Sensor-Internet Share and Search—Enabling Collaboration of Citizen Scientists

Sasank Reddy*, Gong Chen*, Brian Fulkerson*, Sung Jin Kim†, Unkyu Park‡,
Nathan Yau*, Junghoo Cho†, Mark Hansen*, John Heidemann‡
* Center for Embedded Networked Sensing; University of California, Los Angeles,
†Department of Computer Science; University of California, Los Angeles,
‡Information Sciences Institute, University of Southern California

Abstract

Over the last decade, embedded sensing systems have been successfully deployed in a range of application areas, from education and science to military and industry. These systems are becoming more robust, capable, and widely adopted. Yet today, most sensor networks function in isolated patches, each with different mechanisms to deliver data to their users, and often have no formal methods to share data with others. As sensornets become more numerous and their data more valuable, it becomes increasingly important to have common means to share data over the Internet. In addition to simplifying use of a single sensornet, we seek to enable sharing of data across multiple systems, and ultimately *slogging* (sensornet logging), where a single user may discover, process, and republish data from thousands of independently operated sensors. To meet these goals we propose an architecture to interconnect, share, and search sensor data. This paper describes the building blocks of this architecture: sensor stores, search engines, and publishers, joined by a common sensor data streaming protocol. We then detail the research challenges that must be addressed to meet our goal of enabling sensor access to users from scientists, data analysts, to citizen scientists.

Categories and Subject Descriptors

H.3.5 [**Information Systems**]: Information Storage and Retrieval—*Data Sharing*

General Terms

Design, Languages, Management, Standardization

Keywords

Sensor Data, Sensor Networks, Standards, XML

1 Introduction

Sensor networks today are many wireless islands, each isolated and connected through the Internet to its owner, not easily usable by others. The goal of this research is to enable an archipelago of sensors, where both owners can bridge the pieces, and where guests can discover and share. We see a future where, at two extremes, large scientific sensornets are formed of multiple wireless patches bridged by the Internet, and where individual sensors are deployed by casual users or citizen scientists, with their data posted to sensor web logs or slogs. New search and discovery tools will make this data discoverable by many users. Novel disclosure mechanisms will manage access to potentially confidential data, coupled with data validation to check quality, with means to connect live sensor streams to analysis and visualization tools.

Reaching these goals requires an understanding of how to meld evolving sensornet and Internet architectures. Today they are largely disparate. The Internet is a data sharing engine: With a few clicks anyone can establish a web presence; and the recent popularity of blogging, wikis and other Web 2.0 applications offer new frameworks for contributing information. The Internet exploits computational and communication power ranging from an individual's terminal to distributed clusters owned by large organizations. An essential element of the Internet success is that it has found a balance of data sharing controls and tools that allow individuals and companies each to share, discover, and build on information.

Sensornets differ in each of these aspects. Although sensornets increasingly being fielded by scientists, they still require considerable technical expertise to deploy and maintain. They focus on inexpensive sensor nodes and wireless networks, often connect to the Internet by a custom gateway. They exploit new, highly optimized protocols that allow sensors to operate with tiny memories (10–100kB), slow networks (20–200kb/s), and severe energy limitations. Although often operated by scientists motivated to share the data they collect, no consistent approaches exist today to expose this data for discovery and sharing, and no general method exists for controlling access and managing disclosure of data.

This paper makes two contributions to address these challenges. First, we propose an architecture to support sensornet sharing over the Internet, and illustrate how sensor publishers, data stores, republishers, and sensor search engines interact with each and with users (Section 2). Second, we describe the open research challenges to accomplish this architecture, taking as a case study the sharing of sensor data across research groups and citizen scientists (Section 3). Finally, we present system prototypes that are beginning to explore these ideas (Section 4). Our ultimate goal is sharing: We define open protocols that enable interoperation of many sensor streams, each tailored to specific needs, and key protocols to facilitate safe sharing and discovery of these streams.

IPSN DSI '07 Cambridge, Massachusetts, USA

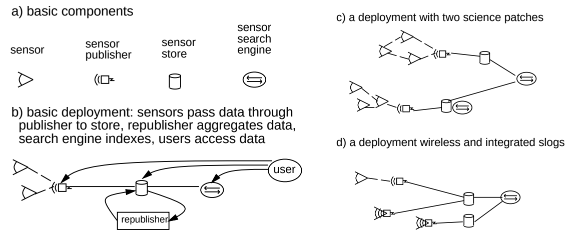

Figure 1. Components and two possible deployments of our sensor/Internet architecture. (Solid lines represent SDSP over Internet links, dashed lines show sensornet-specific wireless links.)

