Building a Semantic Ontology for Internet of Things (IoT) Systems

Ronald DeSerranno, Matthew Mullarkey, and Alan Hevner
Muma College of Business - University of South Florida
4202 E Fowler Ave, Tampa, FL 33620, USA
{ron.deserranno@gmail.com; mmullarkey@usf.edu; ahevner@usf.edu}

Abstract. *The complexity of Internet of Things (IoT) systems requires designers, operators, and users to understand both the structure and semantics embedded in the IoT architecture. A semantic IoT ontology must support a full understanding of system flows, services, and qualities (FSQ). In our research, we will discuss how sematic ontologies are currently utilized in IoT systems and how they could be extended to include semantic definitions for critical flows, services, and qualities in IoT applications. We suggest that a focus on the semantics of the system and the entities in the system be enriched with a definition and understanding of FSQ semantics. Examples of FSQ semantics in an actual IoT environment are presented and discussed.*

1. Introduction

Flow-Service-Quality (FSQ) network engineering provides a framework upon which one can understand the operation of complex networks that operate in asynchronous environments where the system's capabilities, interactions and components are changing over time (Linger, Hevner, Walton, & Pleszkoch, 2004). A focus on FSQ will tend to improve the operation and capabilities of large scale networked systems by ensuring that the designers, operators, and users of the system focus on the flow (of information and control signals), the services provided by the entities, and the quality measures of the interactions within and between entities in the network. Entities in a network are a collection of asynchronously communicating components that use or provide some set of services to satisfy business requirements (Hevner, Linger, Pleszkoch, Prowell, & Walton, 2009). FSQ utilizes flow structures as a bridge between services and the requirements that are invoked on the system. These requirements may have different demands; they may require a certain level of performance, reliability, and security. As such, flow structures will use the quality of these indicators to make decisions about how the flow should execute (Linger et al., 2004). Services are acted on by flows to satisfy business requirements and may themselves execute flows in a recursive manner (Hevner, Linger, Sobel, & Walton, 2002).

Internet of Things (IoT) systems connect physical assets (entities) such as sensors, actuators, and computing devices over a network with a locus of control that monitors or activates the services of each entity as required (Bassi et al., 2013). Within the IoT system is an ontology. A virtual entity is a virtual representation of a physical asset that is present in the IoT model with its definition defined in the ontology. Necessarily, IoT systems must represent each physical entity and its properties with a virtual entity and must describe the nature of information flows between service entities. As the state of a device changes, the system must update the virtual entity and keep it synchronized with the actual asset. A state change can be mono-directional if the asset simply notifies the IoT system to update its virtual state. Alternatively, in complex IoT systems, a state change can be bi-directional whereby a change in the Virtual Entity, an actuator for example, will result in a change in the Physical Asset itself and a confirmation from the asset that the state change has occurred. In this way, the virtual entity initiates an information flow that informs the physical entity to provide a service (e.g. turn on/off, open/close, modify temperature, etc.) and the physical entity, in turn, informs the virtual entity of its change in state. Quality is assessed by comparing the performance of the service to the desired service level requirements and by confirming the accuracy of the flows in both directions.

In engineering complex IoT systems, the individual properties of the entities and the interconnected relationships and flows among entities become increasingly complex to model through virtual schemas that accurately represent what is happening in the system. As we review the nascent level of development of IoT architectural reference models (e.g. ISO, SAE, ARM) that attempt to provide generalizable frameworks of these IoT network systems, we find that the IoT ontologies and the reference models tend to focus on the definition of the entities, their properties and the controls involved in the IoT network (e.g. system structures). Node-based technologies like OPC UA (Open Process Connectivity Unified Architecture) (OPC Foundation, 2016) which were initially developed for industrial automation, are starting to find widespread adoption in other fields including IoT Systems (Schleipen, Sauer, & Wang, 2010). In such a system, the structure of the database is often saved in a triple store format leveraging such technologies as Resource Description Format (RDF) and Web Ontology Language (OWL) (Pessemier, Raskin, Van Winckel, Deconinck, & Saey, 2013).

