

Thing-to-Thing Research Group
Internet-Draft
Intended Status: Informational
Expires: September 12, 2016

J. Strassner
Huawei Technologies
J. Halpern
Ericsson
Q. Wu
Huawei Technologies
March 8, 2016

Semantics and the Internet of Things
draft-strassner-t2trg-semantics-and-iot-00

Abstract

This document examines how semantics help different deployments in an Internet of Things (IoT) environment interoperate. IoT data, device, and system interoperability requires semantics to ensure that the meaning of terms and objects in one device or system are not lost or altered when they are exchanged and used by other devices or systems.

Status of This Memo

This Internet-Draft is submitted in full conformance with the provisions of [BCP 78](#) and [BCP 79](#).

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at <http://datatracker.ietf.org/drafts/current/>.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress".

This Internet-Draft will expire on September 12, 2016.

Copyright Notice

Copyright (c) 2016 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to [BCP 78](#) and the IETF Trust's Legal Provisions Relating to IETF Documents (<http://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the [Trust Legal Provisions](#) and are provided without

warranty as described in the Simplified BSD License.

Table of Contents

1.	Introduction and Motivation	2
2.	Why Information and Data Models Alone Are Not Enough	2
3.	Why Ontologies Alone Are Not Enough	3
4.	A Proposed Architectural Solution	4
5.	Future Issues	6
6.	Security Considerations	6
7.	IANA Considerations	6
8.	References	6
	Authors' Addresses	7

[1.](#) Introduction and Motivation

The Internet of Things (IoT) includes a wide range of devices with a diverse set of functionality. This ranges from "smart" devices [[1](#)] [[2](#)] to simple sensors and actuators that lack any ability to deviate from their pre-programmed functions.

The IoT is currently being populated by diverse devices and systems that are operating as silos. A unified, open system that can support multiple applications cannot be built in a "bottom-up" fashion; this simply encourages more silos. Since different devices will have very different capabilities, standard networking assumptions (e.g., the ability to connect to the Internet anytime) are not applicable. The IoT operates more as a collection of consumers and providers of data than as a typical point-to-point communication system.

[2.](#) Why Information and Data Models Alone Are Not Enough

An information model (IM) [[3](#)] defines a common set of terminology and manageable objects. IMs can be used to define disparate data models (DMs), and maintain coherence between each realization of each common term in each DM. However, IMs and DMs by themselves CANNOT guarantee semantic interoperability.

Semantics in IMs are largely defined by the **descriptive** text associated with the model, rather than by any **formal elements**

contained in the model. This leads to several limitations. First and foremost, it is hard, even for human beings, to determine when two IMs are describing the same concept. This is due to several reasons, including expressing one concept as a set of model elements, having a large number of model elements to understand, and because of the inherent ambiguity of using descriptive, rather than formal, definitions of the model elements. Doing so in an automatic and scalable fashion to support dynamic creation of devices is **much** harder. Models define objects, facts, and values (i.e., data with no persistence) in an extensible manner. However, without formal semantics, reasoning (e.g., subsumption, as used in OWL [16]) is precluded.

To complete the picture, it should be noted that DMs without IMs are even more limiting. Trying for conceptual alignment and reasoning across different data modeling techniques is almost impossible, particularly due to the inability of many DMs to express the rich semantics required by IoT interoperability.

[3.](#) Why Ontologies Alone Are Not Enough

An ontology [4] [5] for a given domain defines a set of formulae that represent the characteristics and behavior of entities in that domain. Ontologies, unlike IMs, use **formal** languages (typically, either first-order logic or some type of description logic); this helps remove ambiguities that can creep into an IM or a DM. However, using ontologies as the sole source of information also has a number of problems.

First, ontologies use an Open World Assumption (OWA), while IMs and DMs use a Closed World Assumption (CWA). An OWA means that the truth of a statement is independent of whether or not it is known to be true. In contrast, a CWA means that if a statement is true, then it is also known to be true. Significantly, this means that if something is not known to be true, then it is false. Put another way, OWA means that the lack of knowledge does NOT imply falsity. This is a key characteristic for enabling inferencing to create new knowledge from existing knowledge.

Second, ontologies focus on the runtime exploitation of knowledge [14], while IMs and DMs are used for realization of knowledge. IMs and DMs provide answers in the form of facts. This is often much simpler than applying logic to reason about whether a particular

attribute has a given value or not. Note that if models are augmented with metadata, then the metadata may be used to "wrap" objects together at runtime, making it possible to add behavior and/or attributes to models at runtime. See, for example, [section 5.7](#) of [15] for how this is used in an IETF IM.

