile` property is set for a participant. Figure 32 illustrates an example process with one mobile participant within a mobile Process and having a Task 2 with an association to a mobile Physical Entity.





**Figure 32: Mobile Participant Collaboration Diagram**

The following table present the additional attribute of the two classes IoTDeviceRole and PhysicalEntityRole. For a complete overview of Participant, please reference [14].

| Attribute Name           | Description / Usage  |
|--------------------------|--|
| isMobile: Boolean [0..1] | An optional Boolean specifying whether the process Participant is mobile. A mobile participant is a participant changing its location over time. Static participants don't need a specific accessing mechanism for resolution and execution reasons. |

**Table 26: IoTDeviceRole and PhysicalEntityRole Attribute**

Below, the XML schema of the IoTDeviceRole and PhysicalEntityRole within the context of the BPMN 2.0 standard is offered.

```

<xsd:element name="iotDeviceRole" type="tIoTDeviceRole" substitutionGroup="partnerRole"/>
<xsd:complexType name="tIoTDeviceRole">
  <xsd:complexContent>
    <xsd:extension base="tPartnerRole">
      ...
      <xsd:attribute name="isMobile" type="xsd:boolean"/>
    </xsd:extension>
  </xsd:complexContent>
</xsd:complexType>

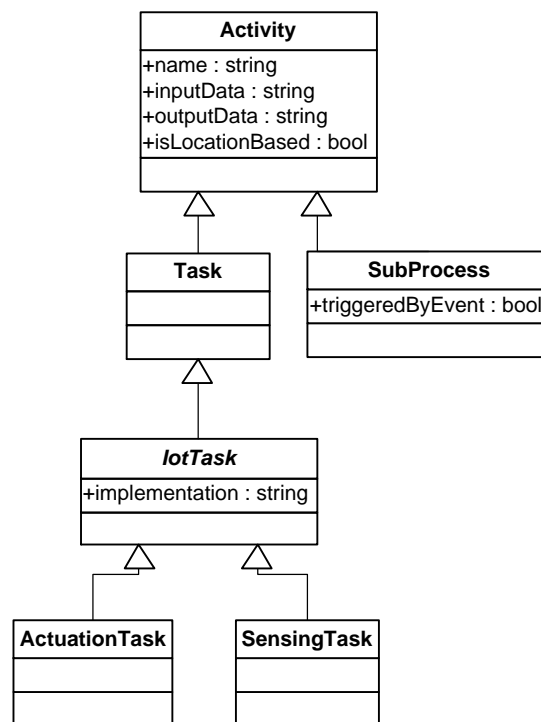
<xsd:element name="physicalEntityRole" type="tPhysicalEntityRole" substitutionGroup="partnerRole"/>
<xsd:complexType name="tPhysicalEntityRole">
  <xsd:complexContent>
    <xsd:extension base="tPartnerRole">
      ...
      <xsd:attribute name="isMobile" type="xsd:boolean"/>
    </xsd:extension>
  </xsd:complexContent>
</xsd:complexType>

```

## Activity

The activity is the abstract super class for all concrete activity types. A location-dependent activity is used in the process model to describe the activity dependence of the mobile behaviour of one or more process participants or entities. A mobile activity is an activity where a participant of type mobile IoT Device is responsible for its execution. By the class relations between IoT Device and IoT Activity, and the already established property `isMobile` of IoT Device, the mobility marker will not be introduced in addition on activity side. Thus, the class activity is simply expanded by the `isMobile` attribute.

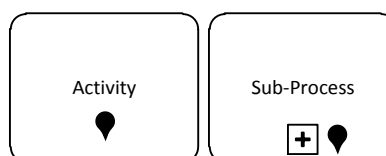
A sub-process in BPMN is an activity whose internal details consist of Activities, Gateways, Events and Sequence flows. It can be presented in a collapsed as well as in an expanded form. BPMN specifies the 5 types of sub-processes Loop, Multi-Instance, Compensation, Ad-Hoc and Compensation + Ad-Hoc. Since `SubProcess` is a sub-class of `Activity` no new property needs to be added. Figure 33 presents the corresponding BPMN dependent Activity Class Diagram



**Figure 33: Activity Class Diagram**

The marker for a task that is location-based must be positioned in the lower middle of the typical task object, which is shown in Figure 34 a). The location-based marker may be used in combination with the loop or multi-instance marker as well as with all types of tasks.

The additional IoT-marker for a location-based sub-process can be used in a complementary form to all types of sub-processes, which is shown in Figure 34 b). While the expanded Sub-Process shows all details in the view, there is no additional introduction by introducing a location-based Sub-Process. In contrast, the collapsed sub-process hides its details and the object uses a marker to distinguish it from a conventional Task. The location-based sub-process marker must be the rotated drop and is positioned at the middle bottom centre of the shape on the right side of the squared plus sign.



**Figure 34: Location-based Activity (a) and Collapsed location-based Sub-Process (b)**

The following table presents the new attribute of `Activity` class. For a complete association overview of `Activity`, please reference [14].

| Attribute Name                  | Description / Usage   |
|---------------------------------|---|
| isLocationBased: Boolean [0..1] | An optional Boolean specifying whether the activity is depending on the location of one or more mobile process participants such as IoT Devices or Physical Entities. |

**Table 27: Activity Attribute**

Below, the XML-schema of Activity within the context of BPMN 2.0 standard is offered.

```

<xsd:element name="activity" type="tActivity"/>
<xsd:complexType name="tActivity" abstract="true">
  <xsd:complexContent>
    <xsd:extension base="tFlowNode">
      <xsd:sequence>
        ...
      </xsd:sequence>
      <xsd:attribute name="isForCompensation" type="xsd:boolean" default="false"/>
      <xsd:attribute name="isLocationBased" type="xsd:boolean" default="false"/>
      <xsd:attribute name="startQuantity" type="xsd:integer" default="1"/>
      <xsd:attribute name="completionQuantity" type="xsd:integer" default="1"/>
      <xsd:attribute name="default" type="xsd>IDREF" use="optional"/>
    </xsd:extension>
  </xsd:complexContent>
</xsd:complexType>

```

### 5.2.7 IoT Process Ratios

In this section we consider a basic conceptual introduction of the process ratios into process modelling for handling some IoT specific aspects. This section includes a first proposition and so far does not represent a mature elaboration.

#### What are process ratios?

A process ratio describes the perspective of the process modeller on the designed business process. With the help of process ratios, the modeller can be enabled to express targets he wants to achieve through executing the process model. The process ratios are primarily related to process models, which were professionally described and the resolution phase is used as a supportive procedure in order to specify the technically process model. The ratios may express any and numerous requirements. A prerequisite is that the formulated requirements are understood by the resolution phase. The resolution phase knows how to interpret and reformulate the requirements so that they can be broken down into a finer requirements level i.e. the resolution of Physical Entities, IoT Devices or IoT Activities. In principle, ratios can contain functional (e.g. the overall processing time shall not exceed 2 s) and non-functional process requirements (process quality shall be reliable).

#### Which typical IoT Process Ratios are needed?

Below some process ratios are listed that might be of particular interest in the IoT domain:

- Maximum number of involved IoT Devices
- Maximum processing time of IoT Devices
- Minimum quality threshold of a process, which contains a breaking down to quality aspects of process components.
  - o Quality of Information
  - o Quality of Service
  - o Quality of Event

The quality of processes could be brought together in [5] by combining the quality characteristics of IoT information and IoT services into the so called IoT Quality Metric (IQM). Thus, the following four main areas are identified that are relevant to the process quality: traceability properties, reliability properties, spatial and temporal properties. Figure 35 shows the Ecore model of IQM metric.

