

# Position Paper: Long Life Web of Things with EXI

Yusuke DOI (TOSHIBA Corporation)

## 1 Introduction: Lifecycle Mismatch in Web of Things

Web of Things (WoT) is one of promising concepts to make our future more connected and convenient. For example, Tokyo urban train system is so famous for its high density and precision. However, once the train system is disturbed by some accident, people in Tokyo experience huge confusion. Intelligent human navigation systems will require real-time information of trains and other information from user's surrounding environment to cope with such exceptional situation, and this is one expected form of open WoT-based application.

WoT shall be used by everybody in daily goods and environment (things), and this makes major difference between 'Web' and 'Web of Things.' The things in general shall have longer lifetime than software, or geeky shiny digital goods. Shall people buy a new refrigerator or washing machine as often as they would buy a new smartphone? Shall your local governments invest in urban infrastructure to keep it up-to-date for the Web technology? In practice, WoT used by everybody shall have some component with longer life cycle with older specification.

In other words, practical WoT is inherently 'out-of-date' in comparison with the Web and rapidly evolving software frameworks on it. Hence, to realize WoT, we need to think how to connect the Things we will produce to always-changing future of the Web.

At some time, we need to control the evolution of the Things. A good thing of innovations is rapid evolution and it makes wide variety of technologies and products. However, the variety will make tons of 'junk Things' if we do not control the evolution because the Web evolve and the Things stay. Of course, strictly-controlled WoT is not what we want. We need to control the evolution with combination of Web-ish freedom and industrial persistence.

## 2 An Approach: Standard-based Industrial WoT for Infrastructures

To control the evolution, we usually make standardization. A standard of data format with a good room for extension is a best practice to control changes and diversion in many industries. Standardization organizations dedicated for application domains (e.g. IEC for electric devices, ASHRAE for large-scale air conditioning) provide good standards for each application domain. On the other

hand, I argue that 'the Standard of every WoT' will be imaginary entity and it does not make sense.

The important role W3C (along with IETF) has taken on is to provide 'standards for other standards.' Namely, XML is adopted as the data exchange format in many domains (except the Web itself, ironically). XML with proper schema is widely used to describe agreement during standardization efforts. It should be same for WoT. Then, I argue we should adopt XML schema also for WoT.

XML schema ('the W3C XML Schema') is difficult to master, but it can provide any balance between extensibility and strictness. In addition to that, the W3C XML schema has been used to build a set of universally defined types (data models) to describe the world. Already existing standards includes ISO, OASIS, IEEE, ZigBee SEP2, IEC, and so on.

WoT should be the next, but apparently it is a difficult path to follow for embedded devices used in the Things.

### **3 Efficient XML Interchange for WoT**

An encoding scheme of XML is already defined in W3C: Efficient XML Interchange (EXI). EXI is another encoding format for XML. Even without memory-consuming entropy-based compression, EXI has efficient coding scheme based on pushdown automaton. Advantage of EXI is maximized if schemas are given and shared between communicating entities. Schemas can be converted into automaton and some static tables (an EXI grammar). Document structure is embedded in the grammar. Hence, all textual representation of tags and attributes can be removed from both data and code. Implementations for constrained embedded devices exists[3, 1, 2].

Although EXI has many advantages to handle universally typed data in embedded devices, programmers need to know much of XML knowledge. To encode and decode EXI data, you and your devices need to know how to convert an XML schema into an EXI grammar.

To make optimized implementation for each Thing easier, the W3C EXI working group is discussing a grammar exchange format. The goal of grammar exchange format is to get rid of XML schema processor engine from EXI processors. Things can forget about details of XML schema and efficient XML data encoder and decoder will become simple.

One implementation can be based on pre-compiled parser from the grammar. Another implementation may dynamically accept grammars with the grammar exchange format. Using the grammar exchange format, we expect we can minimize implementation of Things that are ready to communicate with universally typed data.

### **4 Conclusion and Future Work: Long Live All Things!**

In this position paper, I argued that we need to pay attention to life-cycle gap between the Web and Things that consist of WoT system. WoT seems to be new concept, but it is not something like web, or other software-oriented solutions.

Once you installed a Thing, it will stay as is regardless the Web is evolving or not.

Cost of innovative evolution is waste. If the waste is software, that's not a serious problem. However, huge waste of Things should be considered harmful in various ways. Thus, to control evolution at communication interfaces, industrial/application-specific standard shall be defined. As XML and XML schema are good tools to describe standards, there are many of them based on XML. The standards can be extended to the world of Things as there are several prototype implementations of EXI processors for embedded systems.

Even though EXI is a good approach, there are many open issues not directly being addressed by EXI itself. An issue I'd like to emphasize is life-cycle gap between the Web and Things. Stronger control on communication interfaces to Things may make a WoT system outdated. Weaker control on it may cause interoperability problem. The gap shall be filled somehow with web-based technology such as automated interface matching or data transcoding based on data model engineering.

## References

- [1] Domenico Caputo, Luca Mainetti, Luigi Patrono, and Antonio Vilei. Implementation of the EXI schema on wireless sensor nodes using Contiki. In *Proceedings of 2012 Sixth International Conference on Innovative Mobile and Internet Services in Ubiquitous Computing*, pages 770–774, July 2012.
- [2] Yusuke Doi, Yumiko Sato, Masahiro Ishiyama, Yoshihiro Ohba, and Keiichi Teramoto. XML-less EXI with code generation for integration of embedded devices in web based systems. In *Proceedings of 2012 International Conference of the Internet of Things*, pages 80–87, October 2012.
- [3] Rumien Kyusakov, Jens Eliasson, and Jerker Delsing. Efficient structured data processing for web service enabled shop floor devices. In *Proc. of IEEE International Symposium on Industrial Electronics (ISIE) 2011*, pages 1716–1721, June 2011.