

Towards Modeling Real-World Aware Business Processes

Sonja Meyer, Klaus Sperner, Carsten
Magerkurth

SAP Research Center St. Gallen/Zürich

SAP (Switzerland) Inc.

9000 Switzerland

sonja.meyer@unifr.ch,
klaus.sperner@sap.com,
carsten.magerkurth@sap.com

Jacques Pasquier

Software Engineering Group

University of Fribourg

1700 Switzerland

jacques.pasquier@unifr.ch

ABSTRACT

In this paper we aim at bringing together the Web of Things (WoT) domain with the domain of enterprise business process modeling in order to work towards a Future Internet that includes all layers of networked technology stacks. We suggest introducing new notation concepts to the current business process modeling standards to facilitate modeling WoT aware business processes. We obtain and classify known WoT specific properties of real-world business processes. By means of a sensor based case study we analyze existing business process modeling standards such as Business Process Modeling and Notation (BPMN), Web Service Business Process Execution Language (WSBPEL), Extended Event-driven Process Chain (eEPC) and Unified Modeling Language (UML) in order to extract WoT specific process properties. A final evaluation concludes with the current most fitting process notation for modeling real world processes using WoT technology and suggests extending established approaches by including WoT specific aspects.

Categories and Subject Descriptors

D.2.9 [Management]: Software process models; H.4.1 [Office Automation]: Workflow management; J.1 [Administrative Data Processing]: Business

General Terms

Management, Documentation, Design, Standardization, Languages.

Keywords

Web of Things, Business Process Management, Process Notation

1. INTRODUCTION

In recent years, the so-called Web of Things (WoT) attracted much attention and now begins to gain widespread acceptance as an approach to treat real-world objects and devices as emancipated entities participating in networked applications. Heterogeneous interconnected objects have acquired the ability to sense their physical status and environment, to function as

actuators, and to communicate not only with other objects / nodes, but to reach out to a plethora of other applications via Internet protocols [2]. Initial concepts of WoT architectures already facilitate the integration of WoT systems with the Future Internet and shall form a set of building blocks of the future WoT [1],[13]. While the WoT community has clearly identified the need for interoperability with other internet applications, the domain of enterprise business process modeling so far seems to be more reluctant to address and model aspects of the real world. From a business process perspective, many WoT issues such as uncertainty and unreliability of information, highly dynamic federations of networked components, or even the notion of physical entities as inherent parts of the process or activity modeling are not directly reflected as modeling concepts.

In other words: The “things” in the web of things don’t really exist in the enterprise business process modeling world!

The research work presented in this paper contributes to overcoming the existing obstacles concerning the integration of WoT-technologies and business processes. It is carried out in the European research project IoT-A¹ that seeks to define a reference architecture for the Future Internet of Things that should allow for bridging the gap between the lower layers of a Future Internet and the higher enterprise process layers. A central outcome within the IoT-A project is the definition of a domain model using [7] as a basis, that describes real-world devices in terms of the services they provide, the distinction between devices and entities of interests, and other aspects of the real-world domain. Real-world services would potentially include properties such as coarse consistency, the importance of place and time and limited accessibility.

This paper investigates how the properties of real-world, WoT aware services and the traditional services of the Web of Services (WoS) used for the implementation of business processes differ from one another. The gained properties are classified and ranked regarding their importance of consideration at the level of business process modeling. A detailed analysis shows how the previously defined real-world properties can be modeled by using current BPM notations and highlights the gap to be closed by new approaches. As a final evaluation, a sensor based future shopping business process is modeled using the current notation best fitting with WoT requirements. Finally, this paper closes with a brief summary and an outlook of further research investigations in this field.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

WoT 2011, June 2011, San Francisco, CA, USA.

Copyright 2011 ACM 978-1-4503-0624-9/11/06...\$10.00.

¹ Supported by the EU project IoT-A (FP7-257521).

