

Introduction to Devices Orchestration in Internet of Things Using SBPMN

Alejandro González García, Manuel Álvarez Álvarez, Jordán Pascual Espada, Oscar Sanjuán Martínez, Juan Manuel Cueva Lovelle, Cristina Pelayo G-Bustelo
Computer Science Department, Oviedo University, Oviedo, Spain

Abstract — in this research we try to provide an architecture that allows the orchestration of objects that are part of the Internet of things creating business processes. Internet of Things is still in full development; this implies that there is a lack of standards for its proper implementation. Among these gaps is for example the technology used to allow objects to connect to the network, since there are several options but none seems to end imposed that is why this work try to provide architecture that imposes an alternative solution to this problem. However, it is difficult to provide a common solution to all the objects used in everyday life because of its great diversity, it requires us to classify them and thus create an appropriate architecture for each of the types These architectures are designed to facilitate the devices orchestration in a similar way as is currently done with web services enabling business process modeling.

Keywords — Internet of things, Orchestration, BPEL, SBPMN, SOAP, WSDL, REST, and WADL.

I. INTRODUCTION

MUCH time has passed since the completion of the first connection between computers in 1969, laying the groundwork for Internet. All this time this technology has been constantly evolving encouraged by the continuing advances in hardware and software, and the wide diffusion has had worldwide, from mere military application to be part of our daily lives. In recent years a new trend has emerged, networked objects that are part of our daily lives, the clearest example is mobile phones. This trend is called Internet of Things and is now a rapidly developing field that offers a wide niche research promoted by agencies such as the European Commission [1]. In literature there are examples of how serious our life thanks to the Internet of Things [2] but still no technology exists for doing that.

On the other hand the model-driven architectures seem to be gaining more strength, since the use of modeling techniques is used as a means to build applications simplified. Thus, the main part of the development of business concepts is through the development of the specification of the application, which abstracts the technical details. More and more these applications seem to be based on business processes. In particular the service coordination is gaining increasing acceptance as there are technologies such as BPEL widely consolidated for which is still being investigated as a solution to new problems [3], [4], [5], [6], [7], [8].

This suggests that the Internet of Things can be benefited from the progress in model-driven architecture, facilitating the orchestration of such objects to create business processes with

them. To do this we will rely primarily on the study in [9] which presents a notation for modeling business processes (SBPMN) that appears to be relatively easy and quick for users without technical skills.

II. SMART THINGS - THE INTERNET OF THINGS DEVICES

The aim of the Internet of Things is that all objects are connected to the Internet world, for it is necessary to provide these objects of some intelligence. Ergo incorporate in them certain hardware to enable them to communicate with the outside. These objects are called Smart Things. But we can ask the following questions: Should we treat all objects equally? Is it necessary to use the same technologies to communicate to the outside? Is it profitable to follow a standard procedure for all of them? From our point of view the answer is No. If we start to think about the objects that surround us every day we can find food on base, even our mobile phones. Food does not perform any function and the container is disposable, mobile phones are essentially mini computers today already are capable of connecting to the Internet and perform diverse tasks. We therefore believe that it is necessary to classify these objects. We will propose a taxonomy based on that processing power has and how complex it may be the architecture that can support the object. Following this criterion we classified the objects of the Internet of Things into 3 groups:

High-capacity devices - Type A

These are devices with high processing capabilities, architectures capable of supporting relatively complex and consume considerable bandwidth. For example would be able to publish Web services with SOAP and WSDL architecture. To this group belong a minority group of smart things, for example computers and next-generation devices.

Medium-capacity devices - Type B

These are devices with some processing power and withstand lightweight communications protocols that consume low bandwidth. For example would be able to publish REST web services technology. This group includes most of the machines involved in our daily lives, as they could be appliances.

