Towards an Open Infrastructure for Fast Prototyping Applications on the Web of Things

Vlad Trifa

Institute for Pervasive Computing, ETH Zurich
SAP Research CEC Zurich,
Kreuzplatz 20, 8008 Zurich, Switzerland
vlad.trifa@ieee.org

Abstract. Wireless Sensor Networks provide unprecedented possibilities for monitoring and interacting with the real-world. Unfortunately, the lack of open and simple standards for ad-hoc collaboration between heterogeneous embedded devices makes it difficult to build large-scale deployments. For every particular application requires complex integration work, and therefore technical expertise, effort and time. Following the success of Web 2.0 mashups, we propose to reuse the founding principles of the modern Web architecture as a lightweight middleware. Using Web standards to network all kinds of devices will encourages widespread adoption by maximizing reuse of shared devices and lowering the access barrier for people to use and develop applications on a much wider ecosystem of shared devices. With our approach we push the World Wide Web towards sensor networks so that interacting with devices becomes as easy as typing a URI in a Web browser.

1 Problem statement

Wireless sensor networks (WSNs) offer unprecedented possibilities for monitoring the real-world, and could become an invaluable help in many disciplines. Unfortunately, most research projects have focused on building vertical solutions designed for very specific applications that run as isolated and small scale testbeds. As a consequence, the lack of commonly agreed standards for integration of devices with other applications have resulted in a wide variety of hardware and software platforms. The real value of such applications comes from the sharing and integration of data among heterogenous devices. Without simple and open standards for ad-hoc networking and interaction with embedded devices, building and maintaining large-scale deployments requires extensive effort and expert knowledge. Various middleware have been proposed to facilitate the development of applications for sensor networks (for a recent survey see [1]). Unfortunately, existing approaches are too complex to use and are based on tightly coupled components, which strongly hinders the scalability and evolvability of the whole system. As a consequence, a worldwide Web of sensors (the *Internet of Things*) where billions of "smart" objects can be easily shared and reused, still remains largely unfeasible today.

Based on these observations, the central topic of my research is to design a simple, lightweight, and loosely coupled architecture to build a (globally) scalable network of heterogeneous devices that can interact together without any *a priori* knowledge about each other. The key requirements are to maximize sharing and reuse of services offered by embedded devices, and minimize the time needed to build applications that run on top of them.

As demonstrated by the tremendous success of the Web, loosely coupled approaches offer the greatest scalability, robustness, and evolvability. In recent Web 2.0 applications, the focus has been on the user and user-generated content, as many applications were made available as services accessible through public APIs. This has enabled non-programmers to develop *mashups*, which are hybrid applications that combine Web content from various sources. Because the number of embedded devices with Internet connectivity is growing rapidly, our design goal is to fully leverage the existing and ubiquitous Web standards to build a middleware for embedded devices. Besides, Web standards will allow any device to *speak* the same language as other resources on the Web, making it much easier to integrate the real world with any other Web content. Therefore, physical things can be bookmarked, browsed, googled, and used just like any other Web page.

Tight coupled back-end systems using proprietary protocols will remain the most desirable choice for building high-peformance system with real-time requirements (as for example in the industrial automation or banking domain). These systems can expose their functionality in a high-level abstraction that can be accessible from the web in a plug-and-play manner, that can be used directly over HTTP. However, much simpler loosely coupled approaches are to be preferred for tasks with more modest requirements (which represent most use cases for data monitoring and home application), because of their inherent flexibility and intuitive use. The expected outcome is to lower the entry barrier for developing monitoring applications by allowing people to easily reuse and combine data from sensed by different devices anywhere around the globe.

2 Related work

Early distributed programming abstractions based on Remote Procedure Calls (RPCs) have simplified the development of complex distributed systems [2, 1]. However, these systems can't be used to build evolvable and scalable networks of embedded devices. In a world where billions of smart objects that can sense, process, and exchange information with each other, a common language understood by all devices is crucial. JXTA was an open network computing platform designed for peer-to-peer computing that can be implemented on all kinds of devices [3], but was not based on the Web architecture. More recently, Web services have also been used to interconnect devices on top of standard Web protocols [4, 5].

