IEEE COMMUNICATIONS SOCIETY

Ad Hoc and Sensor Networks Technical Committee (AHSN TC) Newsletter

CONTENTS

PREFACE

The IEEE ComSoc Ad Hoc and Sensor Networks Technical Committee (AHSN TC) aims at sponsoring scientific and technical activities facilitating the dissemination of knowledge in the areas of ad hoc, sensor and mesh networks. In an attempt to make all the TC members as well as the AHSN worldwide community aware of what is going on within our main areas of concerns, this newsletter had been set up*. The newsletter aims at inviting the authors of successful research projects and experts from all around the world with large vision about AHSN-related research activities to share their experience and knowledge by contributing a short news. So, the fifth issue of the AHSN TC Newsletter features one high quality news item gently provided by Dr. Sylvain Cherrier (Université Paris-Est). We thank them as well as all the previous contributors for their effort to make this newsletter successful towards fulfilling its objectives.

Newsletter Co-editor

Yacine Ghamri-Doudane, ENSIIE and Université Paris-Est, France (EiC)
Sidi-Mohammed Senouci, University of Burgundy, France (Co-EiC)
Mohamed Fathy Feteiha, Queen's University, Canada
Burak Kantarci, University of Ottawa, Canada
Damla Turgut, University of Central Florida, U.S.A.

**The AHSN TC Newsletter has been started in 2008, and then re-started in 2012 after 2 years of a stop due to the sad and tragic loss of our friend and colleague Professor Mieso Denko, from the University of Guelph (Canada), who was one of the initiator and an active leader of this Newsletter. In his memory the newsletter had been re-started in 2012 as a tribute to his life's work in research, education and services to the networking community as a whole.*

TC OFFICERS AND NEWSLETTER EDITORS

TC Officers

Names	Affiliation	E-mail
Azzedine Boukerche	University of Ottawa, Canada	boukerch@site.uottawa.ca
Nei Kato	Tohoku University, Japan	kato@it.ecei.tohoku.ac.jp
Jalel Ben Othman	University of Paris 13, France	jalel.ben-othman@univ-paris13.fr

Editor-in-Chief

Names	Affiliation	E-mail
Yacine Ghamri-Doudane	ENSIIE and Université Paris-Est, France	ghamri@ensiie.fr

Co-Editor-in-Chief

Names	Affiliation	E-mail
Sidi-Mohammed Senouci	University of Burgundy, France	sidimohammed.senouci@orange-ftgroup.com

Co-Editors

Names	Affiliation	E-mail
Mohamed Fathy Feteiha	Queen's University, Canada	feteiha@cs.queensu.ca
Burak Kantarci	University of Ottawa, Canada	kantarci@site.uottawa.ca
Damla Turgut	University of Central Florida, U.S.A.	turgut@eecs.ucf.edu

Towards Crowd-Centric Service Composition enabling the realisation of the Internet of Things Paradigm

Sylvain Cherrier

Université Paris-Est, LIGM Lab,
5, bd Descartes Champs sur Marne 77454 Marne-la-Vallée Cedex 2, France.
sylvain.cherrier@univ-paris-est.fr

Introduction

A new world in which our environment provides services, interacts with you and the things you own, that fits to your needs, depending on your choices, your specifications, and that helps you to enjoy it, are the promises of the arising domain of the "Internet of Things". Though, the buzz surrounding this new field of computer technologies is still facing many problematics who need to be resolved in order to enable its real-life raise-up. A global solution, both adapted to the constraints of these new elements (the "Things" or "Objects") and compatible with the existing Internet architecture must be defined to provide a viable framework for the development of this field.

The needs

First, let's give a definition to the Internet of Things. Internet is this well-known interconnection of all public networks that have changed in the last decade our vision of Data Processing. On the other hand, the concept of "Things" is a deliberately fuzzy word used to define our everyday-objects which were added computing capabilities and network connectivity (from phone to light bulb, cars, furnace, door...). The Internet of Things is a gap from a data computing point-of-view to a physical one. It is the foundation of "Home Automation", but also "Smart buildings" and "smart Cities" notions, expanding the scope of computers to the "real (physical) world".

However, the main issue that limits the emergence of this extension of the Internet is the huge heterogeneity of stakeholders involved in the process. Varied networks (in their capabilities and their protocols), high variability of processing power and memory size of involved elements, wide range of physical sensing and actuating actions provided by Objects, the number of parameters is so important that it inhibits the building of common and universal interactions between Objects.

