W3C Workshop on the Web of Things

Enablers and services for an open Web of Devices

Position Paper from DQuid

Gabriella Castelli, Stefano Marzani

Introduction

The Web of Things is here to come: we have technologies that enable our everyday objects to expose real-time data over the Internet and there are plenty of already connected objects. On Thingful[1], a discoverability engine for The Public Internet of Things, you can see where things are, who owns them, and how and why they are used. The increasing availability of connected devices and apps is already changing the way people and things cooperate [2].

In DQuid, we believe that the deployment of smart objects is only the initial step towards the Web of Things, that is a full network of connected devices capable of interacting with other devices and with humans, and possibly involving some intelligence hosted in the cloud. For such a vision to become practical, there is the need for a set of software solutions for building applications that exploits smart objects, enable the interaction between objects and engineer the generated data.

In this paper, after briefly summarizing the current scenario for the IoT, we identify some open challenges and detail the DQuid approach to tackle a specific challenge: the easy of application development for connected objects.

Scenario and Open Challenges

On the basis of the sketched vision for the Web of Things, in this section we present the current scenario of enabling technologies and the key challenges that has to faced to achieve such a vision.

As hardware evolves, a number of solutions to connect devices are already available on the market and almost on a stable state. In particular, we can count on:

1. A variety of plug & play modules to be connected to every kind of objects with low effort (e.g., boards such as Arduino[3], Raspberry PI[4], DQuid IO[5], Xee[6], etc.);
2. Wireless, low consumption and energy-harvesting technologies that enable a full exploitation of sensing and actuating capabilities (e.g., Enocean products[7]);
3. International standards (ISO/IEC 14543-3-10) that lay the pavement for interoperability between devices.

In such a scenario, one may think that realizing the Web of Things is simply a non issue. However, hardware technologies act only as enablers, leaving to software the complexities inherent in managing the generated data and dealing with diverse devices each with its own sensors and actuators. Accordingly, there should be software components capable of:

1. Enabling the implementation of smart applications that interact with objects. This challenge can be faced at different levels of abstraction, for instance Evothings [8] enables to quickly develop mobile app against real devices.
2. Identify relationships between objects (e.g., correlated properties, owners, shared locations, etc.) [9] and making the objects capable of interacting with nearby devices and users as an ecosystem, for instance as promoted by the AllJoin framework [10];
3. Engineer the overwhelming amount of sensed data, process and combine them in real time taking care of privacy issues ([11], [12]);

In the following section we present how DQuid tackles the first challenge. We propose an innovative approach that aims at supporting programmers hiding the complexities of dealing with hardware details and providing a simple yet effective programming API.

The DQuid Approach

DQuid aims at creating a global Web of Things ecosystem where OEMs, companies, developers, makers and users create and consume digital contents.
Such a vision is promoted by the means of a simple paradigm Plug, Tag & Share[13]:

- plug: connecting a DQuid board with a physical object (e.g., a fridge or a toaster) equipped with sensors or actuators, the objects can be connected to the Internet;
- tag: once the object is connected to a board, the user should give the object a digital entity uploading its configuration to the DQuid Server through a configuration app, that implies giving a name to the object and describing which sensors and actuators are connected to the board;

- share: DQuid provides a simple yet effective API to program apps for DQuid objects. Programmers can search for a specific object (e.g., a coffee machine) and download a lightweight library with classes that instantiates the DQuid representation of that specific physical object. Such a programming interface transparently refers to the object configuration, providing access to sensor and actuators available for the object and hiding the complexities of dealing with the physical board configuration.

Conclusions

To achieve the Web of Things vision, it is necessary to provide a full framework of software services that enable to (i) manage data coming from devices, (ii) interact with objects using apps, and (iii) allow objects to interact with each others.
We believe that easy of development of apps for smart objects is a first step towards the realization of such a vision. Accordingly, the DQuid approach is based on the simple Plug, Tag & Share paradigm. At the moment, the DQuid IO hardware enables to connect objects to the Internet, tag them providing a digital identity and a configuration, share it online.

References

[1] Thingful, http://thingful.net/
[2] #IoTH: The Internet of Things and Humans, http://radar.oreilly.com/2014/04/ioth-the-internet-of-things-and-humans.html
[3] Arduino, http://www.arduino.cc/
[4] Raspberry PI, http://www.raspberrypi.org/
[5] DQuid IO, http://www.dquid.com/landing/dquid-io
[6] Xee, http://www.xee.com/
[7] Enocean, http://www.enocean.com/
[8] Evothings, http://evothings.com/
[9] Graphs & The Internet of (Connected) Things, http://neo4j.com/blog/graph-of-things/?utm_source=feedburner&utm_medium=feed&utm_campaign=Feed%3A+Neo4jBlog+(Neo4j+Blog)
[10] AllJoyn, https://www.alljoyn.org/
[11] Data Engineering Is The Bottleneck For The Internet Of Things, http://www.forbes.com/sites/danwoods/2014/04/30/data-engineering-is-the-bottleneck-for-the-internet-of-things/
[12] Will the IoT Crush IT?, http://theinstitute.ieee.org/opinions/question/will-the-internet-of-things-crush-it
[13] Patent n.