Low capacity devices - Type C

These are devices capable of processing very low or negligible, can withstand very simple protocol based on hardware technology with which they were endowed with

intelligence, such as RFID tags. They would be able to offer such a simple protocol by its ID or other simple data. This group includes most of the objects of the Internet of Things.

III. ARCHITECTURE FOR THE ORCHESTRATION

We will propose the characteristics that should be the objects of each of the types according to its processing capacity to be orchestrated by SBPMN. It is important to note that since the objective is not to propose a complete architecture but to establish the bases of what we need to publish objects on the network. All the technology that is needed to succeed in providing intelligence to the objects and communicate these is being investigated in many papers and at different levels [10], [11], [12], [13], and [14]. In fact to begin research we rely on the ability of these objects might have to post something similar to web services. There are investigations as [15] Of particular interest to support our approach as they set an example of architecture applies to any object on the Internet of Things will be able to publish an HTML page or even a WSDL. This investigation is not intended to enter into discussions on whether this architecture is the most appropriate or not, since there is no even a specific standard in order to solve the challenge of communicating objects to the Internet of Things, however, that gives us foot to make a proposal based on service-oriented architecture. We will propose a set of features for each of the types of smart objects that have divided the Internet of Things in terms of its processing capacity, i.e. high capacity devices - Type A, medium-capacity devices - Type B, low capacity devices - Type C.

A. Architecture for high-capacity devices - Type A

This Type A devices are those that we classified as more intelligent, we understand that when an object receives this classification has a relatively high processing capacity and a range of consumption of relatively large bandwidth. For this is the least problematic group because we can rely on already established technologies such as SOAP and WSDL. In the WSDL will define the types and methods offered by that object to then publish them as Web services SOAP, for example a mobile phone may have a method to obtain its location, lock in case of theft or access the calendar, among others (Figure III-1).

We think this is the preferred choice, since technology exists for the coordination of web services based business process modeling (WS-BPEL), this specification in its original version is designed precisely to SOAP Web services with WSDL description services. To this we must add that there is a direct translation between the notation BPMN and WS-BPEL, which we can apply processing in [9] of SBPMN to BPMN to that from a business process carried out in which SBPMN involving Type A smart objects are made the relevant changes to the code is generated automatically to run these processes (Figure III-2).

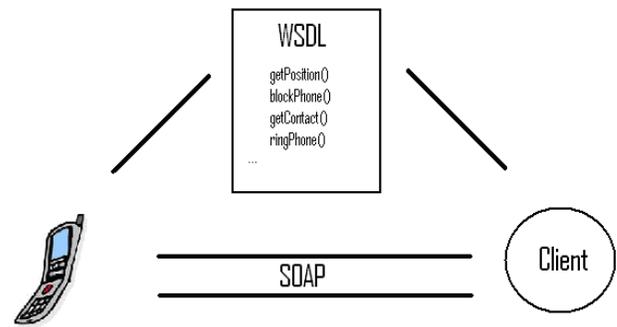


Figure III-1. Example of type A device communication

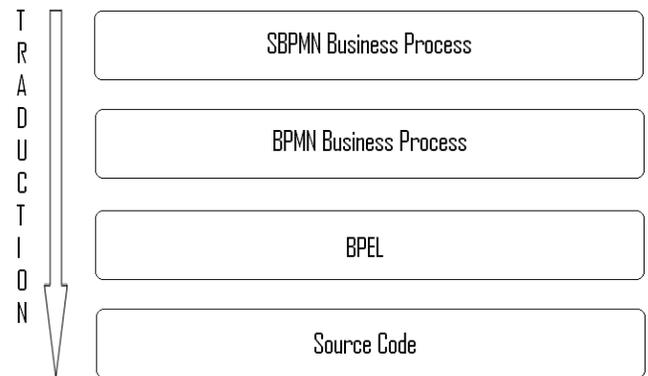


Figure III-2. Translation between technologies for Type A devices

To summarize our proposal for an object can enter the type A is to be able to publish their capabilities abroad in the form of methods for SOAP based web services and described with a WSDL.