2. BPM NOTATIONS: STATE OF THE ART

The European Association of Business Process Management (EABPM) defines Business Process Management (BPM) as a systematic approach to capture, execute, measure, document, monitor and control automated and non-automated processes to reach certain goals [13]. One significant part of the BPM lifecycle is the process modeling. In operational use, two modes of business process modeling are distinguished:

- The business level, which considers the process flow in a few steps without defining detailed execution semantics
- The technical level, which specifies all execution details based on the previous business level description

Business processes can be modeled by many methods and techniques. According to [5] the most commonly used notation standards are BPMN, eEPC and UML. The Object Management Group (OMG) has adopted BPMN as a standard and the long expected version 2.0 was finally released in 2011 [11]. A further standard of the OMG is UML [12], which, besides twelve additional different diagram types, offers the graphically oriented activity diagram suitable to model business processes consisting of several activity elements. One of the most well-known and long adopted standards is the eEPC as part of the ARIS-concept [9], which offers an event focused view on business processes. So far, the latest versions of these standards - BPMN 1.2, eEPC 2.0 and UML 2.3 - only provide graphical flow-chart process notations. Therefore, several metamodels have been developed to convert these notations into executable ones such as WSBPEL 2.0 [4], which is standardized by OASIS and is the leading standard for representing executable business processes. WSBPEL, as an XML based process representation, allows for composing web services to higher value services. These powerful services are usually orchestrated to processes. WSBPEL is processed in a block-oriented way and contributes to a rigid control structure, which is an advantage for the process engine. Furthermore, it supports various aspects such as error or deadlock detection. The block structure of WSBPEL heavily differs from the graph based notations. Due to the totally different concepts, conversions from BPMN 1.2 or EPC to WSBPEL are still severely limited by many problems [16]. The forthcoming BPMN version 2.0 promises to bridge the gap between the notation and the real execution of a business process, therefore rendering WSBPEL obsolete. Consequently, WSBPEL will not be regarded in the further discussion. With the inclusion of defined execution semantics and an XML serialization format BPMN 2.0 is the first notation combining an end-user friendly notation with a detailed technical specification of an executable model in the same process representation [5].

In our approach we focus on conservative modeling standards for complementing existing business processes by a straightforward integration of WoT technology in order to maximize a potential application by industrial stakeholders. We thus deliberately exclude approaches such as stochastic Petri nets that allow for defining business workflows [6] and provide a suitable mathematical model for the control flow of agent-based processes, which are less applied in practice. Web Services Choreography Description Language (WS-CDL) [8] as a choreography language can represent the behavior of individual actors and their process

interaction with one another and are used in addition to modeling standards. Another approach [15] presents executable real-world BPEL processes partitioned to different system types in order to integrate sensor networks with business processes, whose results could contribute to WoT-aware modeling concepts.

3. WEB OF THINGS SPECIFIC CHARACTERISTICS

Current approaches focus on modeling and executing planned processes in a constant enterprise environment. In contrast, standard interfaces of WoT 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 WoT specific properties of this novel technology in the graphical and technical process notation. In order to highlight the difference between current and future business processes and to analyze the WoT-awareness of standard business process notations, we outline a number of WoT specific process properties:

Entity-based concept: While the service is the central concept in the WoS, the *Entity of Interest (EoI)* and its *devices* and *resources* are key concepts in the WoT domain model [7]. EoIs relate to the real-world “things” that one is interested in, devices are hardware that interact with EoIs, e.g. tachometer devices measuring the velocity of a car entity, resources are computational elements, i.e. the sensor software, hosted on the device. A WoT-aware business process modeling language must support an entity-based modeling approach and integrate it with existing modeling concepts of the WoS.

Distributed execution: The automated and semi-automated execution of a modeled business process is one of the key benefits of process modeling. In contrast to having a central process engine in a WoS process, the execution of the process steps is usually distributed over the devices in a WoT-enabled process. The orchestration of these distributed execution activities must be possible with a WoT-aware process modeling language.

Interactions: Business processes in the WoT introduce two additional forms of interactions: The interactions between device-level services of several EoIs and the interactions between WoT and WoS services, provided by e.g. enterprise backend systems. Due to the different nature of WoT- and WoS-services, these interaction forms must be considered during process modeling.

Distributed data: When business processes are realized in the WoS, a central data storage is normally the only data storage. In the WoT it is possible to distribute the data over several data storages, potentially even eliminating a central storage. A modeling language must allow arranging this distribution of data.

Scalability: In WoS business processes there is generally only one central service repository, but in WoT processes multiple EoIs, devices, and resources (e.g. sensors and actuators with its computational elements of a fridge) can appear. The complexity of the modeled process should be independent from the number of EoIs, devices, resources, and services. Additionally, the growing number of devices should not have an impact on the performance of the process execution. Therefore, the modeling language must provide concepts to describe the expected performance even for many devices.

