

A Pragmatic Architecture for Ad-hoc Sensing and Servicing of Industrial Machinery

Nicolaie L. Fantana, *Senior Member, IEEE*, Till Riedel

Abstract— Sensing and servicing industrial machinery using Internet of Things technologies is both important and challenging. In the paper we consider the practical aspects related to the use of wireless sensor nodes as well as RFID for ad-hoc sensing and servicing at an industrial site. An appropriate basic architecture is considered based on web services and DPWS. The architecture, design and prototype implementation are presented and discussed. They take into consideration the needs from the service application, the heterogeneity of devices used and the future extendibility. We show how efficient sensor technology can be combined with standardized web services using a lightweight modular gateway architecture based on message transformation.

Index Terms— Internet of Things; Wireless Sensor Networks; RFID; Web Services; remote sensing; maintenance; service

I. INTRODUCTION

Ad-hoc networked sensing systems can strongly enhance industrial machine servicing by rapid and in situ collection of data from machine, equipment and plant environment [1]. Together with RFID devices wireless sensor networks help to build a decentralized, ad-hoc data and diagnosis infrastructure on site and support service engineers in their tasks. A broad range of wireless technologies has been developed in recent years that enable remote sensing and identification. Integration into an industrial servicing vision comprises seamless access to data from machine identification, from exiting sensing or automation installations, from wireless ad-hoc data collection and enterprise systems via on site data links.

This means that industrial field service applications require means to cope with heterogeneous systems. Furthermore it is desirable from a commercial perspective that industrial servicing applications need to support a certain level of hardware and vendor independence or at least multiple vendor options. A recent editorial [2] nicely summarizes the problem when looking at RFID technology from the technology perspective and the application perspective. There is no single suitable solution in terms of available technology for all wireless sensing problems. Especially an industrial service engineer needs a broad set of tools for his work. The focus when building wireless identification and sensing systems should be on functionality and not on technology. Ideally a system should expose its functional and data centric interface to the outside, to the user. A wide range of concurrent standards is already in place to cover many detailed aspects of wireless sensing. However, they prove rather specific in terms of supported wireless technology and applications. For this reason a more generic approach should not focus on the technical side but on the service it provides to the application. The presented work reflects this in the software architecture of the proposed system.

This work was supported in part by the Federal Ministry of Education and Research as part of the Aletheia research project.

Dr. Ing. N. L. Fantana is with ABB Corporate Research Center, Ladenburg, Germany; (e-mail: Nicolaie.Fantana@de.abb.com).

Till Riedel is with TecO, University of Karlsruhe, 76128 Karlsruhe, Germany (e-mail: riedel@teco.edu).

Wireless nodes for industrial service come with a variety of features. The basic functionality, however, can be split into four functional groups: Identification, Sampling, Processing, Storage and Communication. In this paper we present an approach that builds upon the Device Profile for Web-services (DPWS) to integrate functionality in servicing applications, in order to allow connecting heterogeneous devices like sensor nodes and RFID services to an industrial servicing application. As we have argued in a previous analysis [1] the domain has special requirements that need to be tackled. We show how standardized DPWS services cover the functional groups. E.g. we show how Web Service addressing and discovery perfectly match the identification requirements of ad-hoc measurement application. Additional we define lightweight interfaces to real-time critical services that are essential to enable diagnosis tasks using ad-hoc collected data. Using efficient communication bindings such as proposed in [8] we can further guarantee low communication overhead on the wireless medium and efficient data processing on the node. Further we report on our practical experiences of applying Web Service (WS) interfaces to the application of wireless sensor networks.

II. RELATED WORK

A great amount of wireless sensing research and development has been done in the industrial domain. The focus, however, has been largely in managed installation for either predictive maintenance or wireless control applications. Standards like WirelessHART focus the long-term integration into industrial installations. Reactive servicing has however found considerably less recent attention in the domain of wireless sensor networks although it provides many applications of identification and sensing technology. It is however untypical in the sense that it combines many hybrid technologies to provide a number of different services.