Our vision is based on overspread and standardised interactions between the various components of the IoT, either already known Internet-provided services (such as social networks and web services) or these new smart objects (smart phones, sensors, actuators and so on). Our aim here is to consider each object as a provider of basic services, and then to build a composition of a set of these offered basic services in order to build applications that suits their targeted users. This can be made through two concept we are describing in the following: (i) hardware abstraction of the objects and (ii) an interaction abstraction among the basic services that the objects are offering.

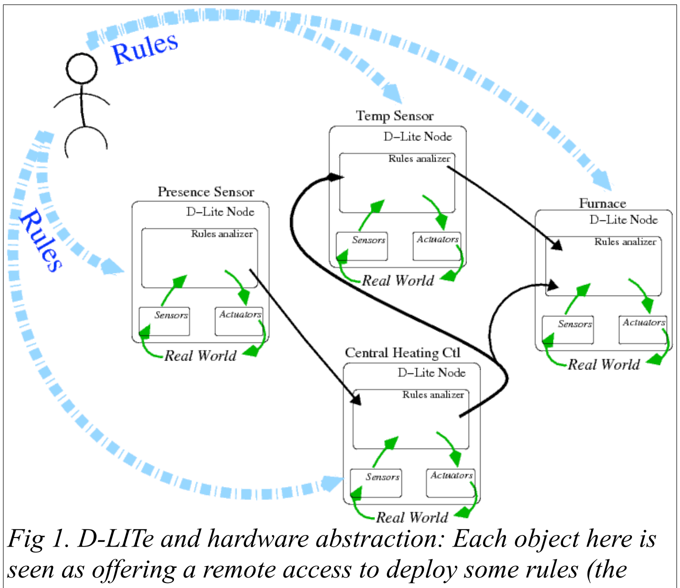

Fig 1. D-LITe and hardware abstraction: Each object here is seen as offering a remote access to deploy some rules (the logic to execute). All objects interact, using the services provided by others. The link with the physical world is offered through an abstraction.

Hardware abstraction

To reach our goal, we need to consider each object as a web services provider. To do so, we propose D-LITe[2] (Distributed Logic for Internet of Things sErvices) that uses the standard IPv6 and REST protocol (an architecture using the

HTTP protocol to allow service interactions). Each object running D-LITe becomes accessible through the Internet. Its capabilities can be discovered using standard verb of REST protocol (GET), the logic the user wants this object to obey can be pushed towards it (PUT), and the object can then interact in a standard way with other D-LITe objects (POST). D-LITe provides a universal view of any object, no matter if its a small and constrained one, or if it is a traditional server in the Internet. All of them can be combined to compose an application, based on interactions between the services they provide (see Fig 1).

This "services" approach is already applied on the Internet when using the SOC (service oriented computing) architecture, and has proven its effectiveness in term of "loosed coupling" (giving the opportunity to change the service or even its implementation without altering its uses. But the large variety of "Objects" characteristics and their capabilities to sense/actuate the real world remains an issue. To give a real universal representation of Objects, D-LITe provides a hardware abstraction layer that hides real object programming under a virtual "feature". With D-LITe, any kind of real (or virtual) element that has a "Button", for example, offers a feature named "Button". Any object that wants to be informed that the button has been pressed will received always the same standardised message, no matter the signal generated by the real button within the object it equips.

This unique combination of services approach and hardware abstraction layer gives all D-LITe object the ability to run generic programming codes. When a programmer designs an application, he combines "features" effects. Then, with the standard use of the REST protocol, he can deploy codes on each object, depending of its features and the role he want this object to have in the whole application (see Fig 1). Then the application starts, using REST protocol to exchange messages and making objects interact through the Internet.

Interactions abstraction

D-LITe when installed on objects gives the programmer a simple and universal view of all its objects, no matter their real characteristics. However, building real applications for the Internet of Things does not require that users will program, or, if it is the case, the coming of this domain will be much slower and limited. Even if the programming model proposed by D-LITe is simple and accessible, the need of applications is so important that we can't assume every user will become a D-LITe programmer.

In a “Services” approach for the Internet of Things, we organise our application by composing elements and making them interacting with each others. D-LITe offers a standard way of describing the role of each Object inside the application. But if we want to give users the ability of building their own applications for their own uses, we need to help them in that composition.

Developing an application using a mash-up is a common way of composing easily elements to obtain a global response to a given problem. To get the same ease of elements composition for the Internet of Things, we looked at the diversity of possible exchanges between objects and how to provide solutions to combine a wide range of basic behaviours.

