Towards Real-World Aware Enterprise Systems

Reflecting the Quality Information of Physical Resources in Services and Processes

Sonja Meyer, Klaus Sperner, Carsten Magerkurth
SAP Research Center St. Gallen/Zürich
SAP (Switzerland) Inc.
9000 St. Gallen, Switzerland
sonja.meyer@unifr.ch, klaus.sperner@sap.com, carsten.magerkurth@sap.com

***Abstract*—In order to facilitate the development of IoT-aware business processes that adapt to real-world context information, we present a novel modeling approach that reflects IoT quality information inherent in physical resources and maps them to the higher layers of enterprise information systems. Quality information is crucial to real-world applications, and is therefore part of standard real-world resource descriptions. Correspondingly, on a service level, quality aspects are referred to as Quality of Service (QoS) and are usually expressed by using Service Level Agreements that are unrelated to the quality information on a resource level. On a process level, quality aspects are currently much less considered, as IoT-aware business processes are still to emerge at a large scale. In order to bridge the gap between physical resources and business process management, we propose to map Quality of Information (QoI) and Quality of Actuation (QoA) common to the resource layer as important, non-functional Internet of Things (IoT) related aspects to the service and process layers in enterprise information systems. By following a bottom up approach, we investigate how standard modeling notations such as USDL and BPMN 2.0 can be integrated and augmented for handling the real world aspects QoI and QoA within IoT-aware business processes. Finally, we exemplify the proposed enhancements by a business process including IoT resources and services.**

Keywords- internet of things; business process management; service description; usdl; process notation; bpmn

I. Introduction

In this paper, we aim for bringing together the seemingly unrelated worlds of the Internet of Things (IoT) and traditional Business Process Management (BPM). IoT applications are comprised of many different and heterogeneous physical resources such as RFID tags, sensors, or manufacturing machinery. In the IoT world, resource-related concepts such as the quality of information (Can I trust the value of a physical temperature sensor?) or quality of actuation (Is the door really closed?) are crucial for building reliable applications.

On the BPM side however, existing process modeling techniques do not necessarily need to consider quality of information aspects, as certain degrees of information quality are simply presupposed and considered as given. Running business processes fulfilling a sufficient degree of quality are certainly a key requirement for organizations, before the inclusion of real-world resources such as sensors or actuators into their business processes can be considered. If erroneous information is provided by sensors or actuators act on faulty information, the business process must react on these quality-related problems accordingly.

Many research initiatives are already concerned with improving IoT resource quality, such as optimizing sensor information by redundant sensor deployments or simply by building better sensors. However, as the real world will always be prone to inaccuracies or physical failures, our approach to deal with non-optimal information and actions in business processes is to acknowledge suboptimal information and actuation qualities in IoT resources and make them transparent in the business process modeling. This way, quality requirements can already be considered at the creation of process models and adaptive processes depending on these measures can be developed.

In order to integrate physical resources and their quality aspects in business processed, we envision an Internet of Things being integrated to enterprise systems with open Internet-based standards, including the Unified Service Description Language (USDL) [1] for describing services and the Business Process Model and Notation (BPMN) for composing and orchestrating IoT-aware as well as non IoT-aware services to business processes [2] in a large-scaled enterprise environment. The central problem in utilizing these open standards for quality modeling at the different layers of enterprise systems is that their underlying concepts, their expressiveness as well as their semantics and meta-models differ considerably, while addressing similar and partly intersecting properties.

By applying a bottom up approach, we introduce the quality concepts of real-world resources to the business process level in two steps: Coming from the low level resource description view based on the results of [3] we first transfer the central aspects from the low-level resource descriptions to the higher level of service descriptions by applying and extending parts of the latest USDL version. Second, we reflect the relevant quality parameters of the USDL service descriptions in an IoT-specific activity level of the business process descriptions that make up the highest level of abstraction. Consequently, we apply and extend the graphical and executable process notation of BPMN 2.0 and create new opportunities for process modelers to deal with IoT quality concepts already in the early phase of creating process models.

II. QUALITY OF INFORMATION: STATE OF THE ART

The exact definition of quality aspects for resources, services and business processes is open to debate. To build a foundation for our research, we have investigated existing definitions and metrics for these levels and grouped them into three according categories.

