2013 IEEE International Conference on Green Computing and Communications and IEEE Internet of Things and IEEE Cyber, Physical and Social Computing

A Domain Model for the Internet of Things

Stephan Haller
Vigience LTD
Horgen, Switzerland
e-mail: haller@vigience.com

Alexandru Serbanati
Computer Science Department
Sapienza University of Rome
Rome, Italy
e-mail: a.serbanati@uniroma1.it

Martin Bauer
NEC Laboratories Europe
NEC Europe Ltd.
Heidelberg, Germany
e-mail: martin.bauer@neclab.eu

Francois Carrez
Centre for Communication Systems Research
University of Surrey
Guildford, United Kingdom
e-mail: f.carrez@surrey.ac.uk

Abstract— **By bringing together the physical world of real objects with the virtual world of IT systems, the Internet of Things has the potential to significantly change both the enterprise world as well as society. However, the term is very much hyped and understood differently by different communities, especially because IoT is not a technology as such but represents the convergence of heterogeneous – often new – technologies pertaining to different engineering domains. What is needed in order to come to a common understanding is a domain model for the Internet of Things, defining the main concepts and their relationships, and serving as a common lexicon and taxonomy and thus as a basis for further scientific discourse and development of the Internet of Things. As we show, having such a domain model is also helpful in design of concrete IoT system architectures, as it provides a template and thus structures the analysis of use cases.**

Keywords: IoT, Domain Model, Internet of Things

I. INTRODUCTION

The Internet of Things (IoT) is a very much talked-about concept today. While there is some common, but fuzzy, understanding that the term "Internet of Things" denotes the convergence of the physical world with the virtual world, there exists no commonly accepted definition of it. Even worse, underlying concepts and terminologies are used differently by different communities. For example, what is the "Thing"? Is it a device, like a sensor or a machine? Or is it a real-world object to be tracked, like some pallet? Both points of view can be observed both in industry as well as in academia. The authors have encountered this issue again and again, always leading to endless discussions and eventually mutual misunderstanding.

Many promises have been made about the benefits of the Internet of Things. Transparent interaction between physical and digital world will enable enterprises to optimize their processes and to become more agile and adaptive to what happens in the real world. Along with the ability of sensing and instrumenting the real world, new businesses with new business models will emerge. This new kind of interaction will also benefit consumers so that they know what they are buying. Advances in Health Care and remote monitoring will enable better care at home, and through smart grids, society will consume less energy. These are just some of the promises; many more exist [5][11][12]. These scenarios require more than just integrating a few devices within the four walls of a company ("Intranet of Things"), but require a technology that enables the cooperation and interoperability between different stakeholders, potentially involved in very different application fields, often called silos. The path leading to such cooperation and interoperability starts with establishing and ensuring a common understanding of the domain one is working in.

Defining a domain model is a first, but important step in this direction. The main purpose of a domain model is to generate a common understanding of the domain in question. According to [13], it should provide a common lexicon and taxonomy, defining the main concepts and their relationships. A domain model needs to be stable, hence it needs to abstract from implementation details: it should separate out what doesn't vary from what does [14].

The domain model for the IoT as presented in this paper has been developed within the IoT-A project [6]. One of the main goals of this project is to develop an *Architectural Reference Model* (ARM) for the IoT, a second full version of which has been released in October 2012 [7]. In addition to the domain model, the full Reference Model (part of the ARM) also includes information, communication, functional and security/privacy/trust models.

The rest of the paper is structured as follows. After describing existing approaches in this field (Section II), we describe in detail our proposed IoT domain model (Section III). This model is then illustrated with two use cases (Section IV), showing how the concepts in the domain model can be applied to concrete application scenarios. Finally, we conclude and give some outlook on future work (Section V).