2 Architecture

In this section, we review the basic architectural components in our proposed system: *sensors* (clustered into sensornets), *sensor publishers* that act as gateways, *sensor stores*, and *sensor search engines*, shown in Figure 1(a). Figure 1(b) illustrates how these components might interact, as sensors feed data with custom protocols to a publisher. Publishers send data to the sensor store, where republishers may process the data and sensor search engines index it, all using a common protocol, SDSP (Sensor data stream protocol). Users typically discover data through sensor search engines, then query the sensor stores for the data itself, or access sensor publishers themselves to adjust parameters or gather more detailed data.

2.1 Sensors and Sensor Publishers

At one extreme, *sensors* are small, inexpensive, and often battery operated wireless devices designed to be placed in the environment, physically close to what they need to sense. Both hardware and software have been an area of intense research, and a range of custom protocols and emerging standards [1] and architectures [2, 3] allow efficient use of minimal sensor resources. Sensors are gathered into *sensornets*, clusters of sensors that share a wireless network. Because the sensornet is the most resource constrained component of our system, we want to allow diversity and customization in the wireless protocols here.

Sensor publishers link sensornets to the Internet, providing a gateway from usually non-IP sensornet protocols to Internet standards. They often therefore translate from a resource-impoverished wireless network to much more reliable Internet (often wired, but sometimes wide-area wireless such as CDPD or satellite). Today the gateway is usually custom built to service a specific application. We instead expect sensor-side of the gateway to be optimized for the sensors, but the Internet side to use SDSP.

This is the “traditional” view of sensornets: Sensors are resource-constrained devices separated from the publisher (think motes and Stargates). With sharing as our ultimate goal, we will support a broader view of embedded sensing. In some cases, sensors will be expensive, capable and wired, while in others, sensors might not be physical devices at all. When data is available through other sources, such as through public web sites, one could construct a virtual publisher that scrapes the website and publishes the data to a sensor store in a standard format. We also anticipate the possible merging of sensing and publication into one device. In Figure 1(d), for example, the top sensor publisher might represent a PC with single wireless camera, while the bottom two integrated sensor/publishers might represent cell phones with audio and video capabilities [4].

2.2 Sensor Stores

Sensors stores provide a safe repository for sensor data. As with web servers and file servers, we expect large sensor stores to be generally well connected and managed, yet other small stores will be run by hobbyists or individual research groups. As with blogging, we expect casual users to select a central storage host to outsource administration and backup. We believe it is essential that *many* sensor stores emerge, providing a range of price, availability, and sharing policies.

2.3 Sensor Republishers

One advantage of sharing sensor data in the Internet with a common protocol is that it becomes easy to add value to data by *republishing* it. Examples of republishing include data aggregation, filtering, statistical estimation, vetting, and error suppression. The key to republishing is that, because sensor publishers and stores are all accessed with the same protocol, republishers access data like any other user, and publish data back to the stores like any other sensor. In effect, they are *virtual publishers*.

As an extended example, consider situations where a sensor stream may be incomplete, perhaps due to sensor outage, read errors, miscalibration or loss in transmission. One republisher might repair the data stream by interpolating missing information from prior values or spatially nearby sensors, then adding a confidence rating to the data upon republishing (NOAA provides such services for amateur weather stations). Another republisher looking at the same flaky data stream may choose to simply annotate the stream as less trustworthy. In both cases, these republishers make the new, annotated data available on a sensor store for others to further build upon, and they add links back to the original stream in the metadata of the republished stream, allowing others to investigate the work. Section 4.2 shows two example republishers we are currently developing.

More interesting republishers may merge data from *different* sensornets. For example, consider two data sources, one that provides information about noise levels in an area, and another that provides data about vehicle traffic. The traffic data may come from a government website and be provided to a sensor store via a *virtual sensor* scraping the site, while the noise levels are measured by cell phones and slogged by individuals. A third party can then investigate the hypothesis that noise correlates with traffic by combining these two streams and republishing the correlation over time.