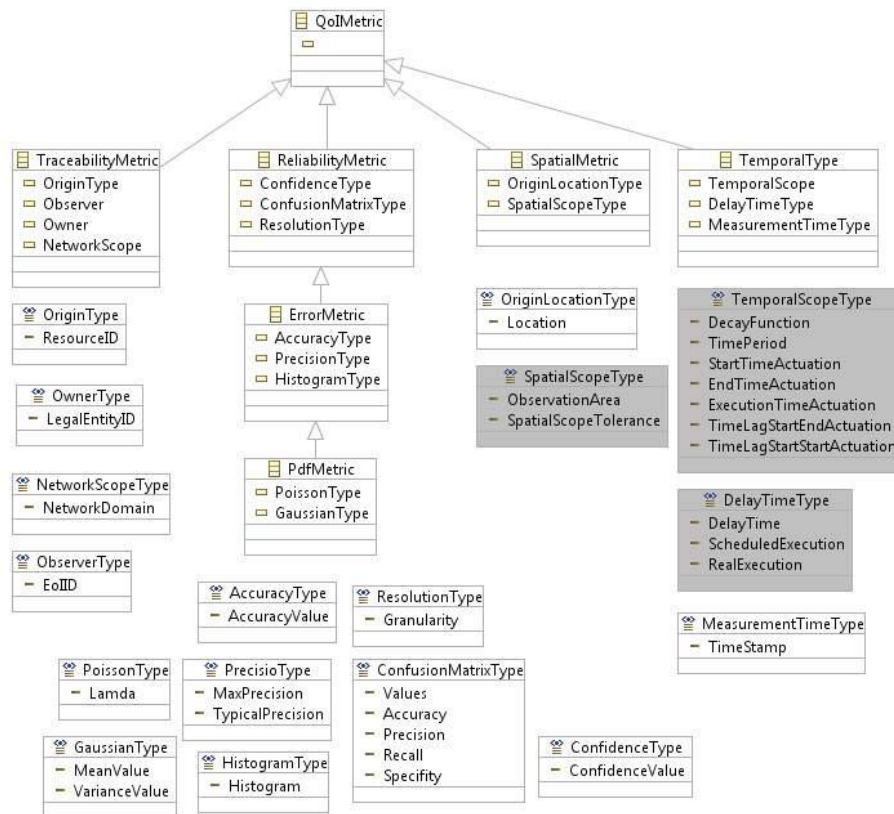


Figure 35: Ecore Model of IQM

### What does business process modelling offer to cover process ratios in the process model?

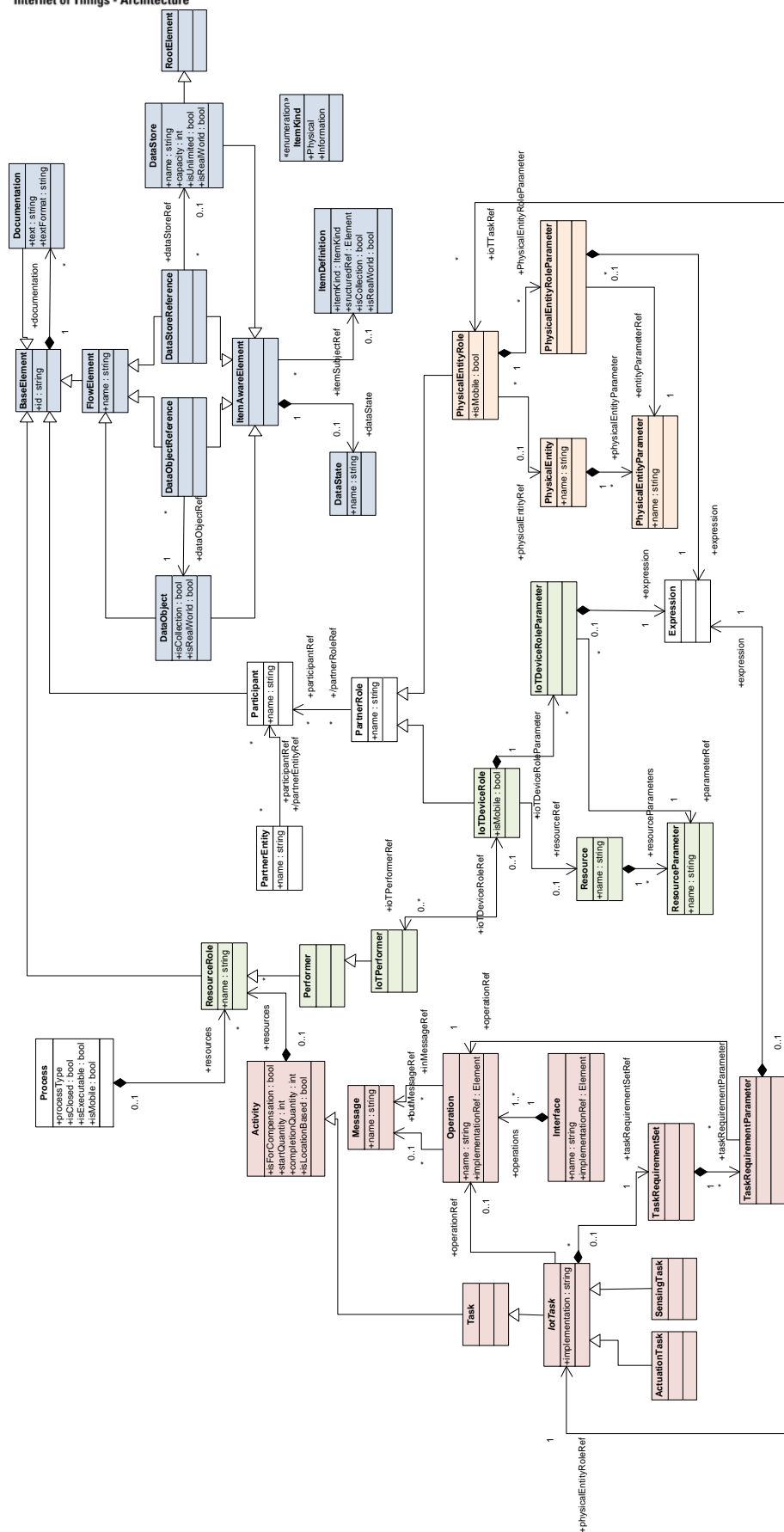
There have been several solutions proposed, such as extending BPMN or EPC meta-models by a goal specification. These solutions cover approaches to define goals not only technically but also to integrate them to the graphical process model. Further research regarding these issues will be undertaken in the future course of this project. The previous project results in terms of process ratios were summarized and published in [5].

## 5.3 From IAPMC to BPMN 2.0 Model Instantiation

### 5.3.1 Complete IAPMC Overview

In this section, the connection of the main components of the IAPMC metamodel is described. Figure 36 presents the metamodel and the relationships between the seven main elements IoT Activity, IoT Device, Process Resource, Physical Entity, Real World Data Object / Store, Mobility and Process IoT ratios, which were introduced in the previous sections. It reveals an excessive dependence between the three components IoT Activity, Process Resource and IoT Device. These are the initial and central components of the IAPMC. Based on the IoT Activity the basic component Physical Entity is introduced. With the component Mobility, mobility aspects of the participants Physical Entity and IoT Device are added and further process flow elements are equipped with location-based markers. The two components Real-World Data Object / Store and Process IoT ratios are independent of the other IAPMC components.





**Figure 37: IAPMC BPMN 2.0 Integration**

## 5.4 Process Examples

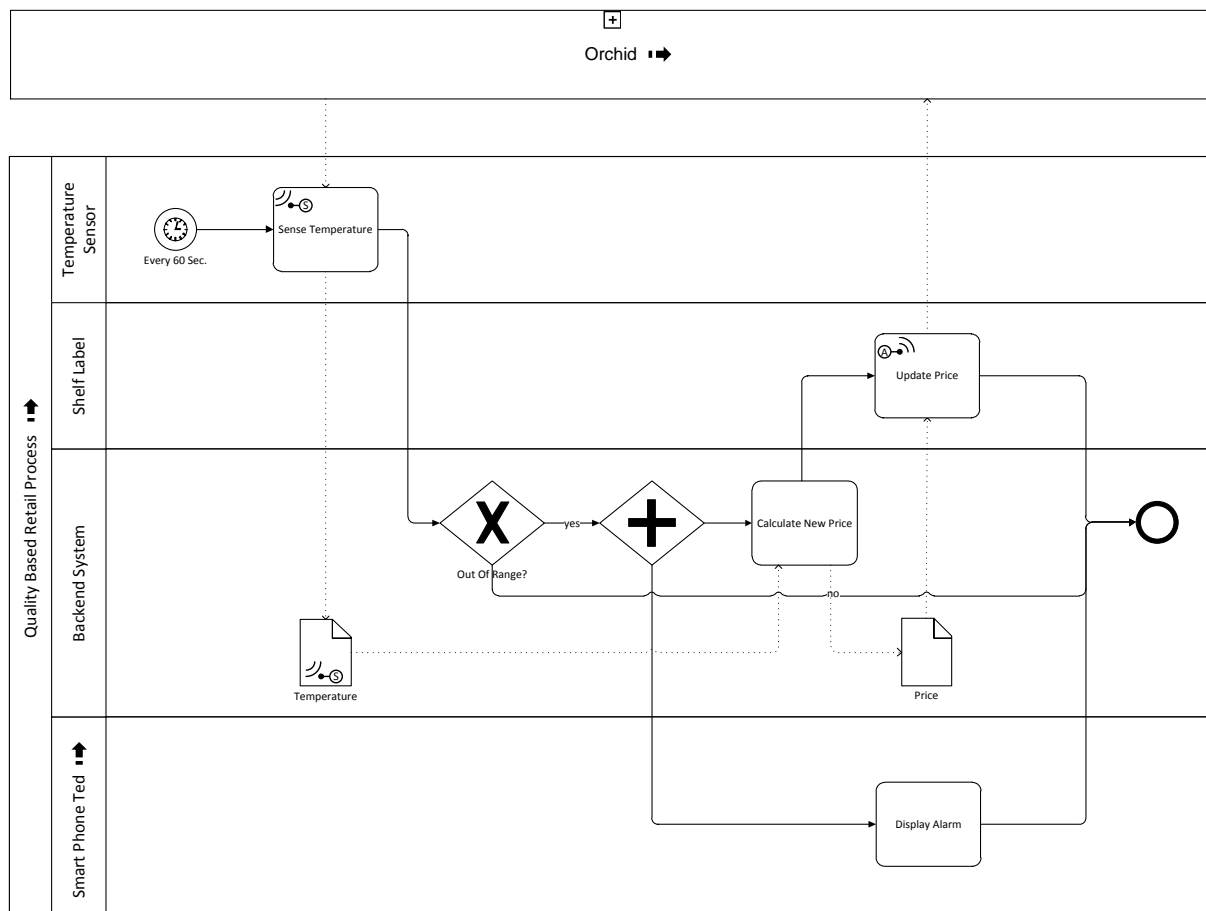
This section selects one sample processes in which the proposed IoT modelling extensions are demonstrated. The presented business processes is not fictitious, but a process that is already implemented within the framework of IoT-A WP7. The sub-section describes the content of the processes and shows the corresponding BPMN 2.0 model in collaboration diagram.