978-0-7695-5046-6/13 $26.00 © 2013 IEEE
DOI 10.1109/GreenCom-iThings-CPSCom.2013.87

II. EXISTING APPROACHES

In the literature, a plethora of definitions of IoT can be found. They are most often in textual form (which usually brings ambiguities) and therefore cannot be used as a primary step in architecting an IoT system [1][3][4]. In addition they often diverge and no real consensus has been reached. However there exist few initiatives that have attempted to come up with a "formal" definition of the IoT domain model. Two major pieces of work by Haller, S. [10] and Serbanati *et al.* [20] have been the "starting points" of the Domain Model we have been developing. To name just a few more, Casagras [2] has proposed an *Inclusive Domain,* with strong focus on RFID technology. Their model is much less complete and comprehensive than the IoT-A one and is an isolated piece of work while the IoT-A one is one component of whole Architectural Reference Model for IoT. In [15] Patel *et al.* extend the Casagras model and propose a complete (but isolated) model comparable to our model. Barnaghi *et al.* proposed an interesting model enriched with Semantics – mainly WS&AN and Knowledge centric– partially based on the SENSEI project results [18]. Both were considered in our work. In addition we are aware of Chinese initiatives in defining Domain Model for IoT, which are unfortunately not well documented or available. The next section presents our IoT Domain Model.

III. THE IOT DOMAIN MODEL

In the domain of the IoT, five core concepts can be discerned:

- **Augmented Entity (AE)**: The combination of the real-world object (Physical Entity) together with its digital representation (Virtual Entity);
- **User**: As the IoT can be regarded as a technical system with some purpose, one needs to look at who uses it and how this relates to the system. Depending on the usage, users vary from human beings -with dedicated roles- to machines, devices, services…;
- **Device**: Hardware to monitor or interact with real-world objects. Devices can also provide the connectivity to IT systems;
- **Resource**: Computational element that gives access to information about, or actuation capabilities on a real-world object;
- **Service**: Services expose the Resources through a common interface and make them (as well as related Devices) available for applications and other Services.

It is important to note that, bridging the physical and digital domains, the IoT domain model contains component of both types. When designing a new complex IoT system, it is worth noting that at the IoT Domain Model level, components can pertain to one or more domains. When approaching the design in a specific domain however (e.g. software design) the domain model can be projected on that specific domain. This process preserves the coherence of complex systems design and enables a better understanding of the different dimensions of a system.

In the following sections, these core concepts and their relationships are explained in more detail, including additional sub-concepts.

A. Users and Interactions

At the base of the domain model there is the requirement of the generic User to interact with a real world object. These objects can be monitored, moved or otherwise manipulated by the User. In order to avoid misunderstandings, the term *Physical Entity (PE)* has been introduced. Examples of PE's include pallets and trucks in supply chain management applications, machinery and tools in manufacturing, ice cream cabinets and vending machines in retail applications, but also animate objects like cows and sheep in herd monitoring, or vines and other crop in environmental monitoring. The PE also corresponds to what has been called Entity of Interest in previous work [10][18].

Traditionally, the interaction between User and PE is carried out directly and physically: observing or touching the real-world object, moving it, or even changing it. IoT introduces a new way to carry out this interaction mediating it through software interfaces and electronic devices. Figure 1 shows the interaction pattern at the base of the IoT paradigm.

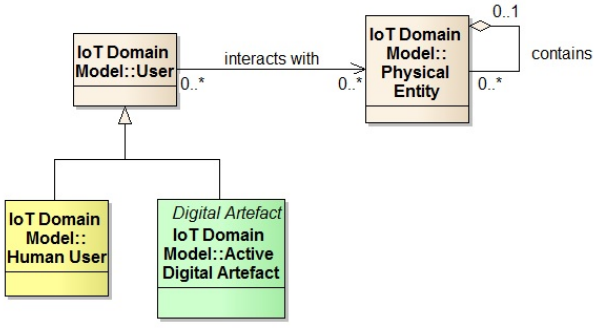

Figure 1. Different types of Users interact with Physical Entities

The introduction of the interaction mediation through digital means has also paved the way for the automation of processes as tasks usually performed by human users can now be delegated to software agents pertaining to the digital world. Thus, in the IoT Users can either be humans, e.g., a warehouse worker moving some pallet, or in *Machine-to-Machine* (M2M) scenarios, an *Active Digital Artefact* (ADA) like a program embedded in a manufacturing robot.

B. Augmented Entity

As mentioned above, the IoT is about the convergence of the real world with the digital one. Thus, a PE always has a virtual counterpart [16], or as we call it, a *Virtual Entity* (VE) to represent it in the digital world of IT systems. A VE must be associated to Resources available at least to some Users. The IoT-mediated interaction of the User with the PE is in

fact an interaction with the VE, e.g., looking up information about a PE or moving it via some actuation Resources, and is achieved through Services, as described later on.