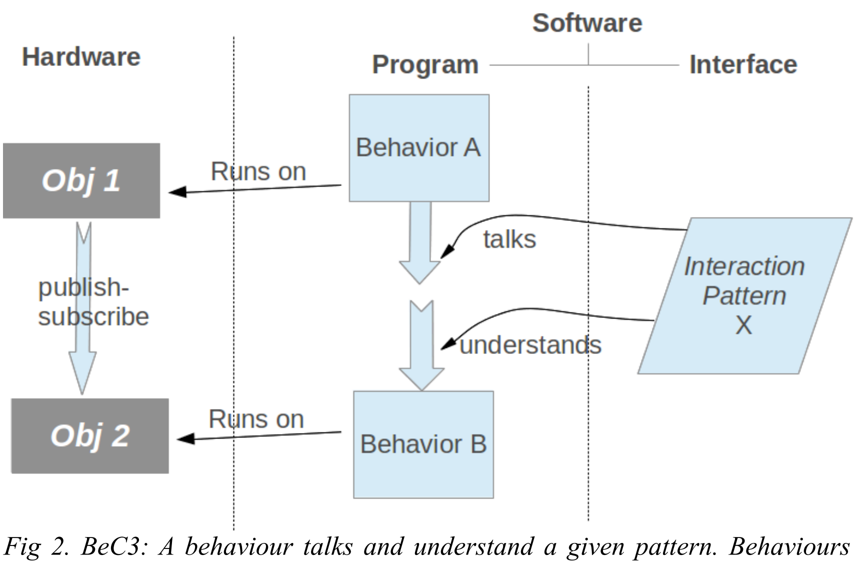

Fig 2. BeC3: A behaviour talks and understand a given pattern. Behaviours run on an object according to the features it provides. The global composition of Behaviours can be checked following each Behaviour constraints.

BeC3[1] (Behaviour Crowd Centric Composition for IoT applications) is what we propose to address this issue. We have identified the specific interactions between objects in common IoT applications. We have obtained a list of "Interaction Patterns" that are mainly used. We have named them and gave them a description for their characteristics. For example, "Boolean Interaction" describes the way one thing may command another one. "Bounded counter" is used to provide gliders or drivers for example, as it describes positions and their evolution over time. Then, each code, called "Behaviour" in Bec3, written for D-LITe will be categorised regarding their needs into "Interaction Patterns". With the model proposed by BeC3, a user can choose for each object a Behaviour (depending on object's feature), and compose all objects together. BeC3 will then check if all needs in terms of "Interaction Patterns" are addressed by the "Behaviours" composition (see Fig 2), and if so, codes will be deployed on Objects and the application will start. If not, the needs that are not covered will be displayed to be resolved by the user.

Finally, to provide a rich base of element giving the user a wide range of combinations, the Crowd orientation of BeC3 gives the possibility to anybody to enrich the Behaviours proposal, and even to provide new D-LITe port on new devices. Fig 1 gives an example of an application made with BeC3 and D-LITe. Using our approach, each object offers a way to program it, and a standard communication with others. In this example, the user describes the interactions between his central Heating controller, his furnace, a temperature sensor as well as a presence sensor. Choosing among “Behaviours” (proposed by other users) for each object (according to its features), this user can build a home automation application that can manage his home depending on the current temperature and his current presence or not.

Conclusion

The Internet of Things and the applications that we will be able to use in the future still faces a major gap in the way we want to interact with the world around us. The “Things” are so heterogeneous and basic that it is still complex to realise such interactions. D-LITe is a solution that allow us to understand each object as a universal representation, and to access this representation in a standard Internet way. By providing such a hardware abstraction layer, D-LITe hides the differences between elements providing the same feature, allowing their replacement indifferently. Then, BeC3 offers to users a simple way of deploying dynamic pieces of code, and to validate their composition following rules that describe each element expectations. In this Newsletter article, we highlighted the basic ideas behind D-LITe and BeC3. For more details about these two complementary mechanisms for building Internet of Things applications, the reader can refer to [1] and [2].

Bibliography

[1] BeC 3: Behaviour Crowd Centric Composition for IoT applications
S. Cherrier, I. Salhi, Y.M. Ghamri-Doudane, S. Lohier, P. Valembois
Springer Mobile Networks and Applications (MONET), Volume 19, Issue 1 , pp 18-32, February 2014.

[2] D-LITe: Distributed Logic for Internet of Things sErvices
S. Cherrier, Y.M. Ghamri-Doudane, S. Lohier, G. Roussel
2011 International Conferences on “Internet of Things” and ,”Cyber, Physical and Social Computing” (iThings/CPSCom), October 2011, Dalian, China.