### 5.4.1 Business Process “Sensor Based Quality Control”

The business process which is shown by Figure 38 models how a temperature sensor monitors perishable goods in a store. The temperature sensor measures are used for estimating the quality of a rare and expensive form of Chinese orchids. Depending on the temperature of the environment, the estimated future quality of the orchids is determined and prices are reduced, even before a perceivable degradation of quality occurs. By applying this sensor based quality control and combining it with dynamic pricing, it is ensured that the goods are sold before quality degradation is likely to occur.

From a business process perspective the diagram shows the quality based retail process comprehending all included participants, process steps of the flow, data objects and gateways. The process pool contains five process participants. Four of the participants are modelled by using swim lanes demonstrating active process participants. They are devices, which are responsible for the activities in their lanes. The whole process is marked as a mobile process resulting of mobile participant “Smart Phone Ted”. Since the resolution could act differently for services of a mobile device the mobile indicator is directly part of the process model. One further process participant, the passive Physical Entity “Orchid” is represented by its own pool. The pool is collapsed because the Physical Entity doesn’t have any responsibilities for process follow elements. In big contrast to the state of the art BPMN 2.0, the diagram contains on the one hand two types of a special IoT tasks and on the other hand one IoT specific data object. The Sensing Task is indicated by the symbol showing the transformation from the physical environment temperature of the orchid to an electrical signal. This information is transferred into the real-world data object “Temperature” in the backend system. Thereby the temperature information gets available for the rest of the business process, but is still marked as a data object having special information properties. The Actuation Task is indicated by the reversed symbol showing the transformation an electrical signal to a physical environmental state of the orchid. Therefore, information in form of the data object “Price” is transferred from the backend system and presented by numbers on the electronic shelf label belonging to the orchid.





**Figure 38: Sensor Based Quality Control**

## 5.5 Conclusions

This chapter introduced the elements of the IAPMC and the BPMN 2.0 implementation. New process modelling patterns were provided, which shall enable modellers to express IoT-aware business processes using current modelling notations like BPMN. Besides introducing concepts for expressing IoT domain key concepts like devices, IoT services and entities, we further provide a lightweight condition based approach for influencing the resolution and execution phase of a business process. With the new elements Real World Data / Store the distributed sensor data within a business process can be expressed, while the quality of the sensing information might be associated linking to external description documents. The mobility aspects focus on the expression of mobile participants like IoT Devices and Physical Entities as part of a mobile process and further provide location-based decisions, gateways and activities. By the introduction of IoT related process ratios we try to improve the issues fault tolerance and scalability. Finally, a meta-model of the IAPMC and its integration in the BPMN 2.0 meta-model is presented. Table 28 presents an overview of the introduced IAPMC elements and its coverage of the main defined weak points. The plus indicates that the corresponding new element provides a contribution to improving the existing weak point, while the bracketed plus a limited contribution and the minus provides no contribution.

|              | Entity based concept | Distributed execution | Interactions | Distributed data | Scalability | Availability / Mobility | Fault tolerance | Quality of information |
|--------------|----------------------|-----------------------|--------------|------------------|-------------|-------------------------|-----------------|------------------------|
| IoT Activity | +                    | +                     | +            | -                | +           | -                       | -               | -                      |

|                                |   |   |   |   |     |   |     |   |
|--------------------------------|---|---|---|---|-----|---|-----|---|
| IoT Device                     | + | + | + | - | +   | - | -   | - |
| Process Resources              | + | + | - | - | +   | - | -   | - |
| Physical Entity                | + | - | + | - | +   | - | -   | - |
| Real World Data Object / Store | - | - | - | + | -   | - | -   | + |
| Mobility                       | - | - | - | - | -   | + | -   | - |
| Process Ratios                 | - | - | - | - | (+) | - | (+) | - |

**Table 28: IAPMC Elements vs. Weak Points**

In the design of the IAPMC, it is important to keep a balance between simplicity and the expressive power. Therefore we keep the introduction of totally new concepts as low as possible and try to reuse existing modelling concepts like the participant for our purposes. On the other side we try to influence the resolution and execution phase by the introduction of conditions that can be formulated in a user-friendly way.

We will further provide some insights to the future technical evaluation of the IAPMC. It is planned to split the evaluation activities into the two strengths expressiveness and usability:

- "Expressiveness" shall test the completeness and coverage of business processes in the IoT domain. Potentially different known approaches might be applied. The ontological model Bunge-Wand-Weber for information systems might be applied in particular to specific domains to examine issues such as the representation of IoT typical scenarios. Another common approach might be the use of workflow patterns that have been proposed in [56] and was already used to evaluate numerous known process modelling standards. In addition to the familiar mapping of these concepts to IAPMC, case studies with modellers could be conducted, which would have to solve a set of IoT specific process modelling tasks. The results of such an analysis could also give information on which IAPMC elements are mainly used. This approach could especially help to find the right balance between expressiveness and suitability of the IAPMC.
- "Usability" shall test the scale on how satisfactory IAPMC might be used by a process modeller for modelling IoT based business processes. We are planning a survey to ask BPMN and IoT experts about their acceptance of the new elements and their opinion on the tool integration of the IAPMC.

## 6. Conclusions and Outlook

### 6.1 Summary

This deliverable reports about the specification and methodology of the concepts for IoT-aware process modelling (IAPM) within the IoT-A project. In the context of this deliverable, the term IoT-aware process mainly refers to the creation of a graphical and executable process model that uses IoT-technology for executing some or all process tasks in form of services. The presented concepts shall enable end-users (such as process modellers) to build new or change current process models according to the integration of IoT-enabled process tasks. The concept follows an overall design science approach of [37] by structuring the IAPM concept design process into the six iterative steps: identification, definition, design, demonstration, evaluation and communication. While the steps identification, definition and design are completely covered through a first iteration cycle, the steps evaluation and communication are partly covered, since they rely partly on a demonstration part that will be addressed during a later project point. The requirements for the IAPM concept are defined within three main categories by applying the requirements acquisition methodology of IoT-A WP6. Mapping the gained requirements to state of the art business process notations, we examine BPMN 2.0 as the most IoT-aware business process modelling approach. As general design principles the cross IoT-A WP2/WP4 view on different processes phases is recognized as well as the capabilities and limitations of the BPMN 2.0 extension mechanism.

According to eight general IoT design principles resulting from the notation examination, the following findings for the IAPM concept are achieved: The IAPM comprises seven new modelling concepts (IoT Activity, Sensing Activity, Process Resources, Physical Entity, Real World Data Object / Store, Mobility Aspect, and IoT Process Ratios), by picking up and extending selected concepts, elements, and artefacts of BPMN 2.0. The selection is performed considering the creation of an IoT-aware process model. In the IAPM BPMN 2.0 representation we have chosen a pragmatic proceeding. By means of seven concept pillars, we finally built a first solid foundation for the IAPM concept. Ontologies can only be applied for specifying process artefacts like IoT Devices, Physical Entities and IoT Tasks. At the moment, this approach seems to be most appropriate in order to achieve a working solution considering IoT aspects. However, as semantic descriptions will evolve and be attached to more and more artefacts, other full-fledged ontological approaches could be applied as well. As a main result a coherent metamodel of the IAPM and its elements, properties, and relationships is presented.

On the programmatic perspective, the IAPM will be implemented by IoT-A D2.4, as part of the tool support. For delimiting the presented contents from resolution und execution purposes, this deliverable describes in detail the interface of IAPM in BPMN 2.0. Thereby, on the one hand programmatic process modelling and serialization support and on the other hand exchange capabilities by XSD files between further BPMN 2.0 Editors or a BPMN 2.0 Execution Engine is provided. The ability to transform the concept into further process modelling languages like EPC, UML or BPEL is a key objective to the IAPM concept, both for executable and documentation purposes. Consequently, this deliverable presents the detailed BPMN 2.0 implementation.