A. *Resource Level*

The IoT domain model [4] introduces three different types of devices: sensor, actuator and tag. In order to facilitate the interaction with such a device, it hosts a software component, the so-called on-device resource. The quality of this resource information of sensors and actuators can be described with the help of a dedicated metadata-model. The aspect of quality is so far not needed for tags.

In [3] two metric concepts are introduced as such a metadata-model for sensors and actuators based on [5]. Thereby, Quality of Information (QoI) is 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”. Therefore, 22 individual properties for the measurement of QoI are defined and grouped into a multidimensional parameter with the four main dimensions time, space, reliability and traceability. For the actuators, [3] defines the Quality of Actuation (QoA) as “any information that describes constraints on how well specific actuation request should be carried out”. Analogously to the QoI, the QoA is measured by different, partly overlapping individual properties, which are grouped into the three main parameters time, space, and reliability.

B. *Service Level*

On service level, semantics allow to define service quality parameters. The Unified Service Description Language (USDL) was defined to describe services from a business and operational point of view, covering technical aspects as well as business characteristics [1]. Since QoI and QoA are non functional aspects beside several functional aspects of a real-world service, we apply USDL for this approach. The current version USDL 3.0 M5 consists of 9 modules, each addressing different aspects of an overall service description. [6] The meta-model of these modules is defined in the Ecore language [7] for meta-modeling [8].

To facilitate the comparison of services regarding their QoS, it is necessary to agree on so-called service levels. A Service Level Agreement is general means to formulate the circumstances under which a service has to be delivered. The USDL Service Level Module describes by non-functional attributes and metrics the level of performance or the quality of a service. USDL provides a generic approach and the service level module definition comes with a basic extension including a basic service level metric; these both are considered as a foundation for this research work. The basic extension is not directly part of the core of the USDL language, but is included into the language specification by default. [9]

C. *Business Process Level*

On process level, there are different approaches considering quality aspects: The works of [10] propose different algorithms for the computation of the QoS of processes and workflows. These focus on the management of the QoS of workflows for organizations, and could be applied as a different, constitutive step. The work of [11] considers the computation of QoS for web service composition based on simulation techniques. This approach could be applied for a closed business process environment, but within an IoT we tackle with an infinite number of devices that potentially could be included into the business process, all providing a different degree of information and actuation quality.

Since all considered quality approaches on process level focus on different goals, none of the propositions can be applied for our intention to comprise several service quality parameters in one single ratio.

III. IOT-AWARE SERVICE LEVEL AGREEMENTS: QOI AND QOA IN USDL

In order to consider the IoT aspects QoI and QoA in the process model, these IoT service properties must first be introduced into the service description. Therefore, we assume an atomic, automated service, which gets a large part of the information it provides directly from an IoT sensor. The USDL service level module provides a generic approach for implementing the two metrics QoI and QoA as part of a Service Level Agreement. Although the basic extension of this module already contains a set of specified service level attributes in metrics, it does not contain all components of the IoT-specific metrics, which are defined in [3].

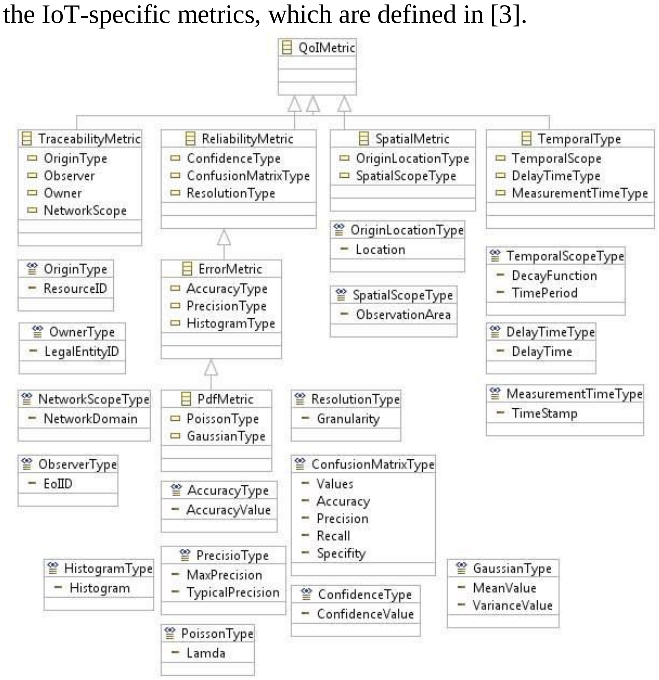


Figure 1. Ecore Model of QoI Metric

A. *Mapping of RDF Models to Ecore*

As a first step for the introduction of IoT related quality parameters into USDL, we define QoI and QoA in the modeling language Ecore using the Eclipse Modeling Framework (EMF). This definition is performed separately for each of the two IoT metrics and independently from the USDL Service Level Agreement Module with the Basic Extension. Since the models of these metrics are available in the Resource Description Framework (RDF) format, we must perform a mapping between those two types of modeling. In this step, we follow the mapping proposal of [12]. Figure 1 shows the resulting Ecore model for QoI.