The CoBIs project already investigated the possibilities for service level interfaces to hybrid wireless sensor networks to business processes in industrial scenarios. Starting with Infineon Sindrion platform [3], which uses Universal Plug and Play (UPnP) interfaces to capture the dynamic aspects of a Sensor Network based service landscape, the CoBIs gateway architecture proposed a transformation framework that allowed the via semantic and syntactic transformation descriptions [4]. Similar work based on a Universal Service Descriptor Language that was used to adapt Bluetooth technology to SOA was already done in the uMiddle framework [5].

In our experience, however, UPnP proved inflexible in terms of integration into a system landscape beyond home media players and home firewalls. The broad platform support and interoperability between implementations was especially troublesome. DPWS was proposed as successor to UPnP, which uses WS technology to provide very similar functionality on a better technological platform. There are recent efforts to apply DPWS to the industrial domain. However, they rather focus on control and automation domain [6] with rather controlled system landscapes and more powerful devices. The strength of self-describing and dynamically discoverable devices becomes especially apparent

in field servicing. Here the service engineer does not have a managed infrastructure at hand as he is in the customer domain. Furthermore many service tasks should be the exception in the 40+ years in the lifetime of a machine, so that reactive maintenance still represents the standard case. Especially old machinery is not equipped with ubiquitous sensors for every type of diagnosis. This introduces the need for ad-hoc deployed infrastructure. We can show that this perfectly fits the design of our architecture.

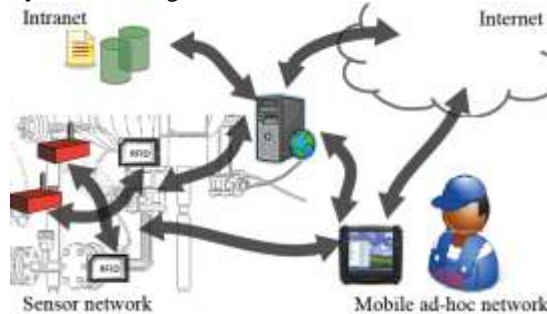


Fig.1 Data flow in wireless sensor solution for field service engineers

III. FIELD SENSING APPLICATION

A system overview of a possible ad-hoc measurement set-up is depicted in Fig. 1. The central part is a mobile device that can range in its capabilities from a smart phone to tablet PC. Its main purpose is to provide the field service engineer with an interface to a service specific online and offline knowledge and case databases as well as to the diagnostic units deployed on site. Wireless diagnostic units are used to analyze customer owned machinery in situ. RFID technology support can range from merely identifying machinery and parts to retrieving measurement from ad-hoc deployable sensor units. For this purpose the mobile device connects to his wireless diagnosis kit and optionally to information sources available through the local infrastructure. This is the envisioned work flow [1]:

- the service engineer identifies machinery,
- retrieves information from knowledge base,
- retrieves a diagnosis program,
- follows instructions for sensor placement,
- uses automatic discovery and like data sources
- uses hand-held visualization and diagnosis
- fixes the problem,
- uploads service and sensor logs to central site

Nodes can be ad-hoc installed or pre-deployed and typically compromise RFID or WSN technology. Currently we are experimenting with a number of different wireless sensor node as well as active and passive industrial RFID tags with and without sensors.

High level service interfaces can hide many non-functional and technical aspects of the underlying technology. This provides the greatest freedom to technology developers and leads to faster technological evolution. At the same time it gives application developers a well-defined technology independent interface. This especially makes sense in the interplay of a rapidly evolving market for wireless sensing and identification solutions and an industry that depends on innovation to provide better service. Web services in general provide a good practice for similar cases where a service can be potentially provided by a variety of servers. DPWS further provides a set of light-weight services that are suitable for an application domain that works with resource constraint service

providers and cannot count on pre-deployed infrastructure.

A. Identification and Discovery

One of the primary concerns of industrial measurement is the unambiguous association of data to data sources. When a service engineer enters an area with pre-deployed measurement units or RFID tags with service histories associated with an installation, he needs to scan the landscape for all devices. He further needs to add sensor nodes, if he needs to gather certain physical parameters that are not covered by the current deployment. WS-Discovery is a very simple standard that allows scanning for devices and services. It can further be extended to search for devices with by custom criteria. Devices can dynamically announce themselves at any time. One important aspect for the servicing use case is that it does not require any infrastructure but works using multicast queries and announcements on peer to peer basis. This discovery scheme can be mapped to a wide range of wireless technologies from RFID batch reading to discovery protocols in wireless sensor networks.

