Service Descriptions and Linked Data for Integrating WSNs into Enterprise IT

Matthias Thoma*†, Klaus Sperner*, Torsten Braun†
*SAP Research Switzerland
SAP (Switzerland) Inc., Althardstrasse 80, 8105 Regensdorf, Switzerland
matthias.thoma@sap.com
klaus.sperner@sap.com
†Communication and Distributed Systems (CDS) Research Group
University of Bern, Neubrueckstrasse 10, 3012 Bern, Switzerland
braun@iam.unibe.ch

***Abstract*—This paper presents our ongoing work on enterprise IT integration of sensor networks based on the idea of service descriptions and applying linked data principles to them. We argue that using linked service descriptions facilitates a better integration of sensor nodes into enterprise IT systems and allows SOA principles to be used within the enterprise IT and within the sensor network itself.**

***Keywords*-wireless sensor network, wsn, enterprise it, business integration, service descriptions, linked data**

I. Motivation

The enterprise IT world nowadays focuses mainly on service oriented architecture and service orchestration and starts to adapt choreography. Business processes are described in modeling languages like BPMN or BPEL. Business process decomposition [1] then decomposes the process into distributed process steps, where some of them can be executed even on physical devices like sensors or a sensor network gateway. Service binding and execution is handled by special enterprise middleware often referred to as enterprise service bus (ESB). Software development for sensor networks necessitates a good knowledge of the underlying technical details, which makes business integration a cumbersome activity. In our experience there is a gap in enterprise integration, which needs to be addressed, between the enterprise SOA world with its general purpose execution engines and process modeling on the one hand, and the sensor network world with its custom (often manually tuned) embedded software on the other hand.

We see the following five drivers in the integration of sensor nets into enterprise IT system:

1) Service Characteristics
   For the integration of a service into an enterprise system it is necessary to have a service endpoint and at least a technical description on how to invoke it. Additionally, it is desired to have also non-functional aspects described, which the execution engine can take into consideration in the binding and execution phase of a business process.
2) Service Discovery
   In a traditional SOA environment a service registry or repository is sufficient for discovering services within an enterprise. For those sensor networks, which are more or less static in nature, where most of the business logic is performed in the enterprise backend system, or if only limited business logic is executed on the nodes, a repository approach is sufficient as well. Nonetheless, in ad-hoc or self-organizing scenarios, where a lot of business logic is executed on the nodes and a backend system might not even exist, self-description of services is a must [2]. In an industrial setting there is often the problem that one or more sensor nodes join a different enterprise context, depending on their location. In transportation scenarios, for example, a sensor network would monitor the goods along the complete supply chain. In case of food it could for example monitor humidity and temperature. A recent research project in this context is Intelligent Container [3]. In such a setup, it is then possible to run small business processes on the sensor nodes, for example, calculating the final price the customer has to pay based on SLA and pricing models stored on the sensor nodes.
3) Specialized Application Fields
   We think that there is no general solution for interacting with real-world entities. While a service oriented approach with web services and SOAP or REST interfaces has worked pretty well in large enterprise IT systems, this is not necessarily the case in other application fields. Sensor network research has so far discovered a lot of different solutions for the specific needs of different applications and deployments. There exist a lot of specialized protocols for WSNs with different advantages and disadvantages (regarding reliability or bandwidth constraints for example), which are tailored for special usages.
4) Constrained Resources
   It is essential for the wide adoption of sensing technology in the enterprise IT that the related technologies come at a low cost. Therefore, we are dealing with devices, which are constrained in terms of memory,

computation and communication. In an industrial setting the usage of constrained devices is desired and often enforced to reduce the total cost.

5) Reconfigurability and Programmability
As more sensors are being deployed by enterprises, the evolution [4], shared use and reuse of already deployed sensor networks will play a crucial role. In a typical enterprise IT system a sensor network will be used for one or multiple tasks for some time. Changing requirements and cost pressure will lead to the need of a constant reconfiguration and shared use of resources. Applications, which time share a sensor network, might need to reflash a node to perform its task, due to the fact that sensor nets are usually memory constrained. Therefore, it is essential that the possible services and their requirements are properly defined.

There are already some approaches for some of the above mentioned drivers, most from the web of things domain. We will discuss these approaches in Section V. To address the aforementioned drivers of WSN integration into enterprise IT systems, we propose the usage of service descriptions, the Resource Description Format (RDF) and linked data.