When we look at complex networks and the performance of IoT systems, we conclude that a fuller understanding of the IoT system – its behavior and performance – can only come from a broader IoT architectural reference ontology that includes not only structural definitions of devices, but semantic understandings of flows, services, and quality measures within and between entities in the system. The rest of this article proceeds with a discussion in Section 2 of IoT ontologies within an existing IoT reference framework, a discussion in Section 3 of flow, service, and quality semantics as they exist within an IoT example, a recommendation in Section 4 of a more complete IoT ontology that more fully describes these complex systems, and, conclusions and future research direction in Section 5.

2. IoT Ontology and the IoT Architectural Reference Model (IoT-A)

The *object-subject-predicate* format used for defining physical assets in an IoT ontology is very useful. We argue that, as the complexity of the IoT systems increases, the capabilities of the ontology are sometimes forgotten and are not fully leveraged. An example of this is with the Internet of Things Architectural Reference Model (IoT-A) developed by the IoT-A (Bassi et al., 2013). The IoT-A is a group of European partners including Siemens, IBM, Alcatel, Hitachi, and several academic institutions. From September 2010 to November 2013 the group produced one of the most comprehensive nascent Reference Architectures developed for IoT platforms. The IoT-ARM is significant because it was developed through the study of existing IoT systems and with input from many leading industrial and academic professionals. IoT-ARM seeks to generalize from and replicate within the reference architecture the nature of working IoT systems.

The IoT-ARM describes physical assets (entities) as resources in the system that can be represented as virtual entities in the model. The IoT-ARM identifies relationships between virtual entities and their physical asset. In IoT-ARM it is explained that the 'associations' between the physical and the virtual entities are defined in the ontology (Bassi et al., 2013, page 144). The model also describes the properties that should be identified for the behavior of each physical entity and the communication of states between physical and virtual entities. What is less clear in the architecture model is a description of the services provided by any given entity, the nature of the information flows between entities, the "control" function of any given entity (and whether control is central or distributed), and the quality measures for the services and flows within the model. Thus, we contend that many of the semantic needs of IoT systems are not addressed in the current reference architectures.

Our observation is not unique to the IoT-ARM. In general, as we review existing IoT ontologies, we find that they are mostly limited to defining "devices, actuators, and equipment" as assets (Mineraud, Mazhelis, Su, & Tarkoma, 2016) as opposed to the flows between them. Within IoT systems, the ontology can and should represent more than just devices (Kumar, 2015). We propose that FSQ terms and definitions should exist in the IoT ontology. In the next section, we will demonstrate how they can be expressed, and how they could be utilized to provide a possible implementation of FSQ ideas in an IoT system architecture.

3. Defining Flow Service Quality Concepts in an IoT Ontology

Designers of IT systems use ontologies to name and define the types, properties, and relationships of entities in the systems. This is often done with modelling languages such as Resource Description Framework (RDF). Within IoT systems we have seen that they are used to define physical entities. The following image shows how a pump may be defined in an IoT ontology using the *object-subject-predicate* format. In this diagram the "type" designation is used to define the related virtual entity.