2.4 Sensor Search Engines

Sensor search engines will index and search sensor data. We expect a few, large (Internet-wide) sensor search engines, augmented by local, laboratory-specific engines. Sensor search engines will facilitate the process of new sensor discovery and exploration by general users based on the their particular need expressed by queries. Search engines aid in browsing and discovery of new sensor streams.

As with sensors and publishers, in some cases it may make sense to merge storage and search in a single device. For example, in Figure 1(c), one science project operating the lower sensornet provides their own sensor store and search engine (minimizing communications costs). By contrast, the group operating the top sensornet uses a public sensor store, and a public sensor search engine indexes both stores.

2.5 Users

We expect that users will interact with this architecture in several ways. Search engines will help users discover new data streams that have interesting content. We expect that new approaches for sensor visualization will be needed as well, particularly to visualize temporal and spatial trends in data. For instance dynamically updated charts and graphs (such as Sparklines [5]) can provide views for the data. Furthermore, we will provide data overlays on maps to enable users to analyze data that has spatial variations.

Once users have discovered interesting data streams we expect they will interact directly with sensor stores to retrieve published (and republished) data.

Basic user interaction will typically be through web-based interfaces to sensor search engines and stores. Equally important is local access to data with tools such as Excel, Matlab, and R, so it is essential that stores and search engines support data export in common formats such as CSV or Excel.

More advanced users may implement republishers. Users with appropriate permissions may communicate directly with sensor publishers, and may interact with them to adjust sensing parameters.

Our description above focuses on how the components of our architecture interact with SDSP and use custom wireless protocols at the sensor and standard web protocols for the user. We also expect that some publishers and stores may choose to make data available through custom web pages and ad hoc, textual descriptions of what they are sensing. These ad hoc approaches complement SDSP by making sensors, data sources, and unstructured information about them visible to current, unstructured web search engines.

2.6 Sensor Data Stream Protocol (SDSP)

Cutting across all aspects of our work is the definition of the Sensor Data Stream Protocol, the protocol that will link publishers to sensor stores and search engines. We also see it as being used by user programs that directly access data, complementing interactive, web access.

At its simplest, this protocol allows one to move data from a publisher to a consumer, a role that could be filled with XML-marked-up data over an RSS-like protocol. Yet a full protocol also needs to support bi-directional communication to allow metadata retrieval sensor parameter adjustment, and event-based triggers.

3 Challenges and Research Issues

In order to enable a marketplace for sensor data exchange and support flexible and efficient sharing, there are few different research areas that need to be investigated further. In this section we look at research questions in the following areas: meta-data sharing, sensor system and data search, and sensor stores.

3.1 Meta-Data Definition and Sharing

Common standards for metadata, that is, the data that identifies the types and formats of the data itself, is essential to promote sharing. Unlike the web, where human-targeted, natural language web pages provide context, raw sensor data is meaningless numbers when not annotated by metadata. Even when the data type is indicated (say, temperature), units, calibration and context are important (Celsius, Kelvin, Fahrenheit; precision; indoors or outdoors).

Although XML and related protocols describe the syntax to share data, and common ways to describe the semantics of that data, they leave considerable latitude in execution. We see two significant beyond basic formatting: use of metadata by non-experts, and metadata for enriched services.

An important goal of our work is to enable citizen science, where relatively casual users may contribute meaningful data. This goal implies we will have a wide range of expertise in those who define sensor metadata. We are considering several approaches to reduce this problem: one is to explore simplified means of specifying metadata, for example, providing a pre-packaged set of metadata for common configurations and sensor types. Second is to enable casual annotations of sensors (text and photos) to encourage third parties to detect and report potential deployment problems, and to let them add metadata to the sensors of particular importance to them.

Finally, we are exploring enriched sensor services, where third parties will annotate, filter, and republish data. These services require new kinds of metadata, including sensor confidence ratings, popularity, and reputation, as well as ways to trace back through republished data to their original sources.

3.2 Sensor Search

Finding data can be difficult for scientists today, and when a large number of casual users share their own sensor data without any central coordination, it is crucial to build an easy-to-use search mechanism to help users navigate and explore the sensor data.

Given the diverse need and sophistication of the potential users, we believe a sensor search engine should support at least two search mechanisms: *exploratory* and *analytic queries*.