In order to later interpret measurements a sensor device needs to clearly identify itself. The DPWS device based metadata exchange clearly identifies model and device on the basis of common attributes. Because the WS-MetadataExchange is extensible it can be augmented with further information that is important to the use case. An example is location information, which is an important part of the identification data. The functional capabilities and the interface of a node are encoded in WSDL as service. The discovery protocol further allows extending the discovery process by using custom query expression to select nodes.

B. Measurements

The major part of any sensing system is the acquisition of measurement data. Document based (in contrast to RPC style) Web Services provide a very fit tool for those needs-. They are by definition data-centric. Firstly they provide the possibility to define typed and documented interface. XMLSchema based technology based formats has been successfully used. Secondly document based Web Services abstract from tight coupling, which makes them well fit for any wireless applications. DPWS uses WS-Eventing with a simple subscription API and event delivery mechanism, which very well fits typical sensor network communication schemes.

C. Storage

Often it is, however, impossible – mostly for reasons of bandwidth and connectivity (e.g. active RFID)– to send immediately send all data to a measurement client. Therefore it also requires local storage. Using document based WS and accordingly crafted interfaces we can use the same data encoding for stored as for immediately sent data which makes it transparent to a later analysis. Storage is an important service in another context as well: Far from all wireless devices in our use case need the capability of actively sensing the environment. A large part of RFID technology is rather deployed to electronically identify parts of the real world. Web services can provide typed interfaces to storage areas on RFID tags e.g. to locally store service histories.

D. Configuration

The process of measurement is as important as the measurement itself especially when talking about digital signal processing systems. Comtrade is a classical format for transient recording, therefore specifies three files to be stored with each measurement the data set itself, a description and the

configuration parameters. Document based web services provide a well-defined way to exchange configuration data with a sensor node. A web service encoding makes it easy to serialize data for persisting as data sets. Also more complicated configuration tasks that involve changing the measurement service interface by e.g. adding processing tasks that, e.g. change the representation of data from the time to the frequency domain, can be realized using DPWS by announcing the new interface via the discovery service.

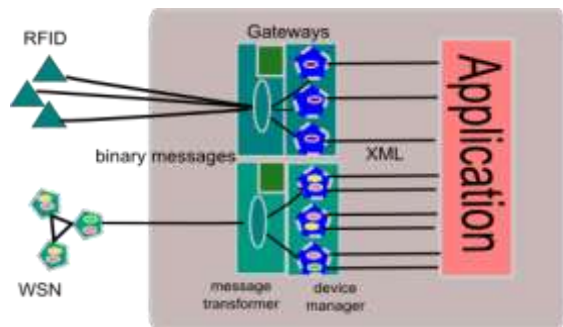


Fig. 2: Simplified architecture diagram of DPWS Gateway

E. Synchronization and Measurement Triggers

Up this point all service fit very well the intuitive hosting relationship between a device and service as it is underlying DPWS. However, especially in sensor networks there are services that must be provided by the network and cannot be logically linked to a single node. We associate such services with the gateway to preserve the device abstraction. A very important example is synchronization and measurement triggers. Here low-level network functionality needs to be exposed to the application. In order to stay abstract enough to include many technologies and retain a loose technology coupling, we propose dual-precision scheme that uses a rather coarse grained time resolution between the client and the measuring nodes but maintains precise timing between different nodes and measurements within a single network according to the technological capabilities.

IV. PROTOTYPE GATEWAY IMPLEMENTATION

The web service gateway itself is build in a modular fashion and is meant to serve as a reference architecture to adapt a number of WSN and RFID platforms to a web service landscape. It utilizes the gSOAP stack and the ws4d Discovery, Eventing and MetadataExchange implementation to run efficiently on embedded devices. The main functionality is captured in three major components that can be easily adapted to different WSN or RFID platforms.

The Device Manager is a generic component manages the proxy instances for devices and their containing services. At its core it is only a network address translation table that is dynamically updated. The platform specific part monitors the underlying network and registers new device. To the outside devices are identified by a URN is generated based on the model and its serial number, that is then resolved to a metadata endpoint. Each endpoint delivers detailed information about the platform, the model and the current device. Additionally each device contains a list of hosted services that is used to announce device specific endpoints. Each service endpoint URL contains both a reference to the device and the service, so that service dispatching can be done based on string matching via a single web server port.