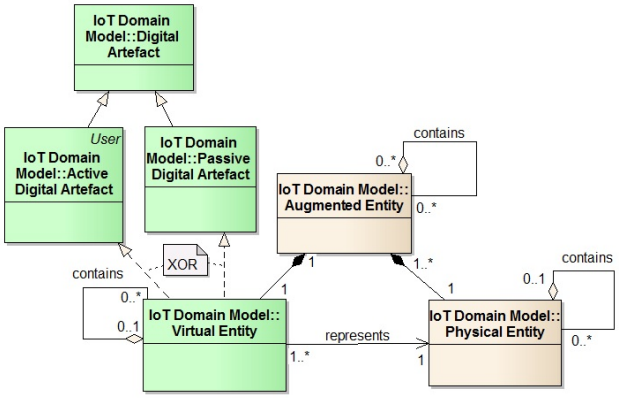

Figure 2. An AE consists of the PE and its digital counterpart, the VE

In a software system, a VE is a kind of digital artefact that represents a PE. It is noteworthy that digital artefacts can be characterized as either passive or active and VEs can be of both types:

- *Passive Digital Artefact* (PDA): A digital representation of something – in this case a PE – stored in a database or similar form;
- *Active Digital Artefact*: Any type of active code or software program. In this case it would usually be some sort of software agent or embedded application.

While ideally one would assume to have just a single VE per PE, in practice this is not the case: firstly, a PE that has a software agent (i.e., an ADA) acting on its behalf would usually also have some information stored about it in a database (PDA). Secondly, there might be separate representations of the same PE within different systems. For example, a returnable container might be represented differently in the asset management system of the container owner and in the supply chain management system of the user of the container for shipping goods.

The features of the PE are reflected as attributes in the VE. The set of attributes is very different for every VE and therefore not in the scope of a domain model. One important attribute worth mentioning however is the location of the PE, as it determines if it shows up in the result set of certain queries or not. The location is also the basis in order to provide location-based services, one of the key features of the IoT.

The AE finally is the composition of a PE with its associated VE and can be considered as the "things" in the Internet of Things. As we can have hierarchies of PE's – e.g., a pallet containing cases –, there can also be hierarchies of VE's as well as AE's.

C. Device

The Device is a piece of computing hardware and is the superclass for the more specific hardware that enables the IoT by establishing the connection to the physical world, actually bridging the digital and physical world. Devices can be physically attached to PE's (e.g., a tag), but they may also be in the environment of PE's (e.g., presence sensor).

Sensors allow the monitoring of PE's, whereas actuators can act on PE's. Tags uniquely identify PE's and can be read by sensors. Devices can be composed of Devices, e.g., a node in a sensor and actuator network may be composed of multiple sensors, actuators and general computing hardware.

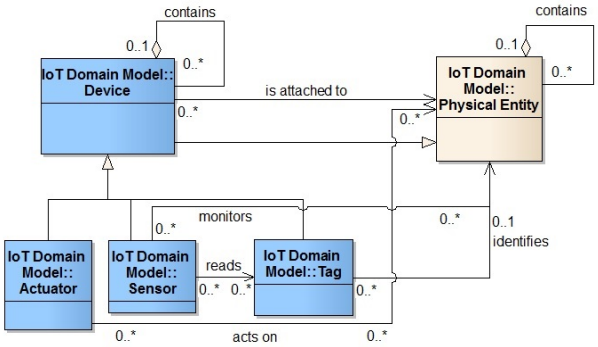

Figure 3. Device types

As a Device is part of the physical world, it is of course itself a PE. The relationships between Devices and PE's are shown in Figure 3.

D. Resource

Resources are software components that implement certain functionalities, i.e., they provide information about PE's or allow the execution of actuation tasks. Resources may be hosted on a Device or they could be hosted anywhere in the network, for example in the case of a processing Resource that derives higher level information by analysing data provided by multiple sensors.

