University of Cincinnati

Date: 11/5/2014

I, Andrew J. Rettig, hereby submit this original work as part of the requirements for the degree of Doctor of Philosophy in Geography.

It is entitled:
Design and implementation of affordable, self-documenting, near-real-time geospatial sensor webs for environmental monitoring using international standards

Student's name: Andrew J. Rettig

This work and its defense approved by:

Committee chair: Richard Beck, Ph.D.
Committee member: Dharma Agrawal, D.Sc.
Committee member: Ishi Buffam, Ph.D.
Committee member: Hongxing Liu, Ph.D.
Committee member: Michael Widener, Ph.D.

12757
Design and implementation of affordable, self-documenting, near-real-time geospatial sensor webs for environmental monitoring using international standards

A dissertation submitted to the Graduate School of the University of Cincinnati in partial fulfillment of the requirements for the degree of Doctor of Philosophy

in the Department of Geography of the College of Arts and Sciences

by

Andrew J. Rettig
November 5th, 2014

M.A. University of Cincinnati June 2010

Committee Chair: Richard Beck, Ph.D.
Abstract

This dissertation documents the design and implementation of a near-real-time geospatial in situ sensor network for monitoring stormwater runoff at the Green Learning Station. The project solves the need by the Environmental Protection Agency and Cincinnati Metropolitan Sewer District for an affordable and standardized network. The project also makes contributions to geospatial standards and sensor web research. This dissertation uses open innovation, including open standards, to help reduce cost and complexities of environmental sensor networking architectures. Article 1 focuses on the technical implementation of the in situ sensor network helping to fill the research gap of applied end-to-end in situ sensing. This gap is further highlighted by the inadequacies within international geospatial standards. Article 2 discusses the greatest hardware challenge within sensor webs, embedded devices. The Green Learning Station project solved this challenge, bridged the gap between sensor protocols and standard communication protocols, with Common-off-the-Shelf (COTS) routers. The modified routers enable the development of the client/server architecture for environmental sensor networking outlined in Article 3. The client software is designed for embedded devices while the web services were designed with the Representational State Transfer (REST) approach. The Green Learning Station design and implementation is unique because of the open innovation approach to geospatial in situ sensor webs by a team of engineers with expertise at every layer of the architecture. This expertise enabled the inclusion of spatial standards and spatial data throughout the architecture. This approach creates a standardized and affordable geospatial sensor network as an example for others to study and expand upon for a variety of monitoring solutions.
Acknowledgments

Thank you to my wife Liz for your support and encouragement for the past four years of my Ph.D. You have believed in me and helped me to overcome the challenges of rigorous research. Thanks to my parents for their continued interest and support for higher education. Thanks to my advisor Dr. Richard Beck for helping to direct my research and giving me opportunities that graduate students dream of. Thanks to Sumit Khanna, my co-author for all three of my papers, for increasing the depth and impact of my research. Thank you to all of my colleagues who have helped or supported my research especially Yongming Cai, Carmen MacCane, Dan Heintzelman, Quinn Wojcik, Kevin Magee, Ben Thomas, Chantal Ivenso, Brian Scholl and everyone else I am forgetting. Thank you to everyone who participated in my dissertation project at the Green Learning Station. Thank you to my committee members, Dr. Hongxing Liu, Dr. Ishi Buffam, Dr. Michael Widener and Dr. Dharma Agrawal, for your contributions to my research. Thank you to my brother Tim and his company InTrust IT for their help with server virtualization. Lastly, thank you God for the passion and persistence to pursue my degree. Thank you.
# Table of Contents

List of Figures........................................................................................................... vi

1. Introduction................................................................................................................... 1


3. Article 2 - An Open Source Software Approach to Geospatial Sensor Network Standardization for Urban Runoff.................................................................................................................. 32

4. Article 3 - Open source REST services for environmental sensor networking .......... 39

5. Conclusion .................................................................................................................... 51

6. Bibliography............................................................................................................... 56
List of Figures

Introduction

Figure 1. Basic system architecture for environmental sensor networking.
Figure 2. Environmental sensor network architecture with three layer research division and spatial data.
Figure 3. Research highlights within the sensor network.
Figure 4. Green Learning Station construction.

Article 1

Figure 1. Green Learning Station site map.
Figure 2. Top view of pavements 2 and 3 (left) and ground view of site (right).
Figure 3. Sensor network data flow diagram.
Figure 4. Seasaw or nutate diagram.
Figure 5. Modified routers with 1-Wire sensor adapters and USB storage.
Figure 6. Sensor measurement and observation information as well as spatial data.
Figure 7. Average temperatures returned from the web service.
Figure 8. Total water volumes per day returned from the web service.
Figure 9. Web application using the permeable paver web services by Yongming Cai, Green Learning Station volunteer.

Article 2

Figure 1. Permeable paver installation for monitoring at the Green Learning Station.
Figure 2. Green Learning Station in situ sensor network.
Figure 3. Open source software architecture for stormwater monitoring sensor Networks.
Figure 4. Verification of OpenWRT and 1-Wire sensor on prototype router.

Figure 5. Modified OpenWRT routers (Netis routers) at the Green Learning Station.

Figure 6. Web application using the BigSense REST services created by volunteer developer Yongming Cai.

**Article 3**

Figure 1. Basic system architecture for environmental sensor networking.

Figure 2. LtSense Client block diagram.

Figure 3. Example of LtSense SensorML v2.0 XML description file.

Figure 4. BigSense web service block diagram.

Figure 5. Various permeable pavers were monitored at the Green Learning Station green technology site.

Figure 6. WebGIS for the Green Learning Station.
Introduction

Protecting our environment requires timely and affordable information. In 1998, U.S. Vice President Al Gore understood this need while articulated the term “Digital Earth” for the visionary concept of a virtual earth to interconnect and to geo-reference the world’s digital knowledge. Interoperable and timely spatial data has great potential for helping to solve and prevent environmental issues. With current advances in technology the Digital Earth continues to come to fruition. Smart devices with sensor and location capabilities are easily available enabling the potential for geospatial sensor networking. For the data from these new technologies to be widely beneficial, spatial standards have to be developed to enable interoperability. The greatest challenge in the construction of environmental monitoring networks is the conversion of a large variety of data streams from diverse sensors, often in proprietary protocols, to international standards in a temporal and spatial context. The purpose of this dissertation is the design and implementation of an open standards-based architecture to monitor stormwater runoff and to enable innovation in the development of geospatial environmental sensor networks (Figure 1).

Figure 1 Basic system architecture for environmental sensor networking
This introduction chapter presents sensor networking research, current trends and the inclusion of spatial data. International spatial standards are introduced along with the Open Geospatial Consortium. GIScience is discussed and explained with the inclusion of geospatial sensor networking research. Open Innovation is defined and emphasized for sensor networking affordability. The Green Learning Station was used as the development test bed and represents an applied application of this architecture.

**Sensor Webs**

Sensor networking research has traditionally been from the top down with an emphasis on military implementations and large sensor platforms. The first sensor network development was done within governmental agencies focusing on large sensor platform integration. One of the first sensor networks was a sonar submarine surveillance network in the Pacific after World War II. This network was followed by military networks for monitoring airplanes and eventually troop movements. These networks relied on trucks or sensor networking stations for monitoring and information relays.

The idea of ‘sensor webs’ was not introduced until the late 1990s. The all encompassing term of ‘sensor web’ was introduced by Kevin Delin of the National Aeronautics and Space Administration (Delin and Jackson 1999). Delin introduced the term as a result of creating a macro intelligence within his sensor network. The term sensor web today is used to not only refer to a macro intelligence within a local sensor network but the connections of sensors to the World Wide Web. However, creating this connection is complex and expensive resulting in the need for standardization. Standardization also began in the 1990s but was not officially
undertaken until 2001 by the Open Geospatial Consortium (OGC). The OGC is an international consortium of nearly 500 entities to promote and establish spatial interface standards. Sensor standards are the focus of the Sensor Web Enablement (SWE) within the OGC. The OGC SWE includes both data encodings and web services for sensor interoperability. The OGC SWE was considered at all levels of architecture however, most of the standards are designed for user level integration and development.

Current trends in sensor networking are for small embedded devices creating the Internet of Things. The term Internet of Things (IoT) was first coined in the late 1990s in Cincinnati by Kevin Ashton while working for P&G. He was interested in using radio-frequency identification (RFID) to manage P&G's supply chains (Ashton 2009). The IoT is the concept of using ubiquitous sensors to connect the physical world to the Internet while eliminating human-entered data (Ashton 2009). The IoT has continued to generate interest with strong participation from both private and community based development. It is listed on Gartner's Hype Cycle in 2014 as the most hyped technology with an expected technical plateau in 5 to 10 years (Press 2014). Unlike traditional sensor networking research, the IoT is based on small embedded devices. The IoT's de facto standards based on the of Representation State Transfer (REST) design philosophy are ideal for in situ sensor networking with simple interfaces, small foot prints and Linked Data (Bizer et al. 2009). These trends within the IoT were considered and incorporated throughout the environmental sensor networking architecture.

Understanding the sensor architecture is important for both design and implementation. Sensor network research separates the architecture into three general layers;
sensor protocols, communication and the server or presentation layer. These research areas are related to the environmental sensor networking architecture as seen in Figure 2.

All three of these layers present significant technical and theoretical challenges when creating a working sensor network. To accomplish the task, in spite of these challenges, engineers focus on the attribute data. The attribute data is emphasized at each layer within the network as the network is created from the bottom sensor layer onward. This dissertation considers the importance of attribute data but also emphasizes spatial data. Spatial data is considered essential to each layer within the network. Spatial data is often a focus at the
presentation layer for the visual focus and search capabilities but this research considers spatial data throughout the network architecture and the resulting methods and techniques.

**GIScience**

Geography in the 21st century continues to expand within GIScience, the research, methods and techniques surrounding space and place technologies. GIScience has been clearly and succinctly defined by David Mark and the University Consortium of Geographic Information Science: “The development and use of theories, methods, technology, and data for understanding geographic processes, relationships, and patterns.” Michael Goodchild describes the three domains of GIScience as human, computer and society (Goodchild 2010). Spatial data infrastructure and sensor standards are primarily represented within the computer and society domains. Developments within this science often involve spatial data while having an interdisciplinary approach. Spatial data is information that identifies geographic location of features and boundaries in space while also including additional attributes such as speed and elevation. The inclusion of spatial data is an integral part of the sensor network from the lower sensing level to the communication level and server level. Alternate standards, storage, description, registration and visualization of the data must now be considered and examined within the network.

**Research**

The research presented here is a blend of three approaches; traditional sensor networking research, GIScience and the Internet of Things. The blending of these approaches
fills the research gap of affordable and standardized connection of in situ sensors to the Internet. Beginning from the sensor protocol layer, the largest engineer divide exists between the sensor protocols and standardized communication protocols. Sensor engineers and scientists are consistently challenged to understand each other’s goals and basic protocols. This divide creates a challenge for getting sensor data into an easily readable format for analysis by scientists. As these differences are flushed out and the parties involved come to an understanding innovation is accomplished. One significant area of innovation involves working with sensor identification. Both sensor engineers and scientists are very interested in uniquely identifying sensors for numerous reasons. The sensor engineer often uses the unique ID to verify the data within their own protocol network. Alternately, scientists attempt to create unique IDs within sensor networks for sensor registration, description and metadata. Combining the needs of both the engineers and scientists can help to bridge the protocol layer to the standardized communication layer. The Green Learning Station project uses the sensor protocol ID for identification and location.

The communication layer provided the most potential for GIScience research within sensor networks. The reason for this is twofold; first traditionally web services and the presentation layer receives the most research and second with the recent development of small embedded devices, as a result of the smart phone markets, there is a variety of opportunities to implement the new technologies. However, considering that there is a lag time for small device development specifically marketed towards in situ sensor networking working within the communication layer is the most challenging. Many of the small devices require waiting periods and are not available for bulk order. It is not uncommon in sensor
networking for scientists to convert devices from an alternative purpose to solve a sensor network infrastructure gap. These devices are then used to solve the challenge of bridging sensor protocols while also storing and transporting the data. Therefore if the scientists are able to overcome the lack of specific embedded devices for sensor networking then the project has great potential. This project used open source embedded software at the communication layer to create an open small device. This development of the small device encouraged standardized spatially-enabled sensor protocols at the embedded device for automated description, registration and transfer in a spatial context.