By analogously applying the mapping procedure to the proposed QoA model we obtain the Ecore model for this metric, which is shown in figure 2.

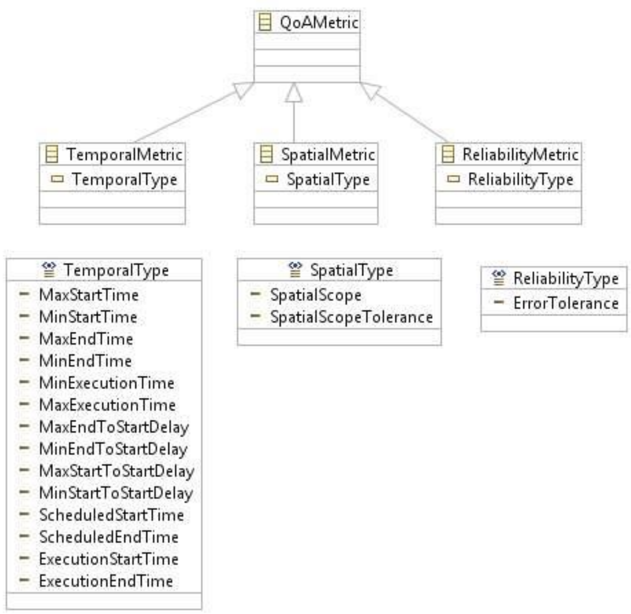


Figure 2. Ecore Model of QoA Metric

B. *Consolidation to One IoT Quality Metric*

While the QoI metric describes the information properties of sensors, the QoA metric states the properties of actuators. The metrics comprise quality information on the provided service of a sensor or actuator, which can also be used to match these properties with the specified service requirements of a process modeler. From a process perspective this means, both Ecore models contain quality properties of an atomic IoT service. As a second step we summarize both quality metrics into one IoT Quality Metric, in order to facilitate the integration with the USDL model. For this purpose, we integrate the parameters of the QoA metric into the more extensive QoI metric.

The class *TemporalScopeType*, that describes the validity of sensor or actuator data, is expanded by the parameters *StartTimeActuation*, *EndTimeActuation*, *ExecutionTime Actuation*, *TimeLagStartEndActuation*, and *TimeLagStart StartActuation*, while the class *DelayTimeType*, which describes the time interval between the real and digital world, is expanded by the parameters *ScheduledExecution* and *RealExecution*. To increase clarity, the *min* and *max* parameters are summarized to single interval parameters. The *SpatialeMetric* of the QoI model includes the parameter type *ObservationArea*, which covers already the QoA parameter *SpatialScope*. The *SpatialScopeTolerance* parameter is added to the *SpatialScopeType*. The parameter *MaxPrecision* exists in the QoI model as well. Figure 3 shows the Ecore model resulting from the aggregation. The customized classes and enumerations are highlighted.