B. Device Architecture medium capacity - Type B

These Type B devices are those that have qualified with a medium capacity, we understand that when an object receives this classification has some processing power and a range of consumption of limited bandwidth. This group is more complex because although we use relatively entrenched standards there is no technological coherence as in Type A. In this case we will use REST and WADL technologies. In WADL will define the types and methods offered by that object to then publish them as Web services, REST, for example, could publish an oven temperature and time schedule (Figure III-3).

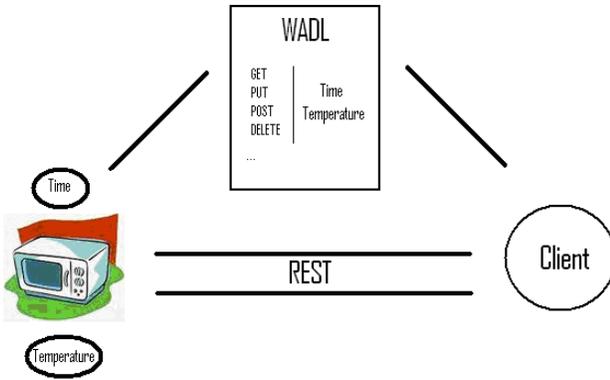


Figure III-3. Example of type B device communication

First we chose REST thinking we need an operation very similar to that in type A devices while taking into account the limitations of Type B. REST technology is lighter than SOAP because it does not add that extra layer above the HTTP protocol and consume less bandwidth by not using any type of packaging in communication as SOAP ago (Figure III-4).

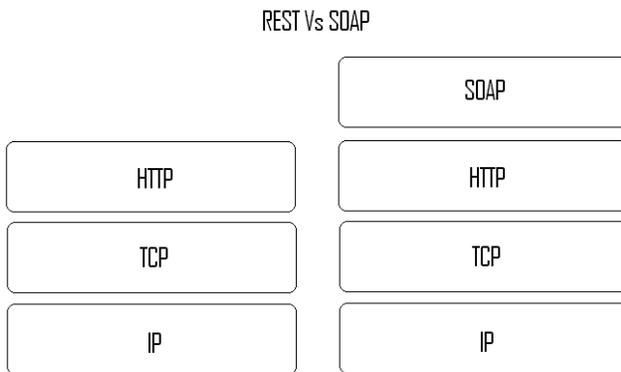


Figure III-4. Protocol stack REST vs SOAP

WADL’s election as a service description of REST services may be somewhat controversial for several reasons:

- WSDL 2.0 can be used with REST services [16]: A clear rivalry between WADL and WSDL 2.0 as both compete as a service description for REST. In work [17] as a comparison of these two technologies coming to the conclusion that they are very similar, but have a few differences. Taking into account these differences we decided to opt for WADL due to:
 - WSDL 2.0 is oriented to interfaces description while WADL is oriented to resource description, which agrees more with the REST philosophy.
 - WADL is simpler, and not a drawback to this research that only supports the http protocol.

Although theoretically the natural evolution of BPEL is to obtain WSDL 2.0, WSDL 1.1 standard is strongly rooted and there are many services in this format. WSDL 2.0 is not yet well established, especially to describe the current API REST services, and there is little evidence that this

situation will change in the future [8].

- There is a discussion about whether they really need a REST service description service as WADL [18] [19]: Theories against using these services are essentially that we not need to define procedures as REST services by relying solely on default HTTP means that their operations are GET, POST, UPDATE and DELETE and data types are defined in XML Schema to which it refers. Of the bids for this research can highlight the use of this kind of services facilitates the self in code. Later we explain the fundamental reason why we have opted for WADL.

