Information Model for an Interoperable Web of Things

Michael Koster*
*ARM Limited, Email: Michael.Koster@arm.com

I. Abstract

A Web Of Things will enable application level interoperability. Application software will make use of sensors, actuators, and other data sources, analogous to the way humans make use of information on the World Wide Web. To make this a reality, hypermedia driven APIs can be annotated with machine understandable hyperlinks. Internet standards exist for web object encapsulation of sensor data with appropriate semantic linking. There are protocols for resource discovery. We are seeing the emergence of architectures for the deployment of IoT services. What is still needed for application interoperability is a common set of high level patterns for the creation, manipulation, and use of descriptive, machine understandable hyperlinks, built on underlying web standard mechanisms and protocols. Information models[1] can be built on a base of reusable concepts, relations, and attributes. These models can be used to create and compose common, reusable resources and application interfaces, promoting broad web scale interoperability.

The W3C could become a venue for standardizing information models that accommodates existing information and data models, enhance the functionality of existing models, as well as standardize new information models.

II. A Web Of Things

The World Wide Web enables anyone to access to any information on any server using any web browser, independent of operating systems, enabled by a core set of protocols consisting of the IP, TCP, HTTP and HTML.

The Internet Of Things results from being able to connect everyday objects to software using networks and internet protocols. This provides an infrastructure for scalable interconnection using the existing internet and new sensor networks.

The next evolution will be to enable application software to understand data from sensors, actuators, and other data sources in the way that people using web browsers understand information on the World Wide Web. The coming Web Of Things will be enabled by application-level interoperability, that is interoperability between application software and network connected things. As with the WWW of information, the Web Of Things will connect any application to any connected thing using a choice of wire protocols.

III. The Machine Interoperability Problem

Humans are able to use web browsers to access information using the relatively simple mechanisms of hyperlinks associated with pictures, text, and other visual cues embedded in web pages and infographics. This is enabled by our ability to rely on visual metaphor, text association, spatial cues, and other high level human cognitive function.

Software needs to be able to access information from sensors using similar simple mechanisms, but software cannot rely on cognitive ability and visual metaphor to select the proper links. What is needed is a higher level of information which includes descriptive attributes sufficient to enable software to automatically discover and link to the appropriate resources.

The existing mime-types are sufficient for web scale interaction, but another layer of metadata is needed to create a machine understandable resource discovery and linking language.

IV. WWW Architecture for Things and Services

The World Wide Web has evolved from its origins in simple HTML, to a rich set of protocols and abstractions that are responsible for the operation of the web at scale. A more recent development in web architecture is the adoption of the REST architecture style. There is a huge advantage provided by REST in the externalization of application state. This enables stateless applications and is key to achieving web scale.

The disadvantage of REST in dynamic systems is the lack of an architecturally well defined asynchronous, or ”push” notification and update mechanism. In practice this is not a problem as there are many mechanisms available, including multi-part GET responses [2], using PUT or POST as commonly done in REST APIs to update endpoints, binding to a message protocol like MQTT [3], and use of the bidirectional socket connection [4].

REST endpoints are referenced by URLs, enabling hyperlinks to be used by application software to discover and interact with resources. Such hypermedia driven APIs enable the decoupling of connected things from dedicated services and applications, enabling broad interoperability.

Position paper for the ’W3C Workshop on the Web of Things - Enablers and services for an open Web of Devices’, 25 - 26 June 2014, Berlin, Germany.

[1] There is a lot of terminology confusion between information models and data models. For this purpose RFC 3444 [1] has been published to provide guidance. As a summary, the main difference between the two terms can be described as follows. The main purpose of an Information Model is to model managed objects at a conceptual level, independent of any specific implementations or protocols used to transport the data. Data models are defined at a lower level of abstraction and include many details intended for implementors and include protocol-specific constructs.

V. Information Models

Application software can automatically discover and link to resources using link metadata, that is metadata embedded within hyperlinks. These hyperlinks have the form of a subject URL pointing to a resource and a collection of relation/attribute pairs. Software examines the relations and attributes in order to select links to suitable resources. Examples of metadata links are found in RFC 6690 [5] and Hypercat [6].

Also known as semantic hyperlinks, there can be many links and relations for a given subject resource. This allows rich descriptions to be created, with layers of context and resource bindings added as needed. The resulting scheme only requires a subset of general web linking, and a subset of the Semantic Web. The focus is on abstraction and virtualization of physical things for the purpose of monitoring and control.

Simple data models and information models can be built from collections of semantic hyperlinks. These data models describe the information at the linked resource sufficiently well to enable application software to correctly select the appropriate resources.