Third, ontologies become less suitable compared to IMs and DMs when the concepts to be represented do not have precise definitions. While work on using different forms of multiple-valued logic (e.g., fuzzy logic, which is different than probability) have been applied to both IMs/DMs and ontologies, in all cases known to the authors, this requires manual coding of relationships and formulae, as well as manual definition of the type of logic used, and how it is extended to include fuzzy reasoning. Furthermore, no standard reasoners that allow the user to define their own type of fuzzy logic are available.

Fourth, networks are inherently collections of heterogeneous entities. Their context and topology change, which causes their behavior to change. This is very difficult to model using ontologies, and typically leads to unsolvable complexity problems.

Finally, the vast majority of equipment data are defined using DMs, which are difficult to translate into ontologies. First, notions of methods, which frequently appear on classes in models, do not have a direct counterpart in ontologies. Classes in models emphasize the operational properties of the object; classes in ontologies reflect the structural properties of the object. We need to incorporate the formal reasoning power of ontologies *without breaking our current uses of DMs*.

[4.](#) A Proposed Architectural Solution

The FOCAL architecture, first defined in [6], combined the use of models and ontologies to build a scalable and extensible knowledge base. Information and data models were used to represent facts, and ontologies were used to represent the semantics required to reason about those facts. The combination of models and ontologies served to deal with the inherent cognitive dissonance that arises from heterogeneous data generated by different platforms, languages, and protocols. For example, models can be used to represent different telemetry data generated by IoT devices, and ontologies can be used to relate these data using one or more semantic relationships (e.g., is-similar-to, is-identical-to, as well as more traditional

linguistic relationships, such as synonyms, antonyms, and meronyms). The semantic processing engine can then use reasoning to determine and infer relationships between data, structures of data, and recognize new data (i.e., data that has not been defined).

In FOCAL, we used the standard Data-Information-Knowledge-Wisdom (DIKW) [7] approach to discover and annotate data. We consider DIKW a continuum; as more information is collected, DIKW elements do not "disappear", but instead are augmented/enhanced, and can move up the continuum. This enables a number of breakthroughs [8], including:

- o heterogeneous data can be recognized as being similar
- o different commands using different languages can be normalized
- o incorrect knowledge can be corrected, and new knowledge can be added to the knowledge base

FOCALE is a variant of Model-Driven Engineering (MDE) [9]. MDE uses an integrated view to more directly translate design intent into implementation through the use of software tooling. Key to such tooling are metadata, Domain-Specific Languages (DSLs), and MDE architectures.

Metadata has been defined as data about data. In MDE, metadata is descriptive and/or prescriptive information about concepts that are not limited to data, but can apply to any managed object. This has been used in industries (though not necessarily networking related) for a long time. For example, the Force.com platform is a metadata-driven development model [10].

Metadata enables all deployment and non-functional behavior for different applications to be derived from the same model and automatically generate configuration and monitoring information. See sections [5.16](#) - [5.20](#) of [15] for how this is used in an IETF IM.

APIs are a hot topic. However, an API is just a static definition, at a particular point in time, of some functionality. What happens if that functionality changes? The API breaks. What is the context in which the API is being used changes? The API may no longer be applicable. Relying on just APIs makes a software design fragile. Instead, if APIs and the data that they use are annotated with appropriate metadata, they can transform application-specific data into a common form used by other applications. Metadata, when used with models and ontologies, creates a self-describing framework that enables data to define contextually-aware application features and

functionality. Such an architecture is needed for IoT, because IoT data does not conform to one "universal schema". In addition, IoT data needs metadata augmentation as a first step to define and understand the hidden semantics that are associated with IoT data.

The underlying issues with IoT are not its volume and velocity, but rather how to extract **value** from collected and processed IoT data [17]. The solution proposed in [17] discovers, through machine-based reasoning, semantic information describing ingested data as well as the context in which the data is produced and received, and then annotates the data using a number of mechanisms, such as [18].

Domain Specific Languages (DSLs) provide a robust way to implement specific configuration and monitoring processes by being formed from models and ontologies that contain metadata [11]. Briefly, the combination of a model and an ontology can be used to represent the syntax and semantics of a grammar (for the DSL). This approach provides an extensible vocabulary, and associated lexicon, for the grammar. When used in an MDE environment, the grammar may be dynamically refined and adjusted to address the needs of the users of the DSL.

[11] uses this approach, along with the concept of the Policy Continuum [19], to define a set of DSLs for different constituencies of users. The Policy Continuum posits that different actors, ranging from business users and product managers, to application developers, to network, compute, and storage administrators, all use policy rules; however, the policy used by each of these actors is different in terms of structure, content, and representation. Writing multiple different languages, with the intent of having them interoperate, is very difficult. In contrast, this approach defines a single conceptual grammar, and then takes pieces of the grammar to build "mini-languages" (i.e., DSLs), each focused on the needs of a specific set of actors. Significantly, this means that changes in the data model can be transformed to a common representation in the information model, which can then be translated into changes in one or more DSLs. Optionally, APIs can also be defined.