First, we expect that the user will often have only vague informational need that is difficult to formulate exactly or he may lack the technical expertise to write a precise query. (e.g., how is the LA weather in summer?) In these cases, the search engine should support *exploratory queries* through Web-based GUI and let the user issue simple queries using keywords (e.g., temperature LA) and/or a few conditions on basic attributes (e.g., location“LA” and type“temperature”). Given these queries, the search engine then returns a ranked list of potentially relevant sensors, so that the user can manually look at the sensor values and their metadata. In addition, the search engine should also provide an easy mechanism to “browse” a set of “relevant” sensors (e.g., other types of sensors that are located in LA or the ones whose values show strong correlation to the one returned) to help this exploration process. Supporting this exploratory search mechanism suggests several research challenges. For instance, a user can look at only a limited number of sensors through this manual process, so it is crucial to develop an effective ranking mechanism to find the best sensors. The popularity of a sensor (which can be measured, say, by the number of times that its data is republished by third-party aggregators), the reliability of the sensor values and the richness of its metadata may be useful metrics for this task. In addition, due to the lack of a central coordination, formats and vocabulary used by metadata may not be consistent, so the search engine should be able to map or translate one metadata to another or fill in missing metadata as much as it can to enable consistent searches on multiple sensor data. The large body of work on schema matching and automatic data tagging will be a good starting point to address this challenge.

Second, when the user's need can be expressed precisely (e.g., the average yearly rainfall in LA), at least in principle, an *analytical query language* is more appropriate. For this scenario, modeling each sensor as a relational table and supporting SQL queries on them can be a reasonable approach. From our initial investigation of this approach, however, we find that crucial extensions to SQL are necessary. For example, the SQL standard requires the user explicitly list all tables (or sensors) in the WHERE clause of the query. In our setting, the user is unlikely to know in advance all relevant sensors (and their corresponding tables) since the hundreds of of sensors shared through the Internet. It is also unrealistic to ask users to list all tables manually in the WHERE clause. Therefore, it is important to extend the SQL to let users declaratively specify *tables* of interest (e.g., all tables corresponding to the sensors of the type "temperature" and the location "LA") as they can do for the *tuples* using the standard SQL.[1] We also find that different sensors exhibit wide variation in their quality and reliability, so it is often important to be able to associate a "quality" or "confidence" value to individual sensor reading (automatically or manually by republishers) and to trace this value during query evaluation and result presentation, so that when the user gets an unexpected result, he will be able to trace back to the source of the problem.

3.3 Sensor Stores

Aggregating data for many users, sensor stores pose several unique research challenges. Most important is to support a range of data disclosure models so users are comfortable sharing their data widely. For instance, say a sensor system provides surveillance images, along with additional context information, for a certain building in UCLA. This information is only needed for people in the UCLA community, specifically the security personal. The administrator of the sensor system might not want users outside of the UCLA community to be able to access the data in any way. Furthermore, even in the UCLA community, certain groups might get more access than others. The security group of UCLA might get access to view all the data and even have permission to edit the data itself, but people that are actually in the building might only get access to a summary representation. Thus, there exists an open research challenge to not only come up with the language for disclosure but also to implement the rules in a efficient, secure fashion.

Sensor stores need to provide a simple, yet robust interface to enable both users and other components in the architecture to publish and obtain data. For instance, the interface needs to support sensor search engines. Information such as meta data regarding a specific sensing project, summaries or snapshots of the actual data, and general statistics about a project, such as noise exhibited, trends, and usage statistics, would need to be made available to sensor search engines. We will also explore the potential of sensor store execution of code from the search engines to allow efficient indexing. Also, users must be able to access the information on sensor stores in a defined fashion. One can imagine having browsable pages for each sensing system that provides visualizations of where the sensors actually located and graphs related to the data that is being published. Furthermore, an interface to obtain sensor data, along with other attributes outlined earlier, is needed.

4 Prototype Implementations

We are currently exploring the concepts described above through several prototype systems. Our main prototype system is SensorBase, a sample sensor store that supports a data archival service in production use by a number of projects today. We also are exploring c-SDSP, a simple implementation of our data sharing protocol that is designed to test the portability of our specification across multiple implementations.

4.1 SensorBase

We have implemented prototype of our sensor store as an extended version of SensorBase (SensorBase.org) [6]. Prior to our work, SensorBase was basically a web interface layered directly over a database backend. While it had an application programming interface to log sensor data (slog), the protocol was complicated and data retrieval was done through a Web interface that was not very intuitive.