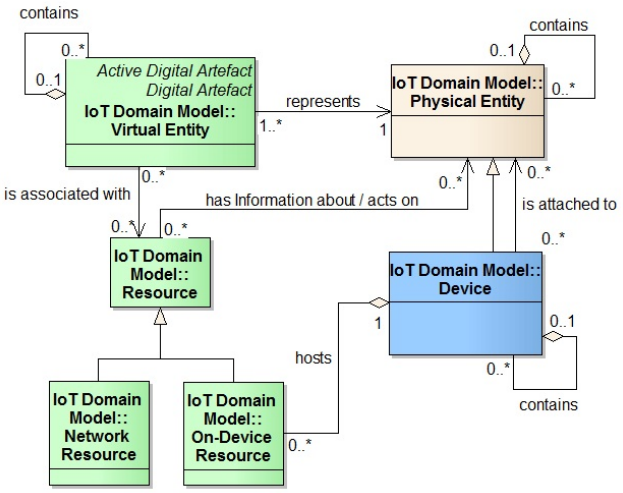

Figure 4. Resources

Resources can be found through associations that link VE's with Resources. These associations can be either relatively static or very dynamic, depending on the relation between the Device and the PE that is represented by the VE. So if the Device is attached to the PE, the association is rather static, whereas if Devices and the PE's only happen to be in the same environment and are mobile, associations may be changing rather dynamically. Resources and their relations are shown in Figure 4. Resources associated with VE's are also responsible for updating the attributes of the VE, or in other words, to make sure that the attributes reflect the true state of the related PE.

E. Service

Resources provide the functionality, but may run on restricted Devices that may not be able to expose a suitable interface enabling users to interact with it. Therefore, the concept of the Service was introduced. It exposes a standardized interface that can be invoked by the user. The Service has the internal knowledge how to access the functionality provided by a Resource and can do additional things like caching results, supporting additional interaction styles, e.g. push/pull, and help achieving scalability, e.g., through load balancing.

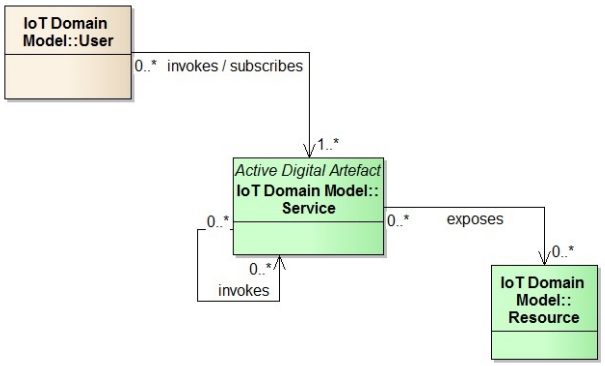

Figure 5. Services enable Users to access Resources

Services can also be hierarchically structured, so that one Service may invoke other Services and combine the results. All relations regarding services are shown in Figure 5.

F. Overview of the IoT Domain Model

The complete domain model we propose is shown in Figure 11 at the end of the paper (for the sake of readability). The concepts shown are coloured according to the form that instances take: hardware is shown in blue, software in green, and animate objects in yellow. Where multiple forms are possible, brown is used.

G. Deployment options

Figure 6 shows different deployment options for Users, Services, Resources and Devices. The leftmost configuration is suitable for a powerful Device that can host Resources, services and even users like active digital entities and applications. User applications may also be hosted anywhere, e.g. in the cloud, using the same API as in the location case. The third deployment configuration is also suitable for more limited Devices. Only the Resource itself is hosted on the Device. The Service can be hosted anywhere, e.g., on a gateway or in the cloud. The Service then communicates with the Resource through some internal access interface and exposes the API to Users.

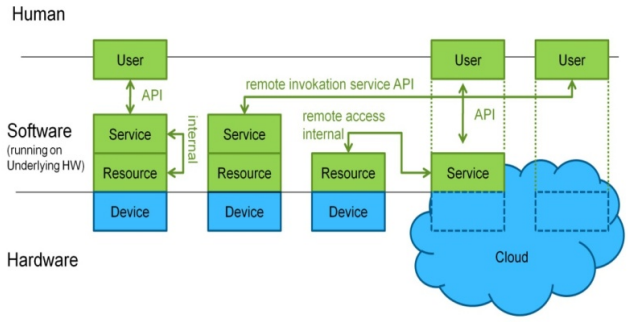

Figure 6. Deployment Options

IV. Applying the IoT Domain Model

The IoT Domain Model as described above enables a clear understanding of the IoT domain, and the exact vocabulary that comes with it facilities discussions between practitioners in the field. On top of that, it also serves as a blueprint for analysts and architects when designing concrete IoT systems. The typical design process will first identify from the requirements set the objects and the actors that need to interact with them along with the interaction functionalities that have to be provided. In terms of the Domain Model, this means that the following scheme will be followed:

1) Identify the Users of the system;
2) Identify the involved PE's and related VE's;
3) Identify the Resources;
4) Map Resources to Device types.

Once this mapping is complete, it can be used as a basis for communicating within the design team and with the stakeholders. Analysts can then delve deeper into the details of the designed system by analysing the all of the system's facets. This activity should be performed in parallel with the development of the system's communication, information and access control models because they are tightly intertwined.

Once these models have been used to define the system-specific mappings, an implementation of the System Functionalities (e.g. Discovery, Lookup, Resolution) is needed [7]. This implementation will cover the steps that are needed before the Endpoint-to-Endpoint communication. Various architecture options are available for implementation purposes in IoT-A deliverables [9].

In order to demonstrate the applicability of the IoT Domain Model, we provide below a concrete example where it is applied to a logistics scenario. We identify the involved

entities and the necessary mappings according to the process outlined.

Logistics is one of the areas where RFID technology is already used to a significant extent. Currently the granularity is often still on the level of pallets rather than on individual items and the tags only allow identification. In the future we can expect item-level granularity. Goods may also carry additional information provided by sensors. Such information may relate to the freshness of the good or the temperature range and the physical pressure it was exposed to during transport and storage. Goods may carry their transportation route and the CO_2 footprint relating to production and transportation.

The resulting information can improve the quality of products as potential damage during production and transportation can be recognized earlier and it can be determined who is responsible for it. Based on the information, the end consumer can make a more informed choice regarding the product to select.

Figure 7 shows users that may be involved in the logistics chain from production to the retail store where the product will be sold. Apart from the human users two ADA's are shown, a smart item that may internally monitor conditions and, for example, trigger some kind of actuation if the storage conditions are outside required limits. Second, an automatic transport agent can steer an automatic transport shuttle in a storage area.

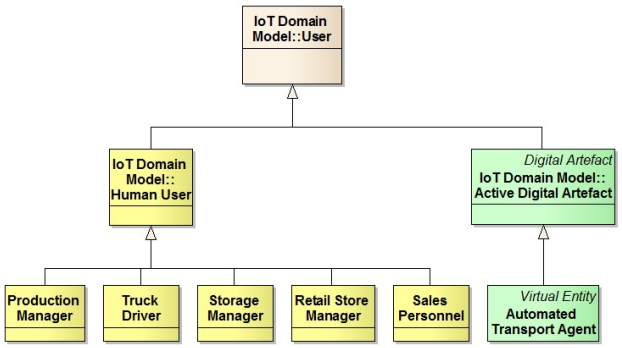

Figure 7. Example of User sub-classing in the design of a logistics system

Figure 8 shows classes of AE's in a logistics scenario, which consist of PE's and their virtual counterparts VE's. These include pallets, items, storage space, transport shuttles within a storage area, and trucks. Most of the PE's are modelled as PDA's, however the smart item and automated transport agents are ADA's that have some intelligence to signal the status of a smart item and to steer an automated transport shuttle in a storage area respectively.

In Figure 9 a number of Resources and Services exposing them are shown. Most of the Resources are On-Device Resources, i.e., represent functionality that is provided by an IoT device. The History Storage Resource however is a Network Resource that may run in the cloud storing history information like the storage condition of an item during production, transport and storage.