II. Concepts and Terminology

In the following section we describe the basic concepts behind our approach. We briefly introduce Linked data, the Resource Description Format (RDF) and Service descriptions. Furthermore we define some additional terminology, as we use it in the following sections.

A. Terminology

The terms enterprise system, wireless sensor network and sensing service are defined in the following for the sake of clarity and completeness.

Enterprise System We use the term enterprise system (often called ERP or Enterprise Resource planning system) as follows [5]: *An ERP is a set of business applications that allows large enterprises to run all phases of an enterprise's operations to facilitate cooperation and coordination of work across the enterprise. The ideal enterprise system could control all major business processes in real time. Enterprise systems have in general high requirements concerning availability, scalability, reliability as well as security and interoperability.*

Wireless Sensor Network A wireless sensor network (WSN) is a network consisting of wirelessly connected small embedded devices (sensor nodes). The devices are equipped with one or more sensors, a microcontroller, radio transceiver, some memory and a power supply. Sensor nodes are often battery powered. Usually they are very restricted devices to reduce the cost per unit.

Sensing Service A sensing service utilizes a sensor network to perform a sensing task. A sensing service offered as part of a SOA architecture usually is a composition of many small sensing services, thus bringing the SOA perspective into the sensor-net itself.

B. Linked Data

The Linked Data principle [6], as introduced by Tim Berners-Lee, is based on the idea of using the Web to create typed links between data from different sources. Linked data is described in a machine-readable way with an explicitly defined meaning. It can be linked to and linked from other external data. Berners-Lee suggested the following rules for publishing data on the Web, forming one single data space (the so called *web of data*):

1) Use URIs as names for things
2) Use HTTP so that people can look up those names
3) When someone looks up a URI, provide useful information, using the standards (RDF, SPARQL)
4) Include links to other URIs, so that they can discover more things

In section III we will apply these four principles to sensor networks.

C. Resource Description Format (RDF)

The Resource Description Format (RDF) [7] is a set of W3C specifications. It is commonly used for representating structured, often distributed, information in a machine readable way. RDF decomposes information into statements, subjects and predicates, which form a directed graph. This very abstract model is sufficient to represent knowledge and process it in an application. RDF is often used in conjunction with ontologies and is often referred to as one of the central building blocks of the semantic web.

D. Service Descriptions

A service description is used to describe the characteristics of a service. This may include non-functional properties as well as a technical interface. It is important to distinguish between the deployed service itself (in a technical sense) and its description. The description, for example, does not need to be hosted on the same device as the service. A service description can exist without a deployed service. This is a necessary precondition for supporting reconfigurability and reprogrammability in an enterprise context.

III. Usage of Service Description Languages and Linked Data for Sensor Services

A. Service Characteristics

The modeling of service characteristics or service properties is essential for inclusion of sensor networks and sensor nodes into business process execution and for the SOA paradigm in general. Services have to be properly described with all their essential properties, so they can be found in a service repository or through some other discovery mechanism. From a business process orchestration point of view there is way more in a service than just its technical interface. The orchestration needs to know about Quality of service parameters, and compared to traditional approaches, it is now necessary to also model aspects which are specific to sensor networks and sensor nodes like an observation area (area covered by a sensor) or an observation schedule (when the area is covered by the sensor).

We propose to use the linked unified service description language (Linked USDL [8]) for describing these properties. Linked USDL has the advantage of being a service description language which goes beyond the technical interface — it consists of different modules which cover functional, operational and business aspects. Additionally, it allows the usage of already existing domain specific vocabularies, because it is modeled in RDF.