We are extending SensorBase to address these problems. First, we added two additional mechanisms to use the system: a SOAP and HTTP POST interface. Furthermore, an XML-based data exchange format was created to slog data. We also revamped the user interface, adding details about meta-information regarding a project, summary representation of the data, and integrating project location with Google Maps. Finally, we export notifications about project and data changes. SensorBase is in active use by 52 different projects, several of which slog data on a minute by minute basis.

4.2 Prototypes of Republishing

We are experimenting with republishing through nest box monitoring and LA Basin temperature collection [7]. The scientific goal of the Nestbox project at UCLA is to monitor the nesting habits of birds. To do this, an imager and a set of sensors monitor light conditions, humidity, and temperature inside bird nest boxes at James Reserve. The sensor readings and images are uploaded every fifteen minutes to SensorBase from these boxes. An external program then pulls these images from SensorBase, allows users to annotate the data by indicating the presence of a bird or the number of bird eggs, then republishes the tags back to SensorBase. Republishing allows us to further automate this process. For instance, we are planning to use image analysis to extract features from the images and compute the probability the nest is occupied by a bird. By republishing a probability of bird presence, human interpretation of the images can be focused on borderline images where image features cannot make a precise decision on the state of the nest box. Figure 2 contains examples NestBox images. In the left and right images the state of the nest

---

[1] Alternatively, to address this issue, we may use a single universal table to keep all data from all sensors. Due to the heterogeneity of the sensor data, this approach raises another set of interesting challenges.

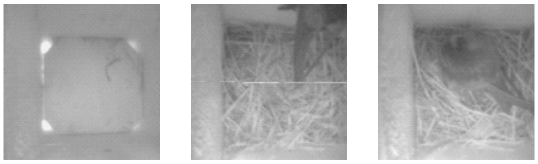

Figure 2. Example Nestbox images, showing when a bird is absent (left), present (right), and an indeterminate case (middle) as the bird leaves the box.

raw	corrected
87.1	87.1 ±0.1
?7.1	87.1 ±0.1
87.?	87.5 ±0.5

Figure 3. Sample captured temperature sensor image (left), with raw and potentially republished corrected data streams (right). (Data is temperature located on the right bottom of the LCD from 3:22pm to 3:32pm, Feb. 18, 2007.)

box is clear, while the state of the middle image is ambiguous. Both the automatic and human interpreted state decisions will be republished back to the common source.

As another example of republishing, we are currently slogging temperature data from several consumer-grade temperature sensors in the LA basin. Computer-attached, all-weather temperature sensors are quite expensive ($100 or more), and a mote-based solution would be more still. Instead, we point a $10 consumer web camera at the LCD display of a $20 consumer-grade temperature sensor, and automatically determine the temperature by processing the images (see Figure 3). While inefficient, we see this approach as leveraging very cheap, commodity hardware and repurposing pre-existing, capable platforms such as PCs and cell phones in order to bring slogging to casual users.

Republishing can play a roll in this process because image-based recognition often has digit-level identification errors. Because temperature is slowly changing, a republisher often could repair errors in high-order digits by inferring them from a prior reading. Alternatively, for low-order digits, a republisher could adjust the known level of precision.

5 Related Work

Related work includes approaches to share data over the Internet, interchange formats, and searching non-textual data.

5.1 Sensor Data Sharing Over the Internet

Much of our inspiration comes from the success of Really Simple Syndication (RSS) and Atom in building the blogging community. These protocols provide XML formats that describe new entries in web blogs, publishing of metadata regarding change in the content of an information source, and aggregating of information from various sources [8, 9].

Several researchers have generalized RSS to sensor data. Simple Sensor Syndication places sensor data over RSS and has shown how users can act in response to these feeds [10]. Numeric Really Simple Syndication (NRSS) extends the ideas of RSS and Atom for syndicating numerical data over the web [11]. It includes fields to provide context and format information regarding the numerical data. Both of these systems capture data sharing, but we strive to go further by formalizing metadata exchange and building search and sharing mechanisms on top.

Several research efforts are exploring how to provide sensor data over the Internet: ArchRock [12], Sensor Web Enablement, and closest to our work, SensorMap [13, 14], IrisNet [15] and GSN [16].