Finally, a first idea for the future evaluation of the contents of the IAPM concept is presented.

### 6.2 Outlook

According to the overall project setup, deliverable D2.4 will evolve in parallel with the development to the IoT-aware process modelling tool as one part of the WP2 tool support. Thus, the WP2 tool implementation shall proof the IAPM concept. A first release is expected due to M20. For the process execution of the IoT-aware process model, IoT-A WP2 decided to not transform the IAPM BPMN 2.0 implementation to BPEL, but to directly execute the BPMN process file using one of the novel BPMN Execution Engines [34]. The direct application of ontologies in the tooling of the process modelling notation in order to formulate requirements to process artefacts will be a further focus.

One future step is to extend and apply the first evaluation ideas to gain feedback to the IAPM concept. Therefore, end-users from different domains (such as IoT and BPM) will model IoT-processes by using the IAPM concept. Any feedback given by these end-users will provide some feedback to further improvements of the concepts and prototypes such as shown in section 0.



**IoT-A**  
Internet of Things - Architecture



IoT-A (257521)

The IAPM concept will be directly transferred to WP3 of the project FI-WARE and serve as a basic IoT-aware modelling proposition.

Within the first IoT-A process year some SAP internal communications were established with business units working on the BPMN 2.0 standardization and associated software implementations in direct interaction with the OMG. The future work might include achieving the standardization of the proposed BPMN IoT-extensions as a long-term objective.

## References

Scientific publications appeared in relation to research work of this IoT-A WP2 deliverable:

- [1] S. Haller, C. Magerkurth, "The Real-time Enterprise: IoT-enabled Business Processes", IAB and IETF Interconnecting Smart Objects with the Internet Workshop 2011, Prague, Czech Republic. Accessible: [http://www.iot-a.eu/public/public-documents/documents-1/1/2/the-real-time-enterprise-iot-enabled-business-processes/at\\_download/file](http://www.iot-a.eu/public/public-documents/documents-1/1/2/the-real-time-enterprise-iot-enabled-business-processes/at_download/file)
- [2] C. Magerkurth, K. Sperner, S. Meyer, M. Strohbach "Towards Context-Aware Retail Environments: An Infrastructure Perspective", Mobile Interaction in Retail Environments (MIRE 2011), Stockholm, Sweden. Accessible: [http://www.iot-a.eu/public/public-documents/documents-1/1/2/towards-context-aware-retail-environments-an-infrastructure-perspective/at\\_download/file](http://www.iot-a.eu/public/public-documents/documents-1/1/2/towards-context-aware-retail-environments-an-infrastructure-perspective/at_download/file)
- [3] S. Meyer, K. Sperner, C. Magerkurth, and J. Pasquier, "Towards modeling real-world aware business processes" in Proceedings of Web of Things 2011, San Francisco, CA, USA, 2011. Accessible: [http://www.iot-a.eu/public/public-documents/documents-1/1/2/towards-modeling-real-world-aware-business-processes/at\\_download/file](http://www.iot-a.eu/public/public-documents/documents-1/1/2/towards-modeling-real-world-aware-business-processes/at_download/file)
- [4] S. Meyer, "Modelando Procesos de Negocios del Mundo Real " in Proceedings of the 6th Iberian Conference of Information Systems 2011, Cavez, Portugal, 2011. Accessible: [http://www.iot-a.eu/public/public-documents/documents-1/1/2/modelando-procesos-de-negocios-del-mundo-real/at\\_download/file](http://www.iot-a.eu/public/public-documents/documents-1/1/2/modelando-procesos-de-negocios-del-mundo-real/at_download/file)
- [5] S. Meyer, K. Sperner, C. Magerkurth "Towards Real-World Aware Enterprise Systems", IEEE MASS 2011, Valencia, Spain. Accessible: [http://www.iot-a.eu/public/public-documents/documents-1/1/2/towards-real-world-aware-enterprise-systems/at\\_download/file](http://www.iot-a.eu/public/public-documents/documents-1/1/2/towards-real-world-aware-enterprise-systems/at_download/file)
- [6] K. Sperner, S. Meyer, C. Magerkurth "Introducing Entity-based Concepts to Business Process Modeling", 3rd International Workshop and Practitioner Day on BPMN, Lucerne, Switzerland. Accessible: [http://www.iot-a.eu/public/public-documents/documents-1/1/2/introducing-entity-based-concepts-to-business-process-modeling/at\\_download/file](http://www.iot-a.eu/public/public-documents/documents-1/1/2/introducing-entity-based-concepts-to-business-process-modeling/at_download/file)

## Further References

- [7] ABPMP, Business Process Management Common Body of Knowledge - BPM CBOK: Leitfaden für das Prozessmanagement: Verlag Dr. Götz Schmidt, 2009.
- [8] Activiti. (2012). Activiti BPM Platform. Available: <http://activiti.org/>
- [9] T. Allweyer, BPMN 2.0 - Business Process Modeling Notation: Einführung in den Standard für die Geschäftsprozessmodellierung, 2nd ed. Norderstedt: Books on Demand, 2009.
- [10] M. Bartonitz, BPMS, BPML, BPEL, BPMN, WMS, XPDL. Available: [http://org-portal.org/fileadmin/media/legacy/BPMS\\_BPML\\_BPEL\\_BPMN\\_WMS\\_XPDL\\_...\\_.pdf](http://org-portal.org/fileadmin/media/legacy/BPMS_BPML_BPEL_BPMN_WMS_XPDL_..._.pdf).
- [11] M. Baker, "Writing a literature review," The Marketing Review, vol. 1, no. 2, pp. 219–247, 2000.
- [12] M. Bauer, V. Huang, M. Strohbach, C. Villalonga, B.J. Zwaag, D2.4 Quality of Information and Actuation: WP2- Sensor Information Service, 2010.
- [13] Bonitasoft. (2012). Bonitasoft Open Source Workflow & BPM software. Available: <http://www.bonitasoft.com/>
- [14] Business Process and Notation (BPMN), Version 2.0, 2012.
- [15] C. Dai and Z. Wang, "A Flexible Extension of WSDL to Describe Non-Functional Attributes: E-Business and Information System Security (EBISS), 2010 2nd International Conference on," e-Business and Information System Security (EBISS), 2010 2nd International Conference on, pp. 1–4, 2010.
- [16] T. Davenport, Process innovation: reengineering work through information technology: Harvard Business Press, 1993.
- [17] D. Domingos, F. Martins, R. Martinho, and M. Silva, "Ad-hoc changes in IoT-aware business processes: Internet of Things (IOT), 2010," Internet of Things (IOT), 2010, pp. 1–7, 2010.
- [18] K. Fakhroutdinov (01.11.2011). UML Course. Available: <http://www.uml-diagrams.org/activity-diagrams.html>.