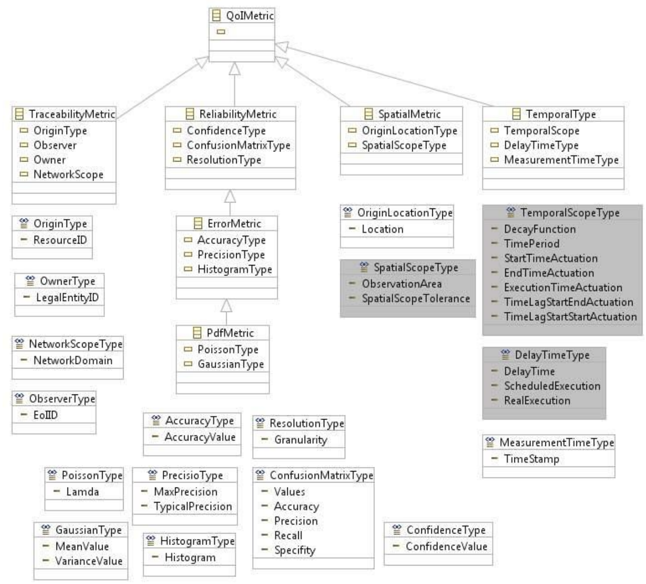


Figure 3. Ecore Model of the IoT-Quality Metric

C. *USDL Integration of IoT Quality Metric*

As a third step, we define the aggregated IoT Quality Metric (IQM) of the previous section as a Service Level Agreement within the USDL Service Level Module, which depends on the USDL Basic Extension. Combining the defined IoT-specific quality properties with further properties of the service layer, like exception handling procedures or privacy policies, we achieve a complete integration into the service layer of the Future Internet. For the Basic Extension of the USDL Service Level Module we obtain the integrated model shown in figure 3.

Service level attributes are part of the service levels and can be out of the three different forms *Constant*, *Variable Reference*, and *Metric*. The IQM can be integrated into the class metric and refers to the observation of a service feature that may change over the life cycle of a service instance. In the Basic Extension the class *Metric* covers already different types of metrics that are completed by the 4 resource-specific metrics of the IQM in order to be IoT-aware. For the Basic Extension of the USDL Service Level Module we obtain by the integration shown in figure 4.

In section V, we present an example of annotating the quality model of a sensing and an actuation service.

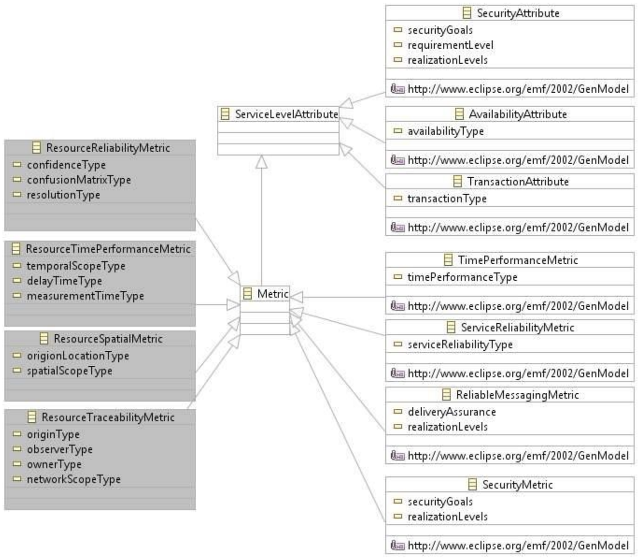


Figure 4. IQM Integration with the USDL Service Level Module

IV. IQM IN BUSINESS PROCESSES

After having transferred the real-world quality aspects from the resource to the service description level, we calculate a business ratio of the quality vector per activity in the process representation based on the USDL file. This enables the process modeler to respond to low-quality real-world activities by modifying the process accordingly, as well as to express the required IQM values per process activity.

A. Ratio Calculation

For the calculation of the process activity ratio $ratio_{IQM}$ we first express the quality parameter defined in section III.B mathematically by

$$IQM = \left(\overrightarrow{tem}, \overrightarrow{spa}, \overrightarrow{rel}, \overrightarrow{tra}\right)^T,$$

which is again composed by the four parameters temporal quality, spatial quality, reliability quality, and traceability quality. In the following, these quality areas are considered separately and the algebraic calculations are shown for the temporal quality vector $\overrightarrow{tem}$ and its range

$$\overrightarrow{tem} = \begin{pmatrix} tem1 \\ tem2 \\ tem3 \end{pmatrix}, \text{with } \begin{matrix} tem1 \in [cl1_{tem}, cr1_{tem}] \\ tem2 \in [cl2_{tem}, cr2_{tem}] \\ tem3 \in [cl3_{tem}, cr3_{tem}] \end{matrix},$$

where temi is the linear transformation of the three different *ResourceTimePerformanceMetric* types and cli_{tem} measures the left boundary and cri_{tem} the right boundary of the codomain.