The main problem we met him at the time of creating the business process with Type B devices and go making changes to the source code that is executed. While the Type A from a business process SBPMN could make a transformation to transform BPMN to BPEL for later in this case the final transformation is not possible because the BPEL only supports SOAP and WSDL originally. However, there are several extensions to meet the new challenges that arise in the coordination of web services and BPEL: BPEL-SPE [3], BPEL4People [4] BPEL4JOB [5], BPEL-DT [6] or BPEL-light [7].

In particular in the work [8] proposes a new extension to allow the use of REST in BPEL. In this research precludes the use of WADL based mainly that most of the REST APIs described using services through human-readable documentation or examples of use because this specification is still recent. Although you get a solution to use REST services for our research we found that this solution is relatively against the fundamental principle of SBPMN notation is abstract the business user of the technical specifications. We understand that this work is necessary to understand the technical documentation of service to BPEL subsequently needed to program the code, while the use of WADL could be implemented automatically as is the case with WSDL in the original specification of BPEL.

Therefore in order to execute business processes involving Type B devices will be necessary to implement an extension of the WS-BPEL for using REST services with WADL. In figure III-5 we can see the evolution from business process to the generated source code execution.

To summarize our proposal for an object can enter the type B is to be able to publish their resources abroad in using Web services with REST and described WADL.

C. Architecture for low capacity devices - Type C

These Type C devices are those that have qualified with a low or almost zero capacity, we understand that when an object receives this classification is not able to post any type of Web service. This group is technologically much simpler to get a chain of devices and processes in accordance with a prearranged agreement. Even so neither will have the technological coherence that we had in Type A. In this case we will provide further details on the hardware necessary to make these objects intelligent type C. The objects will be tagged

with passive RFIDs will be interpreted by a reader to obtain a data string information. For example the packaging of a food product labeled with an RFID could post the code that identifies it and its expiration date (Figure III-6).

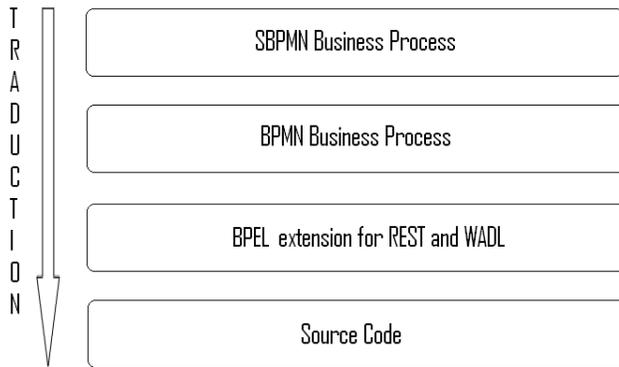


Figure III-5. Translation between technologies for Type B devices

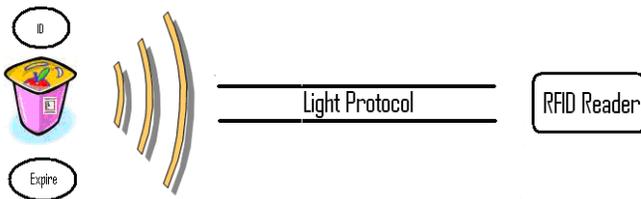


Figure III-6. Example of type B device communication

There is a wide range of RFID tags on the market and greatly varying size and quantity of information they can provide [20]. This influenced the choice of the characteristics of the proposed lightweight protocol. In the relatively open is trying to leave the size that these chains may have obtained by reading the labels. These chains have the following segments:

- ID (mandatory): This is the only mandatory field is the identifier that has been printed on the label for that object.
- Data (optional): This segment represents additional information that the object wants to communicate. Turn is divided into three sections that will be mandatory.
 - Value: The value of data to be transmitted
 - Type: Indicates the type of data, namely, numeric, text, date or Boolean.
 - Description: Briefly describe the data.
- Additional information (optional): full details are to be added to the information transmitted by the object.

By convention establish the character = is the boundary of each of the segments. In Figure III-7 we can see three examples of what these objects could transmit the transmission from simple to more complex.