The ArchRock edge server provides a web front end for a mote-based sensornet [12]. In this system, data can be retrieved via web pages and SOAP-based services; data is read from motes and stored at the gateway. While they provide very capable front-end for a single sensornet, their system does not address issues in data sharing or discovery across different organizations or sensornet patches.

Sensor Web Enablement considers enabling web-resident devices. It includes SensorML [17], TranducerML [18], and observation services for managing sensors and retrieving sensor data, planning services for tasking sensors, notification services, and alert services. This work provides well structure metadata for real-world sensors intended for professionally used data collection. They do not describe mechanisms for search mechanisms (beyond basic data retrieval), or approaches to republish processed or aggregated data.

SensorMap focuses on building a portal that shows real-time sensor data on a map [13]. Their primary contribution is their user interface that adjusts the level of sensor detail to fit map scale. This user interface seems ideal for spatial data, but we plan to also explore non-spatial queries over data. Furthermore, they propose having the publishers represent their data as visual components such as points, lines, regions, and images. Having this UI-centric data representation limits aggregation and analysis methods that can be applied, but enables easy mash-ups of data sources. In addition, SensorMap employs a web crawler to discover new sensor streams from HTML-based keywords. Closely related is the MSRSense toolkit, which streams data from a sensornet to an XML database and into Microsoft Excel for visualization and processing [14]. Both SensorMap and MSRSense provide powerful visualization tools; they leave most questions about access control sharing to the sensornet gateway. SensorMap's discovery process is ideal for web-based sensors; we seek to augment it with sensor-specific approaches that can exploit richer metadata and blog-style linking.

IrisNet considers Internet-side storage of XML-tagged data from PC-connected sensors [15]. It uses a hierarchy of XML databases to enable search over the XML fields. They recognize the need for distributed administration of storage but do not provide a solution. GSN is also middleware design to integrate heterogeneous sensor networks [16]. GSN stores sensor data at *GSN nodes* that gateway and store sensor data. A peer-to-peer network indexes sensor data, allowing efficient discovery of GSN nodes based on data type and range. Like our work, both of these systems are developing protocols for sharing sensor data over the Internet. However we focus on sharing data in sensor stores that can be common to many different users, and on a general approach to data processing and republishing.

5.2 Sensor Data Interchange Formats

Several XML schemes have been proposed to manage sensor data. The Open Geospatial Consortium is developing a SensorML [17]. It captures both the geospatial properties of sensors, as well as sensor-specific metadata such as accuracy and manufacturer, and includes some support for simple data transformations like unit or voltage conversion. TinyML [19] and SDML [13] are alternative sensor data exchange formats that are much more lightweight than SensorML. TinyML targets embedded sensor networks specifically, and SDML focuses on the interface to sensor data. We expect to build upon this prior work for sensor data representation.

5.3 Search of Sensor Data

The sensor search component has been inspired by the general search engines on the Web, which is based on traditional information retrieval techniques [20] with the hypertext-specific enhancements [21, 22]. For an overview of the technologies behind Web search engines, see [23].

There also exist a large body of related work in the database community, ranging from supporting SQL queries on sensors [24] to efficient algorithms for querying temporal data [25]. For example, TinyDB [24] uses SQL to manage the data from a sensor network. Again, our work has been inspired by these but we are addressing the new challenges arising from a large number of uncoordinated and heterogeneous sensors.

6 Conclusion

As sensornet deployments increase, we must find new ways to share and process this information. We have described the approach we are exploring: common elements to share, search, and reprocess sensornet data in the Internet. We do not expect to build the one perfect sensor system, because there is no one such system. Instead, we are defining open standards and services that enable interoperation of many sensor streams, both across separate deployments of a single group, and across independent groups, and ultimately thousands of citizen scientists.

7 Acknowledgements

This work is supported by National Science Foundation (NSF) grants CNS-0626702, “Sensor-Internet Sharing and Search”, and CNS-0627084, “Network Fabric for Personal, Social, and Urban Sensing Applications”.

8 References

[1] IEEE. Ieee standard 802.15.4: Wireless medium access control (MAC) and physical layer (PHY) specifications for low-rate wireless personal area networks (LR-WPANs). Technical Report IEEE 802.15.4-2003, IEEE, 1998.

[2] Joseph Polastre, Jonathan Hui, Philip Levis, Jerry Zhao, David Culler, Scott Shenker, and Ion Stoica. A unifying link absraction for wireless sensor networks. In *Proceedings of the Third ACM SenSys Conference*, pages 76–89, San Diego, California, USA, November 2005. ACM.