The server and presentation layer was the final and most standardized layer of the architecture. Unfortunately, all of the current OGC SWE standards are based on arbitrary web services with Simple Access Protocol (SOAP). SOAP services are more complex with a single entry to point to pass files for instructing the web service. SOAP services are rarely used in the IoT because of their large footprint and single access point design. None of the OGC SWE web services were applicable for enabling an affordable sensor network design. Rather, the Representational State Transfer (REST) design philosophy from the Internet of Things was used for the development of web services (Figure 3).
The server software was developed to receive standardized data transmissions from embedded devices as well as to parse and store the data in a database. The server also publishes the data for consumption by a WebGIS. The REST design philosophy enables the network for linked data while also allowing for incorporation of a spatial data focus. The data can be searched by location as well as by other attributes. Treating the attribute and spatial data with the same emphasis creates the ability to search by thing and/or by location and/or time. These developments emphasize the equality of both the attributes and the spatial-temporal aspects of the data.
**Open Innovation**

There are challenges to blending sensor standards, the IoT and spatial data while working to innovate within the three layers of sensor architecture. Most notable, the hardware and software needed to assemble the network is not easily available. Also the ability to have expertise on every layer of the sensor architecture is very rare. These challenges continue to cause a high entry barrier for sensor networking and the incorporation of spatial methods, techniques and theories, hence the need for standardization. To help overcome the high entry barrier for future research and enable community based development, open innovation (open source software in combination with open standards) was used for this dissertation project. Open innovation by scientists, engineers and entities is the collaborative process of creating value for the project while looking outside of their own resources. Open Innovation is not only important for business but academia as well. Academia is conditioned to be an out flowing idea institution. Open Innovation also causes entities to rethink intellectual property as it pertains to research and development. In particular open source software in combination with open standards can help to lower the entry barrier with shared and community based development. The use of open innovation assists with continued research and development of standardized and spatially-enabled sensor network architectures.

**Green Learning Station Project**

Sensor networking research requires applied sensor design and implementation to test them under real world conditions for technical feasibility, affordability and applicability. This project used the Green Learning Station at the Cincinnati Civic Garden Center as the test bed.
for standardized sensor network development. The Green Learning Center is a green technology site for testing alternative methods for mitigating storm water runoff (Figure 4).

Figure 4 Green Learning Station construction

The site was part of the collaboration between the Environmental Protection Agency and Cincinnati Metropolitan Sewer District to reduce stormwater runoff and combined sewer overflows in Cincinnati. I was project manager for the sensor network infrastructure, including the sensor installation, hardware and software. The project began in the fall of 2010 and ended in the spring of 2012. I assembled the team of 12 volunteers to implement the network. Engineers, scientists, and employees from the University of Cincinnati formed the UC@GS team to achieve the goal of monitoring stormwater runoff from the permeable pavers. The team also expanded the original goal of creating a monitoring network to include contributing to geospatial sensor networking research. With expertise at every level of the sensor architecture, the UC@GS team delivered an affordable near-real-time geospatial sensor
network. The result of this work contributes to the three dissertation articles with implementation of methods and techniques.

**Summary**

Applied sensor networking research was essential for understanding the trends in the research and relevant gaps for proposing and applying geospatial sensor networking research for this dissertation. The top down historical approach helped to focus and guide the research to incorporate current trends. The current trends in sensor networking, including the Internet of Things, provided research that guided development specific for embedded devices. This research enabled our architecture for spatially-enabled environmental sensing. GIScience methods and techniques continued to provide innovations that would dramatically alter a basic attribute based network approach. The simple change of focus from attributes to a dual focus with spatial data dramatically changes the methods, techniques and theories that are applied to the development of sensor architecture as described in the following papers. These methods proved useful not only at the presentation layer but at each level of the network. The result was advances throughout the network architecture with relevant innovations and significant contributions to standardized affordable near-real-time geospatial sensor networks for environmental monitoring.

Open Innovation (open source software combined with open standards and common-off-the-shelf (COTS) hardware) was also very important for this dissertation. Sensor networks require a multidisciplinary approach and the collaboration during the project proved useful for both the design and implementation as well as the accompanied research and publication.
The project provided a solid case study and an environment for testing and developing a new standardized sensor network architecture with a real-world implementation. The project also helped to solve a need expressed by the EPA and the local sewer district of capturing near-real-time stormwater runoff data for analysis and the decision making process on green infrastructure. All three of the chapters presented in this dissertation are published articles on geospatial sensor webs. Chapter 2 was the first publication focusing on a comprehensive discussion of the project and resulting implementation. Chapter 3 highlights the development for bridging the sensor protocol layer to the communication layer. Chapter 4 is an introduction of open source REST services designed specifically for geospatial environmental sensor networking. These chapters are followed by a Conclusions section that briefly summarizes the results of all three papers.
Monitoring permeable paver runoff with an open-innovation geospatial sensor network

Andrew J. Rettig\textsuperscript{a}, Sumit Khanna\textsuperscript{b}, Richard A. Beck\textsuperscript{a}, Quinn Wojcik\textsuperscript{a} & Carmen A. McCane\textsuperscript{a}

\textsuperscript{a}Department of Geography, University of Cincinnati, Cincinnati, OH, USA
\textsuperscript{b}College of Engineering and Computer Science, University of Tennessee at Chattanooga, Chattanooga, TN, USA

Accepted author version posted online: 16 Sep 2014. Published online: 17 Oct 2014.

To cite this article: Andrew J. Rettig, Sumit Khanna, Richard A. Beck, Quinn Wojcik & Carmen A. McCane (2014): Monitoring permeable paver runoff with an open-innovation geospatial sensor network, International Journal of Digital Earth, DOI: 10.1080/17538947.2014.965880

To link to this article: http://dx.doi.org/10.1080/17538947.2014.965880

PLEASE SCROLL DOWN FOR ARTICLE

Taylor & Francis makes every effort to ensure the accuracy of all the information (the "Content") contained in the publications on our platform. However, Taylor & Francis, our agents, and our licensors make no representations or warranties whatsoever as to the accuracy, completeness, or suitability for any purpose of the Content. Any opinions and views expressed in this publication are the opinions and views of the authors, and are not the views of or endorsed by Taylor & Francis. The accuracy of the Content should not be relied upon and should be independently verified with primary sources of information. Taylor and Francis shall not be liable for any losses, actions, claims, proceedings, demands, costs, expenses, damages, and other liabilities whatsoever or howsoever caused arising directly or indirectly in connection with, in relation to or arising out of the use of the Content.

This article may be used for research, teaching, and private study purposes. Any substantial or systematic reproduction, redistribution, reselling, loan, sub-licensing, systematic supply, or distribution in any form to anyone is expressly forbidden. Terms &
RESEARCH ARTICLE

Monitoring permeable paver runoff with an open-innovation geospatial sensor network

Andrew J. Retting*, Sumit Khanna, Richard A. Beck, Quinn Wojcik and Carmen A. McCane

aDepartment of Geography, University of Cincinnati, Cincinnati, OH, USA; bCollege of Engineering and Computer Science, University of Tennessee at Chattanooga, Chattanooga, TN, USA

(Received 5 February 2014; final version received 10 September 2014)

Sensor networks are an essential tool for environmental scientists. As scientists and engineers are beginning to utilize these new methods and devices in their fieldwork, they need to be actively involved in the future of sensor-networking development. Continued sensor network innovation is important for improved standardization, affordability, and interoperability. This article uses a storm water case study to outline an end-to-end open-innovation sensor network. Open innovation by scientists, engineers, and entities is the collaborative process of creating value for this project in permeable paver runoff data and advances within sensor networking. This article focuses on the technical implementation of the near–real-time location and temporally aware sensor network. Data are streamed in near–real-time with subliter precision to the cloud using common off-the-shelf routers. The sensors use Maxim’s 1-wire™ protocol, and the unique digital serial numbers confirm the data. The data retrieved compare residence times within the permeable paver catchment basins and the control basin. Sensor network advances are made by bridging the gap between sensor protocols and communication systems. These advances enable the development of open-source representational state transfer web services. Our successful implementation serves as an example for others to study and expand upon for a variety of monitoring solutions.

Keywords: sensor networks; Internet of things; open geospatial consortium; open innovation; permeable pavers

1. Introduction

1.1. Problem: in situ monitoring of storm water runoff

The urban environment in the twenty-first century continues the science it has sought since the days of the Indus Valley Civilization to cleanse the city of death, disease, and inefficiency. Engineering has created solutions to sanitation, transportation, and drainage. Cities have become delivery and disposal systems supporting a network designed to deliver service to support commerce and culture. However, examining this environment with a ‘City as Infrastructure’ approach makes a new undeniable challenge apparent. Antiquated technology is prevalent throughout the infrastructure of today’s cities. Here in Cincinnati the sewers that carry the urban runoff were installed in 1815, nearly 200 years ago.

*Corresponding author. Email: andrewrettig@gmail.com

© 2014 Taylor & Francis
ago and our water intake system was installed 150 years ago. These are just two of the many aging systems creating new problems. Are these aging infrastructures going to be able to support growing urban populations and handle potential climate changes and stresses? These difficult questions require better information relevant to these infrastructures. This paper addresses the problem of designing and building low-cost (standardized) geospatial sensor networks for monitoring storm water runoff.

Sensor networking is very complex with many moving parts; however, there have been significant technical advances. Innovations within each part of the sensor network infrastructure from the sensors to the sensor nodes and server technologies have helped to improve implementations. In particular, improvements within embedded devices have been driven by the smartphone market and the popularity surrounding the Internet of Things (IoT). The IoT may be defined as having computers sense information without the aid of human intervention expanding the Internet to include objects that are both sensing and interacting with the environment (Gubbi et al. 2013). Gartners IT Hype Cycle has identified the IoT as one of the emerging technologies in IT. In 2011, this interest was further justified as the number of interconnected devices on the planet passed the total number of people (Gubbi et al. 2013). The IoT, as characterized by Atzori, Iera, and Morabito (2010), is composed of three paradigms; sensing, middleware, and the semantic. With sensor-networking research comprising these paradigms, it is essential that the IoT trends are considered and incorporated. Within these paradigms exist particular IoT elements relative to our work, including modular middleware, secure data aggregation, uniquely identifying things, and visualization (Gubbi et al. 2013). Along with the sensors and middleware, the web services are an area needing additional research (Liang, Croitoru, and Tao 2005). By innovating within web services, we are contributing toward the integration of sensor data with Linked Data. Linked data have become the most promising vision for the Future Internet and has been widely adopted by academia and industry (Janowicz et al. 2013). However, currently sensor networking and spatial data infrastructure can only act to a limited degree with the semantic web (Janowicz et al. 2013).

Along with these IoT trends is the importance of interoperability within end-to-end networks. Even with the many moving parts in sensor network implementations, there is overwhelming consensus that in situ sensing needs to provide the ability to share information across platforms through a unified framework (Gubbi et al. 2013; Sempere-Pay’a and Santonja-Climent 2012; Sang et al. 2006). The lack of consistent standards both raises the cost for implementation and creates isolated systems and development. The result is expensive custom in situ implementations even with the dropping cost of smart devices. There exists a standards gap in in situ sensor network standardization from the sensor to the web services (Jirka, Brting, and Stasch 2009). This gap is further exposed in the previous works section as sensor-networking development has traditionally been influenced by top–down large sensor platform development. The result is a challenge for affordable, easy-implementable in situ sensor networks.

Our work is striving to contribute to a modular standardized framework to create low-cost plug-and-play sensors forming in situ sensor networks while incorporating IoT elements. Data are to be streamed in near–real-time with sublitter precision to the cloud using common off-the-shelf routers (COTS) to monitor permeable pavers. We consider our network to operate in near–real-time; however, the time delay is only a couple minutes from observation to publication. The sensors use Maxim’s 1-wire™ protocol, and the unique digital serial numbers confirm the data authenticity and location. The data
retrieved will compare residence times within the catchment basins of the various pavers and the control. Residence time is the average amount of time for a particle of water to travel through the catchment basin. The project also enables open innovation for sensor network advancement. It is vital that scientists as well as engineers play a vital role in guiding the sensor-networking revolution (Hart and Martinez 2006). Software and hardware testing will be incorporated into the project for increased scalability, affordability, and interoperability. The article begins by examining the previous work and study area followed by a methodology section focusing on the technical implementation of a near–real-time location and temporally aware sensor network. Each part of the sensor network architecture will be examined including the sensors, sensor relay, and servers. The article concludes with statistical and qualitative results discussing both the data and sensor advances of the open-innovation network.

