So Many Sensors, So Little Data

Seth Holloway, Drew Stovall, Angela Dalton, Christine Julien
The Mobile and Pervasive Computing Group
The University of Texas at Austin
Austin, TX, USA
{sethh,dstovall,adalton,c.julien}@mail.utexas.edu

ABSTRACT

Future networks are ones in which mobile users are immersed in sensor networks. Application domains such as smart homes, intelligent construction sites, and social networking require users moving through an environment to collect and process data available in that environment. However, even as sensors become increasingly prevalent in our environments, only a handful of applications are available to interact with and process the available sensor data. Poor usability accounts for a good portion of this disparity. In this paper we discuss our directions in creating an architecture for making such sensor data usable by application developers. This has the potential to widen the accessibility of developing mobile applications on embedded sensor networks beyond the niche markets it now enjoys.

Categories and Subject Descriptors

D.2.11 [**Software Engineering**]: Software Architectures; D.2.11 [**Software Engineering**]: Software Architectures—*Domain-specific architectures*

General Terms

Standardization

Keywords

sensor software architecture, sensor web enablement, sensor data over http, web programming

1. INTRODUCTION

Sensors and sensor networks are rapidly becoming entwined with our everyday lives. Sensors embedded in our automobiles provide constant monitoring and feedback about the state of the car. On intelligent construction sites, sensors connected to equipment and workers on the site constantly monitor conditions as people move and equipment is used, enabling real time safety assessment. In our homes, the ability to embed sensors in the structure and our belongings will enable our homes to assist and accommodate us as we move through them on a daily basis. Technological advances have made it possible for almost anyone to own and interact with sensors, however applications that collect and process sensor data have remained almost exclusively research-based, proprietary, and designed for a single purpose. A major impediment to more widespread application availability is the usability of sensors and sensor data. Available sensors are, in general, hard to configure and use, and they transmit their data in an obscure, proprietary format. While the vast majority of sensor network research focuses on communication protocols and performance optimization, in this work, we look at providing a software architecture that enables better integration of sensors and sensor networks with mobile applications. We aim to make data from immersive sensor networks such as those described above available in a usable (meaning both easy to understand and easy to use) way so that the average programmer can easily produce useful applications that leverage the extensive capabilities of sensors.

The range of applications that can be enabled through increased availability of sensor network data is enormous. Smart homes and intelligent construction sites should enable users moving in and around structures to interact with sensors in that structure; social networking applications should allow mobile users to interact not only with each other but with information embedded in the immediately local environment (e.g., a mall or an airport); military applications must allow mobile forces to sense and operate on data collected in real time. Enabling programmers across these applications is difficult as the range of technical experience and intended use varies greatly. However, our approach takes insight from the fact that web programming is highly accessible, as demonstrated by the huge number and variety of websites created by programmers of widely varying skill sets. By making sensor network interaction look and feel like web programming, we open doors to new sensor network application developers. Therefore, our approach builds a software architecture that enables sensor data dissemination based on web programming principles.

Our approach generates a framework through which sensors autonomously and directly post their data "online" through traditional web programming approaches. We (and others) refer to such an approach as *sensor web enablement*. Our approach differs specifically from others in that we provide a loose definition of "online" by not restricting the location of the data posts to a single centralized location. Instead, we assume an underlying network infrastructure that supports communication between the sensors and

SAM'08, May 10, 2008, Leipzig, Germany.

a web server; the web server that receives the sensors' posts could in fact be attached to a mobile device.

2. RELATED WORK

The notion of publishing sensor data online is not entirely new, and several related projects have paved the way for our proposed approach. CoolTown allows networked mobile devices to publish data on the web through a variety of tailored protocols [1]. These posts include information about a device's characteristics (location, for example), enabling a degree of content-based discovery. Another very ambitious project has created a centralized clearinghouse for sensor data generated worldwide [3]. Sensors first communicate their data to a base station supporting a particular sensor network. The base station is connected to the Internet and uses that connection to funnel the sensor data to the clearinghouse. To enable immersed applications such as those described above, we feel that a more decentralized approach that does not rely on persistent connectivity to a specific central point is more suitable.

Other current approaches to publishing and sharing sensor data build on standard web services utilizing SOAP, WSDL, and XML. The Open Geospatial Consortium (OGC) allows governments and companies to take advantage of a massive set of federated devices and sensors. They accomplish this goal using SensorML, a sensor model language that defines an XML schema to identify and use sensors [2]. Microsoft's SenseWeb project [7] has also provided a generic method to push sensor data online. However, both approaches rely on SOAP web services, which for immersive deployments that we target, can be inflexible, slow, hard to maintain and manage, and heavyweight.

The goal of our approach is to enable sensors to autonomously (without relying on a base state for transition or interpretation) post their data to be shared and available to applications. We minimize the interface for both sensors and programmers, relying on a simple but expressive form for data movement (in this case, using HTTP GET and POST commands). Our approach is consistent with REST principles [5] in an effort to be lightweight and flexible. Some work has been done to apply the REST style in the sensor domain [4, 6] although this work violates many REST principles and requires overhead that reduces usability.