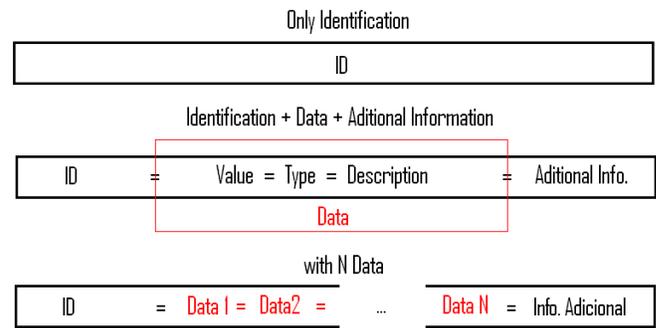


Figure III-7. Example structure of the strings sent by Type C devices

Despite the simplicity of the proposed technology, since there is no need to publish any type of service, we have a problem similar to that of type B and there is no existing technological coherence in Type A. Therefore in order to execute business processes involving C-type devices will be necessary to implement an extension of the WS-BPEL to the correct interpretation of the lightweight protocol. In Figure III-8 we can see the evolution from business process to the generated source code execution.

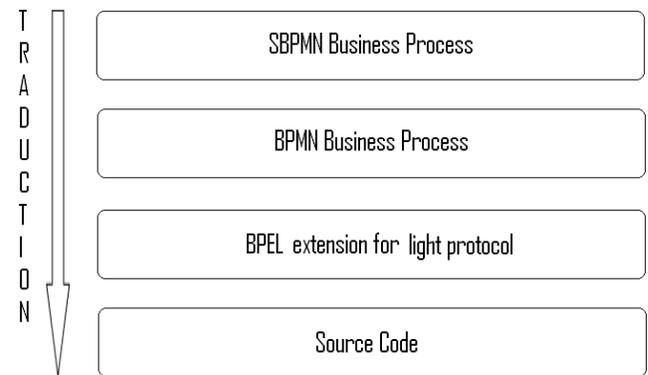


Figure III-8. Translation between technologies for Type B devices

To summarize our proposal for an object can enter into the Type C is that the string read from the label of the object has a structure proposed in this section.

D. Summarizing

We have seen how this classification can save the limitations of the hardware available on the Internet of Things objects by choosing a particular group of technologies. However, this does not mean that objects can not acquire sufficient capacity reserved for technologies lower groups. This is important approached from the perspective of the debate between REST and SOAP [21], [22], [23]. While there is strong disagreement between advocates of one or another, all generally agree that SOAP is oriented procedures while REST is resource-oriented. With the proposed architecture allows a choice to use technology or other information depending on the object you want to publish, as long as they have sufficient capacity to accommodate the technology.

We can see all the proposed architecture summarized in

Table I.

TABLE I
ARCHITECTURE FOR INTERNET OF THINGS DEVICES

	TYPE A	TYPE B	TYPE C
COMMUNICATION	SOAP	REST	Lightweight protocol: Interpretation of the string read
DESCRIPTION	WSDL	WADL	-
ADVANTAGES	SBPMN direct translation of the source code executed.	Functionality similar to type A with lower resource consumption.	Simplicity.
DISADVANTAGES	High capacity. Bandwidth consumption.	It is necessary to implement an extension of BPEL for REST and WSDL.	It is necessary to implement a BPEL extension to the protocol slightly.

IV. SBPMN AND INTERNET OF THINGS DEVICES

In the previous section we have proposed a number of technologies that should be used to perform a SBPMN orchestration in terms of their processing capacity. The next step in this research is defined as represent each of these types with SBPMN.

TABLE II
SBPMN COMMON ELEMENTS FOR ORCHESTRATING INTERNET OF THINGS DEVICES.