Abstraction: As EoIs with multiple devices can appear in the WoT, it is reasonable to abstract these devices to one EoI (e.g. multiple sensors and actuators of a fridge to the EoI fridge). The EoI's devices offer different services. In the WoT the accuracy and availability of such accumulated data can be of much higher quality than the data of each individual device. The modeling language must support this kind of abstraction, in which the services of a device and the EoI are represented.

Availability / Mobility: In the WoS, the availability and configuration of services can be considered static. Due to the mobile nature, that the devices of EoIs often have, the availability of the devices in the WoT can't be guaranteed throughout. This affects the execution time of a process, in which a device and a resource of an EoI is involved. So the disappearance and re-emergence of devices, and upcoming execution delays must be expressible in a WoT-aware business process modeling language.

Fault tolerance: As the availability of devices in the WoT is uncertain, a business process relying on the presence of an EoI and its devices and resources must be able to handle any faults resulting of absence. Accordingly the modeling language must provide a concept to express this fault tolerance.

Flexibility / Event-based: In the WoS processes are mostly rigid and highly structured. In the WoT, the process flow can be influenced by the behavior of the end user, or through an occurring event, caused by e.g. a state change of an EoI [3]. The modeling language must be capable of designing such context-adaptive business processes which vary depending on occurring events from the normal sequence flow [14].

Uncertainty of information: In WoS business processes information is considered to be accurate. In the WoT there is an uncertainty about the information provided by the devices and services. As many devices with different levels of information accuracy may be involved in a business process, the quality of the overall information can vary widely and gets uncertain. In order to deal with this difficulty, the modeling language must provide means to express the certainty of information.

Real-time: In business processes in the WoS there is often no interaction with real world entities, so there is no need to take real-time aspects into account when modeling those processes. As WoT business processes interact with EoIs in the real world, also real-time constraints may apply to these processes. Hence, the expression of a concrete point in time or of a certain period of time for a process step must be possible with a WoT-aware process modeling language.

These WoT properties were obtained by implementing a survey with WoT and BPM experts [10]. In order to rank these characteristics regarding their importance for modeling WoT-aware processes, we divide them into the following three groups:

Group	WoT characteristics	Ranking
Entity-based data and execution	Entity based concepts, distributed execution, interactions, distributed data, real-time	1
Dynamic processes	Availability / mobility, fault tolerance, flexibility	2
Interactions	Scalability, abstraction, uncertainty of information	3

3-1: Classification of identified WoT characteristics

4. ANALYSIS

This section aims to analyze which of the previously defined WoT aspects can be modeled by using existing business process modeling approaches. Therefore, we examine stepwise how the three standard notations BPMN 2.0, eEPC 2.0 and the activity diagram of UML 2.3 cover each of the introduced WoT properties. 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 WoT aspects by the standard business process modeling notations provides an overview of the obtained results.

Entity-based concept: The terminology of the WoS-world is highly focused on modeling processes in an enterprise context. In this environment the technical and human EoI of a particular activity in form of a service is so far less important. All considered notations allow to model actors or roles, but except BPMN and UML where the EoI can be annotated to an activity or modeled using lanes, there is no possibility to specify an EoI. In none of the notations the terms EoI, device and resource are distinguished.

Distributed execution: BPMN and UML offer the ability to annotate an activity with the name or identification of an EoI 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 EoI of a certain EoI type (e.g. all fridges of one manufacturer), the classic capabilities provided through the modeling notation are considered insufficient.

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

Distributed data: All considered notations allow the modeling of data objects which correspond to the WoT-term resource. It is possible to specify a separate data flow from/to a resource to/from a specific activity. Additionally in UML and BPMN, it is possible to annotate the data object with the name of the resource or to place the data object in the appropriate lane which contains all activities corresponding to a certain resource. eEPC does not include this information.

Scalability: To assess the scalability of a business process, information (e.g. number of included devices) about the involved EoIs 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 EoIs, devices, resources and services are involved in a business process and the scalability cannot be evaluated. BPMN and UML offer the opportunity to symbolize different EoI and devices using swim lanes as separated actors. Using this symbolism, different EoIs and devices can be represented. In order to evaluate the scalability of business process further information about the modeled EoI or device needs to be provided during the modeling phase.