[5.](#) Future Issues

There are a number of future issues that need to be worked. These include, but are not limited to:

- o how to harness the enormous repository of Linked Open Data [12] on the web, and to extract useful data for IoT applications, or

- whether Linked Open Data is "merely more data" [13]
- o how to standardize IM to DM mappings/transformations
- o how to standardize adding and managing metadata with IoT data
- o how to develop new architectural patterns for IoT data processing that lend themselves to Big Data and Analytics
- o how different policy paradigms (e.g., imperative and intent-based, or declarative) can be supported

6. Security Considerations

TBD

7. IANA Considerations

This document has no actions for IANA.

8. References

8.1. Informative References

- [1] G. Kortuem, F. Kawsar, V. Sundramoorthy, D. Fitton, "Smart objects as building blocks for the internet of things", IEEE Internet Comput. 14 (2010), pp 44-51
- [2] Horizon 2020 Work Programme 2014-2015, ICT 30 - 2015: "Internet of Things and Platforms for Connected Smart Objects" amended April 17, 2015
- [3] A. Pras, J. Schoenwaelder, "On the Difference between Information Models and Data Models", [RFC 3444](#), January 2003
- [4] N. F. Noy, D. L. McGuinness, "Ontology Development 101: A Guide to Creating Your First Ontology", Stanford Knowledge Systems Laboratory Technical Report KSL-01-05, March 2001.
- [5] S. Staab, R. Studer, "Handbook on Ontologies", Springer, 2nd edition, 2009, ISBN: 978-3540709992
- [6] J. Strassner, N. Agoulmine, E. Lehtihet, "FOCALE - A Novel Autonomic Networking Architecture", International Transactions on Systems, Science, and Applications Journal, vol 3, No 1, pp 64-79, 2007
- [7] J. Rowley, "The wisdom hierarchy: representations of the DIKW hierarchy", Journal of Information Science 33, pp 163-180, 2007

- [8] J. Strassner, S. van der Meer, D. O'Sullivan, S. Dobson, "The Use of Context-Aware Policies and Ontologies to Facilitate Business-Aware Network Management", Journal of Network and System Management 17, pp 255-284, 2009
- [9] D. C. Schmidt, "Model-Driven Engineering", IEEE Computer 2006
- [10] https://developer.salesforce.com/docs/atlas.en-us.fundamentals.meta/fundamentals/adg_intro_metadata.htm
- [11] J. Strassner, "Model-driven DSLs", work in progress
- [12] B. Haslhofer, "Linked Data Tutorial", 2009
<http://www.slideshare.net/bhaslhofer/linked-data-tutorial>
- [13] P. Jain, P. Hitler, P.Z. Yeh, K. Verma, A.P. Sheth, "Linked Data is Merely More Data", Proc. WWW 2009 Workshop on Linked Data on the Web, 2009
- [14] W3C, "Ontology Driven Architectures and Potential Uses of the Semantic Web in Systems and Software Engineering", 2006
- [15] J. Strassner, J. Halpern, J. Coleman, "Generic Policy Information Model for Simplified Use of Policy Abstractions (SUPA)", [draft-strassner-sup-generic-policy-info-model-04](#), Feb 12, 2016
- [16] W3C, "OWL 2 Web Ontology Language Document Overview (Second Edition)", W3C Recommendation, 11 Dec 2012
<https://www.w3.org/TR/owl2-overview/>
- [17] J. Strassner, "Engineering Value from Big Data", presentation at the First International Workshop for Big Data Standards, March 7, 2016
- [18] W3c, "RDFa 1.1 Primer - Third Edition", W3C Working Group Note, March 17, 2015, <https://www.w3.org/TR/xhtml-rdfa-primer/>
- [19] S. Davy, B. Jennings, J. Strassner, "The Policy Continuum - A Formal Model", Proc. of the 2nd Intl. IEEE Workshop on Modeling Autonomic Communication Environments (MACE), Multicon Lecture Notes, No. 6, Multicon, Berlin, 2007, pages 65-78

Authors' Addresses

John Strassner
Huawei Technologies
2230 Central Expressway
San Jose, CA USA
Email: john.sc.strassner@huawei.com

Joel Halpern
Ericsson
P. O. Box 6049
Leesburg, VA 20178
Email: joel.halpern@ericsson.com

Qin Wu
Huawei Technologies
101 Software Avenue, Yuhua District

Nanjing, Jiangsu 210012 China
Email: bill.wu@huawei.com

Strassner et al. Expires September 12, 2016

[Page 7]