1.2. Previous work

The Pacific Ocean is home to some of the most ground-breaking research regarding whales, seismic activity, and sensor networks. This research is a result of one of the first sensor networks ever implemented. The Sound Surveillance System (SOSUS) was a network of underwater acoustic sensors and shore-based listening posts (Amato 1993). SOSUS was implemented in the 1950s as a result of sonar innovation during World War I and World War II for defense and military operations. SOSUS was followed by air defense radars in the 1960s in which multiple sensors were used to detect aircraft and other military assets. Modern research on sensor networks began in the 1980s with the Distributed Sensor Networks program at Defense Advanced Research Projects Agency (DARPA) (Chong and Kumar 2003). The director of the Information Processing Techniques Office at DARPA, R. Kahn wanted to know whether the Arpanet approach for communication could be extended to sensor networks (Chong and Kumar 2003). Kahn’s aspirations were the same goals we are still striving for today – spatially distributed low-cost sensors working together. The sensors implemented at that time were acoustic arrays including a mobile node with a truck equipped with generators and computer hardware. The technology of small yet affordable sensors was not adequately developed (Chong and Kumar 2003). The early research, with technology limitations and a military focus, contributed to a top–down large sensor platform standardization approach.

The comprehensive concept for sensor networks was not introduced until 1997 as ‘Sensor Web’ by Kevin Delin. Delin later defined the comprehensive concept as ‘allows for the spatial-temporal understanding of the environment through coordinated efforts between multiple numbers and types of sensing platforms, including both orbital and terrestrial and both fixed and mobile’ (Delin 2002; Delin et al. 2005). In 1998 the scientific and technology community was inspired by Vice President Al Gore who coined the term ‘Digital Earth.’ Digital Earth was the vision of a multiresolution, three-dimensional representation of the earth with vast quantities of georeferenced data (Gore 1998). While both a comprehensive concept and the vision of a Digital Earth provide excitement and inspiration within sensing, the difference between the vastly different end-to-end architectures within sensing must not be forgotten. In 2001 the Sensor Web standards communication was taken on by the Open Geospatial Consortium (OGC), and today the OGC standards are continually being developed within its Sensor Web Enablement (SWE). However, SWE standards are based on Simple Access Protocols
(SOAP) suited for more complex processes while the developing IoT is using Representation state transfer (REST) web services demonstrating a significant gap in research. It is expected that in 2015 the first standardized RESTful approach to web services for in situ sensor networking will be published. SWE standard research and the IoT are slowly beginning to demonstrate cross-development (Bröring, Remke, and Lasnia 2012; Guinard et al. 2009; Janowicz et al. 2010, 2013; Mazzetti, Nativi, and Caron 2009). Also, an emerging concern in the Sensor Web is the ability to have plug-and-play sensors (Broering et al. 2010; Jirka, Brring, and Stasch 2009). The challenge is to create standards throughout the network from the lower level at the semiconductor protocol level, into the communication intermediary services and finally into the server layer (Bröring et al. 2011). The OGC SWE web services are based at the server layer creating significant distinction among the standards within the network and resulting in challenges for plug and play (Broering et al. 2010). More work needs to include the research of integrating sensor into the SWE services architecture because the details of end-to-end networks have not been specified (Jirka, Brring, and Stasch 2009).

Storm water sensor network research and case studies primarily began in the 1990s. The implementations would focus on three goals: (1) monitoring of storm water, (2) increase efficiency of current infrastructure with real-time controls, or (3) automation of water treatment plants (Wiese et al. 2003; Pfister and Cassar 1999; Kopecny et al. 1999). More recent research has not only focused on alternative storm water solutions such as green technologies but improvements within sensor networking (Sempere-Paya’a and Santonja-Climent 2012; Yu, Behera, and Rochac 2011; Ruggaber, Talley, and Montestruque 2007; Bliss, Neufeld, and Ries 2009). Sensor networks are too complex and expensive, resulting in limited coverage (Sempere-Paya’a and Santonja-Climent 2012). The solution to affordability and complexities is sensor-networking standardization through open innovation. The concept of open innovation (Chesbrough 2003) is well positioned to serve as a mediating, exploratory, and validating approach for sensor networking (Schaffers et al. 2011). Living lab-driven innovative ecosystems help to solve the challenge of sensor networking, urban development policies, and open user-driven innovation (Almirall and Wareham 2008).

2. Experiment

2.1. Study area: Green Learning Station at the Civic Garden Center

Cincinnati is plagued with storm water runoff issues because of the hilly topography, clay soils, and aging sewer infrastructure. The Green Learning Station is part of the Metropolitan Sewer District of Greater Cincinnati (MSDGC) green demonstration projects examining the storm water runoff impacts of green technologies including pervious pavements, bioswales, and rainwater harvesting. The site is located within the service region of the MSDGC within Hamilton County and the city of Cincinnati (Figure 1). Phase 1 of the sensor network implementation was collecting data on the flow and volume of storm water runoff relative to precipitation on the permeable pavers. Permeable pavement allows storm water to infiltrate into either a storage basin or soil (EPA 1999). It has been shown that the use of permeable pavement decreases surface runoff and lowers peak discharge (Pratt, Mantle, and Schofield 1995; Booth, Leavitt, and Peterson 1996; Hunt, Stephens, and Mayes 2002) decreasing the number of combined sewer overflows (CSOs). The Green Learning Station project at the Civic Garden Center
is a response to the municipal system challenges with storm water. More precisely, the Green Learning Station is a test site for green technologies to help reduce CSOs.

2.2. Environment

The purpose of the Green Learning Station research station was to measure residence times of various pervious pavement compared to the control. Six pavement types were used in the construction of this research area and parking lot (Figure 2). Five of the pavements were facing the front of the research facility while one was on the side. Each

Figure 1. Green Learning Station site map.

Figure 2. Top view of pavements 2 and 3 (left) and ground view of site (right).
pavement zone was 7.96 × 13.9 m (26.1 × 45.5 ft) with a 1.5-m (5 ft) concrete barrier separating the zones. The front five zones together were 41.99 × 13.9 m (137.75 × 45.6 ft). Each zone was approximately 113.4 m² (1221 ft²) in area. The whole site was approximately 703.2 m (7570 ft²) in area. Each zone was 0.91-m (3 ft) deep with the bottom consisting of a plastic liner on top of a concrete bed. Above the concrete was a 0.3-m (1 ft) cobble-sized rock, 0.3 m (1 ft) of gravel, and 0.3 m (1 ft) of pea gravel in which the surface material was laid. Additionally, there was a baffled, T-shaped, PVC pipe running down the center line which emptied into the catchment. The slope of each zone was between 10 and 15 degrees. Zone 3 was the control zone as it was filled with nonpervious concrete. The control runoff is routed into the manhole via the holes in the manhole cover. Zones 1 and 2 were filled with pervious asphalt and pervious concrete. Zones 4, 5, and 6 were filled with different types of pervious cobblestone. Per 2.54 cm (1 in) of precipitation, each zone should have yielded approximately 2865.6 liters (757 gallons) of water. The whole site should have yielded 17,764.9 liters (4693 gallons) per 2.54 cm of precipitation assuming no evaporation or transpiration.

2.3. Methodology

Our methodology is summarized in a data flow diagram (Figure 3). Sensor data were collected using single-board embedded systems running the OpenWRT Linux operating system, referred to herein as the sensor relay. OpenWRT was chosen because it is a small embedded Linux distribution designed specifically to work with off-the-shelf home routers. It can run on embedded Linux systems that only have 4-MB to 8-MB of storage capacity. Using off-the-shelf routers as single-board computers allowed us to reduce costs, as home routers are priced for consumer markets and, at the time of this study, were considerably cheaper than equivalent embedded boards such as the Beagle Bone series. OpenWRT had a very large and active support community, and independent developers

![Figure 3. Sensor network data flow diagram.](image_url)
continue to support newer products. Using OpenWRT would allow us to move to

different routers and embedded boards without altering the program code. Another
reasonable alternative that was considered was DD-WRT, which was based on the same
project as OpenWRT, but DD-WRT was excluded as its development moved in a more
commercial approach.

LtSense is a custom application written in the Python programming language. Python
was chosen as our development platform because it is a modern language with a strong
development community, a robust standard library, and it has an embedded interpreter
that fit on the limited space of our sensor relays. Other alternatives, such as Ruby, did not
have OpenWRT packages at the time or were too large to run on embed hardware.
Normally, space constraints on embedded systems result in developers using C/C++, but
our sensor relays had the capacity so that we could make a space trade-off for using a
more modern language like Python.

1-Wire sensors were used for data collection. These sensors were affordable and could
be attached to our sensor relays via Universal Serial Bus (USB). 1-Wire sensors use only
a single cable for multiplexing both power and data readings. They also had existing
driver support in Linux by means of an open-source project known as One Wire File
System (OWFS) that had an active development community. Being able to leverage these
existing drivers allowed us to quickly integrate them into our Python application.

LtSense queried 1-wire sensors wired in series at a preset sample rate of once per
minute. Two sensors were wired to each test site: one temperature and one counter for a
tipping bucket that served as a flow meter. The sensor network was wired, rather than
wireless, in order to simplify our infrastructure. Outdoor Ethernet cable was placed in a
conduit running underneath the permeable pavement connecting the sensors to the relays.
One-wire uses the single wire for both the data and power requirements of the network. In
addition, there were two rain gauges above the primary building at the site wired into our
sensor network to monitor rainfall levels.

Our implementation streams XML from the relay to the web services on the server.
No SWE standards were used in establishing this XML, but as research continues,
distinctions will be made between the higher and lower level protocols to create standards
throughout the network for plug-and-play applicability. Lower level and intermediary
protocols must be approached differently than standardized higher level protocols. The
lowest level is created and implemented to match the design and limitations of the
hardware. For our network the 1-wire protocol was established by Maxim, a semicon-
ductor company. The intermediary protocols were defined within the relay. The
intermediary protocols were chosen for minimalism on the following assumptions:
reduction of data redundancy, reduction of data size for streaming, reduction of local
storage needs, and reduction of local power. Once the data reached our server protocols of
the database, then SWE services could be implemented.

2.3.1. 1-Wire sensors
The primary sensor at each permeable paver test area is for water volume and flow. As
water flows through the meter chamber, a disk seasaws or nutates about its axis
(Figure 4). Each seasaw of the disk allows a known volume of water to pass. A magnet in
the disk trips the reed switch connected to a counting chip sensor measuring the number
of seasaws. The water meter measures the amount of water volume and flow in the
system for each test paver. For pavements 1, 2, 3, and 4, each tip of the rocker indicated a
It is important to note that the seasaw is not measuring runoff itself, but the volume of water that is pumped out of each manhole or pit. The water level must reach a height approximately 0.9 m (1 ft) for the pump to activate and push the water through the sensor. If water volume increases are detected, a minimum volume rise of 227 liters (60 gallons), with a variance of ±76 liters (20 gallons), is recorded with each pumping event. Because of this method of measurement, to gain meaningful values from the data, the volume of water extracted from each pit must be aggregated over a period of time in order to gauge the effect of a rain event.

The rain gauges had tipping buckets manufactured as a RAINEW rain gauge by RainWise with a 1-wire DS2423 counter chip from Dallas Semiconductor, calibrated to measure 0.254 mm per tip of the rocker (1/100th of an inch). At the core of the rain gauge is a pair of buckets connected to a central axis like a seesaw. A funnel collects rain and empties it into one of the buckets. When the bucket fills, it tips over and the other bucket takes its place. When the buckets tip, a reed switch connected to a 1-wire counter chip is briefly closed by a magnet on the bucket. Since the bucket’s volume is known, the count can be used to measure the amount of rainfall.