Abstraction: The notations BPMN and UML offer concepts to abstract activities to higher-level 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 independent 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 modeling of a business process before the real execution. Modeled activities can unexpectedly become unavailable by the mobility of the devices at execution time. Conversely, devices that were not available at modeling time might become available at run time. The first of these two cases can be modeled with all notations except eEPC, since the device is currently available at modeling time. To react to this particular mobile behavior at run time, additional information about the EoI and its devices would be helpful at modeling time such as the average accessibility of the device. This information is 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 modeling 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 EoIs, 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 sequentially to an event. This success mechanism results that a previously carried out activity is directly verified and complicates modeling 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 and do not offer 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. 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 in eEPC 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 notations provide the opportunity to model branches, and thus to verify the correctness of individual information by using error detection mechanisms. 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 sensor, which is influenced by frequent signal failures.

Depending on the description of the information source, the business process could be different. 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 eEPCs 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 modeling some real-time restrictions of WoT-aware business processes.

4.1 Wrapping-up

The graphical representation is quite similar for all approaches - 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 (besides differentiating between process and data flow) different types of interactions on the modeling level. The WoT-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 modeling languages. Fault tolerant behavior 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 WoT-aware business processes. Currently, the aspect of uncertain information is not covered by any language. Of the considered approaches, only BPMN covers the modeling of real-time restrictions.

WoT 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

4-1: Coverage of WoT characteristics by standard business modeling notations

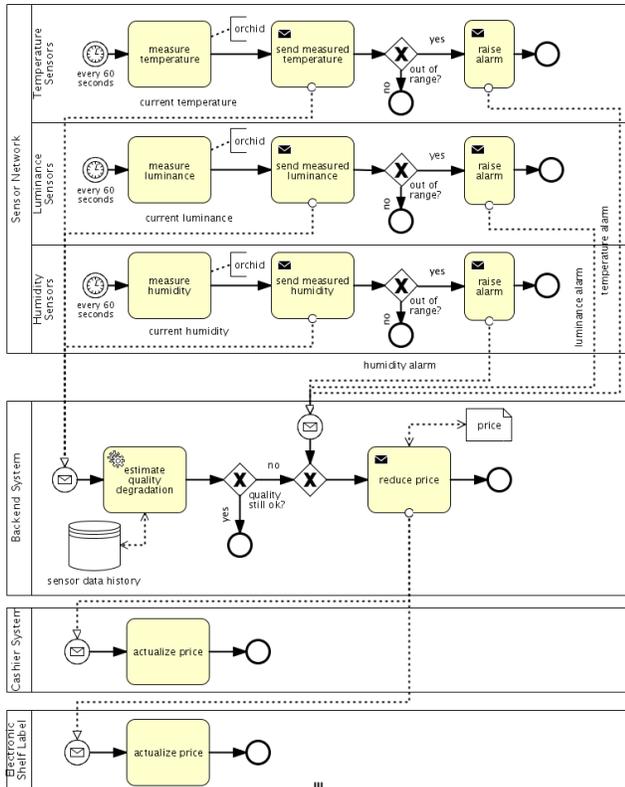
Concluding, the most suitable state of the art approach in order to describe business processes that include WoT-technology and to match the WoT domain model, BPMN 2.0 provides coverage for more WoT specific properties than other approaches. Table 4-1 provides a final overview of the stepwise analysis of this section. In addition to the listed WoT-focused analysis, the newest BPMN

version offers the following further advantages, which are significant for the upcoming research work:

- First end-user friendly modeling notation that is executable by defining execution semantics
- General extensibility for defining new artifacts, which are based on the WoT-aware modeling conventions

5. EVALUATION

In order to evaluate a WoT aware use case for process modeling, we consider the following typical WoT example that implies most of the WoT specific characteristics discussed before: The use case comes from the domain of retail and shows how sensors monitor perishable goods in a store. Measures from several sensors are used for estimating the quality of an expensive form of Chinese orchids. Depending on the luminance, humidity and temperature, the estimated quality of the orchids is determined and prices are reduced before the quality deteriorates. From a user perspective, the use case involves a customer who approaches the shelf with the orchids, when he realizes on the electronic shelf labels (ESLs) their price going down by 10%. Happy about the price reduction, he immediately picks an orchid and continues shopping. From a technical and process perspective, the use case revolves around a network of smart sensors connected to a backend system that continuously monitor certain environmental parameters. The sensors would only send alerts to the backend system on severely inappropriate conditions, but the average conditions over time are still measured to calculate the estimated point in time for a perceivable degradation in quality that leads to a price reduction. When that point in time comes, the backend system reduces the price at the POS by generating a campaign which is then propagated to the cashier system and also wirelessly to the ESLs.