ELEMENT	GRAPHICS REPRESENTATION	DESCRIPTION
Data Object Web Reference		Allows to model data that are processed during a process flow diagram on BPMN. The data objects can represent many different types of electronic or physical. In particular an external object that refers to an external interface
Automatic task		Business process that can not be divided into threads. A basic unit processes. In this case because it automatically means it is automatically executed by a machine.
Textual annotation		Allow further describe the associated item of business process.

Of the proposed elements in the notation SBPMN be used basically three: Task automatically to web reference data and textual annotation. In Table II are explained briefly.

In general we will use these three elements as follows to suit our purposes:

- **Web Reference:** Represents the object involved in the modeling. It may be accompanied by the identifier of the object or some type of name or reference.
- **Automatic Task:** Represent the functionality normally published by the objects.
- **Textual annotation:** Add additional information about the object or its features.

In the following sections we will see how this would be applied to each object type and restrictions will be applied.

A. Type A device representation with SBPMN

To carry out the modeling of business processes involving Type A devices will need a Web reference that represents the device that will be orchestrated and automatic task for each published method.

If we take the example given in Section III-A mobile phone (Figure III-1) and we had a tool that would allow us to model what is proposed in this research, we should be displayed in the component palette similar to what we see in Figure IV-1.

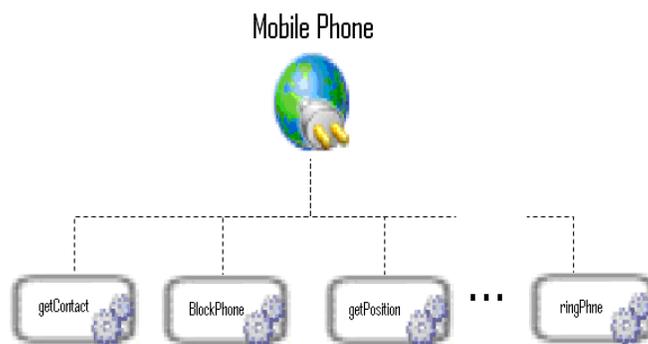


Figure IV-1. Example of components for the orchestration of Type A devices.

We can see the direction in proceedings in this representation due to influence of the technology used (SOAP). In general, these devices can model involving activities, reading, writing or some kind of processing.

B. Type B device representation with SBPMN

The modelling of Type B devices is similar to the type A in terms of components but conceptually different but similar way as do their protocols reported REST and SOAP.

If we take the example given in Section III-B, smart oven (Figure III-3) and we had a tool that would allow us to model what is proposed in this research, we should be displayed in the component palette similar to what we see in Figure IV-2.

As previously discussed this technology is oriented to REST resources. In the palette presented to us the resources published by the device as well as web references the four methods that we have to interact with them in the form of automated tasks. In general, these devices can model involving activities, reading or writing.

C. Type C device representation with SBPMN

Device modeling of Type C is the most conceptually different from the other two, as happened with the proposed

architecture, but only used a different item, the textual annotations.

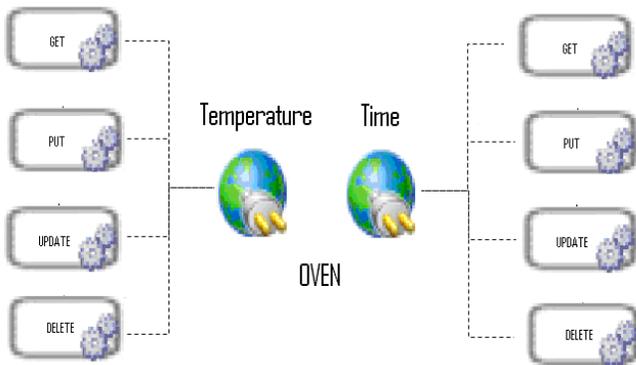


Figure IV-2. Example of components for the orchestration of Type B devices.

Taking the example proposed in section III-C container labeling (Figure III-6) to publish a series of data with the structure presented in (Figure III-7) and we had a tool that would allow us to model what is proposed in this research, we should show in the component palette similar to what we see in Figure IV-3.