2.3.2. Relay queue
Each permeable paver test basin was equipped with its own relay (modified router) that connected the sensors to the network (Figure 5). In the event that the web service was unavailable or there was a network failure, the XML data would queue on the relay until the service was restored. There were two data-queuing options at the relay level. The first was an in-memory queue, and the second involved reading and writing the queue from a SQLite database located on a USB memory drive connected to the embedded system on the relay. The advantage of using the database over the memory queue was that the queue would not persist after power outages. The in-memory queue would be reset, not only if power was cycled to the sensor relay but also if the LtSense application was restarted. Therefore, the database option was preferred and adopted. Calculations for both flow rate and volume were processed on the sensor relay (modified router) based on the count from the tipping sensors. 1-Wire only allowed us to identify a sensor by type (e.g. Temperature or Counting). With this limitation, the sensor relay had to be configured for a specific volume per tip for all counting sensors attached to it. Two counting sensors that counted fluid of different volumes could not be attached to the same sensor relay.
2.3.3. Relay security

Initially, both the sensor relays and the web service were within the same virtual private network (VPN). The web service was hosted at the Geography Department at the University of Cincinnati while the individual sensor relays were housed at the Green Learning Station. The network for the sensor relays at the Green Learning Station was separate from the rest of the center’s primary network, so a VPN tunnel based on OpenVPN was established between the two sites. This initially resulted in a passive security model that depended on sensor relays being on the same network (either real or virtual). The current security model allows for the sensor relay to be on a public network, removing the previous limitation. This is accomplished using RSA (cryptosystem) keys and signature verification. The original XML data, before being transmitted, is signed digitally using a preshared RSA private key. The server can verify the authenticity of the delivered XML payload by using the RSA public key to validate the signature.

2.3.4. Web service

The data were transported in our own XML format to BigSense. BigSense is a web service written in the Scala programming language that collects and aggregates the data.
Scala was chosen as our web service language because it ran on the Java Virtual Machine (JVM), a mature environment used for many large-scale web applications. It added additional functionality and provided a more easily readable and maintainable code than Java, while still maintaining full byte-code compatibility with the JVM. With Scala, we could leverage existing Java architectures and libraries while gaining language enhancements that could reduce development time.

Scala had built-in language support for XML. We decided to use our own custom XML format for the data transfer since the OGC XML were primarily designed for the user-based server level. The web service was also written in such a way that we could add new XML formats in the future should we choose to support them. Once transferred from the relay to BigSense, the XML file is parsed into a Structured Query Language (SQL) database using customized web services. The data are also published by BigSense, and the process is explained further in subsection 2.3.7.

2.3.5. Spatial data
A Trimble differential Global Positioning System (GPS) unit was used to gather latitude and longitude for each sensor location. Our GPS base station was the Cincinnati/Northern Kentucky International Airport (CVG), and the coordinates were accurate to within 13 cm in the horizontal plane. World Geodetic System of 1984 was used as the standard coordinate frame for the location information. The 1-wire unique ID for each sensor was used to create a column within the spatial data to be used as a primary key. Once the 1-wire sensor was installed, the unique ID was used to register the sensor within our network. The unique ID is a 64-bit ROM ID that allows for multiple sensors to be on a single bus. Our temperature and counter sensors within each test paver are on the same bus or relay (modified router). Unique ids associated with the sensors, enable authentication and integration with sensor metadata. If a sensor is replaced, the new unique ID is recognized on the 1-wire bus and can be used for automatic registration. With the possible addition of a GPS-enabled relay, the sensor replacement could be automatically recognized and updated within the system. The ease of replacement is a result of the embedded unique 1-wire ID. Figure 6 shows a list of sensors by their unique identifier, the relays they are attached to, as well as the type of measurement units, type of sensor, and the latitude and longitude coordinates at which they were installed.

2.3.6. Data interpretation
The BigSense web service stored three types of data from the sensor relays: temperature, the current volume, and the current flow rate of water from a counting sensor in the manhole below each pavement test surface. Each counting sensor had a watch battery in it, allowing the count to be retained even after a power outage, regardless of what type of queue was used on the sensor relay. The latest count was always saved to the USB flash memory on the sensor relay, and all calculations for flow rates and volumes were done on the sensor relay itself. Each sensor relay was configured so that counting sensors would be given a bucket size and a unit of measure. Each counter would report to the web service as two different sensors, a volume sensor and a flow rate sensor. Volume was simply calculated as the bucket size multiplied by the current count value. Flow rate was calculated as taking the difference of two measurements at the sensor relays default sample rate, finding the unit rate (1 second), and reporting the data as the units over seconds.
2.3.7 Aggregation

Water volumes and flow rates were dependent on when the pump for the collection pits turned on. After or during rain events, the pits would have to fill up before the pumps would activate, thereby causing the sensors to detect flow. Therefore, there were intervals with zero flow during a mild rain event. To deal with this type of system, the web service allowed for queries to average sensor results over a given amount of time. For example, the query in Figure 7 returns the average temperatures between the date range of 1 May 2012 to 5 May 2012 for interval ranges of 1440 minutes (1 day) in metric units for the US Eastern time zone in a comma-separated value (csv) format. An example with water measurements can be seen in Figure 8. The query returns the total volume of water,

<table>
<thead>
<tr>
<th>Sensor ID</th>
<th>Relay ID</th>
<th>Units</th>
<th>Sensor Type</th>
<th>Latitude</th>
<th>Longitude</th>
</tr>
</thead>
<tbody>
<tr>
<td>BB0008017D091010</td>
<td>1-PermAsphalt</td>
<td>C Temperature</td>
<td>39.13025</td>
<td>~84.4984</td>
<td></td>
</tr>
<tr>
<td>BC0000000F968F1D</td>
<td>1-PermAsphalt</td>
<td>I Counter</td>
<td>39.13025</td>
<td>~84.4984</td>
<td></td>
</tr>
<tr>
<td>420008017D027D10</td>
<td>2-PermConcrete</td>
<td>C Temperature</td>
<td>39.13018</td>
<td>~84.4984</td>
<td></td>
</tr>
<tr>
<td>280000000F861C1D</td>
<td>2-PermConcrete</td>
<td>I Counter</td>
<td>39.13018</td>
<td>~84.4984</td>
<td></td>
</tr>
<tr>
<td>830008017CF0DC10</td>
<td>3-Concrete</td>
<td>C Temperature</td>
<td>39.1301</td>
<td>~84.4985</td>
<td></td>
</tr>
<tr>
<td>D30000000F4E731D</td>
<td>3-Concrete</td>
<td>I Counter</td>
<td>39.1301</td>
<td>~84.4985</td>
<td></td>
</tr>
<tr>
<td>850008017D08C110</td>
<td>4-Paver1</td>
<td>C Temperature</td>
<td>39.13003</td>
<td>~84.4985</td>
<td></td>
</tr>
<tr>
<td>320000000F6D091D</td>
<td>4-Paver1</td>
<td>I Counter</td>
<td>39.13003</td>
<td>~84.4985</td>
<td></td>
</tr>
<tr>
<td>0B0008017CFBB310</td>
<td>5-Paver2</td>
<td>C Temperature</td>
<td>39.12996</td>
<td>~84.4985</td>
<td></td>
</tr>
<tr>
<td>070000000F681B1D</td>
<td>5-Paver2</td>
<td>I Counter</td>
<td>39.12996</td>
<td>~84.4985</td>
<td></td>
</tr>
<tr>
<td>FE0008017D045410</td>
<td>6-Paver3</td>
<td>C Temperature</td>
<td>39.12998</td>
<td>~84.4988</td>
<td></td>
</tr>
<tr>
<td>7C0000000F9F9ED1D</td>
<td>6-Paver3</td>
<td>I Counter</td>
<td>39.12998</td>
<td>~84.4988</td>
<td></td>
</tr>
<tr>
<td>3F0008017D13E110</td>
<td>Bioswale</td>
<td>C Temperature</td>
<td>39.13029</td>
<td>~84.4984</td>
<td></td>
</tr>
<tr>
<td>670000000F963B1D</td>
<td>Bioswale</td>
<td>I Counter</td>
<td>39.13029</td>
<td>~84.4984</td>
<td></td>
</tr>
<tr>
<td>DD0008017CF17610</td>
<td>RainGauge</td>
<td>C Temperature</td>
<td>39.13017</td>
<td>~84.4987</td>
<td></td>
</tr>
<tr>
<td>880008017D03CD10</td>
<td>RainGauge</td>
<td>C Temperature</td>
<td>39.1300</td>
<td>~84.4988</td>
<td></td>
</tr>
<tr>
<td>310000000F87541D</td>
<td>RainGauge</td>
<td>mm Counter</td>
<td>39.12998</td>
<td>~84.4988</td>
<td></td>
</tr>
<tr>
<td>D50000000F681B1D</td>
<td>RainGauge</td>
<td>mm Counter</td>
<td>39.13017</td>
<td>~84.4987</td>
<td></td>
</tr>
</tbody>
</table>

Figure 6. Sensor measurement and observation information as well as spatial data.

Figure 7. Average temperatures returned from the web service.
measured with the counting sensors, for interval ranges of 1440 minutes (1 day) in metric units for the US Eastern time zone in a csv format.

Three functions for aggregation were implemented. Aggregation could be done for average (mean) temperatures, average (mean) flow rates and total volume of water divided by a time interval in minutes. Queries could be performed either by date ranges or by UNIX time stamps in seconds. Several formats were supported including csv, tab delimited (txt), HTML data table (table.html), flat XML (flat.xml) and a proprietary XML format (go.xml). The results from the web service are human readable, but they are not intended to be consumed directly. The HTML data table format is the best for readability and diagnostics, but the reason for supporting terse formats is to allow the web service to be used by other front-end web applications that could use the service to provide both real-time and aggregate data to the end user in readable layouts, graphs, and charts. The basic pattern of the query allows developers to easily see what data are being selected as well as providing a unique location for a given dataset. This is known as a representative state transfer or REST web service.

3. Experimental results

Preliminary rain gauge data analysis was completed for December 2011. These were the first sensors to be statistically verified since the water data aggregated from the pavers were dependent on the rain gauge data collected at the Green Learning Station. Rainfall data were examined for December 2011 for two rain gauges, which will be referred to as ‘D5’ and ‘31.’ In order to determine if D5 and 31 were registering the same rainfall totals, an independent-samples t-test was performed because it compares the mean values (in this case rainfall totals) of two different groups (in this case rain gauges D5 and 31). It is important to note that all statistical assumptions were met (i.e. there were no violation of assumptions) before conducting any of the following analyses. Results from the independent-samples t-test revealed that there was no statistically significant difference in rainfall totals reported for D5 (mean = .218214286535714; standard deviation = .533791841618124) versus 31 (mean = .19250001714286; standard deviation = .489456426302576) in December 2011 (t-calculated = 0.188, p-value = 0.852, σ = 0.05). The magnitude of the differences in the means was very small (eta squared

Figure 8. Total water volumes per day returned from the web service.
Next, we wanted to compare the rain gauge data from D5 to CVG for December 2011. Therefore, a second independent-samples $t$-test was conducted. The results revealed that there was no statistically significant difference in the mean rainfall totals registered between D5 and CVG ($t$-calculated = $-0.103$, $p$-value = 0.918, $\sigma = 0.05$).

Early permeable paver runoff data from the water meters were unreliable due to electromagnetic interference from outside sources. Additional shielding and shorter wires between the sensors and the Ethernet connectors resolved the interference issues. With interference resolved, the counter sensors had no other maintenance issues. Once the sensor relays went through a testing period, all the relays performed consistently without issues. Reliable permeable paver runoff data for analysis were available in late winter and spring of 2012. The MSDGC and the Environmental Protection Agency (EPA) were interested in examining the residence time of each permeable paver and compare them to the control. Figure 9 confirms that the data are producing reasonable results. The volume of water traveling through each permeable paver is relative to the size of the catchment areas (5 mm of rain, left chart, produces 560 liters of water per catchment area, right chart). The right chart in Figure 9 also shows the control pavement having the steepest rising and falling limbs. MSDGC and EPA are currently analyzing the data.