In order to weight the individual parameters of the vector with one another, we move the left boundary of the range in the origin and set the individual ranges to length 1:

$$\overrightarrow{tem}_n = \overrightarrow{crl}_{tem}^T \, E \left(\overrightarrow{tem} - \overrightarrow{cl}_{tem}\right),$$

$$\text{with } \overrightarrow{crl}_{tem} = \begin{pmatrix} \frac{1}{cr1_{tem} - cl1_{tem}} \\ \frac{1}{cr2_{tem} - cl2_{tem}} \\ \frac{1}{cr3_{tem} - cl3_{tem}} \end{pmatrix} \text{ and } \overrightarrow{cl}_{tem} = \begin{pmatrix} cl1_{tem} \\ cl2_{tem} \\ cl3_{tem} \end{pmatrix}.$$

By weighting the individual components of the vector with the vector $\vec{g}_{tem}$, we obtain

$$\overrightarrow{tem}_{ng} = \vec{g}_{tem}^T \, E \, \overrightarrow{crl}_{tem}^T \, E \left(\overrightarrow{tem} - \overrightarrow{cl}_{tem}\right).$$

From this the ratio of the temporal quality vector can be determined by

$$ratio_{tem} = \frac{1}{|\vec{g}_{tem}|} \sum_{i=1}^{3} temi_{ng} .$$

The ratio calculation of the other quality parameters is performed analogously to the calculation of the temporal quality ratio. With the calculation of the weight vector $\vec{g}_{IQM}$ between the four quality parameters, the overall ratio of the process activity can be obtained:

$$ratio_{IQM} = \frac{1}{|\vec{g}_{IQM}|} \sum_{i=tem}^{tra} ratio_i$$

which takes values in the range of $[0, 1]$. An IQM ratio of 1 means that the information provided by the resource is absolutely perfect, while a IQM of 0 means that the information provided by the resource is completely defective.

B. Modeling Phase

For the usage of the IQM ratio during the modeling of business processes we distinguish between the following cases.

First, we consider the case that a new real-world activity is added to the process model. At modeling time no information about the service, which will be bound to the activity during the service resolution phase, is available. Nevertheless, if the modeler wants to ensure that the service, which will be bound to the activity, fulfills a certain threshold of IQM, he can annotate the desired threshold to the activity.

Second, a certain threshold of IQM for an activity, which is newly added to the process model, can be represented to the modeler. Prerequisite for this case is the direct integration of the service repository into the process modeling environment. To provide the threshold to the modeler, the modeling environment reads the USDL descriptions of the appropriate services from the service repository and calculates the IQM ratios as it is defined in the previous section. When saving the model to a file, the modeling environment writes the minimum of the calculated IQM ratios as the guaranteed threshold into the process model.

Like in the previous case, prerequisite for the third case is the integration of the service repository into the process modeling environment. The modeler selects a service that will be bound to the activity in the resolution phase. The modeling environment reads the IQM value from the service description and writes it into the process model.

Fourth, the modeler specifies the minimum threshold of a modeled activity itself manually. The threshold of the activity in the process model is overwritten with the new value.

C. Ratio Integration in BPMN 2.0

In order to store the calculated or required IQM value in the process model, two types of extensions must be realized.

On the one hand, the standard XML schema representation of the activity in the BPMN 2.0 process model needs to be extended to provide the ability to annotate the two IQM values. On the other hand, the symbolism of the existing BPMN 2.0 activity will be expanded to represent these two values in the process model as well. To distinguish real-world process activities from traditional process activities, we first introduce a real-world activity. At the moment we investigate if a detailed separation between sensor- and actuator-specific activities is necessary.

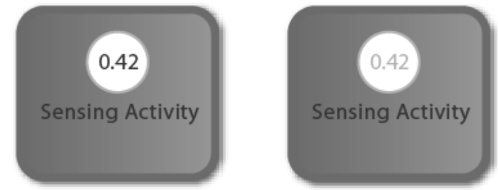
The BPMN 2.0 meta-model is extensible by default, but the extensions are limited to some restrictions that need to be fulfilled in order to be BPMN 2.0 compliant. [13] An activity of BPMN 2.0 can be out of atomic and non-atomic nature. In this case, we consider an atomic activity that is also called task and means that the process step is not broken down into a finer level of detail. This atomic activity corresponds to the atomic service on service level.