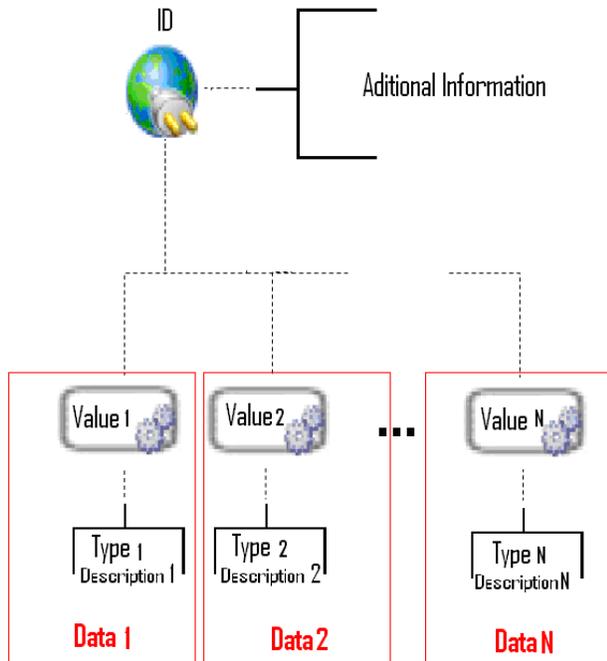


Figure IV-3. Example of components for the orchestration of Type C devices.

For such devices see the additional information and description and type of data allows the user to have enough information of the elements to be used despite not having a service description. Remembering the proposed structure for strings (Figure III-7) shows how to create a Web reference to the segment ID an annotation concerning this textual reference, with additional training segment (if it exists) and automatic activity with a personal annotation for each segment

of data that is sent. In particular, the activity will read the data value and the information that appears in the annotation text will be the data type and description, which are the segments that are subdivided Data segment. As a last point to note that these devices allow only read operations.

V. CONCLUSION AND FUTURE WORK

In this work we have laid the groundwork for continuing research on the devices orchestration in Internet of Things:

- We have proposed Taxonomy for the Internet of things objects.
- We have proposed architecture for each of the types described in the taxonomy. These architectures also provide some flexibility if those objects have sufficient resources may use the technology you want, this is important from the point of view of the debate between SOAP and REST because we can use one type or another depending on whether procedures aim to publish articles or rather offering resources.
- We have proposed SBPMN representation for each of the types described in the taxonomy.

Among other points of future development can include:

- Extending BPEL for web services and REST-based technologies WADL.
- Extension BPEL to orchestrate objects that communicate with the proposed lightweight protocol.
- Development environment for BPEL orchestration and the proposed extensions to SBPMN.
- Development of a series of pilot applications in different platforms to enable the orchestration of devices, Internet of Things in real time through a simulated environment.
- Testing the usability of these applications with business users.

REFERENCES

- [1] ITU Report. "The Internet of Things". 2005.
- [2] EC-EPoSS Workshop Report "Internet of Things in 2020 A ROADMAP FOR THE FUTURE". 2008.
- [3] IBM; SAP. "WS-BPEL Extension for Sub-Processes". <http://xml.coverpages.org/BPEL-SPE-Subprocesses.pdf>. 2005.
- [4] SAP; IBM. "WS-BPEL Extension for People". http://download.boulder.ibm.com/ibmdl/pub/software/dw/specs/ws-bpel4people/BPEL4People_white_paper.pdf. 2005.
- [5] Tan, Wei; Fong, Liana; Bobroff, Norman. "BPEL4JOB: A fault-handling design for job flow management." Springer Berlin / Heidelberg. 2007.
- [6] Habich, Dirk; Richly, Sebastian; Grasselt, Mike; Preissler, Steffen; Lehner, Wolfgang; Maier, Albert. "BPEL-DT – dataaware extension of BPEL to support data-intensive service applications". <http://ftp.informatik.rwth-aachen.de/Publications/CEUR-WS/Vol-313/paper7.pdf>. 2008.
- [7] Nietzsche, Jörg; van Lessen, Tammo; Karastoyanova, Dimka; Leymann, Frank. "BPELlight". Springer Berlin / Heidelberg. 2007.
- [8] Pautasso, Cesare. "BPEL for REST". <http://www.jopera.org/files/bpm08-bpel4rest.pdf>. 2008.
- [9] Fernández, Héctor; Palacios-González, Elías; García-Díaz, Vicente; Pelayo-Bustelo, Cristina; Sanjuán Martínez, Oscar; Cueva Lovelle, Juan Manuel. "SBPMN — An easier business process modeling notation u for business users". 2009.