Message Transformations are encapsulated in precompiled libraries for each service. To create a library the code generator reads the WSDL file and creates an acceptor automata for each message part. A call is split into separate transformations so that the transformation itself is stateless. According to the direction of the message the generator applies the according parsing and serialization templates to generate a transformation between the data formats. This works if both formats adhere to the same tree grammar. Specifically we are using it as grammar based compression technique to convert verbose XML payload to optimized binary representations. This saves communication overhead and can easily be parsed on a sensor node. Furthermore we generate optimized data type encodings for numerical values based on interface restrictions like minimum and maximum values and convert them to network byte-order. We provide an eclipse plug in for generation and automatic building of service transformation.

The Call and Subscription Handler is used to provide callback routing to the client. In order save network bandwidth and node memory subscriptions are pooled in the gateway. When an event arrives the all subscriber endpoints are individually called via a Web Service callback. The web service side of subscription handling was implemented using the ws4d WS-Eventing service. Call using explicit reply addresses are handled similar to events the arrival of a reply message. For anonymous replies the call's socket context is also maintained.

V. PROTOTYPE EVALUATION

The general gateway is designed to be largely independent of different hardware platforms. As a core it provides a generic message transformation infrastructure. We needed to customize transport, routing and discovery for our target platforms and we employed customized target code generation for the server stubs on the sensor node side. We started out with porting the Particle Computer 2/29 platform with the SSimp 2/02 generic sensing board to our system. We had employed the hardware in early prototypes of our industrial service application. The original system used an IPv4/UDP to AwareCon Bridge. The resource efficiency (4MIPS pic18f6720 MCU with 4KB RAM and an ultra-lowpower 869MHz TR1000 radio) and the variety of present sensors make it a good choice for the evaluation of the system and allow us to port existing applications to the WS interfaces.

The second platform we newly developed better reflects concrete requirements of the application domain. We integrated a 16bit dsPIC33FJ128GP706 controller together with a variety of typical sensor needed for machine diagnosis. The board includes 3-axial MEMS acceleration sensor, a piezo body sound microphone, a MEMS microphone. It further comprises a separate communication board which runs Contiki OS with a 6LoWPAN stack on a Jennic JN5139 2.4GHz 802.15.4 DSSS radio SOC with an additional USB transceiver. Both the communication board and the 2GB SPI flash memory can be driven from the DSP-Board via DMA. This allows us to drive communication, storage and sampling concurrently at a maximum rate. This platform allows us to validate the whole spectrum of services that can be provided via Web Services. The service stubs for both platforms were automatically generated using the techniques described in [8].

We implemented both measurement services on sensor nodes as well as measurement clients using the proposed

techniques. The implementation was part of an ongoing effort to port existing platform specific API based wireless measurement application (see Fig. 3). The DPWS gateway replaces the current interface to network scanning, node configuration, data series measurement and serialization.

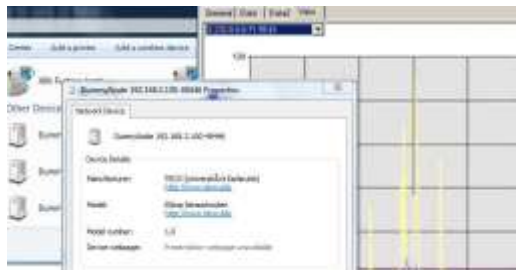


Fig 3. Dynamically discovered nodes and application prototype

A. Performance

Performance is one of the primary concerns when using verbose techniques like web services. We already showed in [8] that by using XML to binary translators we can use verbose interfaces without putting extra overhead on the sensor node or the wireless communication channel. This also applies data to RFID tags as long as part of the work is done by the gateway. This is not as critical because the message load of sensor network is typically bounded by its bandwidth. With our new 6lowpan node we can support a maximum stream payload of 15kbyte with the pPart it is even only 2kbyte on a single network: A more critical part turned out to be actually not the sensor network side of things but the typical Web Service client stacks. A gateway principally cannot make any assumptions about a client in contrast to the other way around. Especially when using callbacks, the clients, however, act as server. With one or more clients it is very difficult to maintain the overall performance of the gateway. With a maximum speed of 13byte every 13ms of binary payload on the producer side, in our system the gateway and client would run at about 1-2% processor utilization each on a 1.6GHz Pentium Core2 system (currently scales linear with the number of clients). We tried the same event rate on the Materna DPWS Explore 2 the throughput could drop to 5 messages per second and the processor utilization while creating a huge message queue in the gateway.