- [19] P. Ferreira, R. Martinho, and D. Domingos, "IoT-aware business processes for logistics: limitations of current approaches," in *INForum 2010: II Simposio de Informatica*, L. S. Barbosa and M. P. Correia, Eds, 2010.
- [20] J. Freund, B. Rücker, and T. Henninger, *Praxishandbuch BPMN*: Hanser, 2010.
- [21] T. Gardner, Ed, *UML modelling of automated business processes with a mapping to BPEL4WS: Proceedings of the first european workshop on object orientation and web services at ECOOP*: Citeseer, 2003.
- [22] N. Glombitza, S. Ebers, D. Pfisterer, and S. Fischer, "Using BPEL to Realize Business Processes for an Internet of Things: Ad-hoc, Mobile, and Wireless Networks," in *Lecture Notes in Computer Science, ADHOC NOW 2011*, H. Frey, X. Li, and S. Ruehrup, Eds.: Springer Berlin / Heidelberg, 2011, pp. 294–307.
- [23] M. Hammer and J. Champy, *Reengineering the corporation: A Manifesto For Business Revolution*: Harper Business New York, 1994.
- [24] A. ter Hofstede, M. Adams, N. Russell, and W. M. P. Aalst, *Modern Business Process Automation: YAWL and its Support Environment*. Berlin, Heidelberg: Springer-Verlag Berlin Heidelberg, 2010.
- [25] imeikas. (2012). *BPMN2 Visual Editor for Eclipse*. Available: <https://github.com/imeikas/BPMN2-Editor-for-Eclipse/wiki>.
- [26] D. Jordan, J. Evdemon, A. Alves, A. Arkin, S. Askary, C. Barreto, B. Bloch, F. Curbera, M. Ford, and Y. Goland, "Web services business process execution language version 2.0," *OASIS Standard*, vol. 11, 2007.
- [27] JBoss Community. (2012). *JBoss Application Server 7*. Available: <http://www.jboss.org/jbossas>.
- [28] JBoss Community. (2012). *jBPM*. Available: <http://www.jboss.org/jbpm>.
- [29] N. Kavantzas, D. Burdett, G. Ritzinger, T. Fletcher, Y. Lafon, and C. Barreto, *Web services choreography description language version 1.0*, 2004.
- [30] N. Kefalakis, J. Soldatos, N. Konstantinou, and N. R. Prasad, "APDL: A reference XML schema for process-centered definition of RFID solutions," *Journal of Systems and Software*, vol. 84, no. 7, pp. 1244–1259, 2011.
- [31] M. Kloppmann, D. Koenig, F. Leymann, G. Pfau, A. Rickayzen, C. von Riegen, P. Schmidt, and I. Trickovic, "Ws-bpel extension for people-bpel4people," *Joint white paper*, IBM and SAP, 2005.
- [32] O. Kopp, M. Wieland, and F. Leymann, Eds, *External and Internal Events in EPCs: e 2EPCs*: Springer, 2010.
- [33] T. Kozel, Ed, *BPMN mobilisation: Proceedings of the European conference of systems, and European conference of circuits technology and devices, and European conference of communications, and European conference on Computer science: World Scientific and Engineering Academy and Society (WSEAS)*, 2010.
- [34] S. Meissner, "IoT-A Project Deliverable D2.3 – Orchestration of distributed IoT service interactions", 2012.
- [35] S. Meyer, *IoT-A WP2 Project Survey*. SAP Research, Zurich, 2011.
- [36] C. Ouyang, W. van der Aalst, M. Dumas, and A. ter Hofstede, "Translating bpmn to bpel," 2006.
- [37] K. Peffers, T. Tuunanen, M. Rothenberger, and S. Chatterjee, "A design science research methodology for information systems research," *Journal of Management Information Systems*, vol. 24, no. 3, pp. 45–77, 2007.
- [38] J. Recker, "BPMN modeling—who, where, how and why," *BPTrends*, vol. 5, no. 3, pp. 1–8, 2008.
- [39] G. Martin. et al. "IoT-A Project Deliverable D2.1 - Resource Description Specification", 2012.
- [40] C. Seel, J. Schimmelpfennig, D. Mayer, and P. Walter, "Conceptual modeling of complex events of the Internet of Things: EChallenges, 2010," *eChallenges*, 2010, pp. 1–8, 2010.
- [41] Signavio GmbH. (2012). *Signavio Core Components by Signavio GmbH*. Available: <http://code.google.com/p/signavio-core-components/>.
- [42] Signavio. (2012). *BPM + SaaS by Signavio. Business Process Management*. Available: <http://www.signavio.com/>.
- [43] B. Silver, *BPMN method and style*, 2nd ed. Aptos, USA: Cody-Cassidy Press, 2011.



- [44] B. Silver, "Executable BPMN 2.0", in BPMS Watch, ed, 2011. (Link: <http://www.brsilver.com/2011/11/07/executable-bpmn-2-0/>)
- [45] The Apache Software Foundation. (2012). Apache Tomcat. Available: <http://tomcat.apache.org/>.
- [46] The Eclipse Foundation. (2012). Eclipse Modeling Framework Project. Available: <http://www.eclipse.org/modeling/emf/>.
- [47] The Eclipse Foundation. (2012). Eclipse. Available: <http://www.eclipse.org/>.
- [48] The Eclipse Foundation. (2012). Graphiti - a Graphical Tooling Infrastructure. Available: <http://www.eclipse.org/graphiti/>.
- [49] The Eclipse Foundation. (2012). Model Development Tools BPMN2. Available: <http://www.eclipse.org/modeling/mdt/?project=bpmn2>.
- [50] I. Trickovic. (2008, 26.11.2010). BPMN 2.0 - Getting started. Available: <http://www.sdn.sap.com/irj/scn/go/portal/prtroot/docs/library/uuid/50db5a00-78bc-2b10-8ba4-dfc9b66e3789?QuickLink=index&overridelayout=true>.
- [51] Unified Modeling Language (UML), Version 2.4, 2011.
- [52] P. Valiente and H. van Heijden, "A method to identify opportunities for mobile business processes," Proc. Working paper series in Business administration stockholm school of economics, 2002.
- [53] W. van der Aalst and A. ter Hofstede, YAWL: yet another workflow language: Elsevier, 2005.
- [54] W. van der Aalst and C. Stahl, Modeling business processes: A petri net-oriented approach. Cambridge, Mass: MIT Press, 2011.
- [55] W. van der Aalst, A. ter Hofstede, and M. Weske, "Business process management: A survey," in Business process management, Berlin: Springer, 2003, pp. 1019-1019.
- [56] W. van der Aalst, A. ter Hofstede, B. Kiepuszewski, and A. Barros, "Workflow patterns," Distributed and parallel databases, vol. 14, no. 1, pp. 5–51, 2003.
- [57] H. Volzer, "An Overview of BPMN 2.0 and Its Potential Use," 2010, p. 14.
- [58] J.W. Walewski et al., Project Deliverable D1.2 – Initial Architectural Reference Model for IoT, June 2011, "[http://www.iot-a.eu/public/public-documents/documents-1/1/1/d1.2/at\\_download/file](http://www.iot-a.eu/public/public-documents/documents-1/1/1/d1.2/at_download/file)".
- [59] M. Weske, Business process management: concepts, languages, architectures: Springer-Verlag New York Inc, 2007.
- [60] S. White, "Using BPMN to model a BPEL process," BPTrends, vol. 3, no. 3, pp. 1–18, 2005.
- [61] C. Wolf, P. Harmon: The state of Business Process Management – 2010, BP-Trends-Report, February 2010, [http://bptrends.com/survey\\_landing.cfm](http://bptrends.com/survey_landing.cfm).
- [62] Workflow Management Coalition, XPD L Support and Resources. Available: <http://www.wfmc.org/xpdl.html>.
- [63] J. Ziemann and J. Mendling, Eds, EPC-based modelling of BPEL processes: a pragmatic transformation approach. Italy: Citeseer, 2005.



## Glossary

The following table describes the definitions of the main concepts relevant for the work in WP2. It consists of the official terms of the project, as they are referenced at <http://www.iot-a.eu/public/terminology> as well as the WP2 specific terms that are relevant for WP2 deliverables, but not necessarily for the work done in other work packages. These WP2-specific terms such as Business Process Management, Business Process Execution, Service Composition, Service Orchestration, Service Choreography, Complex Event Processing, etc. mostly relate to the Future Internet technologies and higher level enterprise systems that WP2 is concerned with.

| Term                             | Definition  | Source           |
|----------------------------------|---|------------------|
| Active Digital Entity            | Any type of active code or software program, usually acting according to a <i>Business Logic</i> .  | Internal         |
| Actuators                        | "An <i>actuator</i> is a mechanical device for moving or controlling a mechanism or <i>system</i> . It takes energy, usually transported by air, electric current, or liquid, and converts that into some kind of motion."  | [Sclater2007]    |
| Address                          | An address is used for locating and accessing – "talking to" – a <i>Device</i> , a <i>Resource</i> , or a <i>Service</i> . In some cases, the ID and the Address can be the same, but conceptually they are different.  | Internal         |
| Application Software             | "Software that provides an application <i>service</i> to the <i>user</i> . It is specific to an application in the multimedia and/or hypermedia domain and is composed of programs and data".   | [ETSI- ETR173]   |
| Architectural Reference Model    | The IoT-A architectural reference model follows the definition of the IoT reference model and combines it with the related IoT reference architecture. Furthermore, it describes the methodology with which the reference model and the reference architecture are derived, including the use of internal and external stakeholder requirements.  | Internal         |
| Architecture                     | "The fundamental organization of a <i>system</i> embodied in its components, their relationships to each other, and to the environment, and the principles guiding its design and evolution".   | [IEEE-1471-2000] |
| Architecture Vision              | "A high-level, aspirational <i>view</i> of the target <i>architecture</i> ."  | [TOGAF9]         |
| Aspiration                       | " <i>Stakeholder</i> Aspirations are statements that express the expectations and desires of the various <i>stakeholders</i> for the <i>services</i> that the final [ <i>system</i> ] implementation will provide."   | [E-FRAME]        |
| Augmented Entity                 | The composition of a <i>Physical Entity</i> together with its <i>Virtual Entity</i> .   | Internal         |
| Association                      | An association establishes the relation between a <i>service</i> and <i>resource</i> on the one hand and a <i>Physical Entity</i> on the other hand.  | Internal         |
| AutoID and Mobility Technologies | "Automatic Identification and Mobility (AIM) technologies are a diverse family of technologies that share the common purpose of identifying, tracking, recording, storing and communicating essential business, personal, or product data. In most cases, AIM technologies serve as the front end of enterprise software <i>systems</i> , providing fast and accurate collection and entry of data. AIM technologies include a wide range of solutions, each with different data capacities, form factors, capabilities, and "best practice" uses. AIM technologies also include mobile computing devices that facilitate the collection, manipulation, or communication of data from data carriers as well as through operator entry of data via voice, touch screens or key pads. | [AIMglobal]      |
| Business Logic                   | Goal or behaviour of a system involving Things serving a particular business purpose. Business Logic can define the behaviour of a single Thing, a group of Things, or a complete business process.   | Internal         |
| Business Process Execution       | Business Process Execution (BPE) consists of the manual, semi-manual and automated execution of business processes over actors. An important precondition for the automated execution of business processes is the technical specification of the business process model by technical analysts. The relevant industry standards for describing an executable business process model are BPEL and BPMN. Common IT-Systems supporting the execution of a huge number of automated processes use a process execution engine as a central controlling component.  | Internal         |
| Business Process Management      | Business Process Management (BPM) is a structured approach that employs methods, policies, metrics, management practices and software tools to manage and continuously optimize the activities and processes of an organization. Process management uses an iterative process revision cycle  | Internal         |