With advances in computing technology, tiny Web servers could be embedded in most devices [6]. The idea of each thing having its own Web page is appealing because Web pages could be indexed by search engines, then searched and accessed directly from a Web browser. Early approaches to linking objects with the Web were based on physical tokens (such as barcodes or RFID tags) [7]. In the Cooltown project [8], each thing, place, and person have an associated Web page with information about them. Other systems for integration of sensor systems with the Internet have been proposed [9–11], but none of them reuse the Web as application layer. An early gateway system similar to ours has been proposed [12]. The SenseWeb project [13] is a platform for people to share their sensory readings using Web services to transmit data on a central server. Pachube [14] offers a similar community Web site for people to share their sensors but uses more open data formats. Unfortunately, these approaches are based on a centralized repository and devices need to be registered before they can publish data, thus are not sufficiently scalable. Prehofer et al. [15] recently proposed an web-based middleware that is similar to our approach, however Internet is used only as a transport protocol, and no references to use a fully Web-like approach is not mentioned.

The goal of most previous applications, was to only offer an online representation of real things for humans (real-time sensor data displayed on a HTML page). To our knowledge, no attempt seamlessly integrate devices into the Web to enable sharing and reuse of their functionalitieshas been mentioned. However, existing most of these Web-based approaches use HTTP only to transport data between devices, whereas HTTP is in fact an application protocol. Projects that specifically focus on re-using the founding principles of the Web as an application protocol are still lacking. Creation of devices that are Web-enabled *by design* would facilitate the integration of physical devices with other content on the Web. As pointed out in [16], in which case there would be no need for any additional API or descriptions of resource/function. Early promising attempts to use REST on devices are found in [17, 18], unfortunately seem to have been abandoned. The approach found in [19] has a very similar approach to ours, but focuses mainly on the discovery of devices, unfortunately a more systematic approach and system evaluation are lacking.

3 Approach and methodology

However, most of these approaches are a legacy of RPC systems, therefore are not flexible enough for highly dynamic environments where new, unknown devices that continuously appear and disappear. Unfortunately, they are not sufficiently loosely coupled to be useful when it comes to dealing with the constraints and requirements of mobile embedded devices, in particular for ad-hoc interaction with new devices with unknown properties. Most of these approaches are based on tightly coupled solutions, where each element had full knowledge about the other peers and the functions they offered. Unfortunately, this approach is not sufficient to deal with the constraints and requirements of mobile embedded devices, in particular for ad-hoc interaction with new devices with unknown properties.

The central element of my research is be the design, implementation, and evaluation of an infrastructure to interconnect embedded devices that is built according to the founding principles of the modern Web architecture, in particular **REST** (REpresentational State Transfer) defined in [20]. The idea behind REST is that the each important concept of an application, in our case properties and functionalities of devices, are represented as Web resources that can be accessed using a globally unique resource identifier (URI) and HTTP.

A core research question is *how can common Web standards (HTTP, XML, or RSS) be adapted and used to push the Web onto physical devices to build the Web of Things*? As opposed to the *Internet of Things* where the Internet is used only as a mere protocol to transport "proprietary" data between devices; the Web of Things HTTP is used directly as the application protocol. In a Web-Oriented Architecture, the focus is on resources instead of services or functions. This brings the second question which is: *how can the data and functionalities offered by networks of low-power embedded devices be easily made available as standard Web resources*, thus facilitating reuse and integration with other applications?

Essentially, RPC and REST will both yield similar results. While, the traditional RPC systems will allow to do more powerful things (but in counterpart is more complex to use and to maintain); the REST would allow to develop simpler things and is much simpler to implement (thus suited for both fast prototyping of physical mashups and for ad-hoc interaction in ubicomp scenarios).