4. Discussion

Our sensor network implementation helps to solve the problem of designing and building low-cost (standardized) sensor networks for monitoring storm water runoff. It should be noted that the experimental configuration included manholes for collecting storm water runoff as designed by the EPA and MSDGC. Measuring runoff was dependent upon the activation of the water pump within the manhole. The pump would activate when the water depth exceeded 1 ft and stop when the water level returned to the 1-ft level. If runoff from the storm events did not activate the pump, the runoff would not be measured. This monitoring limitation may not allow for estimation of the total rainfall amount or the residence time, which are both important variables for CSOs. Limitations were also encountered with the use of the Maxim DS2423 1-wire counter chip. Both storing and calculating the count function was intensive and resulted in a limitation of one counting chip per sensor relay. However, using an alternative sensor or including additional sensors is made relatively easy with the use of the open-source software OWFS.

Home routers were chosen as sensor relays initially due to their low costs as single-board computers. In recent years, the introduction of the Raspberry, Pi has greatly reduced the price in the single-board market. Embedded systems like the Raspberry, Pi, the Beagle Bone, and others have several advantages such as general input/output pins, built-in analog-to-digital converters, and significantly higher storage capacity. The larger capacities mean these systems are not limited to small Linux distributions such as OpenWRT, but can run full-sized distributions such as Ubuntu and Arch. In addition to single-board computers, the Arduino has made its way into the market as a very low cost programmable microcontroller. This could help facilitate creating self-calibrating sensors, simplifying the role of the sensor relay. For example, an Arduino could be attached to both a compass and a windy direction sensor and programmed to determine which angle is North. The Arduino could then report that data to a sensor relay without the relay software needing to be specifically configured for that sensor.
Figure 9. Web application using the permeable paver web services by Yongming Cai, Green Learning Station volunteer.
The OGC SWE standards were considered for all levels of the network. The XML format for the data was originally chosen for the possible inclusion of OGC standards for the data feed. With the OGC standards primarily focused on the server and user-level standards, none of the OGC standards were implemented. Communication from the relay to the server required maximum efficiency considering energy and data size restrictions resulting in custom XML formats. Current developments are focused on using JavaScript Object Notation, an easier to use alternative for the data feed. SWE SOAP services were considered for the server but were abandoned for REST web services. REST provides a smaller footprint on the relay and server while lowering the development entry point for application integration. The REST web services were easy for the development team to query the data and integrate into the web application. The only issue with the web services was the importance to incorporate a spatial component into all the services. The web application development required additional programming to create a connection between the sensor data and sensor location without a consistent spatial component. These frustrations demonstrated the need for a standard on spatial data inclusion in a RESTful approach. The OGC and Steve Liang are currently hoping to publish a standard in the fall of 2014 for spatial data inclusion in a RESTful API. Standardization of the API would have simplified our development of both the services and the applications.

The Consortium of Universities for the Advancement of Hydrologic Science, Inc’s (CUAHSI) WaterML-encoding standard also was considered at the server level as an additional service. CUAHSI recently created the most relevant XML standard, WaterML, for hydrology data publication at the server level, but this standard was not yet OGC compliant in the fall of 2011 (Ames et al. 2009) during the network installation. This standard was not implemented considering the lack of OGC compliance of WaterML 1.0. Database conversion to the CUAHSI’s schema was possible once WaterML 2.0 was released. WaterML 2.0 was approved as an OGC standard in the fall of 2012.

5. Conclusion

Our open-innovation geospatial sensor network bridged the lower level sensor protocols into the communication layer with the development of a COTS-modified router as a sensor relay. The work contributed to sensor research and the IoT with modular relays, secure data feeds, unique identification of sensors, and web service aggregation for visualization. In addition, these advancements created affordability and scalability by utilizing and creating open-source software. These developments further enabled the implementation and testing of sensor standards and the utilization of the IoT RESTful web services approach. The simple and easy data access through the web services enabled our team to easily monitor the system during the testing and implementation phase. This access to the near–real-time data also promoted continued improvements to the web services such as improved aggregation, data format inclusion, and unit conversion. Our data demonstrated that an affordable, scalable, and interoperable system was within the capabilities of a well-rounded team of engineers working with supporting sensor and hardware providers.

The open-innovation approach to geospatial sensor networking provided some significant insights and advances. Scientists and engineers gained interest and extensive experience during the implementation. During construction, the electrician and plumber were included in meetings to understand the sensor installation and advice on the infrastructure implementation. These meetings resulted in the construction manager
designing a sensor housing for installation in the permeable pavers for Phase 2 of the Green Learning Station. Phase 2 is to install pressure sensors into the catchment areas to more closely monitor the runoff process. Also, a biology professor approached us with interest in expanding our network into water quality and green roof monitoring, the 1-minute time interval would enable a new scale of modeling for his work. The WaterML 2.0 integration interested the civil and environmental engineers at the university. The semiconductor company, Maxim Integration, donated all the sensors for the project expressing interest in the possibilities to expand their sensor integration capabilities. The creative process of the project continued to enable and motivate innovation throughout the geospatial sensor network.

The result is a project that has continued to create opportunities to expand beyond its original research focus of storm water monitoring. We have recently been approached by a scientist to deploy sensors in extreme remote environments with electricity and sporadic Internet. These deployment limitations require a GPS, adjustable sensor intervals, and large data storage. The creativity within open innovation begins to develop alternative solutions to difficult challenges faced within sensor networking. The flexibility of the geospatial sensor network continues to be applied to various projects around the university and the community. Future work will discuss these implementations with advances in plug-and-play and extreme remote environment sensor networking.

References


An open source software approach to geospatial sensor network standardization for urban runoff

Andrew J. Rettig a,⇑, Sumit Khanna b, Dan Heintzelman c, Richard A. Beck a

a Department of Geography, University of Cincinnati, Cincinnati, OH 45221, United States
b College of Engineering and Computer Science, University of Tennessee at Chattanooga, Chattanooga, TN 37403, United States
c School of Planning, University of Cincinnati, Cincinnati, OH 45221, United States

Abstract

In this paper, we implement a geospatial sensor network for monitoring a green technology stormwater runoff site. The sensor network uses OpenWRT, an embedded Linux operating system, and other open source software, to create a modified router for reading Maxim’s 1-Wire™ protocol, queuing and transferring standardized sensor data while enabling location and time. The modified router created the bridge between the sensor protocols and the middle-level software to provide reliable data to both the sewer district and the Environmental Protection Agency. Representational State Transfer (REST) is used in the design philosophy of the client and server open source software for transferring the data from the embedded systems to the server level for storage and publication. The use of open source software not only creates a more affordable network but lowers the entry barrier to sensor networking and enables developers for continued innovation and standardization.

1. Introduction

Stormwater and wastewater management has been a crucial part of urban planning since the beginning of urban development. Engineers design city infrastructures to either carry water away through city streets and water ways or capture the water for reuse. Stormwater is the result of water runoff from impermeable surfaces during precipitation, in contrast to wastewater that is discarded after human use. It is not uncommon for stormwater and wastewater to be combined into a single system, combined sewer systems. Peak flows during rain events can cause these systems to overflow. Accurately predicting stormwater volume with variable rainfall is crucial to preventing flooding. However, stormwater is rarely measured due to a lack of affordable and standardized sensor networks. In the rare cases where stormwater is measured, it is at a few fixed locations. Alternative means are beginning to be created for more widespread measurement throughout the city. Distributed fixed and in some cases mobile stormwater measurement and water quality stations are being created to supplement fixed locations (Ruggaber, Talley, & Montestruque, 2007 and Sempere-Paya and Santonja-Climent, 2012). But complete city coverage at the household level is impractical both financially and technically, primarily due to a gap between sensor protocols and middle-level software. Sempere-Paya and Santonja-Climent, 2012 described the next step for urban wastewater sensor and management systems as the merging of “Internet of Things” and “Smart Cities” for providing the automation of data. The first step in their plan was for a modular architecture that is not tied to the equipment manufactures at geographically disperse locations.

In this paper, engineers and researchers from the University of Cincinnati implemented a modular architecture for monitoring stormwater runoff. The geospatial sensor network uses the embedded Linux operating system, OpenWRT, for reading, storing and transferring sensor data. The open source implementation bridges the gap between sensor protocols and middle-level software enabling innovation within an end-to-end sensor network. The structure of this paper includes: previous work in sensor networking highlighting open source developments, an introduction to the study area and project, and the technical architecture of the network focusing on the OpenWRT router. The OpenWRT router enables the affordable and scalable modular stormwater sensor networking implementation so the paper describes building the embedded Linux kernel and flashing the router.
2. Previous work

2.1. Sensor networking

Traditionally sensor network innovation and development occurs within military research. The Sound Surveillance System (SOSUS), considered one of the first sensor networks, is a network of underwater acoustic sensors that was installed in the 1950s as a result of research from World War I and World War II (Chong & Kumar, 2003). SOSUS today is part of a larger military sensor network. As research continued into the 1980s, work was done under the Defense Advanced Research Projects Agency (DARPA). Today, the most advanced sensor networks are still being developed by the military but with increasing affordability, sensors and sensor networks are economically deployable. With the development of open source modified routers with a GNU/Linux Operating System, user community innovation will help accelerate and improve deployment. Sensor deployment faces many challenges including: power consumption and battery issues, storing of data, streaming data, connectivity, wifi meshing, sensor availability, and weather and environmentally resistance. Allowing greater user control, incentive and innovation will continue to improve sensors, sensor networks and standardization. Bob Gourley founder Crucial Point LLC, top 100 “Tech Titans” list and former member of US Department of Defense, cyberdefense group recently stated “Open source tools are becoming the infrastructure that every company is putting themselves on,” (Vance, 2013).

Open source software, for sensor networks, has been developed by several research groups. The Web Service Access Framework (OX-Framework) has been developed by 52° North to assist with clients connecting to the Open Geospatial Consortium (OGC) services. The Framework assists with the formidable task of trying to connect a variety of different sensors to the client for visualization (Bröring, Jürrens, Jirka, & Stasch, 2009, Jirka, Bröring & Stasch, 2009). Effective clients are needed for comparison and analysis of the data. The OX-Framework offers a reusable design that can be altered. Quality Assurance/Quality Control (QA/QC) is very important with sensor networks as complications arise at various levels within the network. Workflow systems can help manage the data, as well as can be used for data exploration and analysis. Examples of open source data management systems are VizTrails, Taverna and CUAHSI information system (Ames et al., 2009; Callahan et al., 2006 and Hull et al., 2006). Analytical tools such as R, Kepler or GCE Matlab are often incorporated into the data flow as well (Alintas et al., 2004). The most relative open source development within sensor networking is Data Turbine. Data Turbine is a robust real-time streaming data engine with three basic parts; First Data Turbine connects to sources, or sensors, to stream the data; Second, it connects to servers for storing the data; Third, it connects to sinks that receive the data for quality assessment and quality control (QA/QC) or visualization (Fountain, Tilak, Shin, & Nekrasov, 2012). Recently, Data Turbine was ported onto an android platform sensor pod for easier sensor integration and additional processing closer to the sensors. We chose not to use the android platform of Data Turbine for a more affordable OpenWRT Linux based system suitable for wired Ethernet connections.

OpenWRT is GNU/Linux for embedded devices with a writable file system with package management called ipkg, Itys Package Management System. OpenWRT was a development specifically for writing a GNU/Linux firmware for a router, specifically the WRT54G Linksys router (Innes, 2005). Linksys released the WRT54G router in 2004 running an embedded version of Linux. Once Linksys released the code, the WRT54G router code be customized and enhanced by the open source community (Innes, 2005). However, OpenWRT was written as a new complete firmware for the router. OpenWRT was different from other firmware for the WRT54G, because it did not attempt to be an all-inclusive system with only a minimal Linux install (Innes, 2005). This firmware approach has attracted many programmers over the years and attracted our engineering team with its applicability as a sensor data logger and transmission device.