- [10] Balkesen, Cagri. "An internet of things infrastructure". <http://www.inf.ethz.ch/personal/cagri.balkesen/pdf/epc.pdf>. EPCEGlobal. 2008.
- [11] Arnold, Hermine. "UCODE EPC Gen 2 Objective Product Information". http://www.gs.ttw.org/twct/web/epc_rfid2006/JoelO.pdf. Philips. 2006.
- [12] IPSOAlliance white papers. <http://www.ipso-alliance.org/Pages/DocumentsAndWhitePapers.php>. 2010.
- [13] MIT. "Internet 0: Past, Present, and Future". <http://cba.mit.edu/events/04.09.I0/>. 2004.
- [14] Organization dedicated to research in pervasive computing. <http://www.uidcenter.org/index-en.html>. 2010.
- [15] Urien, Pascal; Chabanne, Herve; Pépin, Cyrille; Bouet, Mathieu; Cunha Daniel, Guyot, Vincent., Pujolle Guy; Paradinas, Pierre; Gressier, Eric; Susini, Jean Ferdinand. "The T2TIT research Project. Introducing HIP RFIDs for the IoT". http://www.cs.kuleuven.be/conference/WoSSIoT2007/Jean-Ferdinand.Susini_T2TIT.pdf. 2007.
- [16] Mandel, Lawrence. "Describe REST web service with WSDL 2.0". <http://www.ibm.com/developerworks/webservices/library/ws-restwsdl/>. Rational Software Developer, IBM. 2008.
- [17] Toshiro, Takase; Makino, Satoshi; Kawanaka, Shinya; Ueno, Ken; Ferris, Christopher; Ryman, Arthur. "Definition Languages for RESTful Web Services: WADL vs. WSDL 2.0". <http://download.boulder.ibm.com/ibmdl/pub/software/dw/specs/ws-wadlwsdl/WADLWSDLpaper20080621.pdf>. 2008.
- [18] Burke, Bill. "To WADL or not to WADL". <http://bill.burkecentral.com/2009/05/21/to-wadl-or-not-to-wadl/>. 2009.
- [19] Gregorio, Joe. "Do we need WADL?". <http://bitworking.org/news/193/Do-we-need-WADL>. 2007.
- [20] Abraham, Cherian; Ahuja, Vinay; Kumar, Ghosh, Pakanati, Praveen. "Management using Passive FRID Tags: A Survey". <http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.94.6599&rep=rep1&type=pdf>. 2002.
- [21] Zur Muehlen, Michael; Nickerson, Jeffrey; Swenson, Keith. "Developing web services choreography standards—the case of REST vs. SOAP". Decision Support Systems Volume 40, Web services and process management. 2004.
- [22] He, Hao. "What is a Service-Oriented Architecture". <http://www.xml.com/pub/a/ws/2003/09/30/soa.html>. 2003.
- [23] Zimmermann, Olaf; Pautasso, Cesare; Leymann, Frank. "Restful web services vs. "big" web services: making the right architectural decision". International World Wide Web Conference. 2008.