Web services for Wireless Sensor and Actuator Networks

Guido Moritz
University of Rostock,
Rostock 18051, Germany
guido.moritz@uni-rostock.de

Abstract—**This paper describes the efforts of Dipl.-Ing. Guido Moritz to fill the missing gap of application layer protocols in 6LoWPANs. To achieve long term solutions, flexibility and scalability are main design criteria. The author argues that beside architectural principles and discussions (SOA vs. REST), SOAP Web services are a proper candidate. But solutions tailored already during design time like the emerging CoAP protocol suite are also among the top priorities of the author.**

6LoWPAN, IPv6, wireless, Devices Profile, Web Services, SOAP, XML, REST, SOA, CoAP

I. INTRODUCTION

The author of this paper has graduated at the University of Rostock. Currently he is a PhD candidate at the Interdisciplinary Department at the University of Rostock. The interest of the author is in device centric communication, with special focus on Ambient Assisted Living (AAL) scenarios. The researcher and the research team around the author are forming the core group of the Web Services for Devices (WS4D[1]) initiative.

WS4D is a nonprofit group of researchers and developers and provides a platform for concentrating efforts in the area of device centric communication. WS4D focuses on open standards. Main scope is the Devices Profile for Web Services. The Devices Profile for Web Services (DPWS) can be used to realize both SOAs and RESTful [1] deployments that fit into device centric applications and thus enables the application of SOA and REST in the area of networked devices. DPWS was developed to enable secure Web service (WS) capabilities on resource-constraint devices. DPWS has an architectural concept that is similar but different to the Web Service Architecture (WSA) to fit better into device scenarios. The main difference is the multicast and unicast service discovery with WS-Discovery that does not require any central service registry such as UDDI. But the usage of services on devices is similar to the service usage in WSA, whereby DPWS devices can be directly integrated into WSA based enterprise systems [2], [3], [4]. Hence a DPWS enabled device is a compliant Web service and is based on existing WS-* specifications. Additionally DPWS also features dynamic device and service description, eventing and security.

[1] http://www.ws4d.org

WS4D has published several open source tools and implementations of DPWS for different classes of devices, ranging from embedded Linux to resource rich server system. With the emerging 6LoWPAN (i.e. IPv6 over Low power Wireless Personal Area Networks) protocols, IP enabled WSN provide a common layer for highly interoperable device communication by using matured concepts, technologies and protocols. But further efforts on application layer on top of 6LoWPAN are still an urgent need. Thus, the author of this paper investigates in filling the missing gap of application layer protocols and argues that DPWS is capable of being applied also in highly resource constrained networks like 6LoWPANs.

The intention is to reuse existing protocols and technologies following the tenet of 6LoWPAN protocols. But these protocols and technologies need adaptations and enhancements. The vision is the development of technology to realize a manufacturer-independent Web of Objects. This includes creation, deployment and management of distributed applications and thus objects sharing their resources. Objects (sensors, actuators, devices) are developed to be open, thus share functionality to enable new vendor independent applications potentially unintended during design and development time. The value of the network rises exponentially with the number of objects that contribute (c.f. Metcalfs Law).

II. APPROACH

The challenges of applying SOAP Web Services are mainly based on the expensive bidirectional message flow of SOAP and the related heavyweight data representation in XML. Hence the efforts of the author concentrate on different areas as described in the following subsections. The interests are widespread but required to achieve long term cross domain solutions.

A. Architectural Design and Style

For future IP based wireless communication of smart cooperating objects, RESTful resource-oriented architectures based on HTTP and SOAs implemented by DPWS are proper candidates. Both provide basic functionalities to meet requirements not only of single application scenarios but to be applied as platform independent cross domain technologies. Nevertheless, the high degree of extensibility and flexibility coupled with missing documentation and partly high learning curve leads to pitfalls for both of them. The underlying architectures are most remarkable difference between both, but while based on partly same protocols and technologies, discrepancies are pre-programmed as described by Moritz et al. in [5].

B. *Protocol Design*

Beside architectural design criteria, the author investigates in deep evaluation of existing protocols. WS-* protocols are designed in accordance with the principle “flexibility over optimization”. Thus there is much room left for optimizations to apply the WS-* framework also in resource constrained environments. Principal considerations are described by the author of this paper in [6]. To get rid of data representation overhead, Moritz also analyses optimized platform independent data representations like EXI (Efficient XML Interchange) and FI (Fast Infoset) as described in [7], [8].