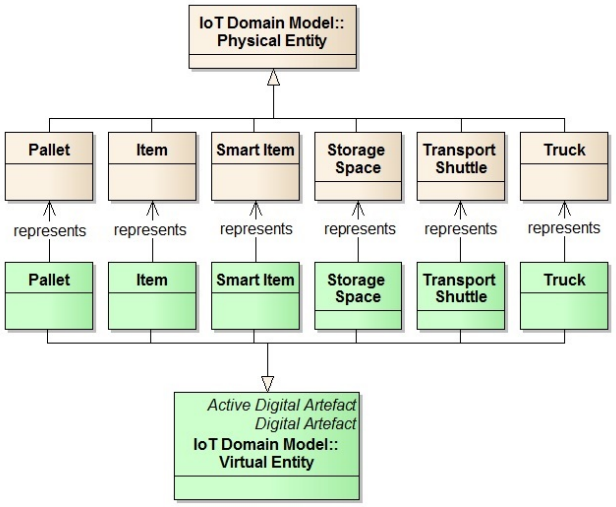

Figure 8. Examples of Augmented Entities in a logistics scenario

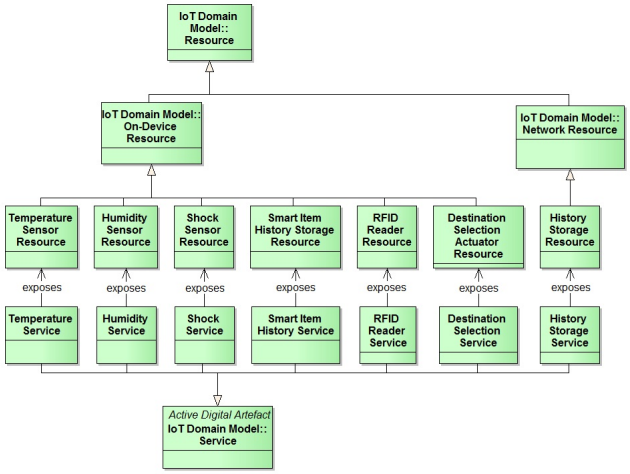

Figure 9. Examples of Resources and Services in a logistics scenario

Figure 10 shows sub-classes of devices that are relevant in a logistics scenario. There are a number of different sensors that may be deployed in storage areas and trucks, but may also be attached to smart items themselves. This is shown through the "contains" relationships between the smart items and the respective sensors. The transport shuttle also consists of sensors and actuators. As the transport shuttle is represented as an Active Digital Artefact, the respective functionality may not (directly) be exposed, but rather used internally for steering the transport shuttle. A higher-level Resource is exposed through a Service for selecting the destination as is shown in Figure 9

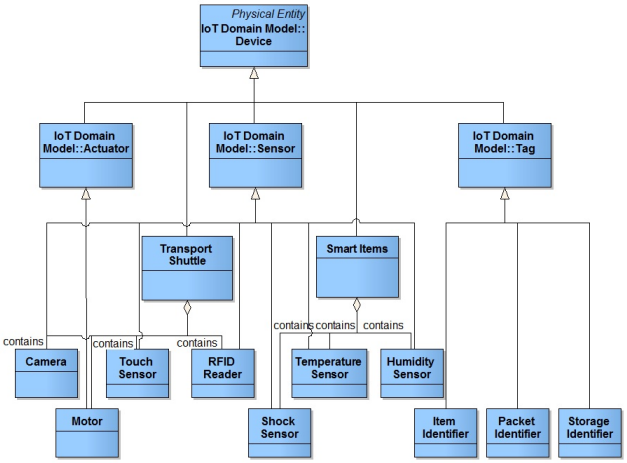

Figure 10. Examples of Devices in a logistics scenario

Other examples have been considered by IoT-A and will be available shortly in the "IoT-A book" (please check on the IoT-A web site). It is also worth mentioning that in order to demonstrate the compatibility of our Domain Model with existing IoT architectures we did a few "Reverse-mappings" of those architectures to the IoT Domain Model [8], mapping their concepts and relationships to ours.

V. CONCLUSION

If we allow ourselves a short metaphor we could argue that a Domain Model in the context of system architecture plays a similar role as do ontologies in the context of semantic and knowledge sharing: it defines a point of reference for understanding a domain and agreeing on its definition, using some formalism; in our case UML diagrams. The Domain Model presented in this paper represents an attempt of defining IoT in formal terms. IoT-A goes much further than just defining a Domain Model. Our Architectural Reference Model provides in addition a whole set of models (briefly mentioned at the beginning of this paper) and a comprehensive set of Views and Perspectives [17] in addition to a large set of guidelines that can be used to derive concrete IoT architectures out of an abstract meta architectural framework (i.e. the ARM). The IoT-A project has established a lot of contacts in Europe and world-wide (China, Taiwan, Japan, Australia, Brazil,..) and is actively presenting and discussing the Domain Model with those overseas IoT communities, the obvious objective being to reach consensus on the definition of the IoT Domain. We believe in IoT-A agreeing on such a model is the first step on the path leading to the raise of an eco-system of compatible and interoperable IoT systems. The final Version of the ARM will be available by end of June 2013 on the IoT-A public website [8].