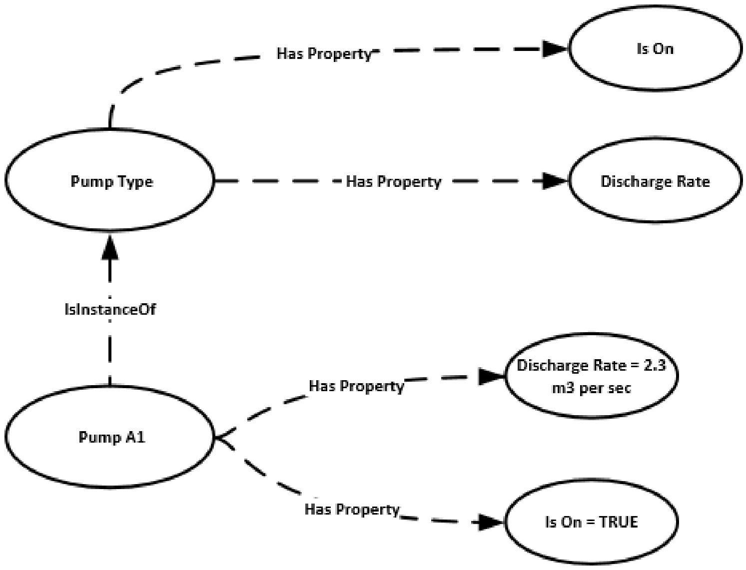

Figure 1. Example of a Virtual Entity defined in an Ontology for an IoT system.

In the IoT ontology, the type can be used to generalize from the specific as a category or each type can be associated with a distinct virtual-physical entity pairing.

In addition to defining assets, it is possible to define services of entities within the ontology. The simple example below defines a service, ‘Set Value’ provided by the entity that can be used on condition (true/false). In this example, a condition has been set up to monitor the fill level of a tank. If the fill level exceeds 95%, the condition will have its ‘Is Active’ state turn to true. The Set Value flow uses the ‘Is Active’ state of the condition to determine if the workflow should start. When it does start, the Set Value workflow changes its target property - the ‘Is On’ property for a pump - to TRUE allowing the fill level of the tank to come down while preventing overflow. In this way, the IoT ontology offers a definition of the service provided by the physical entity.

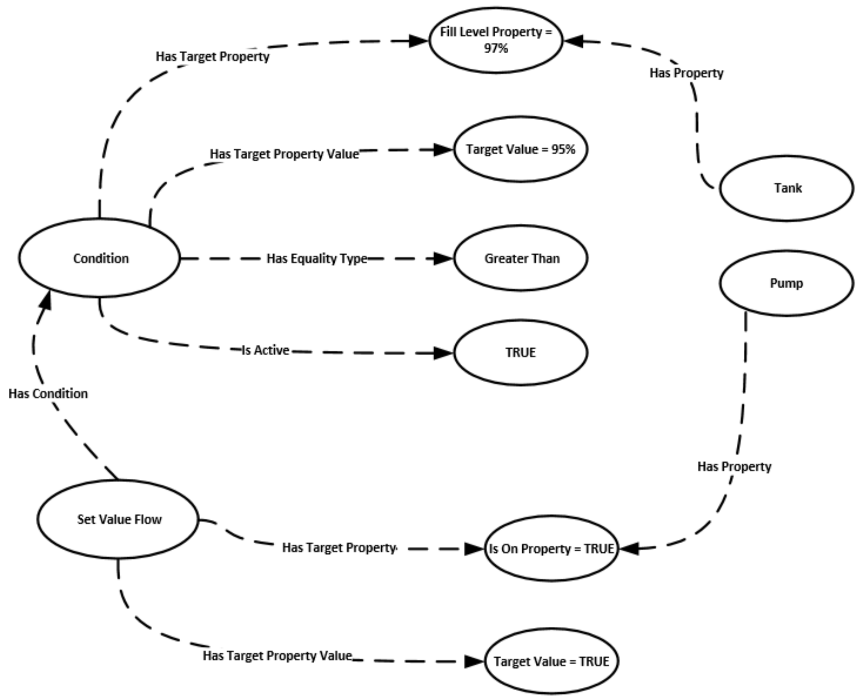

Figure 2. A Condition as a Trigger and a ‘Set Value’ Flow in the Ontology.

The signal that triggers the condition that enables and performs the service is one type of “flow” in the IoT ontology. Flow in this case is a “control” mechanism that changes an entity’s state. Flow can also be information provided by an entity to the IoT system about its state or its properties. In Figure 2, we see that the entity “reports” its “Has Target Property” to its virtual entity. In a centrally controlled IoT system, the virtual entity is then expected to share the new state with an IoT system manager. In a distributed IoT system, the flow could provide information in a peer-to-peer communication.