C. *Implementation*

A core question for applying the proposed solution is about feasibility on resource constrained sensor and actuator platforms. The author of this paper and the research group has developed several implementations of DPWS, which have past the OASIS WS-DD [9] interoperability tests. The newest development, which is based on the work of Moritz, is the uDPWS[2] (micro DPWS) implementation that fits in the resource constraints of mote class devices like Crossbow TelosB with tens of kB RAM and ROM. Current implementation efforts are concentrating on an EXI parser implementation to be integrated in uDPWS. The currently existing prototype (not published yet) of the EXI parser is implemented in C and has a footprint of round about 10-20kB for schema-informed mode in both bit and byte aligned mode, but without EXI compression.

D. *Standarization*

Based on the research efforts, Moritz and the surrounding research group are participating in standardization efforts, whereby only open standards are supported. The University of Rostock is voting member at OASIS WS-DD [9] and has contributed significantly in shaping the core protocols DPWS, WS-Discovery and SOAP-over-UDP. Furthermore, the author contributes directly to the IETF working groups 6LoWPAN and CoRE. The CoAP protocol is in particular interest of Moritz, as it is based on different architectural design principals than the own approach.

III. Conclusion

The work and interests of the author of this paper are widespread and mainly focus on protocol design, not necessarily coupled with architectural design criteria. But beside the direct protocol design work, Moritz has a clear vision of future applications known under terms like Smart Objects, Cyper Physical Systems and Internet of Things. Thus, the authors implementations are used to realize visionary demonstrators like the world wide first DPWS enabled pulse oximetry oxygen saturation (SpO2) sensor running over 6LoWPAN and embedded in a complex AAL demonstrator[3].

[2] http://code.google.com/p/udpws/

[3] http://en.wikipedia.org/wiki/OSAMI-D

References

[1] R. T. Fielding, “Architectural styles and the design of network-based software architectures,” Ph.D. dissertation, 2000, chair-Taylor, Richard N.

[2] P. Spiess, S. Karnouskos, D. Guinard, D. Savio, O. Baecker, L. Souza, and V. Trifa, “Soa-based integration of the internet of things in enterprise services,” in Web Services, 2009. ICWS 2009. IEEE International Conference on, 2009, pp. 968 –975.

[3] D. Savio and S. Karnouskos, “Web-service enabledwireless sensors in soa environments,” in Emerging Technologies and Factory Automation, 2008. ETFA 2008. IEEE International Conference on, 2008, pp. 952 –958.

[4] H. Bohn, F. Golatowski, and D. Timmermann, “Dynamic device and service discovery extensions for ws-bpel,” in Service Systems and Service Management, 2008 International Conference on, 2008, pp. 1–6.

[5] Moritz, G.; Zeeb, E.; Prueter, S.; Golatowski, F.; Timmermann, D.; Stoll, R.; , "Devices Profile for Web Services and the REST," Industrial Informatics (INDIN), 2010 8th IEEE International Conference on , vol., no., pp.584-591, 13-16 July 2010

[6] Moritz, G.; Zeeb, E.; Pruter, S.; Golatowski, F.; Timmermann, D.; Stoll, R.; , "Devices Profile for Web Services in Wireless Sensor Networks: Adaptations and enhancements," Emerging Technologies & Factory Automation, 2009. ETFA 2009. IEEE Conference on , vol., no., pp.1-8, 22-25 Sept. 2009

[7] Moritz, G.; Timmermann, D.; Stoll, R.; Golatowski, F.; , "Encoding and Compression for the Devices Profile for Web Services," Advanced Information Networking and Applications Workshops (WAINA), 2010 IEEE 24th International Conference on , vol., no., pp.514-519, 20-23 April 2010

[8] Moritz, G.; Timmermann, D.; Stoll, R.; Golatowski, F.; , "encDPWS - message encoding of SOAP Web Services," Pervasive Computing and Communications Workshops (PERCOM Workshops), 2010 8th IEEE International Conference on , vol., no., pp.784-787, March 29 2010-April 2 2010

[9] Web Services Discovery and Web Services Devices Profile (WS-DD), OASIS Open, http://www.oasis-open.org/committees/ws-dd/, 2009.