ACKNOWLEDGMENT

The authors would like to thankfully acknowledge the support for this work provided by the European Commission within the FP7 project IoT-A (FP7-257521). A special thanks goes to Martin Strohbach (NEC) and SAP for their contributions to the Domain Model.

REFERENCES

[1] Auto-ID: "What is the Internet of Things" available at http://www.im.ethz.ch/education/HS10/AUTOIDLABS-WP-BIZAPP-53.pdf

[2] Casagras: "RFID and the Inclusive Model for the Internet of Things" http:// http://www.grifs-project.eu/data/File/CASAGRAS FinalReport (2).pdf

[3] CERP-IoT: in Chapter 1 of " Internet of Things Strategic Research Roadmap" available at http://www.internet-of-things-research.eu/pdf/IoT_Cluster_Strategic_Research_Agenda_2009.pdf

[4] CISCO: "The Internet of Things: How the next Evolution of the Internet is changing everything" available at: http://www.cisco.com/web/about/ac79/docs/innov/IoT_IBSG_0411FINAL.pdf

[5] IoT European Research Cluster (IERC) Cluster Book: "The internet of Things 2012: New Horizons". ISBN 978-0-9553707-9-3.

[6] IoT-A Website: http://www.iot-a.eu

[7] IoT-A D1.4: "Converged Architectural Reference Model for the IoT V2.0". Available at http://www.iot-a.eu/public/public-documents

[8] IoT-A D1.5: "Final Architectural Reference Model for the IoT V3.0" available on 30th June at http://www.iot-a.eu/public/public-documents

[9] IoT-A Deliverables of WP2 and WP4 at http://www.iot-a.eu/public/public-documents

[10] Haller, S. "The Things in the Internet of Things", Poster at the Internet of Things Conference, Tokyo (IoT 2010). Available online at www.iot-a.eu/public/news/resources/TheThingsintheInternetofThings_SH.pdf.

[11] Haller, S., Karnouskos, S. and Schroth, C. "The Internet of Things in an Enterprise Context", in J. Domingue, D. Fensel und P. Traverso (Eds.), "First Future Internet Symposium - FIS 2008", LNCS 5468, Springer Verlag 2009, pp. 14-28.

[12] Libelium: "50 sensor application for a better world". Available at: http://www.libelium.com/top_50_iot_sensor_applications_ranking/

[13] Mueller, G. "A Reference Architecture Primer", 2008. [Online]. Available at www.gaudisite.nl/referencearchitectureprimerpaper.pdf,

[14] Oldfield, P., "Domain Modelling", 2002. Available online at: www.aptprocess.com/whitepapers/DomainModelling.pdf,

[15] Patel, P., Pathak, A., Teixeira, T. and Issarny, V. "Towards Application Development of the IoT". ACM/IFIP/USENIX 12th International Conf. (2011) available at: http://hal.archives-ouvertes.fr/docs/00/71/12/66/PDF/a5-patel.pdf

[16] Romer, K., Mattern, F., Dubendorfer, T. and Senn, K. "Infrastructure for Virtual Counterparts of Real World Objects" Department of Computer Science, ETH Zurich, 2002 (Available online)

[17] Rozanski, N. and Woods, E. "Software Systems Architecture – Working with Stakeholders Using Viewpoints and Perspectives", Addison Wesley, 2011

[18] "SENSEI Architecture Whitepaper" at http://www.ict-sensei.org/index.php?option=com_content&task=view&id=63&Itemid=58

[19] De, S., Barnaghi,P., Bauer, M. and Meissner, S. "Service Modelling for the Internet of Things". Proceedings of the Federated Conference on Computer Science and Information Systems. Pp. 949-955,