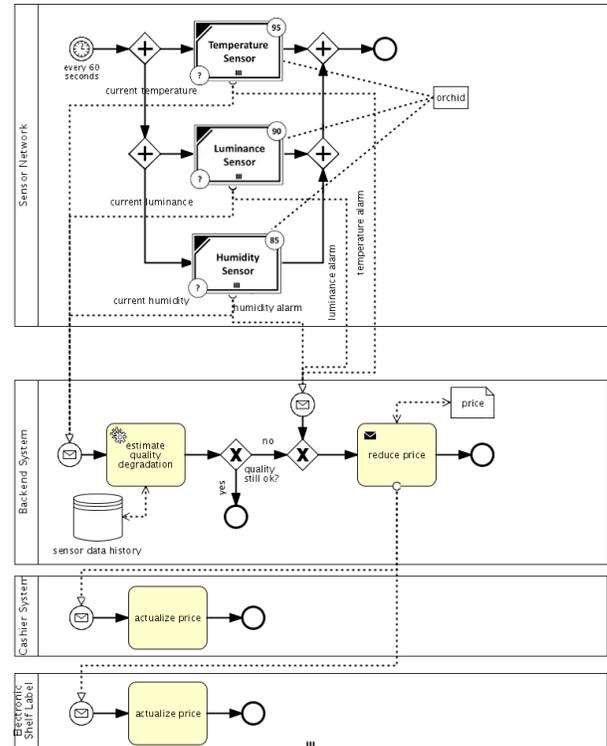


5-1: BPMN 2.0 notation of the example process

Figure 5-1 shows a BPMN 2.0 collaboration diagram of the described process. Each device has its own sub-process with start and end events. As sending messages is very common in the WoT and BPMN 2.0 forbids sending messages between lanes of a pool, the participants of the process are arranged in separated pools. The different types of sensors are grouped as lanes in a pool to show, that they form a sensor network. Alternatively it would have been possible to model all participants as lanes in one pool.

Evaluating the traditional BPMN 2.0 process model, it becomes apparent that some important WoT characteristics such as the entity-based concept, the uncertainty of information, the availability aspect, or the modeling of device internal services could not be represented appropriately. We therefore propose potential WoT specific extensions to the BPMN standard. We have not yet reached a mature state of how these extensions should finally be realized, also with respect to potential constraints, side effects, or execution / runtime implications. Nonetheless, we want to open the discussion about WoT specific BPMN extensions with a redefined process model shown in figure 5-2 that explicitly models the EoI and the sensor network surrounding it. Its key elements are:

- Explicit notion of the EoI and its related devices (“EoI Device Stencil”)
- Modeling of the certainty of information provided by the devices from “0 to 100%” (“Upper device annotation”)
- Explicit notion of the availability / potential fault of the devices; a potential failure of the device is indicated by “?” (“Lower device annotation”)
- Device internal services, which are activities provided by a device (“Interactions”)



5-2: BPMN 2.0 notation including WoT characteristics

As figure 5-2 demonstrates, by introducing a dedicated kind of “EoI Device Stencil” as a new artifact that implicitly includes the individual and formerly lane-based sensor activities, the complexity of the standard BPMN model can be reduced as both the lanes and activities count is reduced. Furthermore, the upper and lower device annotations enhance the expressiveness of the model with the WoT specific characteristics of uncertainty and availability inherently available as modeling constructs, potentially also supporting the respective executional aspects of the model. A more subtle, but from a WoT perspective rather important extension relates to the cardinality and explicitness of the EoI itself (in our case the orchid) which has to be modeled as a standard annotation for each relevant activity in standard BPMN, whereas by extending the annotations to allow for 1..n relationships to activities we can explicitly model that several devices and activities belong to the same EoI.

When comparing both models we can clearly perceive the reduced complexity and improved expressiveness of the WoT enhanced model despite the rather modest WoT specific extensions we proposed as an example for augmenting BPMN. The extensions relate to only some aspects identified in Section 3. We do not yet take execution aspects into account and do not base the presented model improvements on any formally solid procedures, but again, we are currently at the stage of a BPMN gap analysis of WoT aspects and do not yet propose concrete extensions, however with the above revised model we want to show in which direction future enhancements to BPMN could go. Our own work will now focus on elaborating on the gaps that BPMN currently has in terms of modeling WoT specific characteristics and coming up with an appropriate set of extensions.