We will analyze the requirements of different scenarios for different domains to identify a set of common patterns and requirements for different types of applications. More particularly, based on previous work in event driven architectures and stream processing, we will implement and analyze how the low systems can be implemented and how to bridge it with the web. My core research topics can be summarized in four parts.

Web-enabled embedded devices The first step is to investigate the use of REST on embedded devices with IP connectivity and propose a systematic/formal methodology and practical guideline for developing RESTful devices. In particular analyze the tradeoff between expressiveness and code complexity/size.

Traditionally, low-power communication protocols are used for sensor networks and their limited resources prevent them to implement a complete Web server. To enable web-based interaction with such devices, we propose the concept of *Smart Gateways* to offer a unified Web interface to embedded devices. A gateway is a lightweight and modular software components that enable Web-based interactions with all kinds of embedded devices that do not have a TCP/IP connection, thus hiding the complexity of heterogenous devices and network protocols they use. We will explore the possible mapping mechanisms between the loosely coupled world of Web resources, and the tightly coupled approaches found sensor networks (programming paradigms and protocols). Additionally, gateways *augment* the base capabilities of sensor networks by taking care of the computational overhead required to share physical devices over the Internet, and provide advanced functionalities that are not available on the devices themselves. Finally, gateways are designed to be linked together to form hierarchical trees that could be easily mapped to physical locations, thus provide support for location-based services on the Web where devices can be searched according to their location.

Dynamic devices discovery Discovery of new devices is an essential component of most distributed computing paradigms, in particular when sharing and using resources on open networks. Most solutions for discovery can only locate devices that use the same protocol on the local network, and a centralized repository to store information about new devices is not scalable. We are exploring lightweight discovery solutions where devices send minimal information about them (e.g. an EPC code or a URI where data about that device can be retrieved), and the gateway could retrieve a resource descriptor from the homepage of the constructor for example. Several possibilities for dynamic discovery and search in the context of devices augmented with semantic information about their capabilities have been proposed, as for example mRDP [21]. We will take inspiration of early work to derive a simple and generic discovery mechanism for RESTful devices.

Messaging and Eventing A major success factor behind Web 2.0 has been RSS. By allowing people to subscribe to changes on their favorite Web pages RSS has enabled the *live* Web. More recent standards such as ATOM and ATOM Publishing Protocol (AtomPub) are simple and flexible HTTP-based standards to create, update and fetch Web resources. Unfortunately, ATOM is not adapted for data streams and real devices. However, Web technologies have become mature and powerful enough to be used in dynamic contexts and for real devices, for example to perform stream processing [22]. We propose to further investigate this research direction to enable the creation of Web-based complex stream processing and event-driven systems that transform data from real devices and sensor networks into a similar mechanism built on top of Web technologies. Such a messaging mechanism will allow to program applications for physical devices in a declarative way, similarly to the way Yahoo pipes[1] are used to aggregate and filter feeds based on their content and properties. We will compare the requirements of home entertainment and industrial systems, and evaluate against them the use of Web technologies to built mashup applications for these different domains.

System evaluation To test and validate the proposed approach, a realistic testbed needs first to be built. Currently we are developing an large-scale ecosystem of heterogeneous RESTful devices that will all be interconnected on this Web of Things. The goal is to use a large variety of devices in terms of functionality, network protocols, and system resources available. Additionally, a simulator to

[1] `http://pipes.yahoo.com/`

implement and test large deployments with thousands up to millions of devices will be necessary to evaluate the feasibility of our approach on a large scale. Based on our requirements, the following factors will be evaluated:

- **Simplicity:** the proposed approach must be transparent and easy to use to be adopted. For this purpose, the testbed will be made available on the Web, and people will be invited to use it to build their own mashups, and user studies will be performed to evaluate the usability of our system.
- **Scalability:** we will analyze the scalability of our approach and what factors improve the scalability of large-scale deployments. In particular we will quantify the effects upon scalability induced by dynamic environments and devices, in comparison to static documents on the Web.
- **Interoperability:** we will integrate a large number of different devices that provide different types of information, and test how easily devices can be made to communicate with each other without human intervention.
- **Performance:** we will evaluate the raw performance of our smart gateways, in terms of the overhead introduced by the gateways and how much augmentation of low-power devices can be beneficial from various perspectives (computational, energy, response times, etc.)