VI. Web Objects, Resource Directories, and Catalogs

Web objects encapsulate semantic links and other metadata along with observable data from sensors at a web endpoint pointed to by a URI. A web object may map to a physical object, or a web object may be composed of resources from other objects.

A web object may encapsulate a number of resources and their associated metadata, where each resource is a distinct URL from the base URI. There may be a common structure to the data and metadata endpoints, for example the /well-known/core resource in CoAP objects that contains semantic links.

To make the objects discoverable by applications, the object's semantic links need to be accessible by the application. One way is to allow the application to read embedded metadata directly from the object. More likely, there will be a middleware platform that the object's semantic links can be uploaded to, which is also reachable by the application software. An example is the CoRE Resource Directory [7], which provides capabilities for registering device metadata to make it available to applications for discovery by relation and attribute. Resource catalogs are another name for similar functionality, see [6].

An example of an IPSO [8] application object, an information model for a temperature sensor, is shown in Table I and in Table II. IPSO objects use the Open Mobile Alliance (OMA) Lightweight Machine-to-Machine (LightweightM2M) template [9].

TABLE I
Example: IPSO Object (Temperature)

Object	Object ID	Object URN
IPSO Temperature	3303	urn:oma:lwm2m:ext:3303

TABLE II
Example: IPSO Resource Info (Temperature)

Name	Resource ID	Access Type	Data Type	Units
Sensor Value	5700	read	Float	Celsius
Min Measured Value	5601	read	Float	Celsius
Max Measured Value	5602	read	Float	Celsius
Min Range Value	5603	read	Float	Celsius
Max Range Value	5604	read	Float	Celsius

VII. Hypermedia driven API composition

The information model for an object can contain information about the objects structure and resource types. This can enable APIs to automatically be composed by middleware and automatically consumed by application software.

Additional metadata can indicate context, such as geographical location, and bindings, such as message protocols and event handlers, as well as access control information.

Applications can use templates based on models to facilitate discovery and linking to resources by attribute and relation, including context and bindings. Consider the following examples:

1) Calculate average temperature within a region by linking to all of the resources that expose temperature measurements and are within a geographically bounded region.
2) Turn on a light in the room I am presently occupying. Applying a filter based on a template would return objects and resources that satisfy the relations, as well as additional metadata that specify properties, for example engineering units to enable auto-conversion.

VIII. The Role of the W3C

Existing standards exist that offer the infrastructure for connecting Things to the Internet. What is needed are information models for diverse application domains. Although standardization efforts on defining information and data models have been started in other fora already the W3C might offer a unique venue for defining additional information models by involving a wider community. The author hopes that the workshop will offer a possibility to discuss how much interoperability of information models will be desired and how to avoid overlap with ongoing efforts.

References

[1] A. Pras and J. Schoenwaelder, "On the Difference between Information Models and Data Models," Internet Engineering Task Force, RFC 3444, Jan. 2003. [Online]. Available: http://www.rfc-editor.org/rfc/rfc3444.txt

[2] Z. Shelby, K. Hartke, and C. Bormann, "Constrained Application Protocol (CoAP)," Internet Engineering Task Force, Internet-Draft draft-ietf-core-coap-18, 06 2028, work in progress. [Online]. Available: http://www.ietf.org/internet-drafts/draft-ietf-core-coap-18.txt

[3] IBM, "MQ Telemetry Transport (MQTT) V3.1 Protocol Specification," Aug. 2010. [Online]. Available: https://www.ibm.com/developerworks/library/ws-mqtt/

[4] I. Fette and A. Melnikov, "The WebSocket Protocol," Internet Engineering Task Force, RFC 6455, Dec. 2011. [Online]. Available: http://www.rfc-editor.org/rfc/rfc6455.txt

[5] Z. Shelby, "Constrained RESTful Environments (CoRE) Link Format," Internet Engineering Task Force, RFC 6690, Aug. 2012. [Online]. Available: http://www.rfc-editor.org/rfc/rfc6690.txt

[6] "HyperCat," Sep. 2013. [Online]. Available: http://wiki.1248.io/doku.php?id=hypercat

[7] Z. Shelby, C. Bormann, and S. Krco, "CoRE Resource Directory," Internet Engineering Task Force, Internet-Draft draft-ietf-core-resource-directory-01, 12 2011, work in progress. [Online]. Available: http://www.ietf.org/internet-drafts/draft-ietf-core-resource-directory-01.txt

[8] "IP for Smart Objects (IPSO) Alliance," May 2014. [Online]. Available: http://www.ipso-alliance.org/

[9] OMA, "OMA LightweightM2M v1.0," Dec. 2013. [Online]. Available: http://technical.openmobilealliance.org/Technical/release_program/lightweightM2M_v1_0.aspx