3. Methods

3.1. Study area

Stormwater runoff and resulting combined sewer overflows are one of the largest water polluters in the nation (EPA, 2012). Cincinnati has recently negotiated a federal consent decree (legally binding agreement between city, state, and federal agencies) that sets combined sewer overflow pollution reduction goals to two billion gallons by 2018 (EPA, 2013). Our geospatial sensor network was developed in conjunction with the Green Learning Station Project (GLS) at the Cincinnati Civic Garden Center. The Green Learning Station Project was part of the Metropolitan Sewer District of Greater Cincinnati (MSDGC) green technologies examination of pervious pavements, bioswales and rain water harvesting for reducing stormwater runoff (Fig. 1). Green infrastructure sites have been implemented throughout the city as “early success” sites to examine their reliability and effectiveness to meet the two billion gallon reduction goal (EPA, 2013). Our team of engineers, UC@GS, was formed in conjunction with the Cincinnati Civic Garden Center and the University of Cincinnati to develop a real-time and location in situ sensor network for monitoring stormwater runoff from the test site. The sensor observations would be used for evaluating the green technologies by the MSDGC and the Environmental Protection Agency.

3.2. Site design

The sensor network to be implemented would be an end-to-end network, monitoring the water flow and quantity of the 5 permeable pavers and 1 control pavement. Weather stations were also incorporated in the network, installed on site to record the rainfall for evaluating the permeable pavers. Fig. 2 is a diagram of sensor monitoring infrastructure for water quantity and flow at the Green Learning Station. Our prototype stormwater monitoring sensor network consists of: a water pump and water meter in each manhole, a counting sensor on each meter, a temperature sensor in the manhole, Cat5 burial grade cable to the Green Learning Station, a USB to 1-Wire convertor, a USB hub, an optional GPS, a modified router, Ethernet connection, and an offsite server cluster (Fig. 2).

3.3. An open source sensor network architecture for stormwater runoff

Sensor networks have three generalized levels of infrastructure, (1) a low-level physical layer of sensors and monitoring infrastructure that translates the sensor data into a common format for storage and transmission, (2) a middle-level layer of hardware and software that receives the data and parses the data in a database, and (3) a high-level hardware and software for distributed visualization and analysis in the form of web services (Fig. 3). The low-level is composed of a variety of proprietary and open protocols for communicating with the sensors. This level of infrastructure includes sensors connecting to a modified router that receives and translates those data from a variety of formats to a standard XML format and transfers the data to the server. This low-level set of functions is necessary to construct the next generation of plug-and-play sensors. The low-level of our technology stack used
a customized version of the OpenWRT operating system on a home Internet router to translate proprietary sensor data streams to a common XML format for transmission to our database. Our modified router enabled us to develop custom software at all levels of the network to improve in situ sensor networking software. We created LtSense, using the GPLv3 open source license, to read the sensor data, package it in XML and transmit it using HTTP as our lower-level.

Open source hardware was also considered including Arduino (Hribernik, Ghariri, Hans, & Thoben, 2011 and Boonsawat, Ekchamanonta, Bumrungkhet, & Kittipiyakul, 2010) but the modified routers satisfied the technology requirements while allowing us to work with an established bulk manufacturer. Our modified OpenWRT routers fill the same sensor-to-network gap in end-to-end in situ sensor networks as the Data Turbine “sensor node”, but use embedded classic Linux on less expensive hardware with physical connectors appropriate for our stormwater application. The GNU/Linux Operating System on our OpenWRT routers provides all of the freedoms of open source software. Our OpenWRT routers are able to connect directly to the lowest sensor level protocols. For our project, the Green Learning Station Project, we accessed the data directly from the Maxim 1-Wire protocol. This low level connection even allows us to create a complete loop, in that we can control the sensors at the 1-Wire level as well. These freedoms allow for control of the sensors and the data including innovation of the sensor connection, data storage and streaming at the low-level. This enables space and time awareness as a GPS enabled OpenWRT router is easy to implement.

The middle-level of hardware and software, comprises the infrastructure that receives the data from the modified router (and connected sensors) and parses the data into a database on a...
server. Our middle-level receives the data from the modified router via HTTP over wired Ethernet connections. The middle-level was initially composed of our open source BigSense software in SCALA on a Windows server that received and parsed the XML format sensor data to a PostgreSQL database on a server running the Linux operating system. Our high-level hardware and software level provides the search and retrieval of visualization and analysis of data based on sensor parameter such as temperature, rainfall, and stormwater flow volume, by date and time and location of acquisition via REST services for simplicity and interoperability. The high-level used our open source BigSense software running on a Linux server to extract data from the database in XML, tab delimited, comma separated value, and HTML formats. BigSense also converts sensor data between Standard and Metric units of measure and provides separated value, and HTML formats. BigSense also converts sensor data to extract data from the database in XML, tab delimited, comma separated value, and HTML formats. BigSense also converts sensor data between Standard and Metric units of measure and provides separated value, and HTML formats.

Our innovation efforts are related to a continuing international effort for sensor network standardization by the OGC with the Sensor Web Enablement (SWE). However, at the time we began this project those standards were not yet specific enough to implement for stormwater monitoring. Our implementation provides a starting point for future OGC compliant sensor networks for stormwater monitoring. Our work contributes to the innovation within REST services to be utilized by the OGC (Bröring, Remke, & Lasnia, 2012; Janowicz, Anne Bröring, & Thomas Everding, 2010; Janowicz et al., 2013 and Mazzetti, Nativi, & Caron, 2009). Rapid advances in the capabilities of small and inexpensive networkable computers with internal or external sensors are creating a need for expanded OGC standardization within the Internet of Things.

3.4. Low-level sensors

The 1-Wire counter sensor DS2423 was installed for getting the sub-liter flow and volume rates from water pumps beneath a variety of permeable pavement types in order to determine their effectiveness with regard to stormwater management. This sensor was also used for the wind direction, wind speed and rain volume on the weather station. The DS18B20 1-Wire temperature sensor was also used on the weather station. In addition, the DS18B20 was used in the man holes to verify Ethernet cable connectivity when the water pumps were idle.

Maxim’s 1-Wire Public Domain Kit, a C program API for accessing low level 1-Wire bus protocol algorithms was utilized for sensor reading while 1-Wire drivers established the connection. Initially, an open source program, 1-Wire File System (OWFS) was used for testing and evaluation. OWFS is a suite of programs that is designed to make the 1-Wire bus and its devices easily accessible (OWFS, 2013). Conduct and wires were installed at the station for physically connecting the sensors to the modified router. Once connected routers could control, as well as, receive information from the sensors. The observation information from the sensors could be stored locally but primarily the data was streamed to the server. Observation information was able to be recorded at 60 s intervals. Our sensor-to-network OpenWRT routers with our LtSense software provided an affordable, low-power, low-carbon emission, WiFi enabled, scalable solution for stormwater sensor networking that is accessible to the general public.

3.5. Low-level sensor-to-network OpenWRT router with LtSense

Most current sensor networks consist of high cost, specialized embedded systems used to collect sensor data. Our work has shown that these expensive systems could potentially be replaced with much lower cost commodity embedded sensor storage and transmission devices such as our OpenWRT routers with LtSense. We tested common-off-the-shelf (COTS) routers under $100 USD that were marketed towards non-technical consumers for this purpose (Fig. 4). Routers are layer-3 devices in the Open Systems Interconnection model (OSI) used to send and receive traffic between multiple computer networks. Our OpenWRT modified routers, more accurately referred to as residential gateways, began to enter people’s homes with the emergence of residential broadband Internet service in the early 2000s. Routers allowed multiple computers to share the same Internet connection and typically used only one IP address using Network Address Translation (NAT), that made all devices attached to the route appear to be only one device to the rest of the network. Today, most routers come with additional features that make them ideal for connecting sensors to networks such as support for wireless devices using Wi-Fi, quality of service (QoS) to ensure low latency traffic gets a higher priority, wireless security and mobile broadband connectivity.

There are many companies that manufacture these routers including Cisco-Linksys, Netgear, Belkin and D-link. Although they appear to be specific purpose devices, many of these routers run on general purpose embedded processors, with ARM or MIPS architecture and a Linux operating system. Routers often vary in their central processing unit type, storage capacity, memory and other features. Analyzing each potential router’s datasheet was essential in finding the best cost to performance when choosing a router to use as a general purpose device for processing and transmitting sensor data. Memory and processing power are important for linking sensors to networks. For the device to meet our needs we needed it to also have a USB connection and for the manufacturer to be willing to work with us directly. Lastly, with these needs met, it was essential to be able to flash OpenWRT onto the router as the operating system for running LtSense. We chose an inexpensive, bulk ordered, Netis WF-2410 with 16 MB of storage and 64 MB of memory with a customized embedded Linux Kernel that supports Wifi and USB to satisfy our needs.

3.5.1. OpenWRT on the Netis router

To install OpenWRT on the Netis router, it was necessary to find an operating system kernel that supported the chipset built into the router. The Netis router is built upon the Ralink RT305X chip-set. We modified the Asus RTG-32 OpenWRT software (http://wiki.openwrt.org/toh/asus/rt-g32) to work on the Netis router. A serial to USB cable was used to connect the main board back to a laptop. Putty, a communications terminal program, initiated a serial connection. Uboot, the boot loading program, enabled the flashing of firmware via a Trivial File Transfer Protocol (TFTP) connection. Modifications were necessary to get OpenWRT to use the USB and additional flash storage on the Netis. With the help of open source developers such as Sergiy Gurjev, who had worked on adding USB support to the Asus RT-G32, a working USB patch was created for the Netis. USB support allowed an interface with the 1-Wire sensors, storage drives, GPS devices for location, 3G/4G dongles for mobile internet and cameras. The flash map was also rewritten enabling the full use of the 16mb of flash on board as compared to the 4mb on the Asus. This additional space was needed for extra drivers, utilities, file system support, and software libraries.

Lastly, OpenWRT provides software packages for many of its dependencies including Python 2, One Wire File System (OWFS), distribute and libusb. Python 2 was used for programming the modified router transmission. OWFS was used initially to read 1-Wire sensor data. Distribute is used to install additional Python packages not present as OpenWRT packages such as rsa. Libusb
was used for accessing the usb devices. With these added features our COTS (common off the shelf) routers had now been purposed as scalable open sensor storage and transmission devices (Fig. 5).

3.6. Middle-level

Our modified router with open source software enables the bridging of the gap between sensor protocols and middle-level software. The Maxim driver and 1-Wire API are utilized to retrieve the sensor data on the modified router. This allowed for the implementation of programmable data translation and forwarding software, Lt Sense. LtSense is a Python 2 application that was developer for the purpose of reading data from sensors and encoding that data in XML for transmission to a service for storage. BigSense is a web service, developed in conjunction with LtSense, to provide the means of receiving and storing the sensor data from LtSense. BigSense was written in Scala and runs on a Tomcat 7 web application server. BigSense was developed to consume sensor data sent to it by LtSense. The web service uses REST in its design philosophy. The entry point to the service is via a series of actions. There are actions for posting sensor data to the web service in supported formats, optionally using RSA keys for digital signature based security. The posted sensor data received by BigSense is parsed into a relational database for storage and retrieval. Our success with LtSense and BigSense resulted in continued development of the BigSense REST service.

3.7. High-level

BigSense development expanded to also provide a means to query sensor data. BigSense published the data creating transparent Universal Resource Identifiers (URIs) that are both transparent and flexible. The queries provided both the specific measurements, as well as, aggregate queries that show averages of the data over time. Data can be queried by specific sensors, dates and times. BigSense uses a modular framework that is designed to support multiple types of formats, relationship databases, unit conversions, security mechanisms and storage back ends. The REST services URI approach simplified testing with an accessible high-level architecture. Developers quickly and easily integrated the sensor data into a user friendly web site. BigSense provides the overarching integration of our sensor networking by allowing the user to search and retrieve and visualize real-time, distributed, networked sensors for stormwater monitoring. Lastly, the REST services also included spatial data of all the sensors enabling a geospatial sensor network. This allowed our application developer to create innovative web Geographic Information System (GIS) applications such as the one shown below for quick and easy to understand visualization and analysis (Fig. 6).

4. Results

The lower-level open source software and modifiable router enabled 1-Wire sensor connectivity. This connectivity and data
translated = the Cincinnati/Northern Kentucky International Airport (t-calculated = –0.103, p-value = 0.918, \( \sigma = 0.05 \)). The sensor network implementation satisfied all of the temporal and spatial data requirements for the real-time stormwater volume and flow for the Metropolitan Sewer District of Greater Cincinnati (MSDGC) and the EPA. Both the MSDGC and the EPA are currently analyzing the Green Learning Station data to evaluate green infrastructure implementation for stormwater mitigation.