6. CONCLUSION

With this paper we gave an introduction to the field of WoT-aware business process modeling. We presented a state of the art analysis of available business process modeling notations focusing on the graphical business process representation. First, we discussed several characteristics of WoT aware business processes emphasizing on the difference between WoT and WoS aware processes. Second, we analyzed and evaluated the current BPM notations BPMN 2.0, eEPC 2.0 and UML 2.3 in terms of their capability to cover the defined WoT characteristics. We identified that BPMN is most suitable for modeling WoT-aware business processes. Third, we opened the discussion for extending current BPM notations by new concepts in order to enable the application of WoT-technology within business processes from a modeling perspective.

In the future, we will continue to work on a general and language independent modeling concept in order to model WoT-aware business processes. Further research will deal with the application of this WoT-aware reference concept to a concrete business process modeling notation. As an obtained result of this paper, we plan in particular to apply the reference concept to BPMN 2.0 in the frame of the EU funded IoT-A project in order to come with concrete and verified WoT extensions. Besides these conceptual challenges, the project focuses on the development of a process editor in order to provide tool support for modeling WoT-aware business processes.

7. ACKNOWLEDGMENTS

The authors would like to thankfully acknowledge the support for this work provided by the European Commission within the FP7

project IoT-A. The authors would also like to thank Stephan Haller for his support.

8. REFERENCES

- [1] IoT-A (Internet of Things – Architecture). <http://www.iot-a.eu/public>, 2011.
- [2] SENSEI (Integrating the Physical with the Digital World of the Network of the Future). <http://www.sensei-project.eu/>, 2010.
- [3] Fraunhofer ISST. <http://www.saperionblog.com/lang/de/wie-workflows-auf-ihre-umgebung-reagieren-konnten-oder-kontextadaptives-bpm/1133/>, 2010.
- [4] Alves, A., Arkin, A., Askary, S., Barreto, C., Bloch, B., Curbera, F., Ford, M., Golan, Y., Guizar, A. and Kartha, N. Web services business process execution language version 2.0. OASIS Standard, 2007.
- [5] Freund, J., Rücker, B. and Henninger, T. *Praxishandbuch BPMN: Incl. BPMN 2.0*. Hanser Fachbuchverlag, 2010.
- [6] Ferscha, A., “Qualitative and Quantitative Analysis of Business Workflows using Generalized Stochastic Petri Nets”, in *Proceedings of Workflow Management - Challenges, Paradigms and Products*, 1994, 222-234.
- [7] Haller, S. “The Things in the Internet of Things”, Poster at the (IoT 2010). Tokyo, Japan, November 2010.. [Online]. Available: http://www.iot-a.eu/public/news/resources/TheThingsintheInternetofThings_SH.pdf, [Accessed Jan. 24, 2011].
- [8] Kavantzaz, N., Burdett, D., Ritzinger, G. and Lafon, Y. *Web Services Choreography Description Language Version 1.0, W3C Candidate Recommendation*. Technical report, November 2005.
- [9] Keller, G., Nüttgens, M. and Scheer, A. *Semantische Prozeßmodellierung auf der Grundlage "Ereignisgesteuerter Prozeßketten (EPK)"*. Inst. für Wirtschaftsinformatik, 1992.
- [10] Meyer, S. *IoT-A WP2 Project Survey*. SAP Research, Zurich, 2011.
- [11] *OMG Business Process Model and Notation 2.0 Specification*, 2011.
- [12] *OMG Unified Modeling Language Specification*, 2010.
- [13] Schmidt, G. *Business Process Management Common Body of Knowledge- BPM CBOK: Leitfaden für das Prozessmanagement* herausgegeben von der European Association of Business Process Management, 2009.
- [14] Saidani, O. and Nurcan, S. *Towards context aware business process modelling*. Citeseer, 2007.
- [15] Spiess, P., Vogt, H. and Jutting, H. *Integrating sensor networks with business processes*. Citeseer, 2006.
- [16] Weidlich, M., Decker, G., Großkopf, A. and Weske, M. *BPEL to BPMN: the myth of a straight-forward mapping*. On the Move to Meaningful Internet Systems: OTM 2008, 265-282.