The following example for a room temperature service illustrates the usage of Linked USDL for the description of the basic properties of a sensing service:

```
<> rdf:type usdl:ServiceDescription ;
   dcterms:title "USDL service description for a
        temperature sensor";
   dcterms:creator :Matthias_Thoma ;
   owl:versionInfo "0.1";
   dcterms:contributor  [
     a foaf:Person ;
     foaf:name "Matthias Thoma";
     foaf:firstName "Matthias";
     foaf:lastName "Thoma" ] ;
   dcterms:created "2011-09-29"^^xsd:date .

:SensorTemperatureService a usdl:Service, msm:Service;
  usdl:hasNature usdl:Automated;
  usdl:hasServiceModel <http://research.sap.com/svc/
       sensors> ;
  dcterms:title "Temperature sensor service"@en;
  usdl:hasProvider :SAP_SENSOR_GROUP;
  usdl:hasInteractionProtocol :ip_1;
  usdl:hasImplementation <http://research.sap.com/bin/tsen
       .bin>;
  usdl:hasDocumentation <http://research.sap.com/doc/
       techspec.pdf> ;
  usdl:hasLegalCondition :termsAndConditions .

:termsAndConditions a legal:TermsAndConditions;
  dcterms:title "Terms and Conditions"@en;
  dcterms:description "Defines terms of use, liability,
       safety and so on."@en;
  legal:hasClause [ a legal:Clause;
    legal:name "Liability";
    legal:text "Legal, Warrenties etc."@en ] .

<http://research.sap.com/bin/tsen.bin> a usdl:Artifact;
  usdl:artifactType usdl:Software;
  dcterms:title "Binary for the sensor sofware" .

:SAP_SENSOR_GROUP a gr:BusinessEntity;
  foaf:name "SAP Sensors Service";
  foaf:homepage <http://www.sap.com>;
  usdl:legalForm "AG" .

:ip_1 a usdl:InteractionProtocol;
  dcterms:title "Read sensor value";
  dcterms:description "Read sensor data"@en;
  usdl:hasTechnicalInterface :SAPDataInterface_1 ;
  usdl:hasInteraction [
    a usdl:Interaction;
    dcterms:title "Get sensor data"@en ;
    usdl:hasOutput [ a usdl:Parameter;
                     gr:unitOfMeasurement "kelvin";
                     rdfs:label "temperature"@en ],
    ] .
```

The description starts with the general properties of the service. In the following detailed description of the actual SensorTemperatureService the underlying service model, the implementation, and the documentation are linked, and the legal usage conditions are referenced. The legal conditions, the implementation, and the service providing business entity are specified in the next two sections. In the last section the interaction protocol of the service is specified. Here the input and output parameters of a service are described: The sensor service has one output parameter for returning the measured temperature.

USDL also allows to attach service offerings to these sensor network services, which can be used for internal accounting or selling sensor network services in a shared sensor network. The following example adds a price plan and a service level profile to the SensorTemperatureService.

```
:offering a usdl:ServiceOffering;
   usdl:includes :SensorTemperatureService;
   usdl:hasPricePlan :price_plan_1;
   usdl:validFrom "2012-01-01"^^xsd:date;
   usdl:validThrough "2012-12-31"^^xsd:date;
   usdl:hasServiceLevelProfile :sl_profile_1.
```

The price plan could, for example, be defined on a sensor network level with a fixed setup price and an additional monthly fee for using that sensor service.

```
:price_plan_1 a price:PricePlan;
   dcterms:title "Price Plan"@en;
   price:hasPriceComponent [ a price:PriceComponent;
     dcterms:title "Fixed component";
     dcterms:description "Initial fee for setup";
     gr:hasCurrency "EUR";
     gr:hasCurrencyValue "15.00"^^xsd:float;
    ], [ a price:PriceComponent;
     dcterms:title "Monthly rate";
     dcterms:description "Monthly fee for service usage";
     gr:hasCurrency "EUR";
     gr:hasCurrencyValue "1.00"^^xsd:float;
     gr:unitOfMeasurement "per month"
    ] .
```

The RDF approach makes the utilization of existing vocabularies easy, as well as adding domain specific aspects. At this point, for sophisticated applications, and maybe even advanced reasoning within business process execution, we envision sharing or mapping these vocabularies between the service and the business process modeling and execution tools.

B. Service Description and Constrained Devices

We now apply the linked data principle to constraint devices. The URI can define either an individual sensor or a sensor network, depending on the application and the described service. This URI must be accessible via HTTP and must deliver a set of service descriptions. It can point to the sensor node itself, to a gateway, a repository or to some other place in the internet. It is not necessary that the endpoint has its implementation at the same sensor node, which it represents. The only requirements are that the URIs name actual things and (if feasible) to use HTTP, so that a lookup is possible. The service description and the corresponding technical interface and additional descriptions, as well, can be either on the sensor, off the sensor node or even both. This allows to keep only a minimum of required information on the sensor, and it allows to link to more comprehensive specifications, which can be stored off the sensor-node. This decoupling of service description, technical interface and access is essential for the usage on constrained devices. It allows cost reduction, and usage on devices, which are not HTTP enabled, but use proprietary protocols.