5. Conclusion

Geospatial sensor networking has traditionally had a high level entry barrier. Sensor networks require a complex architecture with compatible sensors, hardware and software. Even with the availability of these resources, basic implementation can be overwhelming. However, we believe with continued open source involvement these challenges can begin to be made manageable. Our work within this stormwater runoff sensor network creates a modified OpenWRT router enabling the first step in the modularization of stormwater runoff sensor networking. Our OpenWRT router is beginning to enable widespread monitoring to help decision makers identify areas in most need of stormwater runoff reduction. Each year combined sewer overflows have resulted in 850 billion gallons of pollution of untreated wastewater and stormwater (NPDES, 2004). Modularization is not only important to stormwater runoff but all in situ sensor networking creating a more affordable network and helping to bridge the gap between lower level sensor protocols and middle-level software. Modularity by definition is suggesting a loosely coupled networking system where components may be separated and recombined. The result is networks where sensors may be added and devices may be replaced. Modularization also creates a scalable network with endless applications. Developers can utilize the hardware and software to expand upon our solution or adapt it for alternate applications with a freedom and affordability rarely found within sensor networking.

Eventually, our goal is that sensor networking is simplified to a plug-and-play implementation. The open source approach is essential to the enablement of plug-and-play by creating developments within standardization. Open source development within infrastructure across various medium is ideal for the user community involvement. The modified OpenWRT router continues innovation opportunities by allowing for OGC SWE standard testing and development within end-to-end in situ sensor networking. Initially our XML format for data transfer is not an OGC SWE standard nor is our REST web services. However, the newly approvedSensorML 2.0 standard has applicability for sensor registration and description. Continued innovation within low-level data transfer is essential. Currently standards are based upon Simple Object Access Protocol (SOAP) web services while the Internet of Things primarily utilizes REST services. Future work will focus on our web services and bridging the gap between traditional geospatial sensor networking and the current linked data approach utilizing the semantic web. For more information on the stormwater project please visit the Green Learning Station online at <http://green-learningstation.org/>. Updated information on LtSense and BigSense may be found at <http://bigsense.io/display/overview/BigSense.io>.

References


Accepted pending revisions for Applied Geography (Applied Geography Conference special issue)

37th Applied Geography Conference

Wednesday (10/15/2014) to Friday (10/17/2014)
Open source REST services for environmental sensor networking

Andrew J. Rettig, Sumit Khanna, Richard A. Beck

Department of Geography, University of Cincinnati, Cincinnati, OH 45221, United States
College of Engineering and Computer Science, University of Tennessee at Chattanooga, Chattanooga, TN 37403, United States

Abstract

The greatest challenge in the implementation of environmental sensing networks is converting a large variety of data streams from diverse sensors, often in proprietary protocols, to international standards such as Extensible Markup Language (XML) with Open Geospatial Consortium (OGC) XML tagging and web service standards. Implementing standards throughout the architecture will not only enable interoperability and reduce cost but will allow scientists to contribute to sensor network innovation. This article introduces open source Representational State Transfer (REST) services created specifically for environmental monitoring. OGC standards are suggested to help guide future community development for sensor description and registration. This article contributes to the design and implementation of affordable, self-documenting, near-real-time geospatial sensor webs for environmental monitoring using international standards.

Keywords: Environmental sensor networks, Open source software, REST, Open Geospatial Consortium, Web of Things

Introduction

What would computer networking be today without the standardized communications protocols interconnecting networks to form the Internet? A significant part of the Internet’s success has been made possible by volunteer adherence to software (e.g. TCP/IP and HTTP) and hardware standards (e.g. CAT6 and RJ45) (Hovav et al., 2004). This resulted in computers with standardized software that were easily connected to a standardized communications network. This approach decreased costs while enabling interoperability for manufacturers and end-users to help create the global information network. Before the mainstream Internet existed, proprietary networks such as CompuServe and AOL connected millions of subscribers delivering limited and controlled services. These services were provided, controlled and monitored by the provider with proprietary protocols. However, these same proprietary controls limited flexibility, functionality and innovation within the system. Without the Internet, these proprietary networks would have had a significantly smaller effect on our society and economic system.

Protecting our environment requires timely and affordable information. In 1998, U.S. Vice President Al Gore articulated the term “Digital Earth” for the visionary concept of a virtual earth to interconnect and to geo-reference the world’s digital knowledge. Traditionally sensor networking has occurred from the top down in military applications (Diaz et al., 2012) and more recently in corporate implementations. These proprietary installations have created a controlled development regime with significant advances only within the confines of tethered device releases. Proprietary “in situ” sensor networks are providing similar monitored and controlled services for complete sensor networking solutions. Although these providers are advancing sensor technology, in many cases, they are limiting innovation as well. Proprietary
development rarely promotes the development of standards and interoperability within sensor networking. Sensor networking must utilize the business practice of open innovation to enable sensors that are implementable by the general developer. Open innovation suggests that entities not only look beyond their own borders for external ideas but rethink intellectual property as it pertains to research and development. Open source software lowers the entry barrier to sensor networking enabling innovations including new applications of the sensors and standards within the network (Bitzer and Schröder, 2003; Bitzer, 2004). Standardization is essential for low-cost collaboration and rapid innovation of environmental sensor networks.

The Open Geospatial Consortium (OGC) has been the global forum for the development of standardized sensor networks in the form of Sensor Web Enablement (SWE) since 2001 (OGC, 2014). SWE consists of web accessible sensor networks and data that can be discovered, accessed and possible controlled using open standard protocols and interfaces (Botts et al., 2008). The OGC was chosen to host the standards for sensor networks because location is essential to sensor data and often the only common field between heterogeneous systems. Similar to the Internet, serious debates must be waged on each part of the most basic protocols with widespread testing to make sensor network standards widely applicable.

Current innovation and applications are being developed between the OGC and the Web of Things, web integration of the “Internet of Things” (IoT). The IoT is the connection of the physical world to the Internet through ubiquitous uniquely identifiable sensors. However the Internet of Things is still being limited by the lack of clear, standardized and interoperable community protocols (Guinard & Trifa, 2009). Guinard and Trifa provide guidelines for integrating real world objects into the Web creating the Web of Things (WOT). They suggest that HTTP becomes an application layer as devices become available through a RESTful API. Our work applies these guidelines to Environmental Sensor Networking (ESN). A RESTful approach is used for client and server software that record, store and transfer the sensor data. The open source software client and server platform is developed independent of the hardware. A modulation approach allows increased innovation and reduced cost throughout the network (Sempere-Payá & Santonja-Climent 2012). This approach allows for the incorporation of OGC standards were applicable and uses recent advances from the WOT. Hart et al. (2009) states, ESNs will produce a revolution in all aspects of earth system and environmental sciences similar to that generated by the use of satellite remote sensing in the 1990s. Whether, a Digital Earth or Hart’s revolution neither will occur within environmental science without the help of scientists being involved in every level of the sensor network implementation and development.

This article begins with a discussion of previous research before introducing an open standards-based architecture to enable innovation in the development of environmental sensor networks. The two general categories of web services are discussed to compare OGC web services to current WOT web services. A case study describes the working implementation of the open source software architecture.

**Previous Research**

**Web Services**
Shortly after the challenge was presented to create the Digital Earth, web services were beginning to establish themselves with standardization. With web services, the Web was in its next stage of evolution, in that software components could discover other software components (Roy & Ramanujan, 2001). The World Wide Web Consortium (W3C) categorizes web services into two general categories: 1) arbitrary web services and 2) REST services (W3C, 2014). Arbitrary web services consist of services through a single access point. All of the operations of the web service are executed through this single end point. The operations executed by the service are custom for each individual service (W3C, 2014). Almost all arbitrary web services utilize the Simple Object Access Protocol for standardization (SOAP). SOAP standards were developed in the late 90s and became the de facto standard for arbitrary services in early 2000.

In contrast, REST services achieve integration in a more lightweight and simpler manner, and focuses on resources (Guinard & Trifa, 2009). The concept of the REST web service was introduced in 2000 by Roy Fielding. REST uses the stateless nature of the Hyper Text Transfer Protocol (HTTP) (Guinard & Trifa, 2009). HTTP was originally intended to transfer pages of data to web browsers to be presented for humans. However, it's grown to be a protocol used to marshal data as well between applications, in the form of web services, which are consumed by other network services. The original GET and POST requests are the ones most commonly used in standard web browser requests, but the inclusion of DELETE and PUT allow for more complex operations where the Uniform Resource Identifier (URI), used to reference a piece of data, also relates to what that data is and what operations are being performed upon it. The real benefit of REST is its enterprise level integration and this contributes to the visionary multilayer Digital Earth (Janowicz et al., 2010; Guinard et al., 2009). The RESTful API lowers the entry barrier for developers allowing them to quickly query and understand the data for application integration. The developer friendly interface and the smaller client and server footprint has resulted in the wide spread adoption of the RESTful approach for the Web of Things (WOT).

**OGC Web Services**

Along with the standardization of web services, sensor networking standards were beginning to be suggested and established. Then in 2001, these sensor network standards were incorporated into the work of the Open Geospatial Consortium (OGC) in the Sensor Web Enablement (SWE). With the growth of GIScience, it has become apparent that it must take a leadership role in analysis and understanding with the majority of data embedded in space (Farmer & Pozdnoukhov, 2012). OGC SWE standards were developed using arbitrary web services with the SOAP protocol primarily for satellite remote sensing and other large sensing platforms.

However, in the last 5 years, OGC SWE research has begun to use REST services bridging the gap between SWE services and the WOT. Therefore, many of the SOAP-based SWE standards such as sensor data encoding standards using XML are now applicable to REST services. Specifically, OGC Observation and Measurement (O&M) standard defines XML schemas for observations and for features involved in sampling when making observations publishable (Cox,
2011). An example is a RESTful proxy that has been developed to extend SWE SOAP services and to replace the extensive filtering capabilities of the original Sensor Observation Service (SOS) (Janowicz et al., 2010). Also, the new SensorML v2.0 standard defines XML schemas for focusing on the process of measurement and observation while also including the ability to define the physical characteristics and functions of the sensors and actuators (Botts & Robin, 2014). Our work expands upon the integration of OGC with RESTful approaches with a client/server architecture for in situ environmental sensing. Our work develops BigSense, open source RESTful services implemented on the server, and LtSense, an open source client running on the embedded devices. The RESTful services will be introduced as well as the embedded client.

**Architecture**

Environmental monitoring traditionally required the recording of data through first hand observation or logging devices. However, environmental monitoring is being revolutionized by ESNs (Hart & Martinez, 2006). Architecture should be based on requirements emerging from specific usage cases (Mazzetti et al., 2009). An ESN (Fig. 1.) is comprised of an array of sensor nodes and a communication system that allows data to reach a server (Hart & Martinez, 2006). In situ ESNs typically require little processing or complexities at the sensing level. Understanding the environmental sensor to server infrastructure is essential for open source developers in this field of research. The RESTful client, LtSense, provides a basic architecture for communicating with the sensor protocols, archiving the data locally if necessary and transporting the data in a temporal and spatial standard to the server. RESTful web services on large traditional servers at the University of Cincinnati run our BigSense software to provide database, storage, retrieval and visualization functionality. This basic open source architecture will assist innovation within environmental monitoring. The RESTful approach allows us to use OGC standards modified for WOT at every level of the network including the publication of the data at the server level.

![Figure 1. Basic system architecture for environmental sensor networking](image-url)
LtSense (pronounced Little Sense) is a client application written in Python that can be installed on embedded devices and is licensed under the GNU GPL3. It retrieves data from sensors and transmits data to web services such as BigSense using the HTTP POST method (Fig. 2.). LtSense polls for new data at configurable sample interval, and then queues sensor data in a SQLite database. Optional RSA signature verification can be enabled to provide for the authenticity of the data. Standard Secure Socket Layer connections (SSL/TLS) can also optionally be used for confidentiality. LtSense also has various options for its functionality and is continually being expanded (Figure 2.). LtSense has been utilized for both 1-Wire sensors and transmitting images from USB web cameras.