B. Development

Aside from many teething problems of web service stacks one big advantage of using Web Services in our applications was that we could speed up the development cycle. On one hand we were able to use the rather mature Windows Communication Foundation for client programming, which gave a great deal of comfort. On the other hand we were much better able to provide functional mockups of sensing technology ("dummy nodes") before the real hardware was ready. This allowed us to concurrently develop and test functionality on both client and server side.

C. Interfacing in Hybrid systems with Web Services

Another very practical issue when dealing with Web Services still is the interoperability of different implementations. It is still much easier to stay within one framework such as the Windows Communication Foundation, than to implement server and client on two different systems. Although thanks to recent efforts to support rather uncommon things like the UDP discovery it was easy to get started, however, we had to put big amount of work simple things like

asynchronous messaging SOAP HTTP to work on all sides. It seems that even simple standards like WS-Addressing or WS-Eventing allow many practical variations and still need much manual work. This is one reason, why we believe that a gateway approach using much simpler communication schemes on the sensor nodes is the way to go. One downside of access by proxy is that subscriptions and call context cannot be easily maintained when a node moves between multiple gateways, which do not currently seem to be a problem in the application. Even if binary SOAP bindings like EXI will find their way into the web service stacks and enables the direct use of sensor node web service e.g. via 6lowpan [7], full standard compliance still seems a bigger problem on resource restricted devices. In our application end-to-end Internet scale access to any node is not a requirement but may even lead to security problems. In our gateway system we can have a trusted gateway that can provide access control and integrate with existing systems. Similar considerations have to be made when providing access to nonprogrammable nodes like passive RFID tags or proprietary wireless sensors. Transparently to any client application a gateway can provide consistent interfaces.

VI. CONCLUSION AND FUTURE WORK

We presented, an approach and an architecture based on web service technology which can be applied to the ad-hoc industrial sensing and servicing domain using Internet of thing devices. Our first prototypical implementation of web service enabled measurement clients, wireless sensor nodes and gateways proves promising way give a field service engineer access to a variety of wireless networked sensing and information tools. Challenges still exist e.g. related to the integration of various system like classical RFID into our gateway architecture, how to implement transaction mechanisms and caching to allow a better decoupling between client applications and wireless devices. Localization is an issue especially if it has to be for ad-hoc systems and at low cost. Irrespective of the technological challenges of localization in industrial settings, we believe that DPWS discovery and meta-data exchange allows many extensions in those directions. We see web service based approaches an open technology that allows us to easily integrate many hybrid systems for applications in industrial sensing and servicing.

REFERENCES

- [1] N. Fantana and T. Riedel, "Servicing industrial machinery" INSS 2009, pp. 70ff.
- [2] M. Roberti, "Why RFID Vendors and Users Speak Different Languages," RFID Journal, Feb. 2009.
- [3] Y. Gsottberger, et al., "Sindrion: a prototype system for low-power wireless control networks" IEEE MASS 2004. pp. 513ff.
- [4] "Architecture for Collaborative Business Items," Architecture of Computing Systems - ARCS 2007, 2007, p. 142ff.
- [5] J. Nakazawa, H. Tokuda, W.K. Edwards, and U. Ramachandran, "A bridging framework for universal interoperability in pervasive systems," Proceedings of the 26th IEEE International Conference on Distributed Computing Systems, 2006, pp. 3ff.
- [6] F. Jammes and H. Smit, "Service-oriented paradigms in industrial automation," IEEE Transactions on Industrial Informatics, vol. 1, 2005, pp. 62ff.
- [7] Guido Moritz, "DPWS for 6LoWPAN," Dec. 2009.
- [8] Till Riedel, Nicolaie Fantana, Markus Scholz, Christian Decker "A Model Driven Internet of Things", INSS 2010