As an example, a sensor node could keep information about its service endpoint, its operations, its sensing area as well as some quality of service information on the node. Additional information, like accounting aspects, comprehensive technical information or different service level agreements, could then be linked to some enterprise repository. This repository then could link to a further repository of the nodes manufacturer for very detailed technical specifications. Another scenario are nodes, which do not have a SOA accessible service endpoint. It that case a SOA integration is still possible, because the service description can be accessed by the client, and this description delivers an endpoint somewhere on a gateway and thus can easily be integrated in a SOA environment as well.

C. *Reconfigurabily and Reprogrammability*

In this section we introduce the idea of service-level reconfigurability. The usage of service descriptions allows reconficurability and reprogrammability. As more sensors are being deployed by enterprises, the time-sharing, on-demand usage, and evolution within the lifetime of sensor networks will emerge. For this reason, there needs to be a link between the provided services, the corresponding code or binaries, and middleware or business process in order to support time-sharing, evolution and on-demand service level reconfiguration. This should not be confused with the traditional meaning of reconfigurability in sensor networks, which is usually associated with (dynamic) reconfiguration of a net, for example, when nodes join or leave it. This kind of reconfiguration is handled within the service.

As part of our service description we can directly link to one or more binaries (for example, for different platforms) in a binaries repository. The actual service delivery and binary deployment is future work that we plan to do as part of a more comprehensive sensor network service delivery and sensor network landscape management platform.

IV. Enterprise IT Integration

In our opinion the most important benefits of using service descriptions are to be seen at the level of the enterprise IT integration. In this section we present a high level architectural overview on how to link service descriptions and enterprise IT using SOA principles.

In Figure 1 we show one possible partitioning of a composed service with its different sub-services deployed

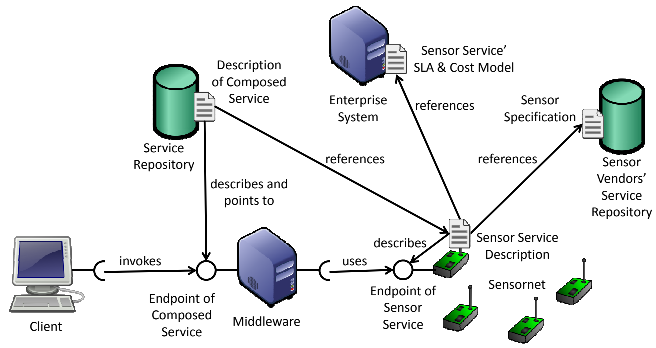

Figure 1. Typical sensor network integration scenario in an enterprise environment

on the sensor nodes. The composed service, which forms a sensor network service, has its accessible service endpoint on an enterprise middleware system. It references to the sensing services on the individual sensor nodes.

The sensor service descriptions on the individual nodes describe their technical service endpoint, which is on the same device in this scenario. Nonetheless, this is completely flexible. The service description there may, for example, link to some service endpoint on the gateway and to a service description on the gateway.

Furthermore, as also shown in Figure 1, the sensor nodes' service descriptions link to further information on an enterprise system (for example SLAs) and even to detailed comprehensive sensor specifications at some repository of the sensor vendor.

The whole architecture follows SOA principles. The sensor network services up to the middleware follow a classic repository based SOA approach. The sensor network beyond the gateway also follows SOA principles. Service descriptions and service endpoints allow, for example, the usage of individual sensor nodes and compositions thereof just like any other web service from the perspective of a business process execution engine.

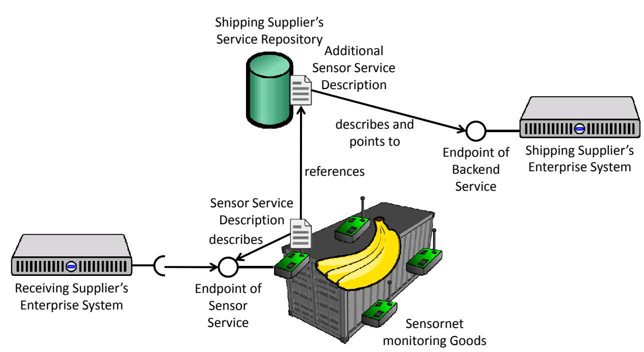

Figure 2. Sensor network migration in an enterprise context

On the other hand, in Figure 2, we show a different scenario. This time the ERP processes queries and accesses the sensor nodes directly and there is no explicit gateway. This scenario is quite common in a sensor network migration context. In transportation of goods, for example, we might have a sensor network monitoring one ore more shipping containers. These containers move along different stations in the supply chain and therefore their sensor networks need to be integrated into several enterprise backend systems. These backend systems get direct access to the sensor network and integrate them into their own network. Additionally, the description might reference to the original vendors backend system for additional information. For example, the Service Level Agreements could be stored there and is accessed for billing. Nonetheless, the sensor network needs to stay accessible even when this additional information is not available.