In order to evaluate and validate the utility and performance of a Web-based middleware for interconnecting devices, a thorough quantitative and qualitative comparison of a Web based architecture versus more tightly coupled approaches will be also performed for the mentioned factors. In particular, a thorough investigation of the requirements and properties for such a system (what properties are essential and which constraints can be relaxed).

4 Preliminary results

We have developed a simple RESTful architecture to interact with low-power devices and have implemented it on low-power sensor nodes that can POST data directly into real enterprise systems. This allows people to *browse* devices and navigate the resources they offered directly with their browser. Based on that, we have showed that REST is a surprisingly easy-to-use interaction paradigm for fast prototyping simple applications that combine data from different devices.

We have also built a complete working prototype for the Smart Gateway that supports automated discovery of low-power devices, and implemented drivers for SunSpot, TMotes, and RFID readers. A simple discovery mechanism has been implemented that allows to retrieve semantic information about each device and make it available on the Web (both devices properties and the resource they expose). Whether the Web page is physically located on the device or proxied by the gateway is completely transparent to the user, therefore offers a standard and uniform Web interface to users. On top of the, we have implemented a simple HTTP-based eventing mechanism that allows users to subscribe to events generated by devices using keywords (i.e. you can subscribe to a RSS feed on any gateway that will contain all the events from any device that contain the keyword). After the current testing phase, we are planning to release the gateway source code as an open-source project and build a community around to motivate people to RESTify their devices.

Finally, to test large-scale applications we have built a simulator based on multi-agent systems that simulates RESTful devices that are directly accessible through a smart gateway. In the near future, we will extend the simulator to fetch real sensor readings and device metadata from real devices through Pachube [14], thus to completely integrate our simulator with the Web.

5 Conclusion and future steps

With the large amount of devices that will be connected to the Internet of Things, manual management and configuration of each device will simply be impossible. Therefore, we are exploring Microformats[2], where simple semantic information (tags) is directly embedded within the web pages generated by a device. This makes the information about a device to be simultaneously viewable by a human with a Web browser and processed by a machine, i.e. the page *is* the interface. Although the Semantic Web is not my research topic, a central design goal for the gateway is to support *by design* existing semantic Web technologies.

The Web as it exists today completely lacks the real-time aspect of our everyday lives and that matters today. Enabling billions of embedded devices to become citizens of the Web will contribute to the creation of a real-time *Web of Things*. This research could benefit first and foremost Web developers. The development of a newer generation of Web Services suited for mobile and embedded devices that are Web-enabled by design will facilitate the development of hybrid Web applications that blend the real world with virtual information. Additionally, the sensor network community could also profit from because integration of data from physical devices with any other Web resource will be straightforward. In the long run, extending the Web to integrate physical devices will reshape the Internet into a versatile collection of physical and virtual resources that can easily (re)combined at run-time to solve any task at hand.

References

1. Wang, M.M., Cao, J.N., Li, J., Dasi, S.K.: Middleware for wireless sensor networks: A survey. Journal of Computer Science and Technology **23**(3) (May 2008) 305–326
2. Römer, K., Kasten, O., Mattern, F.: Middleware challenges for wireless sensor networks. SIGMOBILE Mob. Comput. Commun. Rev. **6**(4) (2002) 59–61
3. Traversat, B., Abdelaziz, M., Doolin, D., Duigou, M., Hugly, J., Pouyoul, E.: Project JXTA-C: Enabling a Web of Things. In: Proceedings of the 36th Annual Hawaii International Conference on System Sciences. (2003) 282–290
4. Jammes, F., Mensch, A., Smit, H.: Service-Oriented device communications using the devices profile for web services. In: Proceedings of the 21st International Conference on Advanced Information Networking and Applications Workshops - Volume 01, IEEE Computer Society (2007) 947–955