In this very simple example, the IoT ontology also describes the quality of the service provided by the entity and the quality of the information flow performed by the control function. If the entity is “informed” to change state, its response to that direction will indicate either the quality of the “command” or the quality of the execution of that command by the entity. Both can be compared to the properties that denote “good” quality outcomes in the IoT system. In the next section, we explore the concept of IoT “management” that connects the entity to its state by way of flows at an accepted understanding of quality.

4. Semantic Ontology and IoT System Management

We have established that flows can be defined in an ontology, but how do they execute? What services produce the actual work and when? In IoT-ARM a process exists for managing flows between entities and the desired functioning of the IoT system. In Figure 3, we use the IoT-ARM framework to identify the process management activities performed through process modeling and execution that connects the virtual model entity to the service provided by the physical asset.

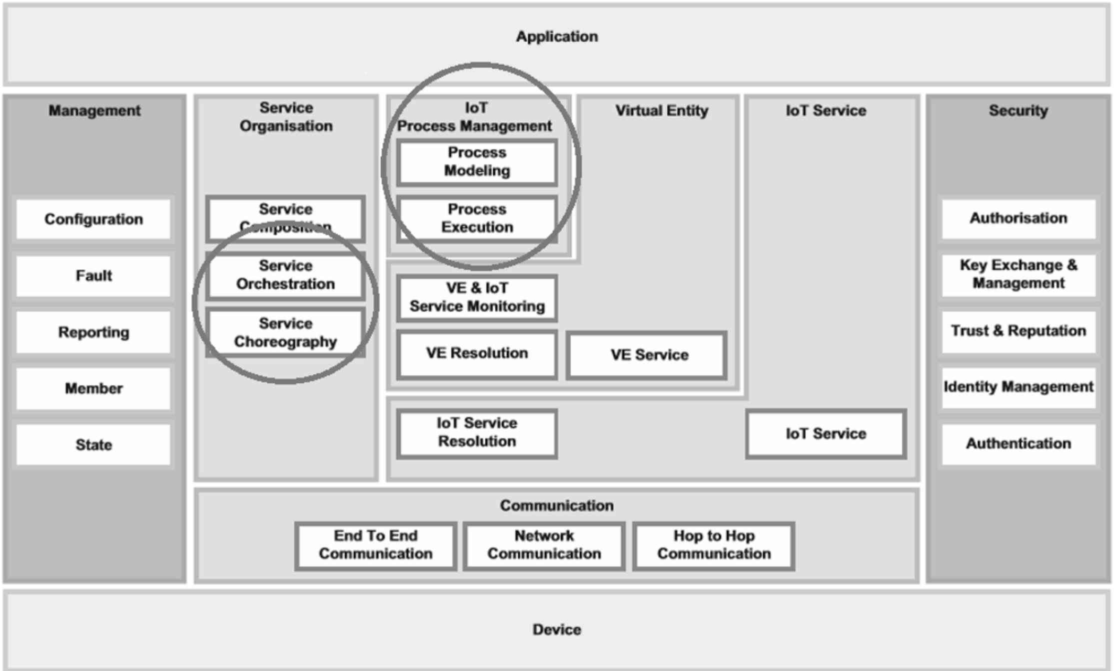

Figure 3. Functional Decomposition View of the IoT Reference Architecture (Bassi et al., 2013).

IoT Process Management is responsible for the execution of the flow models by recognizing the properties of individual entities, invoking services, and allocating resources within the IoT system to execute a given change in entity/system state. Previous research in Flow-Service-Quality (FSQ) (Hevner et al., 2009) suggests an FSQ manager that is responsible for managing the flows. The FSQ Manager initiates an 'instance' of a flow in the IoT system. Once initiated, the FSQ Manager monitors the flow execution, updates the current state of the system dynamically during execution, and determines a final system state upon completion of the flow. Quality requirements of the flow are continuously monitored and decisions to continue or terminate the flow are made based on whether the flow achieves its required levels of quality in the system.