V. Related Work

The problem of integrating sensor networks and enterprise systems is well known in the literature. Gomez et. al. [5] propose an additional layer called Enterprise Integration Component (EIC), which is a generic mediation layer between enterprise systems and the WSN middleware. Our approach also allows the usage of a middleware like the EIC, but it additionally enables the usage of SOA principles on the side of the sensor network itself.

Glombitza et. al. [9] propose the usage of standard web service technologies. The problem of the high XML payload is solved by a light-weight transport protocol (LTP) for XML. They also aim for using SOA principles thus allowing the integration with BPEL, nonetheless their approach is based on WSDL and covers only the technical aspects of SOA.

Further approaches for integrating (lightweight) web technologies into sensor networks are, for example, CoAP [10] and Tiny Web Services [4]. CoAP implements a REST-based web transfer protocol build upon a subset of HTTP, which is optimized for constrained devices. On the other hand, Tiny Web Services implement a small (usable on nodes with only 48k of ROM and 10k of RAM) SOAP/WSDL based web server. We limit ourselves to the aspect of service description only. Therefore, we are independent of the actual transport or application layers and the used architectural style (SOAP or REST, for example).

Sensor description languages are used to model the characteristics, as well as the input and output parameters of sensing services. SensorML [11], for example, is an XML based modeling language, which allows specifying each sensor by its meta-data. It allows to model processes that are linked together through inputs and outputs.

Service descriptions, on the other hand, have attracted a lot of attention in the context of the internet of services. The most well known standard is the WS-* family which centers around the Webservice Description language (WSDL) [12]

The idea of using linked data for service descriptions has received wider attention recently. There are approaches known as *linked service* (e.g. [8], [13] and [14]), which contribute to the web of data by introducing or applying ontologies for service descriptions and discovery. iServe [15], for example, aims to support service publishing and discovery in a better way.

Most of the existing service description languages mainly focus on technical interfaces. We currently use an RDF-based version of USDL (Unified Service Description Language), which addresses — in addition to the technical aspects — business-related properties, capabilities and non-functional characteristics [16]. Nonetheless, any kind of RDF-based service description language could be used.

While our approach is inline with the ontology-based linked service approaches, which try to contribute to the web of data, we focus purely on services on and for sensor networks and sensor nodes, their capabilities and their integration into current enterprise IT systems. For example, it would be completely fine within our approach to just use our service description as a means to access a purely technical WSDL interface.

VI. Further Work

We plan to explore new ways to integrate sensor networks, and other building blocks of the Internet of Things, in a service oriented way into enterprise IT systems. Our research currently focuses on developing and evaluating an integrated business modeling and service delivery framework. Additionally, we foresee the need for landscape management, which makes evolution and time-sharing of corporate sensor networks possible.

Currently, there is no high-level service description language tailored towards embedded devices. We will explore how Linked USDL can be adapted for the description of services provided via small embedded devices. This includes the technical challenges of accessing RDF on sensor nodes, compressing RDF files, discovery mechanisms and handling of binary software code.

First steps on extending BPMN with concepts from the Internet of Things domain have already been done [17]. We will investigate how to integrate BPMN and our envisioned sensing service delivery framework. Research challenges in the field of time-sharing and evolution, as well as related business models will be addressed.

VII. Conclusion

In this paper we have shown the usefulness of service descriptions and their combination with the linked data principle for business process integration. Nowadays, the service oriented paradigm is predominant in an enterprise context. Therefore it is necessary to allow SOA integration in an standard compliant way. In this work we have chosen a representation based on RDF and shown the potential in combining embedded devices, such as sensor networks and sensor nodes, with techniques known from the semantic web.

ACKNOWLEDGMENT

The research on this topic received funding from the European Commission under grant 257521 (IOT-A) and grant 285248 (FI-WARE).