|                                |  |                   |
|--------------------------------|--|-------------------|
|                                | known as the BPM lifecycle, which enables the continuous improvement of business processes. Depending on the methodology the following lifecycle phases are distinguished: Definition, Modeling, Simulation, Deployment, Execution, Monitoring, Analysis, Optimization and Discovery. In the IoT-A project we mostly focus on parts of the lifecycle phases Modeling, Deployment and Execution.  |                   |
| Business Process Modeling      | Business Process Modeling is the activity of representing processes of an enterprise by abstracting the business process in a graphical or non-graphical model for progressing with one of the other BPM lifecycle phases in any order. Modeling a business process is typically performed by business analysts. The industry relevant standards are UML, EPC and BPMN. WP2 focuses on extending business process modelling approaches by IoT-specific characteristics.  | Internal          |
| Complex Event Processing       | Complex event processing (CEP) describes a method of processing a multitude of events triggered by various information sources, and consecutively derive more meaningful events from a set of low level events. In IoT-A CEP is considered as an approach to bridge real-world events to the execution of business processes. We use the term event processing in the narrowest sense, i.e. analysis of events and subsequent decision making in terms of taking actions in real-time, are not considered as event processing, but are rather considered part of the business logic. | Internal          |
| Constrained Network            | A constrained network is a network of devices with restricted capabilities regarding storage, computing power, and / or transfer rate.   | Internal          |
| Controller                     | Anything that has the capability to affect a <i>Physical Entity</i> , like changing its state or moving it.  | Internal          |
| Credentials                    | A credential is a record that contains the authentication information (credentials) required to connect to a resource. Most credentials contain an user name and password.   | Internal          |
| Device                         | Technical physical component (hardware) with communication capabilities to other IT systems. A <i>device</i> can be either attached to or embedded inside a <i>Physical Entity</i> , or monitor a <i>Physical Entity</i> in its vicinity.  | Internal          |
| Digital Entity                 | Any computational or data element of an IT-based system.   | Internal          |
| Discovery                      | Discovery is a <i>service</i> to find unknown <i>resources/entities/services</i> based on a rough specification of the desired result. It may be utilized by a human or another <i>service</i> . Credentials for authorization are considered when executing the discovery.  | Internal          |
| Domain Model                   | "A domain model describes objects belonging to a particular area of interest. The domain model also defines attributes of those objects, such as name and identifier. The domain model defines relationships between objects such as "instruments produce data sets". Besides describing a domain, domain models also help to facilitate correlative use and exchange of data between domains".  | [CCSDS 312.0-G-0] |
| Energy-harvesting Technologies | " <i>Energy-harvesting</i> (also known as power harvesting or energy scavenging) is the process by which energy is derived from external sources (e.g., solar power, thermal energy, wind energy, salinity gradients, and kinetic energy), captured, and stored. Frequently, this term is applied when speaking about small, wireless autonomous <i>devices</i> , like those used in wearable electronics and wireless <i>sensor networks</i> ."   | [Wikipedia EH]    |
| Future Internet                | Future Internet is a summarizing term for worldwide research activities dedicated to the further development of the original Internet. In the IoT-A project we focus mostly on the Internet of Things and the Internet of Services as major constituents of the Future Internet that are consolidated within WP2.  | Internal          |
| Gateway                        | A Gateway is a forwarding element, enabling various local networks to be connected.  | Internal          |
| Global Storage                 | <i>Storage</i> that contains global information about many <i>entities of interest</i> . Access to the <i>global storage</i> is available over the <i>Internet</i> .   | Internal          |
| Human                          | A human that either physically interacts with Physical Entities or records information about them, or both.  | Internal          |
| Identity                       | Properties of an entity that makes it definable and recognizable.  | Internal          |
| Identifier (ID)                | Artificially generated or natural feature used to disambiguate things from each other. There can be several Ids for the same <i>Physical Entity</i> . The set of Ids is an attribute of an <i>Physical Entity</i> .  | Internal          |
| Information Model              | "An information model is a representation of concepts, relationships, constraints, rules, and operations to specify data semantics for a chosen  | [Autol]           |



|  |  |                |
|--|--|----------------|
|  | domain of discourse. The advantage of using an information model is that it can provide sharable, stable, and organized structure of information requirements for the domain context. The information model is an abstract representation of entities which can be real objects such as devices in a network or logical such as the entities used in a billing system. Typically, the information model provides formalism to the description of a specific domain without constraining how that description is mapped to an actual implementation. Thus, different mappings can be derived from the same information model. Such mappings are called data models."  |                |
| Infrastructure Services                      | Specific services that are essential for any IoT implementation to work properly. Such services provide support for essential features of the IoT.   | Internal       |
| Interface                                    | "Named set of operations that characterize the behaviour of an entity."  | [OGS]          |
| Internet                                     | "The <i>Internet</i> is a global system of interconnected computer networks that use the standard <i>Internet</i> protocol suite (TCP/IP) to serve billions of <i>users</i> worldwide. It is a network of networks that consists of millions of private, public, academic, business, and government networks of local to global scope that are linked by a broad array of electronic and optical networking technologies. The <i>Internet</i> carries a vast array of information <i>resources</i> and <i>services</i> , most notably the inter-linked hypertext documents of the World Wide Web (WWW) and the infrastructure to support electronic mail. Most traditional communications media, such as telephone and television <i>services</i> , are reshaped or redefined using the technologies of the <i>Internet</i> , giving rise to <i>services</i> such as Voice over <i>Internet</i> Protocol (VoIP) and IPTV. Newspaper publishing has been reshaped into Web sites, blogging, and web feeds. The <i>Internet</i> has enabled or accelerated the creation of new forms of <i>human</i> interactions through instant messaging, <i>Internet</i> forums, and social networking sites. The <i>Internet</i> has no centralized governance in either technological implementation or policies for access and usage; each constituent network sets its own standards. Only the overreaching definitions of the two principal name spaces in the <i>Internet</i> , the <i>Internet</i> -protocol address space and the domain-name system, are directed by a maintainer organization, the <i>Internet</i> Corporation for Assigned Names and Numbers (ICANN). The technical underpinning and standardization of the core protocols (IPv4 and IPv6) is an activity of the <i>Internet</i> Engineering Task Force (IETF), a non-profit organization of loosely affiliated international participants that anyone may associate with by contributing technical expertise." | [Wikipedia IN] |
| Internet of Things (IoT)                     | The global network connecting any smart object.  | Internal       |
| Interoperability                             | "The ability to share information and services. The ability of two or more systems or components to exchange and use information. The ability of systems to provide and receive services from other systems and to use the services so interchanged to enable them to operate effectively together."   | [TOGAF 9]      |
| IoT Service                                  | Software component enabling interaction with resources through a well-defined interface. Can be orchestrated together with non-IoT services (e.g., enterprise services). Interaction with the service is done via the network.   | Internal       |
| Local Storage                                | Special type of <i>resource</i> that contains information about one or only a few <i>entities</i> in the vicinity of a <i>device</i> .   | Internal       |
| Location Technologies                        | All technologies whose primary purpose is to establish and communicate the location of a <i>device</i> e.g. GPS, RTLS, etc.  | Internal       |
| Look-up                                      | In contrast to <i>discovery</i> , <i>look-up</i> is a <i>service</i> that <i>addresses</i> exiting known <i>resources</i> using a key or <i>identifier</i> .   | Internal       |
| M2M (also referred to as machine to machine) | "The automatic communications between <i>devices</i> without <i>human</i> intervention. It often refers to a <i>system</i> of remote <i>sensors</i> that is continuously transmitting data to a central <i>system</i> . Agricultural weather sensing <i>systems</i> , automatic meter reading and <i>RFID</i> tags are examples."  | [COMPDICT-M2M] |
| Microcontroller                              | "A <i>microcontroller</i> is a small computer on a single integrated circuit containing a processor core, memory, and programmable input/output peripherals. Program memory in the form of NOR flash or OTP ROM is also often included on chip, as well as a typically small amount of RAM. <i>Microcontrollers</i> are designed for embedded applications, in contrast to the microprocessors used in personal computers or other general purpose applications. <i>Microcontrollers</i> are used in automatically controlled products and <i>devices</i> , such as automobile engine control <i>systems</i> , implantable medical <i>devices</i> , remote controls, office machines, appliances, power tools, and toys. By reducing the size and cost compared to a design that uses a separate microprocessor, memory, and input/output <i>devices</i> , <i>microcontrollers</i> make it economical to digitally control even more <i>devices</i> and processes. Mixed signal <i>microcontrollers</i> are common, integrating analog components needed to control non-digital electronic <i>systems</i> ".   | [Wikipedia MC] |