A system that is consistent with the established values for its state, the communication of flows, and the provisioning of services can be considered a highly functional IoT. Inevitably, the quality of a system is a major concern for complex architectures (Cardoso, Sheth, Miller, Arnold, & Kochut, 2004). The FSQ Manager will function to direct the activities of the entities and manage flows in an IoT system.

As the FSQ manager executes instructions from the flow, it interacts with the Service Orchestration function and Service Choreography function. These components execute IoT services, coordinate tasks, and report back various quality indicators to the

Process Execution function (FSQ Manager) (Bassi et al., 2013). The FSQ Manager can take the quality and status information it receives, and update quality of service ('QoS') properties that exist in the current state of the system. The FSQ Manager can also change how the flow will progress, or terminate the flow based upon the specific IoT system's rules. The IoT ARM has a Virtual Entity Service as a functional component that allows clients access to the virtual entities in the ontology (Bassi et al., 2013). Clients interested in tracking the progression of the flow and its quality parameters can receive notifications such as when the flow completes from the Virtual Entity Service. This is a requirement defined in FSQ literature (Hevner et al., 2009) and demonstrated in Figure 4.

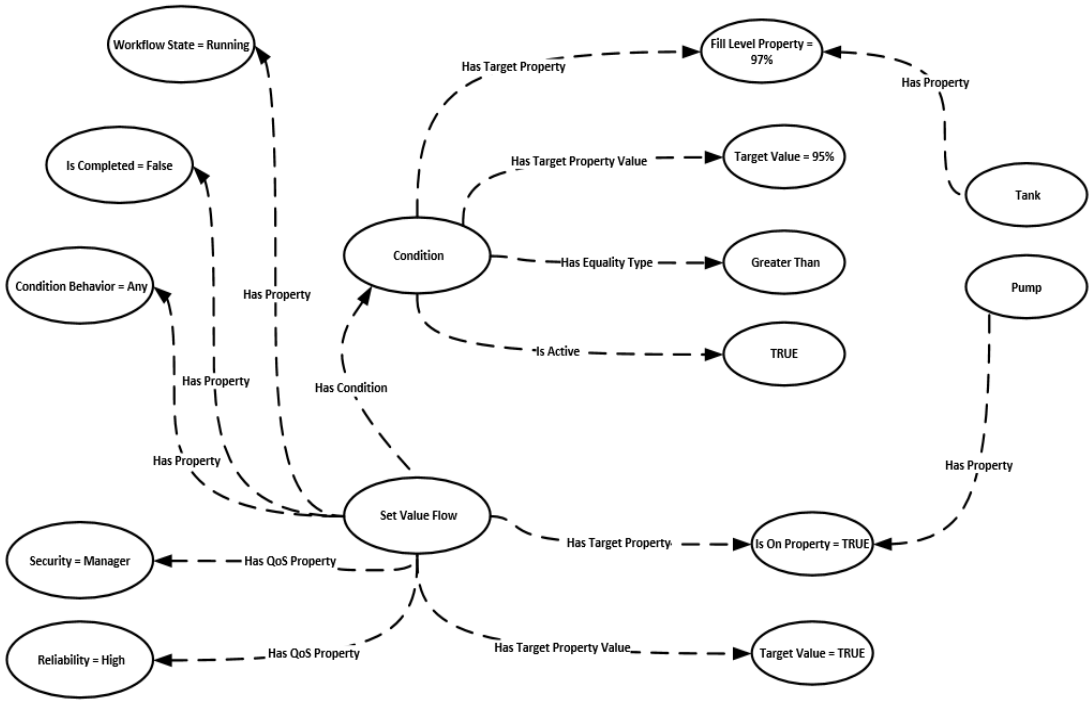

Figure 4. Diagram of a Flow, QoS Properties, the Condition to Trigger the Flow and the Virtual Asset that is Modified.