REFERENCES

[1] S. Haller, S. Karnouskos, and C. Schroth, "The internet of things in an enterprise context," in *Future Internet FIS 2008*, ser. Lecture Notes in Computer Science, J. Domingue, D. Fensel, and P. Traverso, Eds. Springer Berlin / Heidelberg, 2009, vol. 5468, pp. 14–28.

[2] L. Wanner, A. Hoeller, A. de Oliveira, and A. Frohlich, "Operating system support for data acquisition in sensor networks," in *Emerging Technologies and Factory Automation, 2006. ETFA '06. IEEE Conference on*, sept. 2006, pp. 582 –585.

[3] J. D. Gehrke, C. Behrens, R. Jedermann, and E. Morales, "The intelligent container- toward autonomous logistic processes," 2006.

[4] N. B. Priyantha, A. Kansal, M. Goraczko, and F. Zhao, "Tiny web services: design and implementation of interoperable and evolvable sensor networks," in *Proceedings of the 6th ACM conference on Embedded network sensor systems*, ser. SenSys '08. New York, NY, USA: ACM, 2008, pp. 253–266. [Online]. Available: http://doi.acm.org/10.1145/1460412.1460438

[5] L. Gomez, A. Laube, and A. Sorniotti, "Design guidelines for integration of wireless sensor networks with enterprise systems," in *Proceedings of the 1st international conference on MOBILe Wireless MiddleWARE, Operating Systems, and Applications*, ser. MOBILWARE '08. ICST, Brussels, Belgium, Belgium: ICST, 2007, pp. 12:1–12:7.

[6] T. Berners-Lee, "Linked data - design issues," 2006.

[7] J. J. Carroll and G. Klyne, "Resource description framework (RDF): Concepts and abstract syntax," W3C, W3C Recommendation, Feb. 2004, http://www.w3.org/TR/2004/REC-rdf-concepts-20040210/.

[8] T. Leidig and C. Pedrinaci, "Linked usdl," 2012. [Online]. Available: http://www.linked-usdl.org

[9] N. Glombitza, D. Pfisterer, and S. Fischer, "Integrating wireless sensor networks into web service-based business processes," in *Proceedings of the 4th International Workshop on Middleware Tools, Services and Run-Time Support for Sensor Networks*, ser. MidSens '09. New York, NY, USA: ACM, 2009, pp. 25–30. [Online]. Available: http://doi.acm.org/10.1145/1658192.1658197

[10] Z. Shelby, K. Hartke, C. Bormann, and B. Frank, "Constrained application protocol (coap), internet-draft," 2011.

[11] "Sensor model language," Open Geospatial Consortium, Tech. Rep., 2005.

[12] E. Christensen, F. Curbera, G. Meredith, and S. Weerawarana, "Web services description language (wsdl)," W3C, W3C Note, Mar. 2001, http://www.w3.org/TR/wsdl.

[13] S. Dietze, H. Yu, C. Pedrinaci, D. Liu, and J. Domingue, "Smartlink: A web-based editor and search environment for linked services," in *The Semanic Web: Research and Applications*, ser. Lecture Notes in Computer Science, G. Antoniou, M. Grobelnik, E. Simperl, B. Parsia, D. Plexousakis, P. De Leenheer, and J. Pan, Eds. Springer Berlin / Heidelberg, 2011, vol. 6644, pp. 436–440.

[14] S. Speiser and A. Harth, "Taking the lids off data silos," in *Proceedings of the 6th International Conference on Semantic Systems*, ser. I-SEMANTICS '10. New York, NY, USA: ACM, 2010, pp. 44:1–44:4. [Online]. Available: http://doi.acm.org/10.1145/1839707.1839761

[15] C. Pedrinaci, D. Liu, M. Maleshkova, D. Lambert, J. Kopecky, and J. Domingue, "iserve: a linked services publishing platform," in *Workshop on Ontology Repositories and Editors for the Semantic Web*, ser. ORES 2010, 2010.

[16] A. Charfi, B. Schmeling, F. Novelli, H. Witteborg, and U. Kylau, "An overview of the unified service description language," in *Web Services (ECOWS), 2010 IEEE 8th European Conference on*, dec. 2010, pp. 173 –180.

[17] K. Sperner, S. Meyer, and C. Magerkurth, "Introducing entity-based concepts to business process modeling," in *3rd International Workshop and Practitioner Day on BPML*, 2011.