|   |  |                   |
|---|--|-------------------|
| Network resource  | <i>Resource</i> hosted somewhere in the network, e.g., in the cloud.   | Internal          |
| Next-Generation Networks (NGN)                              | "Packet-based network able to provide telecommunication <i>services</i> and able to make use of multiple broadband, QoS-enabled transport technologies and in which <i>service</i> -related functions are independent from underlying transport-related technologies"  | [ETSI TR 102 477] |
| Observer  | Anything that has the capability to monitor a <i>Physical Entity</i> , like its state or location.   | Internal          |
| On-device Resource  | <i>Resource</i> hosted inside a <i>Device</i> and enabling access to the <i>Device</i> and thus to the related <i>Physical Entity</i> .  | Internal          |
| Operator  | The operator owns administration rights on the services it provides and/or on the entities it owns, is able to negotiate partnerships with equivalent counterparts and define policies specifying how a service can be accessed by users.  | Internal          |
| Passive Digital Entities                                    | A digital representation of something stored in an IT-based system.  | Internal          |
| Physical Entity   | Any physical object that is relevant from a user or application perspective.   | Internal          |
| Perspective (also referred to as architectural perspective) | "Architectural perspective is a collection of activities, checklists, tactics and guidelines to guide the process of ensuring that a <i>system</i> exhibits a particular set of closely related quality properties that require consideration across a number of the <i>system's</i> architectural <i>views</i> ."   | [ROZANSKI2005]    |
| Privacy   | Information Privacy is the interest an individual has in controlling, or at least significantly influencing, the handling of data about themselves.  | Internal          |
| Process Execution Engine                                    | The process execution engine (PEE) is the central component with the complete process overview, which handles the specified technical process model. The PEE decides which service calls take place under which conditions and controls the process by orchestrating services and resources via interfaces. Automated processes are impacted by activities of human operators, services, and their descriptions, as well as resources and their descriptions. The PEE supports the process controlling with key indicators for evaluating and monitoring the process.  | Internal          |
| Reference Architecture                                      | A reference architecture is an architectural design pattern that indicates how an abstract set of mechanisms and relationships realises a predetermined set of requirements. It captures the essence of the architecture of a collection of systems. The main purpose of a reference architecture is to provide guidance for the development of architectures. One or more reference architectures may be derived from a common reference model, to address different purposes/usages to which the Reference Model may be targeted.  | Internal          |
| Reference Model   | "A reference model is an abstract framework for understanding significant relationships among the entities of some environment. It enables the development of specific reference or concrete architectures using consistent standards or specifications supporting that environment. A reference model consists of a minimal set of unifying concepts, axioms and relationships within a particular problem domain, and is independent of specific standards, technologies, implementations, or other concrete details. A reference model may be used as a basis for education and explaining standards to non-specialists." | [OASIS-RM]        |
| Resolution  | Service by which a given <i>ID</i> is associated with a set of <i>Addresses</i> of information and interaction <i>Services</i> . Information services allow querying, changing and adding information about the thing in question, while interaction services enable direct interaction with the thing by accessing the <i>Resources</i> of the associated <i>Devices</i> . Based on a priori knowledge.   | Internal          |
| Resource  | Computational element that gives access to information about or actuation capabilities on a <i>Physical Entity</i> .   | Internal          |
| Requirement   | "A quantitative statement of business need that must be met by a particular <i>architecture</i> or work package."  | [TOGAF9]          |
| RFID  | "The use of electromagnetic or inductive coupling in the radio frequency portion of the spectrum to communicate to or from a tag through a variety of modulation and encoding schemes to uniquely read the <i>identity</i> of an RF Tag."  | [ISO/IEC 19762]   |
| Self-* properties   | Self-* properties subsumes the desired capabilities of information systems to be self-configuring, self-organizing, self-managing, and self-repairing.   | Internal          |
| Sensor  | A sensor is a special <i>Device</i> that perceives certain characteristics of the real world and transfers them into a digital representation.   | Internal          |



|  |   |                  |
|--|---|------------------|
| Security   | "The correct term is 'information security' and typically information security comprises three component parts: 1.) Confidentiality. Assurance that information is shared only among authorised persons or organisations. Breaches of confidentiality can occur when data is not handled in a manner appropriate to safeguard the confidentiality of the information concerned. Such disclosure can take place by word of mouth, by printing, copying, e-mailing or creating documents and other data etc.;2.) Integrity. Assurance that the information is authentic and complete. Ensuring that information can be relied upon to be sufficiently accurate for its purpose. The term 'integrity' is used frequently when considering information security as it represents one of the primary indicators of information security (or lack of it). The integrity of data is not only whether the data is 'correct', but whether it can be trusted and relied upon; 3.) Availability. Assurance that the systems responsible for delivering, storing and processing information are accessible when needed, by those who need them. | [ISO27001]       |
| Service  | "Services are the mechanism by which needs and capabilities are brought together"   | [OASIS-RM]       |
| Service Choreography                                 | Service Choreography is a form of Service Composition in which the participating services interact without being coordinated by a central component. The messaging schema between the elementary services is defined from a global point of view outside the involved services.   | Internal         |
| Service Composition                                  | Service Composition denotes the composition of loosely coupled elementary services to form a higher-order composite service to provide additional functionality by re-using existing functionality. IoT-A focusses on hierarchical service composition that follows a black box approach keeping the composition implementation opaque to service consumers.  | Internal         |
| Service Composition Model                            | A Service Composition Model is a formal graphical or textual representation of a Service Composition.   | Internal         |
| Service Orchestration                                | Service Orchestration is a form of Service Composition in which the coordination of the involved services is performed by a central component. This component is called the orchestrator and defines the messaging schema, needed to fulfil the composite service.  | Internal         |
| Stakeholder (also referred to as system stakeholder) | "An individual, team, or organization (or classes thereof) with interests in, or concerns relative to, a <i>system</i> ."   | [IEEE-1471-2000] |
| Storage  | Special type of <i>Resource</i> that stores information coming from <i>resources</i> and provides information about <i>Entities</i> . They may also include <i>services</i> to process the information stored by the <i>resource</i> . As <i>Storages</i> are <i>Resources</i> , they can be deployed either on-device or in the network.   | Internal         |
| System   | "A collection of components organized to accomplish a specific function or set of functions."   | [IEEE-1471-2000] |
| Tag  | Label or other physical object used to identify the <i>Physical Entity</i> to which it is attached.   | Internal         |
| Thing  | Generally speaking, any <i>physical object</i> . In the term ' <i>Internet of Things</i> ' however, it denotes the same concept as a <i>Physical Entity</i> .   | Internal         |
| Unconstrained Network                                | An unconstrained network is a network of devices with no restriction on capabilities such as storage, computing power, and / or transfer rate.  | Internal         |
| User   | A <i>Human</i> or any <i>Active Digital Entity</i> that is interested in interacting with a particular physical object.   | Internal         |
| View   | "The representation of a related set of concerns. A <i>view</i> is what is seen from a <i>viewpoint</i> . An architecture <i>view</i> may be represented by a model to demonstrate to stakeholders their areas of interest in the architecture. A <i>view</i> does not have to be visual or graphical in nature".   | [TOGAF 9]        |
| Viewpoint  | "A definition of the perspective from which a <i>view</i> is taken. It is a specification of the conventions for constructing and using a <i>view</i> (often by means of an appropriate schema or template). A <i>view</i> is what you see; a <i>viewpoint</i> is where you are looking from - the vantage point or perspective that determines what you see".  | [TOGAF 9]        |
| Virtual Entity                                       | Computational or data element representing a <i>Physical Entity</i> . Virtual Entities can be either Active or Passive <i>Digital Entities</i> .  | Internal         |
| Wireless communication                               | "Wireless communication is the transfer of information over a distance without  | [Wikipedia WI]   |



|  |  |                          |
|--|--|--------------------------|
| technologies                           | the use of enhanced electrical conductors or "wires". The distances involved may be short (a few meters as in television remote control) or long (thousands or millions of kilometres for radio communications). When the context is clear, the term is often shortened to "wireless". Wireless communication is generally considered to be a branch of telecommunications." |                          |
| Wireline communication technologies    | "A term associated with a network or terminal that uses metallic wire conductors (and/or optical fibres) for telecommunications."  | [setzer-messtechnik2010] |
| Wireless Sensors and Actuators Network | "Wireless sensor and actuator networks (WSANs) are networks of nodes that sense and, potentially, control their environment. They communicate the information through wireless links enabling interaction between people or computers and the surrounding environment."  | [OECD2009]               |



