Summary of AllSeen Alliance Work Relevant to Semantic Interoperability

Summary written by Dave Thaler (all errors are mine :)

1 Introduction

The AllSeen Alliance uses an XML format (see https://wiki.allseenalliance.org/irb/xmlv2) for specifying data models, which is capable of modeling properties, methods, signals, each with complex types, metadata, human-readable descriptions, etc. There are already a number of in-market devices (https://allseenalliance.org/certification/certified-products-directory lists ones that have been certified) that support a number of AllSeen standard data models.

An Interface Review Board reviews data models before publication, and it has published guidelines for data models.

The AllSeen Alliance assumes that a translator (or "gateway", or "bridge") will be used to support various other protocols, since no single protocol will work for all devices today. The AllJoyn protocol is designed for use within a proximal network (mainly due to the discovery protocol being a local subnet protocol… the communication itself does not preclude remote), with a gateway being used to translate to cloud protocols or other proximal protocols such as non-IP based protocols. https://wiki.allseenalliance.org/gateway/gateway provides specs and information on AllSeen Alliance experience with providing gateways to various other protocols. Experience (and code) exists with gatewaying to Z-Wave, BACnet, XMPP, and other protocols.

2 Data Model Introspection

The AllSeen Alliance believes that not only are data models required, but data models should be capable of including human-readable descriptions of each resource and action, in multiple languages.

There are a number of scenarios that need the ability to get and use data model(s) including human-readable descriptions:

1. End users need descriptions of resources and actions in their language of choice, for creating If-This-Then-That (IFTTT) style rules.
2. Administrators need descriptions of interfaces and sometimes their members, for configuring security policies (e.g., ACLs). The advantage of being able to put descriptions in data models is that it avoids gratuitously introducing a separate format of metadata about the data model.
3. Data models need to be reviewed for consistency by the organization publishing the spec (in the AllSeen Alliance, this review is done by the Interface Review Board).
4. A developer wants to get the official data model (e.g., from the AllSeen Alliance website for an AllSeen standard data model, or from a vendor's website for a vendor-specific data model) for purposes such as algorithmic generation of code to talk to such a device.

5. A developer wants to get the actual data model(s), which may be standard or proprietary, from a device itself, for purposes such as code generation. One cannot guarantee they will be available anywhere else, especially for non-standard data models or extensions.

Some scenarios might entail multiple components. For example, scenario 1 requires getting descriptions in an end-user's language of choice, while scenario 6 requires getting a data model description from a device itself for code generation. This does not imply that the data model description from the device itself needs to include human-readable end-user descriptions in every possible relevant language. It merely means that whatever is obtained from the device must be sufficient to allow acquiring descriptions, potentially from elsewhere (this concept is called an Interface Description Server). Another example would be where the device's data model includes (say) English descriptions, and some online translation service is consulted to translate text into the requested language, which may be necessary for supporting a diverse customer base with many languages spoken.

3 Data Model Guidelines

The IRB guidelines for writing data models are at https://wiki.allseenalliance.org/irb/interface_design_guidelines_1.1. While some are specific to the AllJoyn XML syntax, a number of guidelines are more generic and would apply to data models in general, especially when a gateway to other protocols is employed.

For space reasons, I will cite just a few of many examples:

- "Do not reinvent the wheel": The guidance is specifically about semantic interoperability, stating, "Design to ease bridging to standard or widely deployed non-AllJoyn schemas if they exist (Bluetooth profile, ZigBee profile, IPSO, MIB, MOF, etc.). That is, where such schemas are known to exist, try to be as consistent as possible in order to aid in bridging, and reference relevant ones in your specification."
- "Avoid alternative marshaling schemes": This guideline, and the IRB guidance text associated with it, argues *against* using binary blobs, or blobs of data in a format (whether XML, JSON, CBOR, ASN.1, or whatever else) other the primary one for the data model. Doing so can make semantic interoperability (as well as the scenarios mentioned earlier) significantly more difficult due to the need for custom code.
- "An interface must encapsulate a single, coherent block of information and functionality.": This guideline, and the IRB guidance text associated with it, argues *against* having resources with a mix of optional properties. Instead, for maximum interoperability, it is important that a client be able to easily tell what set of properties/actions are supported by a device, and that grouping sets of associated properties is important to simplify applications and aid in reasoning about security and correctness. Thus, resources that have a bunch of individually optional parameters are problematic, whereas having a set of resources each with all properties being mandatory are better.