[2] `http://www.microformats.org/`

5. Priyantha, N.B., Kansal, A., Goraczko, M., Zhao, F.: Tiny web services: design and implementation of interoperable and evolvable sensor networks. In: SenSys '08: Proceedings of the 6th ACM conference on Embedded network sensor systems, New York, NY, USA, ACM (2008) 253–266
6. Boriello, G., Want, R.: Embedded computation meets the world wide web. Commun. ACM **43**(5) (2000) 59–66
7. Ljungstrand, P., Redström, J., Holmquist, L.E.: Webstickers: using physical tokens to access, manage and share bookmarks to the web. In: DARE '00: Proceedings of DARE 2000 on Designing augmented reality environments, New York, NY, USA, ACM (2000) 23–31
8. Kindberg, T., Barton, J., Morgan, J., Becker, G., Caswell, D., Debaty, P., Gopal, G., Frid, M., Krishnan, V., Morris, H., Schettino, J., Serra, B., Spasojevic, M.: People, places, things: web presence for the real world. Mob. Netw. Appl. **7**(5) (2002) 365–376
9. Gibbons, P., Karp, B., Ke, Y., Nath, S., Seshan, S.: Irisnet: an architecture for a worldwide sensor web. Pervasive Computing, IEEE **2**(4) (2003) 22–33
10. Balazinska, M., Deshpande, A., Franklin, M., Gibbons, P., Gray, J., Nath, S., Hansen, M., Liebhold, M., Szalay, A., Tao, V.: Data management in the worldwide sensor web. Pervasive Computing, IEEE **6**(2) (2007) 30–40
11. Inc., O.G.C.: Ogc sensor web enablement: Overview and high level architecture. White paper OGC 07-165 (2007)
12. Schramm, P., Naroska, E., Resch, P., Platte, J., Linde, H., Stromberg, G., Sturm, T.: A service gateway for networked sensor systems. Pervasive Computing, IEEE **3**(1) (Jan.-March 2004) 66–74
13. Kansal, A., Nath, S., Liu, J., Zhao, F.: SenseWeb: an infrastructure for shared sensing. IEEE Multimedia **14**(4) (2007) 8–13
14. Haque, O.: Pachube. Online at `http://www.pachube.com`
15. Prehofer, C., van Gurp, J., di Flora, C.: Towards the web as a platform for ubiquitous applications in smart spaces. In: Second Workshop on Requirements and Solutions for Pervasive Software Infrastructures (RSPSI), at Ubicomp 2007. (2007)
16. Wilde, E.: Putting things to REST. Technical Report UCB iSchool Report 2007-015, School of Information, UC Berkeley (November 2007)
17. Drytkiewicz, W., Radusch, I., Arbanowski, S., Popescu-Zeletin, R.: prest: a rest-based protocol for pervasive systems. In: Proc. of the IEEE International Conference on Mobile Ad-hoc and Sensor Systems. (Oct. 2004) 340–348
18. Luckenbach, T., Gober, P., Arbanowski, S., Kotsopoulos, A., Kim, K.: TinyREST - a protocol for integrating sensor networks into the internet. in Proc. of REALWSN (2005)
19. Stirbu, V.: Towards a restful plug and play experience in the web of things. Semantic Computing, 2008 IEEE International Conference on (Aug. 2008) 512–517
20. Fielding, R.T., Taylor, R.N.: Principled design of the modern web architecture. ACM Trans. Internet Techn. **2**(2) (2002) 115–150
21. Vazquez, J.I., de Ipiña, D.L., Sedano, I.: SoaM: A Web-powered Architecture for Designing and Deploying Pervasive Semantic Devices. IJWIS - International Journal of Web Information Systems **2**(3-4) (2006)
22. Dickerson, R., Lu, J., Lu, J., Whitehouse, K.: Stream feeds: an abstraction for the world wide sensor web. In: Proceeding of the 1st Internet of Things conference (IOT), Zurich, Switzerland (2008)