Data Models for the Internet of Things

Michael J Koster
August 15, 2012

The title of this is "Data Models for the Internet of Things". Perhaps it should be "Data Models for a Semantic Web of Things" because that describes what I believe is going to be the next need for a broad interoperability standard. First I need to set the background and state some assumptions.

What is the Internet of Things?

Here is a broad definition: *People connecting to things, things connecting to things, and people connecting to people at a granular scale.*

A thing can be a sensor that encodes observations, or some other source of information about the physical world around us, or a display to inform us, or it may be an actuator to control some feature of our environment. For example, a thing might be a measurement from a temperature sensor, or a weather forecast for tomorrow, or the current location of a friend.

Connecting at a granular scale means that the information can be used at either large scale or small scale, or anywhere in between. It's also known as scale-less. It means, for example, that single sensor data can be used as a single observation, or data from multiple sensors can be aggregated over time for a larger view of the phenomenon. Mash-ups are examples of data interaction on a granular scale.

Many of these things are currently observable on the internet, but they don't easily *connect* in meaningful ways to people or to each other. Nor do they always function well on a granular scale.

The Internet of Things is, more specifically, a set of tools and technologies which enable the internet to be used to connect people to things, things to things, and people to people on a granular scale.

The World Wide Web

This isn't meant to be a history lesson, but more of an exposition of a process.

In 1990, a number of tools and techniques were integrated to create a World Wide Web of documents on top of the existing internet. The first prototype consisted of a document server (http) transmitting HTML documents to a simple client on a NeXT workstation which was able to display the HTML in outline format with images and other hyperlinked resources rendered in separate windows.

The prototype code was openly published for people to play with and, over time, the system of rough consensus and running code was adopted for creating and extending internet and particularly world wide web standards.

The platform that is "the web" is today far richer than the HTML + http + URL hyperlinks of the first prototype; there are multiple revisions of HTML, dozens of other markup standards, WS, REST, LAMP, AJAX, Ruby On Rails..., all of which use http and the underlying network stack to basically accomplish the same thing: wiring user controls and displays in the web browser to application logic on the server or more recently "in the cloud". Now we have smartphones and tablets with skinny GUI tools as a very nice alternative to bloated browserware, but in the end it's still controls wired to server or cloud applications as the main user interaction paradigm, with back end communication between servers (Machine to Machine, or M2M) doing most of the work of integrating and filtering of data streams.

Web pages for light switches. Yay.

I believe that we are at a point in the development of the Internet of Things which is analogous to where the Internet, specifically the World Wide Web, was in 1990. At that time, the Internet existed and there were a number of applications hosted including email (POP, IMAP), relay chat (IRC), and document file transmission (UUCP).

The Internet of Things today looks a lot like the good old World Wide Web. Analogous to using *uucp* for sharing documents on the internet of 1989 (it was better than quarter inch tape...) we now have systems where networks of sensors e.g. smarthomes, are interacted with on web pages and smartphone apps hosted by cloud servers. Sure it's fun and useful, but the connectivity and ultimately the ability of the system to evolve is limited by use of the legacy interaction mechanism.

The sensor and low level network technology is also available today. It's in the early adopter phase waiting for the number of connections to increase to the point where Metcalfe's law becomes relevant. Low power radio, 6LoWPAN, CoRE, CoAP, and related efforts are creating a broad base for a low level infrastructure of sensing and control. Personal sensors like Nike Fuel are on the market. Kickstarter is flush with gadgets that connect to the internet, gadgets that sense and also gadgets that passively inform through soft lights or lava-lamp style readouts.

The problem is that each new gadget comes with it's own back end cloud hosted service that only connects to a select set of other things on the same service (and of course twitter). There is no easy way for a user of these gadgets to mash up their own set of interactions or even have their displays and controls all in one place. We might say that the interfaces are in *silos*, stuck in vertical enclosures that hide the contents from other silos.

We could think about some intermediary aggregators on the web to connect data sources from multiple cloud hosted services, but each service has it's own coding and semantics, quickly becoming a big data flavor of problem if we want to solve it in a general way.

Toward A Semantic Web of Things

The technology to accomplish the higher level goals, connecting people to things and things to things at a granular scale, exists today. It exists in a component form and in the form of hundreds of separately designed proprietary back end systems that all use common tools and design patterns but lack the common data models needed to successfully interoperate.

What's missing is a set of tools and techniques, analogous to the early web tools, for creating a system that can connect anything to anything else that makes sense. Note that this is not about connecting everything to everything else. Though we want to allow large scale deployments, this is more about connecting things in a **very selective** manner.