[3] Ramesh Govindan, Eddie Kohler, Deborah Estrin, Fang Bian, Krishna Chintalapudi, Om Gnawali, Sumit Rangwala, Ramakrishna Gummadi, and Thanos Stathopoulos. Tenet: An architecture for tiered embedded networks. Technical Report 56, University of California, Los Angeles, Center for Embedded Networked Sensing, November 2005.

[4] D. Siewiorek, A. Smailagic, J. Furukawa, A. Krause, N. Moraveji, K. Reiger, and J. Shaffer. SenSay: a context-aware mobile phone. *Wearable Computers, 2003. Proceedings. Seventh IEEE International Symposium on*, pages 248–249, 2003.

[5] James Byers. Sparkline PHP Graphing. http://sparkline.org/, 2006.

[6] D. Estrin, R. Guy, K. Chang, M. Hansen, , and E. Graham. Sensorbase. http://www.sensorbase.org, 2006.

[7] CENS. CENS 2006 Annual Report. http://research.cens.ucla.-edu/, 2006.

[8] Dave Winer. RSS 2.0 Specification. http://blogs.law.harvard.-edu/tech/rss, 2002.

[9] Wikipedia. Atom Syndication Schema. http://wikipedia.-com/atom, 2006.

[10] M. Colagrosso, W. Simmons, and M. Graham. Demo abstract: Simple sensor syndication. In *SenSys '06: Proceedings of the 4th international conference on Embedded networked sensor systems*, 2006.

[11] Agilent Labratories. Numerical RSS: A protocol for Syndicating Numeric Data. http://nrss.org, 2005.

[12] Alec Woo. Demo abstract: A new embedded web services approach to wireless sensor networks. In *Proceedings of the Fourth ACM SenSys Conference*, page 347, Boulder, Colorado, USA, November 2006. ACM.

[13] Suman Nath, Jie Liu, and Feng Zhao. Challenges in building a portal for sensors world-wide. In *First Workshop on World-Sensor-Web*, Boulder,CO, October 2006. ACM.

[14] A. Woo, S. Seth, T. Olson, J. Liu, and F. Zhao. A spreadsheet approach to programming and managing sensor networks. *Proc. of the Fifth Int. Conf. on Info. Processing in Sensor Networks*, pages 424–431, 2006.

[15] Phillip B. Gibbons, Brad Karp, Yan Ke, Suman Nath, and Srinivasan Seshan. Irisnet: An architecture for a worldwide sensor web. *IEEE Pervasive Computing*, 02(4):22–33, 2003.

[16] Karl Aberer, Manfred Hauswirth, and Ali Salehi. A middleware for fast and flexible sensor network deployment. In *VLDB*, pages 1199–1202, 2006.

[17] Mike Botts, editor. *OpenGIS Sensor Model Language (SensorML), Implementation Specification.*

[18] IRIS Corporation. Transducer Markup Language. http://www.-transducerml.org.

[19] N. Ota and W.T.C. Kramer. TinyML: Meta-data for Wireless Networks.

[20] Gerard Salton and Michael J. McGill. *Introduction to modern information retrieval*. McGraw-Hill, 1983.

[21] Sergey Brin and Lawrence Page. The anatomy of a large-scale hypertextual web search engine. In *Proceedings of the International World-Wide Web Conference*, Brisbane, Australia, April 1998.

[22] J. Kleinberg. Authoritative sources in a hyperlinked environment. *Journal of the ACM*, 46, 1999.

[23] Ricardo A. Baeza-Yates and Berthier A. Ribeiro-Neto. *Modern Information Retrieval*. ACM Press / Addison-Wesley, 1999.

[24] Samuel R. Madden, Michael J. Franklin, Joseph M. Hellerstein, and Wei Hong. Tinydb: an acquisitional query processing system for sensor networks. *ACM Trans. Database Syst.*, 30(1):122–173, 2005.

[25] Reza Sadri, Carlo Zaniolo, Amir Zarkesh, and Jafar Adibi. Optimization of sequence queries in database systems. In *Proceedings of the ACM SIGACT-SIGMOD-SIGART Symposium on Principles of Database Systems (PODS)*, Santa Barbara, California, May 2001.