5. Conclusions and Future Research

Flow-Service-Quality (FSQ) semantics provide a useful extension of an IoT Architectural Reference Model (IoT-ARM) in ways that more fully develop our understanding of the IoT Ontology. Reference models are intended to evolve as new methods, architectures, and understanding come to light (Cloutier et al., 2010). In this paper, we propose that FSQ ideas fit into the IoT-ARM architecture and that ontologies can be used describing flow, services, and quality in IoT systems. We further suggest that the IoT ARM reference architecture could be evolved through the inclusion of FSQ concepts into the IoT Process Management.

As future research, the ideas reported in this paper will be used to improve the reliability, functionality, and adaptability of complex systems including IoT systems. This will have positive effects not only on IoT systems but other similar systems like those managing Smart Cities (Gaur, Scotney, Parr, & McClean, 2015). Our future plans include projects that will incorporate FSQ concepts in an extended semantic ontology for IoT reference architectures. Based on this innovative architecture model, we plan to implement a FSQ Manager in an active IoT application environment. We hope to measure improvements in our abilities to design, operate, and use the IoT system capabilities in real-time operations.

References

Bassi, A., Bauer, M., Fiedler, M., Kramp, T., Van Kranenburg, R., Lange, S., & Meissner, S. (2013). Enabling things to talk. *Designing IoT Solutions With the IoT Architectural Reference Model*.

Cardoso, J., Sheth, A., Miller, J., Arnold, J., & Kochut, K. (2004). Quality of service for workflows and web service processes. *Web Semantics: Science, Services and Agents on the World Wide Web, 1*(3), 281-308.

Cloutier, R., Muller, G., Verma, D., Nilchiani, R., Hole, E., & Bone, M. (2010). The concept of reference architectures. *Systems Engineering, 13*(1), 14-27.

Gaur, A., Scotney, B., Parr, G., & McClean, S. (2015). Smart City Architecture and its Applications Based on IoT. *Procedia Computer Science, 52*, 1089-1094.

Hevner, A., Linger, R., Pleszkoch, M., Prowell, S., & Walton, G. (2009). Flow-Service-Quality (FSQ) Systems Engineering. *Systems Analysis and Design: Techniques, Methodologies, Approaches, and Architectures, 15*, 11.

Hevner, A., Linger, R., Sobel, A., & Walton, G. (2002). The flow-service-quality framework: unified engineering for large-scale, adaptive systems (pp. 4006). Los Alamitos, CA, USA, USA: IEEE.

Kumar, V. (2015). Ontology Based Public Healthcare System in Internet of Things (IoT). *Procedia Computer Science, 50*, 99-102.

Linger, R. C., Hevner, A. R., Walton, G., & Pleszkoch, M. G. (2004). Flow-Service-Quality (FSQ) engineering: foundations for high-assurance network systems development (pp. 265). Los Alamitos, CA, USA, USA: IEEE.

Mineraud, J., Mazhelis, O., Su, X., & Tarkoma, S. (2016). A gap analysis of Internet-of-Things platforms. *Computer Communications*.

OPC Foundation. (2016). OPC Unified Architecture: Interoperability for Industrie 4.0 and the Internet of Things. Retrieved from https://opcfoundation.org/wp-content/uploads/2016/05/OPC-UA-Interoperability-For-Industrie4-and-IoT-EN-v5.pdf

Pessemier, W., Raskin, G., Van Winckel, H., Deconinck, G., & Saey, P. (2013). A practical approach to ontology-enabled control systems for astronomical instrumentation. *arXiv preprint arXiv:1310.5488*.

Schleipen, M., Sauer, O., & Wang, J. (2010). *Semantic integration by means of a graphical OPC Unified Architecture (OPC-UA) information model designer for Manufacturing Execution Systems*: na.