3. POSTING SENSOR DATA TO THE WEB

The goal of our approach is to make sensor data usable from two directions. First, it is important that it is simple for sensors (and therefore people who develop *sensing* applications) to post their data in a shared space. Second, developers of immersive mobile applications must be able to access and use stored sensor data in a simple way. To aggregate and synthesize data in the most straightforward manner, we rely on the enormous body of research and technology that has already been developed to support the HTTP protocol.

In our approach, the user desiring to consume sensor data creates a standard web application with a variety of simple forms that accept data. Identifiers for these shared spaces (i.e., URLs) are distributed to sensors which subsequently use standard HTTP methods to transmit the desired sensor values to the pages and therefore into the user's application. The distribution of the URLs and data posts relies on the availability of the appropriate underlying discovery and routing protocols. We discuss the related challenges a bit more in the next section. Sensors can also read data from the URLs they are provided; for example, a sensor could retrieve new configuration parameters such as the desired sensor fidelity or reporting frequencies.

Our approach allows for a flexible backend built with standard web programming tools and techniques. This piece runs on the higher-powered mobile device or stationary server. By building on standard web technology, we ease the transition from sensor data to application by allowing developers to directly incorporate sensor data into existing sites or create new more sophisticated sites or applications that consume the data. Applications receiving multiple types of sensor information could aggregate or interpolate the data received, providing more abstract representations to users. Received sensor data can even be used as context to enable adaptive applications that respond to environmental conditions. The availability of the sensor data in a web format simply makes the development task in all cases easier.

Ideally, sensors that participate in this scheme will require little to no tinkering from the user. At a minimum, however, the sensors must support a lightweight and complete HTTP stack to support interaction with the framework. Beyond that, simple configuration such as transmitting the necessary URLs can be handled through facile web interfaces that send the URL information to the sensors either through a physical connection or over the air.

To reduce the amount of data sent and conserve battery power we expect that sensors would employ simple logic to periodically sample data or post only if a threshold is met. These uncomplicated algorithms will likely be preprogrammed on many sensors, though ideally manufacturers would provide configuration options that users could access without in-depth knowledge.

4. DIRECTIONS

In this paper, we have proposed a software architecture that eases the complexity of incorporating sensor data in mobile and pervasive applications. Our approach is focused on system usability, both from the perspective of programming the sensors to adhere to the architecture and from the perspective of enabling applications to use it.

To move this project past the idea stage, there are several avenues to investigate more carefully. First is the manner in which HTTP connections are created and maintained from sensors directly to the posting location, given the potentially unreliable network that supports them. Existing approaches have avoided this challenge by connecting only the base station of a sensor network to the centralized web location. Because our aim is instead to enable sensors to post directly without the need for a base station, we may become encumbered by unpredictable links, and our approach must account for this possibility.

In addition, our current implementation relies on simple one-hop communication approaches. Implementing our architecture over mobile discovery and routing protocols will give us good insight into how expressively stating URLs can enable both the sensors and applications to influence the discovery process. For example, applications can use content information to determine to which sensors the requests for posts are pushed (i.e., which sensors the configuration messages described in the previous section target), and in-

formation in the URLs can influence when, where, and how data from sensors is posted to mobile applications.

5. ACKNOWLEDGMENTS

The authors would like to thank the Center for Excellence in Distributed Global Environments for providing research facilities and the collaborative environment. This work was funded, in part, by the National Science Foundation (NSF), Grant # CNS-0620245. The views and conclusions herein are those of the authors and do not necessarily reflect the views of the sponsoring agencies.

6. REFERENCES

[1] J. Barton, T. Kindberg, H. Dai, and N. Priyantha. Sensor-enhanced mobile web clients: an XForms approach. *WWW*, pages 80–89, 2003.

[2] M. Botts. SensorML website: http://vast. uah. edu.

[3] K. Chang, N. Yau, M. Hansen, and D. Estrin. SensorBase.org-A Centralized Repository to Slog Sensor Network Data. *DCOSS/EAWMS*, 2006.

[4] W. Drytkiewicz, I. Radusch, S. Arbanowski, and R. Popescu-Zeletin. pREST: a REST-based protocol for pervasive systems. *MASS*, pages 340–348, 2004.

[5] R. Fielding and R. Taylor. Principled design of the modern Web architecture. *TOIT*, 2(2):115–150, 2002.

[6] T. Luckenbach, P. Gober, S. Arbanowski, A. Kotsopoulos, and K. Kim. TinyREST: A Protocol for Integrating Sensor Networks into the Internet. *REALWSN*, 2005.

[7] A. Santanche, S. Nath, J. Liu, B. Priyantha, and F. Zhao. SenseWeb: Browsing the Physical World in Real Time. *Demo Abstract, IPSN06*, 2006.