The extensibility and scalability of LtSense could be further increased with the incorporation of the newly published OGC SWE SensorML v2.0 standards. SensorML, a process and measurement standard, can be implemented within an XML description file (Fig. 3.) and sensor registration. The description file example is included as a preliminary step towards understanding and implementing standardized registration. After implementing the description file, it could be retrieved from LtSense as needed for inventory management, location information or general understanding of the process and components included. The primary information for the description file is system description, observed property and station location. The example is a single sensor at a static location, for multiple sensors additional outputs could be included. The SensorML standard includes various embedded standards within its schema. Geography Markup Language (GML) is used for creating the unique id for the resource, shown in Figure 3. as “DevSite01” and for the basic description. The observed property (Fig. 3. Line 9) includes a resolvable definition to specify the measurement for interoperability across scientific communities. The definition utilizes the NASA Semantic Web for Earth and Environmental Technology (SWEET) ontology. Ontology is defined vocabulary within a domain enabling the sharing of information through a clear conceptual model. Similar to standards and often embedded in them, ontologies are meant to be utilized whenever possible (Simperl et al., 2013).

For the station location, there are numerous options for representing spatial location in SensorML, including the Vector element (Fig. 3.) or options within GML. As development on
LtSense continues GML can be used to represent various spatial information and coordinate systems as well as stationary to mobile sensors. Ontologies are essential for location identification as well. In Figure 3., SensorML ontologies are used within the Vector element. SensorML ontologies are currently being expanded and developed. Sensor specific ontologies are becoming increasing important to create sensor dictionaries of terms while defining the relationships between them (Botts Innovative Research, 2014). Working on sensor ontology is geographic theory and ontology is essential to geographic theory within GISience (Farmer and Pozdnoukhov 2012). Lastly, although more complex, sensor registration uses many of the same elements from SensorML as sensor description making sensor registration the next logical development. The SensorML v2.0 standards with structured encodings and resolvable namespaces are an essential guide for interoperable open source development within sensor description and registration processes as sensor inclusion expands. For the latest development on LtSense visit https://github.com/sumdog/LtSense.

BigSense

BigSense is an open source web service, licensed under the GNU GPL3 that is designed to record and present data from sensor networks at the server level (Fig. 4). BigSense is written in Scala and runs on Tomcat while utilizing the RESTful architecture for environmental monitoring data. It supports queries based on date and timestamp ranges. Aggregation support is also built into the API as well as the presentation of data in various formats (XML, Comma Separated Values).
BigSense stores all its data internally in metric and UTC time, but the API provides for both unit and time zone conversions, to allow for consistent queries across multiple platforms. BigSense can also verify LtSense data using the optional RSA signatures mentioned previously. Secure Sockets Layer (SSL) can optionally be used in the front end web server to secure data communications. BigSense also supports multiple relational databases for its storage including MySQL 5, Microsoft SQL 2008/2012 and Postgres 9. BigSense has a documented RESTful API to allow for developers to easily create clients and front ends for both storing and retrieving data. Not only does BigSense provide the necessary architecture for the recording and easy sharing of data, it also provides a community based development to begin the incorporation of OGC OM standards into the RESTful namespace (Janowicz et al. 2010). Access to the BigSense repository is found at https://github.com/sumdog/BigSense. Additional information may be found at BigSense.io.

**Case Study**

We implemented and successfully tested our RESTful web services architecture for an environmental sensor network for the Cincinnati Civic Garden Center, the US Environmental
Protection Agency (EPA) and the Cincinnati Metropolitan Sewer District (CMSD). The EPA and the CMSD required a sensor network to monitor stormwater runoff from permeable pavers at a green technology testing site, the Green Learning Station. Six pavement types were used in the construction of the research site including one control pavement. The goal of the project was to measure the residence times of the various pavers compared to the control. Basins were constructed to capture the runoff under each permeable paver type. Manholes, with water pumps and meters, were used to capture and pump the water at the base of each drainage area. A near-real-time in situ sensor network was required for recording rainfall and stormwater runoff data from six water meters and two rain gauges. Additional temperature data was also collected with the rain gauge and in the manholes. For the implementation, Ltsense was installed on modified embedded Linux routers for monitoring permeable paver stormwater runoff. 1-Wire sensors were installed on the water meters and the rain gauge tipping buckets and wired to the modified routers. Approximately 30 1-Wire sensors and 15 routers were installed on the site to thoroughly test the sensor network infrastructure. After minimal initial testing, Ltsense performed exceptional while recording the sensor data, storing the data in the SQLite if necessary and forwarding the information to BigSense. SQLite within the Ltsense application allows the storage of the data until an Internet connection is established. Data storage is only limited by the storage space on the embedded device making Ltsense ideal for intermittent connectivity. Ltsense recorded the data from the open source software 1-Wire File Service running on the router and transported the data to the server in an XML file. The site successfully recorded data every 60 seconds on stormwater runoff, rainfall and temperature for 12 months. A rain gauge data analysis was completed for December 2011 with no statistically significant difference between recorded data and the Cincinnati/Northern Kentucky Airport rainfall data.

BigSense was also used for the EPA and CMSD green technology site. Ltsense sent data in XML format, over the HTTP protocol, to the BigSense service, hosted at the University of Cincinnati. Upon receiving the data, BigSense parsed the data and stored it in a SQL database; data which could then be retrieved and aggregated via the web service. A webGIS application was developed using the RESTful API. The spatial application allowed for searching, chart display and downloading of the data while utilizing the BigSense RESTful API. The completed sensor network for the Green Learning Station satisfied all the requirements of the EPA and the CMSD. The network provided accessible near-real-time data with 1 minute time intervals for evaluating the permeable pavers.
Conclusion

The challenge within sensor networking is converting a large variety of data streams from diverse sensors. This challenge includes numerous moving parts such as converting sensor data from sensor protocols to middleware software to sensor registration and data encoding standardization. This article focuses on in situ sensor networks for environmental monitoring. Our work builds on the real world environmental monitoring basic client/server architecture by providing the open source LtSense and BigSense software. Standardized data encodings are suggested to guide development of the client description file, registration file and REST namespaces. The data encoding challenge will only begin to get easier as the entry point barrier for both scientists and developers is lowered. The continued exposure to the Open Geospatial Consortium Sensor Web Enablement standards will increase the understanding and importance of standards while testing their applicability. The implementation of sensor networks will slowly get easier with understanding from a comprehension standpoint and the establishment of open source software platforms. The Sensor Web framework can be fun and excited as the interoperability of sensors begins to come to fruition. Similar to the Internet, a generative Sensor Web will create innovations and solutions that have not even been considered. Only through open innovation will Environmental Sensor Networking revolutionize all aspects of earth system and environmental sciences. Future work will continue to exam sensor networking infrastructure from a GISience perspective with spatial data as an essential component. The spatial data focus enables alternate methods and techniques within the infrastructure such as seamless location based searching at the presentation layer. Our software development will also enable expanded environmental studies as an affordable and scalable solution. LtSense is currently being installed on BeagleBones for deployment in the Arctic for global warming studies and Africa for expanded weather monitoring and coverage with BigSense receiving the data from international deployment.

REFERENCES


Conclusion

The challenge within in situ sensor webs is converting a large variety of data streams from diverse sensors into standardized data encodings. This dissertation has helped to solve this challenge by testing and implementing international geospatial standards within an in situ sensor network architecture. Each layer of the architecture is evaluated for the applicability of the geospatial standards. An open innovation approach is incorporated into a real-world sensor network project to promote collaboration, open source software and low cost hardware. The Green Learning Station in situ sensor network served as this real-world test bed and is helping to fill the need by the Environmental Protection Agency (EPA) and the Cincinnati Metropolitan Sewer District to (CMSD) monitor a stormwater runoff green technology.

The Green Learning Station project is a response to the combined sewer overflow (CSO) problem in Cincinnati. The topography of Cincinnati, large areas of impervious surfaces, frequent large rain events and clay soils often results in large stormwater runoff events. The antiquated sewer systems in Cincinnati use a single pipe system for both stormwater and sewage. During significant rain events the combined sewers are unable to control the large runoff events resulting in overflows. The EPA and CMSD have created numerous green technology sites to test stormwater runoff mitigation. Sensor networking is used on these sites to gather data for examining and mapping the effectiveness of the various technologies. The Green Learning Station was the first fully operation site in Cincinnati with near-real-time data from the affordable end-to-end geospatial sensor web. The three articles presented in Chapters 2, 3 and 4 document the new network architecture.
**End-to-End Geospatial Sensor Web**

The first article introduces the problems, networking cost and complexities within in situ monitoring of stormwater. The article introduces open innovation as the solution to affordable and standardized networks. Open Innovation encourages both a collaborative approach and the rethinking of intellectual innovation with open source software, open hardware and open standards. All three are needed for affordable in situ sensing. The article includes a complete end-to-end sensor network technical discussion of the Green Learning Station project. The article also begins to expose the inadequacy of current OGC SWE standards for in situ implementations. Lastly, experimental results on the rain gauge data are also presented, verifying the quality of the sensor data being recorded for the test site.

**Bridging the Gap; Sensors and the Internet**

The second article was written to expand upon ways to bridge the gap between sensors and the Internet for the Green Learning Station project. The most challenging hardware development for end-to-end networks is finding and programming an adequate embedded device to link sensors to the Internet and then transmit the data in a standardized format. Engineers and scientists struggle to find open embedded devices for sensor network infrastructure. Even with the recent development of the Raspberry Pi, Arduino and the BeagleBone open hardware devices, bulk order devices continue to be an issue. For this prototype, open source software was written to flash Common-Off-the-Shelf (COTS) routers to transform them into simple devices for translating proprietary sensor data streams to standardized XML data streams on the Internet. These modified routers were able to connect to the sensors to
the Internet by storing and transferring the data to the server. These modified routers are the first step towards a modular architecture for an urban runoff sensor system. This platform enabled the development of additional open source software for transferring, storing and publishing the sensor data at the data server(s) for visualization and analysis in a spatial and temporal context. These low-cost embedded Linux-based sensor-to-Internet devices made the rest of the standardized client/server architecture possible and helped to make the overall sensor network affordable and reproducible.

**Environmental Sensor Web Architecture**

The third article introduces REST web services that were specifically designed for environmental sensor webs. The article reviews current approaches to web services and provided the reasoning and benefits of the RESTful design approach for in situ sensor networking. The open innovation approach described in the two previous articles is continued with standardized web services to reduce complexity and cost. These open source web services simplify the challenge of client and server software development. The sensor description XML encodings, including sensor type and location, from the OGC SWE are introduced as an example to guide future community based development of spatially-enabled environmental sensor webs. The article incorporates applicable standards while justifying alternative solutions specific for in situ environmental sensing.
Lessons Learned

Installing geospatial sensor webs is difficult because of the cost and the complexity (lack of standardization) of the many components. Open innovation and the incorporation of current trends (common hardware and evolving standards) such as the Internet of Things helps to overcome these challenges. Spatial data is essential and must be considered at every layer of the sensor web. International geospatial, computing and telecommunications standards are important ways to improve upon current, often proprietary and expensive, sensor networks. At the user level, a simple interface for software developers as well as scientists assists with understanding, analyzing and mapping the data. However, scientists and engineers not only want standardized networks but are excited by the sensor web innovations and the resulting possibilities in scientific research. Incorporating these lessons into a successful geospatial sensor network can result in an affordable network with lower cost and skill entry barriers for scientists and other end-users. This dissertation is new and unique because of the critical approach to the design and implementation of sensor webs. Assembling a team of expertise at every level within the architecture afforded us the ability to improve within each area of sensor webs. The result is a complete innovative system that we can be proud of.

Future Work

Future research is needed for additional end-to-end geospatial sensor web case studies with real-world implementations to test and evaluate standards. The environmental sensor web architecture is applicable to a wide variety of monitoring solutions. Continued applied research will enable additional sensors to be incorporated into the software and hardware
design, such as GPS for mobile sensing. Future research in the communications layer will include development and testing of the sensor description and registration. Research in the presentation layer is needed for standardized spatial RESTful services and an open source user interface. Future work within geospatial sensor webs is exciting and fun as the interoperability of spatially-aware sensors begins to come to fruition.
Introduction Bibliography