## XML Schema with IAPMC BPMN Extensions:

```

<!-- actuation task extension -->
<xsd:element name="actuationTask" type="tActuationTask" substitutionGroup="flowElement"/>
<xsd:complexType name="tActuationTask">
  <xsd:complexContent>
    <xsd:extension base="tIotTask"/>
  </xsd:complexContent>
</xsd:complexType>

<!-- /actuation task extension -->

<!-- IoT Device extension -->
<xsd:element name="iotDeviceRole" type="tIotDeviceRole" substitutionGroup="partnerRole"/>
<xsd:complexType name="tIotDeviceRole">
  <xsd:complexContent>
    <xsd:extension base="tPartnerRole">
      <xsd:sequence>
        <xsd:element ref="iotDeviceRoleParameter"
minOccurs="0" maxOccurs="unbounded"/>
      </xsd:sequence>
    </xsd:extension>
  </xsd:complexContent>
</xsd:complexType>
<!-- /IoT Device extension -->

<!-- IoT Device extension -->
<xsd:element name="iotDeviceRoleParameter" type="tIotDeviceRoleParameter"/>
<xsd:complexType name="tIotDeviceRoleParameter">
  <xsd:complexContent>
    <xsd:extension base="tBaseElement">
      <xsd:sequence>
        <xsd:element ref="expression" minOccurs="1" maxOccurs="1"/>
      </xsd:sequence>
    </xsd:extension>
  </xsd:complexContent>
</xsd:complexType>
<!-- /IoT Device extension -->

<!-- IoT Resource extension -->
<xsd:element name="resourceRef" type="xsd:QName"
minOccurs="0" maxOccurs="1"/>
<xsd:element name="iotPerformerRef" type="xsd:QName"
minOccurs="0" maxOccurs="unbounded"/>
<!-- /IoT Resource extension -->
</xsd:sequence>
<!-- mobility extension -->
<xsd:attribute name="isMobile" type="xsd:boolean"/>
<!-- /mobility extension -->
  </xsd:extension>
</xsd:complexContent>
</xsd:complexType>
<!-- /IoT Device extension -->

<!-- IoT Device extension -->
<xsd:element name="iotDeviceRoleParameter" type="tIotDeviceRoleParameter"/>
<xsd:complexType name="tIotDeviceRoleParameter">
  <xsd:complexContent>
    <xsd:extension base="tBaseElement">
      <xsd:sequence>
        <xsd:element ref="expression" minOccurs="1" maxOccurs="1"/>
      </xsd:sequence>
    </xsd:extension>
  </xsd:complexContent>
</xsd:complexType>
<!-- /IoT Device extension -->

<!-- IoT Resource extension -->
<xsd:element name="iotPerformer" type="tIotPerformer" substitutionGroup="performer"/>
<xsd:complexType name="tIotPerformer">
  <xsd:complexContent>
    <xsd:extension base="tPerformer">
      <xsd:element name="iotDeviceRoleRef" type="xsd:QName" minOccurs="0"
maxOccurs="1"/>
    </xsd:extension>
  </xsd:complexContent>
</xsd:complexType>
<!-- /IoT Resource extension -->

<!-- actuation/sensing task extension -->
<xsd:element name="iotTask" type="tIotTask" substitutionGroup="flowElement"/>
<xsd:complexType name="tIotTask">
  <xsd:complexContent>
    <xsd:extension base="tTask">
      <xsd:sequence>
        <xsd:element ref="taskRequirementSet" minOccurs="1"
maxOccurs="1"/>
      </xsd:sequence>
    </xsd:extension>
  </xsd:complexContent>
</xsd:complexType>
<!-- IoT Entity extension -->

```



```

    minOccurs="1" maxOccurs="1"/>
    <xsd:element name="physicalEntityRoleRef" type="xsd:QName"
    <!-- /IoT Entity extension -->
  </xsd:sequence>
  <xsd:attribute name="implementation" type="tImplementation"
  default="##IoTService"/>
  <xsd:attribute name="operationRef" type="xsd:QName" use="optional"/>
  </xsd:extension>
</xsd:complexContent>
</xsd:complexType>
<!--/actuation/sensing task extension -->

<!-- IoT Physical Entity extension -->
<xsd:element name="physicalEntityRole" type="tPhysicalEntityRole"
substitutionGroup="partnerRole"/>
<xsd:complexType name="tPhysicalEntityRole">
  <xsd:complexContent>
    <xsd:extension base="tPartnerRole">
      <xsd:sequence>
        <xsd:element ref="physicalEntityRoleParameter"
        <xsd:element name="physicalEntityRef" type="xsd:QName"
        <xsd:element name="iotTaskRef" type="xsd:QName"
        </xsd:sequence>
        <!-- mobility extension -->
        <xsd:attribute name="isMobile" type="xsd:boolean"/>
        <!-- /mobility extension -->
      </xsd:extension>
    </xsd:complexContent>
  </xsd:complexType>
  <!-- /IoT Physical Entity extension -->

  <!-- IoT Physical Entity extension -->
  <xsd:element name="physicalEntityRoleParameter" type="tPhysicalEntityRoleParameter"/>
  <xsd:complexType name="tPhysicalEntityRoleParameter">
    <xsd:complexContent>
      <xsd:extension base="tBaseElement">
        <xsd:sequence>
          <xsd:element ref="expression" minOccurs="1" maxOccurs="1"/>
        </xsd:sequence>
        <xsd:attribute name="entityParameterRef" type="xsd:QName"
        use="required"/>
      </xsd:extension>
    </xsd:complexContent>
  </xsd:complexType>
  <!-- /IoT Physical Entity extension -->

  <!-- IoT Physical Entity extension -->
  <xsd:element name="physicalEntity" type="tPhysicalEntity" substitutionGroup="rootElement"/>
  <xsd:complexType name="tPhysicalEntity">
    <xsd:complexContent>
      <xsd:extension base="tRootElement">
        <xsd:sequence>
          <xsd:element ref="physicalEntityParameter" minOccurs="0"
          maxOccurs="unbounded"/>
        </xsd:sequence>
        <xsd:attribute name="name" type="xsd:string" use="required"/>
      </xsd:extension>
    </xsd:complexContent>
  </xsd:complexType>
  <!-- /IoT Physical Entity extension -->

  <!-- IoT Physical Entity extension -->
  <xsd:element name="physicalEntityParameter" type="tPhysicalEntityParameter"/>
  <xsd:complexType name="tPhysicalEntityParameter">
    <xsd:complexContent>
      <xsd:extension base="tBaseElement">
        <xsd:attribute name="name" type="xsd:string"/>
      </xsd:extension>
    </xsd:complexContent>
  </xsd:complexType>
  <!-- /IoT Physical Entity extension -->

```

```

<!--sensing task extension -->
<xsd:element name="sensingTask" type="tSensingTask" substitutionGroup="flowElement"/>
<xsd:complexType name="tSensingTask">
  <xsd:complexContent>
    <xsd:extension base="tIotTask"/>
  </xsd:complexContent>
</xsd:complexType>
<!--/sensing task extension -->

<!--actuation task extension -->
<xsd:element name="taskRequirementParameter" type="tTaskRequirementParameter"/>
<xsd:complexType name="tTaskRequirementParameter">
  <xsd:complexContent>
    <xsd:extension base="tBaseElement">
      <xsd:sequence>
        <xsd:element ref="expression" minOccurs="1" maxOccurs="1"/>
      </xsd:sequence>
      <xsd:attribute name="operationRef" type="xsd:QName" use="optional"/>
    </xsd:extension>
  </xsd:complexContent>
</xsd:complexType>
<!--/actuation task extension -->

<!--actuation task extension -->
<xsd:element name="taskRequirementSet" type="tTaskRequirementSet"/>
<xsd:complexType name="tTaskRequirementSet">
  <xsd:complexContent>
    <xsd:extension base="tBaseElement">
      <xsd:sequence>
        <xsd:element ref="taskRequirementParameter" minOccurs="0"
maxOccurs="unbounded"/>
      </xsd:sequence>
    </xsd:extension>
  </xsd:complexContent>
</xsd:complexType>
<!--/actuation task extension -->

```