Making Internet-Connected Objects readily useful*

Manfred Hauswirth
DERI Galway
Ireland

Dennis Pfisterer
University of Lübeck
Germany

Stefan Decker
DERI Galway
Ireland

Abstract

To turn smart objects into truly Internet-connected Objects, IP-based access to them is required as a unifying network layer. As the smart objects landscape is very heterogeneous in terms of hardware capabilities/constraints and network protocols the necessary efforts to achieve this are significant and come from a variety of areas, e.g., CORE, 6LoWPAN, ROLL, to name a few. In this position paper we try to make the case that for making objects readily useful for application development, we should not stop at the network layer but also address some of the integration issues at the higher layers. Specifically, we envision encapsulating device and service semantics in a special *semantic layer* that spans and integrates the Internet and the embedded world.

1 A complete stack?

For a moment, let us assume that we have succeeded in making smart objects accessible via IP on a large scale. Would this be sufficient? On the one hand, we could now build applications which could communicate using a unified network stack and without having to worry about getting data from/to Internet-Connected Objects (ICOs). On the other hand, we would still have to know a lot about the underlying, heterogeneous network as the uniform network interface does not make resource constraints and hardware limitations go away. To that end we see the need that ICOs need to be equipped with **light-weight means to communicate their constraints and operating conditions**, maybe through a new form of ICMP specifically for ICOs.

Now, if we assume this problem to be solved, the next question which comes to mind is the one of **discovery**. Discovery is a quite overused notion and can mean many things: from mapping names to routable addresses to finding services (either general model or only enabling to model specific services, either based on simple fields/keywords or semantic descriptions, etc.), up to finding entities based on their meaning, e.g., for being able to send a message to a "room" or a "house". Specifically in the world of ICOs the requirements for discovery change significantly. For example, it is more important to be able to discover objects based on their (possibly changing) geo-spatial location or their temporal/geo-spatial "location". I.e., one is only interested in objects which are in a place of interest in a certain time interval. Also the redundancy introduced by multiple objects at a location of interest opens up new questions for discovery: return all found objects, only the "relevant" ones (however relevancy may be defined), etc.? A follow-up question is to which objects to talk to. For example, if we can trust the temperature sensors in a room, then we only need one (or k) to be available for a correct reading. This *semantic representation* of the real-world may influence discovery and routing approaches. At the moment it is unclear which functionalities should be part of the Internet infrastructure and which should be implemented by the applications.

Again, assuming all these questions have been addressed successfully, the next question is how to enable meaningful **access to the ICOs, i.e., services**. Here, initiatives like the IETF's CORE working group are already on the way and provide important work to guide the development. CoAP, a draft by the CoRE working group, deals with Constrained Restful Environments and provides exactly the subset

*The work presented in this paper was supported (in part) by the Líon project supported by Science Foundation Ireland under Grant No. SFI/02/CE1/I131 and the European Union under contract number ICT-2009-258885 (SPITFIRE).

of HTTP methods (GET, PUT, POST, and DELETE) that is required to offer RESTful web services on ICOs. If we manage to address all the issues we have already pretty sophisticated access mechanisms at our hands. We essentially can plug-in ICOs into service-oriented architectures (SOAs), i.e., make them available for business processes. But is this sufficient? So far, we have essentially only talked about the interface side of things to a large extent. However, it is equally important **to understand the data** which comes out and goes into these interfaces. As the number of ICOs will by far exceed the number of Internet hosts, we will have to include mechanisms to (1) exchange data in a uniform format, which is as flexible as IP (at the networking level) to serve a wide range of requirements and applications and to (2) provide means for automatic interpretation and understanding of the transported data (semantics).

Such a **data exchange layer** will be crucial to enable scalability from an application's point of view as nobody will be able to deal with the number of ICOs – and the even larger amount of data they produce – efficiently and scalable without such a layer. The necessary technologies are already being developed and deployed: Linked Data and the Resource Description Format (RDF) are accepted standards in the Web and provide a general model. It is flexible enough to express whatever is required by applications through simple subject-predicate-object terms. These triples are the basic units of information, which can be connected arbitrarily and enable an Internet-wide database model of information. However, while this is the right way to proceed as can be seen by the tremendous uptake on the Web, these technologies need to be condensed into lighter forms, very much in the same spirit as CoAP was done for the service side. Currently, CoAP provides service descriptions that describe the interface (using WADL) but not its semantics. With light-weight semantics, ICOs will be first-class citizens of an Internet-wide semantic database that can easily be indexed, searched, and used using standard Web technologies such as SPARQL.

At this point it seems appropriate to add some brief remarks on the old services vs. data discussion: In our opinion, the relation between services and the data they transport can be seen a bit like the view of electrons as particles or waves – there is no definitive answer, but a spectrum of answers and one should be able to use the conceptualization which solves a problem best. Essentially, the question boils down to where to put the expressivity – into the interface or into the data. In the first case, it may be hard to decide what exact functionalities should be externalized in the interface. Too much, then the interface becomes too heavy; too little, then the interface needs to be changed rgularly. On the other hand, an interface hardwires some of the semantics which makes the execution more efficient. If a system requires higher flexibility and needs to support unforeseen use, then the interface should be kept minimal and the necessary flexibility is supported through the data model. If the data model in itself should be flexible, then semantics must be introduced. The system will be more flexible but this requires "runtime interpretation" at certain costs. The point we are trying to make here is, that the decision where to position a future ICO infrastructure is to be determined. Our feeling is, that a larger amount of flexibility will be required than currently being offered.

2 Conclusions

In this brief position paper we tried to highlight some of the relevant issues to make ICOs readily usable for applications. In our opinion, this means also to provide simple, yet powerful and generally usable abstractions for the architectural levels on top of the core networking layers. What and excatly were needs to be determined. But we belief that a similar approach to the one which guided the design of the networking layer itself should be taken to consolidate the humongous number of choices into a generally usable substrate.