The following code extraction shows a task that was defined in BPMN 2.0 Core and has been enhanced by the two elements "IQMCalculated" and "IQMManual" defined outside the core in order to store the calculated IQM ratio as well as the manually defined IQM ratio permanently in the process model.

```
<xsd:schema ...>
   ...
   <xsd:group name="IQM">
   <xsd:sequence>
   <xsd:element ref="IQMCalculated" minOccurs="1"
        maxOccurs="1" />
   <xsd:element ref="IQMManual" minOccurs="1"
        maxOccurs="1" />
   </xsd:sequence>
   </xsd:group>
   ...
</xsd:schema>
```

Figure 5. Code Extraction of BPMN 2.0 XML Schema Extension

The proposed symbolism is based on the standard activity in BPMN 2.0. The new activity contains the IQM ratio, making it recognizable as a real-world specific activity. Figure 6 a) shows a manually entered IQM ratio by the process modeler as a minimum requirement to the activity bind service, while figure 6 b) illustrates an automatically generated IQM value from the service descriptions available in the service repository.



a) b)

Figure 6. Graphical Representation of IQM Ratio in BPMN Activity

V. APPLICATION EXAMPLE

To illustrate our proposed approach, we present a business process with an activity, in which the quality aspect from the resource level is reflected.

A process modeler creates a new process model for the receipt of goods in the company BigRetailer. To model the identification of incoming goods at the receipt point of the company, he adds the real-world activity *Identify_Goods* as a first step to the process model.

After the sensor-based identification, the process executes the non-IoT activity *Record_Goods* to record the identified goods in the goods receipt of the BigRetailer IT-system. The posting of goods is a sensitive business transaction, because the material management not only modifies the stock, but also enters the values of the received goods to the respective financial sub-accounts of the BigRetailer company. For the execution of the activity *Identify_Goods* in this business process, different sensor devices and subsequent different services are available. Depending on the different sensor devices the IQM of the on-device resources and of the services, which can be bound to the activity *Identify_Goods*, vary. Consequentially, the correctness of the amount of goods, which is recorded in the IT-system, would also vary. Figure 7 shows the process model.

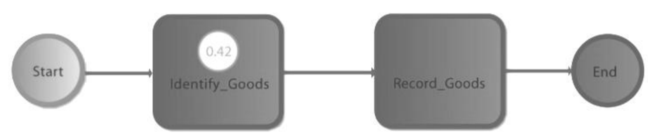


Figure 7. Process Model of Goods Receipt Point

From the USDL descriptions of the services, which are available for the activity *Identify_Goods* at design time, the modeling environment calculates the value 0.42 for the minimum IQM and displays it as the IQMCalculated ratio of the activity. Because the company BigRetailer requires a minimum IQM ratio of 0.85 for material accounting processes, which is bigger than the 0.42, the modeler overrides the automatically calculated value with the required IQM ratio IQMManual. The values IQMCalculated and IQMManual are stored in the BPMN process model. Figure 8 shows a simplified extract of the BPMN process model for this operation.

```
<bpmn:definitions id="ID_1" ...>
   ...
   <bpmn:extension mustUnderstand="true"
   definition="bpmn:QualityOfInformation"/>
```

```
 ...
<bpmn:task name="Identify_Goods" id="ID_2">
   <bpmn:IQMCalculated name="UsdlCalculation"
   id="ID_3">0,42
   </bpmn:IQMCalculated>
   <bpmn:IQMManual name="ManualDemand"
   id="ID_4">0,85
   </bpmn: IQMManual>
</bpmn:task>
```

Figure 8. Simplified BPMN Model of the Real-World Activity

At execution time one considered receipt point receives several truckloads of grit. Two different sensors are available in order to identify the amount of the received good: A weight scale (A) that provides accurate measurements, but requires a lengthy swaying process, and an optical sensor (B) that allows for roughly estimating the load level without stopping a truck. Depending on the contractual conditions, different accuracy levels might be sufficient for the goods receipt process. For each sensor the atomic service *Identify_Goods* is provided. Both on-device resources have different quality properties and therewith different quality properties in the descriptions of the atomic service. Figure 9 shows the two IQM values of sensor A and sensor B.