[20] Serbanati, A., Madaglia, C.M. and BiaderCeipidor, U., "Building Blocks of the Internet of Things: State of the Art and Beyond", in Deploying RFID - Challenges, Solutions, and Open Issues, ISBN 979-953-307-026-0, InTech, 2011

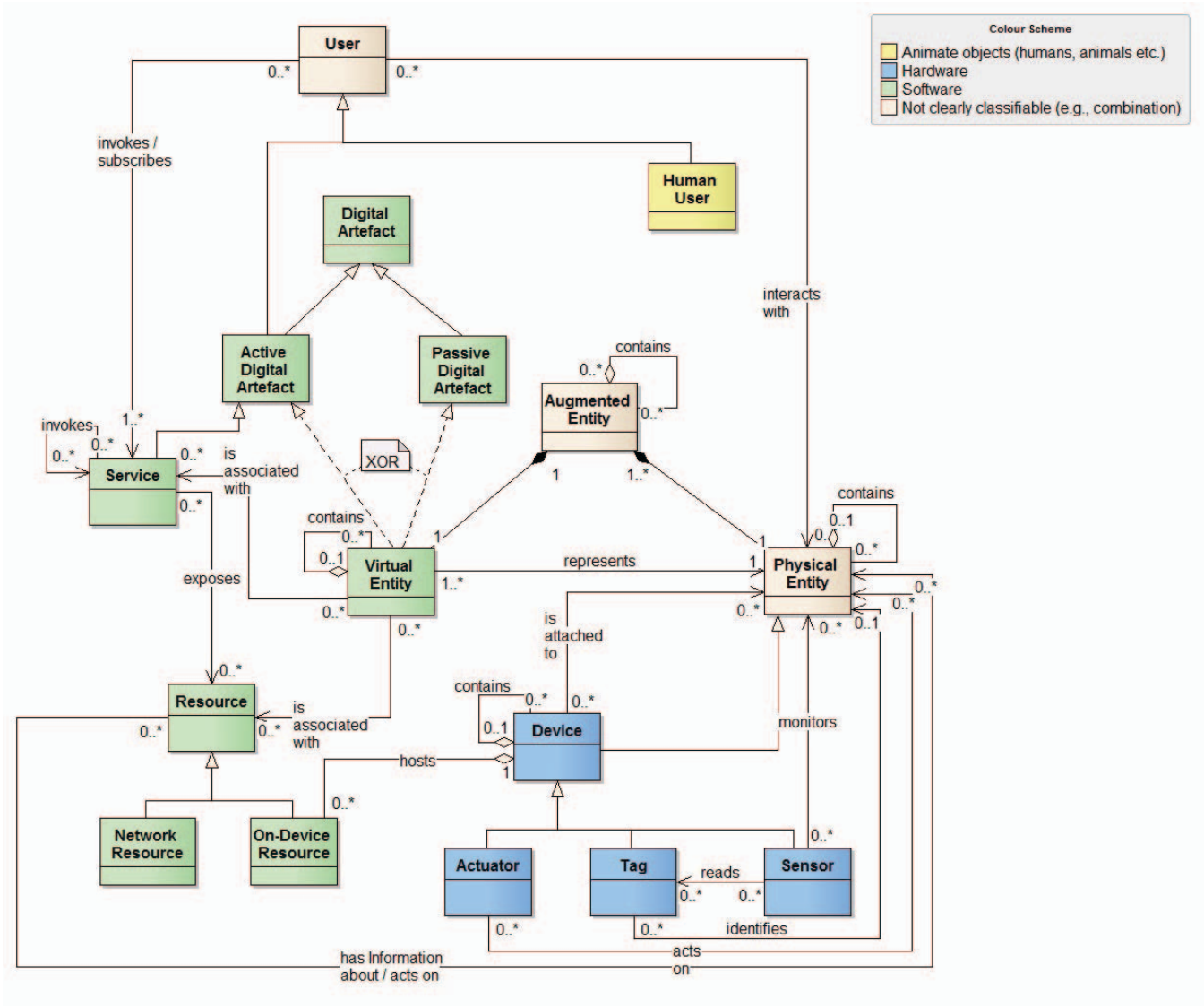

Figure 11. The full Domain Model for the Internet of Things