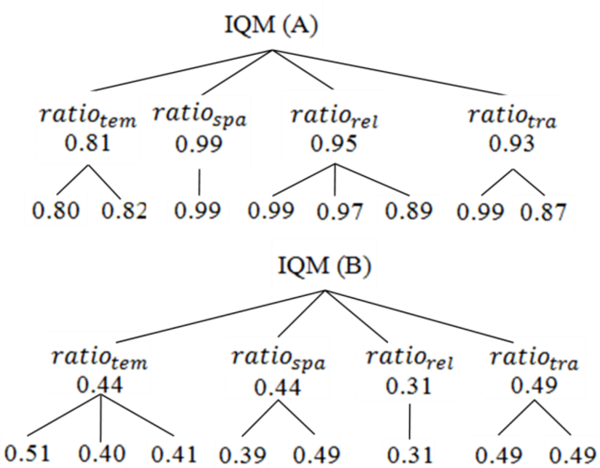


Figure 9. IQM Values of Sensor A and Sensor B

By applying the IQM ratio algorithm with the weighting vectors $\vec{g}_{tem} = \vec{g}_{spa} = \vec{g}_{rel} = \vec{g}_{tra}$ we obtain the ratio values:

a) $ratio_{IQM}(\text{sensor A}) = 0.92$
b) $ratio_{IQM}(\text{sensor B}) = 0.42.$

Only the sensor A of the two available devices satisfies the quality requirements of the process model and is used in the final business process execution environment.

VI. CONCLUSION

The imperfect quality of IoT resources such as sensors and actuators is a significant obstacle to successfully include real-world resources in business processes. In this paper, we have shown how to improve this situation considerably with a new bottom-up modeling approach that introduces quality aspects to the business process level. In order to reach that goal, we initially introduced new quality aspects to the service layer and extended the USDL Service Level Module. Utilizing an appropriate abstraction algorithm we made the quality parameters of the USDL description available to an IoT-specific activity level of the business process description, thus extending BPMN 2.0 modeling concepts.

Our future work will include a broader application of the IoT Domain Model presented in [4]. Further research will deal with the stepwise implementation of the identified IoT-specifies [2] in the process modeling domain.

ACKNOWLEDGMENT

The authors would like to thankfully acknowledge the support for this work provided by the European Commission within the FP7 project IoT-A, contract number: 257521. The authors would also like to thank their colleagues Stephan Haller and Matthias Thoma at SAP Research working on related topics in our research team.

REFERENCES

[1] SAP AG, Internet of Services. Available: http://www.internet-of-services.com/, (May 30, 2011).

[2] 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.

[3] M. Bauer, V. Huang, M. Strohbach, C. Villalonga, B.J. Zwaag, D2.4 Quality of Information and Actuation: WP2- Sensor Information Service, 2010.

[4] J. W. Walewski, D1.2 - Initial Architectural Reference Model for IoT, 2011. Available: http://www.iot-a.eu/public/public-documents/project-deliverables/1/1/D1%202_Initial_architectural_reference_model_for_IoT.pdf/at_download/file, (June 30, 2011).

[5] C. Villalonga, D. Roggen, C. Lombriser, P. Zappi, and G. Tröster, “Bringing quality of context into wearable human activity recognition systems,” Quality of Context, pp. 164–173, 2009.G.

[6] World Wide Web Consortium (W3C), W3C Unified Service Description Language Incubator Group. Available: http://www.w3.org/2005/Incubator/usdl/, (May 26, 2011).

[7] F. Budinsky, S. Brodsky, and E. Merks, Eclipse modeling framework: Addison-Wesley Boston, 2003.

[8] A. Barros, U. Kylau, and D. Oberle, Unified Service Description Language 3.0 (USDL) Overview. Available: http://www.internet-of-services.com/fileadmin/IOS/user_upload/pdf/USDL-3.0-M5-overview.pdf, (May 25, 2011).

[9] A. Barros, U. Kylau, and D. Oberle, Unified Service Description Language (USDL) Service Level (SLA) Module. Available: http://www.internet-of-services.com/fileadmin/IOS/user_upload/pdf/USDL-3.0-M5-servicelevel.pdf, (May 29, 2011).

[10] J. Cardso, Ed, Adaptive algorithm to predict the QoS of Web processes and workflows: International Conference on Computational Intelligence: Citeseer, 2004.

[11] G. Silver, A. Maduko, R. Jafri, J. Miller, and A. Sheth, Eds, Modeling and simulation of quality of service for composite Web services: Proceedings of the 7th World Multiconference on Systemics, Cybernetics and Informatics: Citeseer, 2003.

[12] IBM Corporation, Application Programming Interface. Available: http://www.cise.ufl.edu/mirrors/eclipse/modeling/mdt/eodm/javadoc/0.9.0/org/eclipse/eodm/rdf/transformer/package-summary.html, (May 29, 2011).

[13] Object Management Group (OMG), Business Process Model and Notation (BPMN) Version 2.0, 2011.