Sensors meet World Wide Web: searching and organizing world’s real-time information

Feng Zhao
Microsoft Research
http://research.microsoft.com/zhao

Keynote at IEEE International Conference on Sensor Networks, Ubiquitous and Trustworthy Computing (SUTC’06), Taiwan, June 5, 2006
Challenges for the Sensornet Community

• Great to see startups begin to tackle practical problems, collecting low-hanging fruits

• But, the research community must set the sight further
  – Every field needs grand challenges to focus energy and drive progress, so does sensornet
  – Here is my take on the grand challenges; you can substitute it with your favorite …
Ubiquitous Sensing Infrastructure and Reality Browser:
Query physical world, live and up close, from anywhere
Potential Apps

• Environment
  – Volcano, underwater, rainforest

• Education
  – K-12 interactive learning, virtual laboratory

• Leisure
  – Virtual travel, sunny spot tracker (for Seattle!), what is the temperature at my favorite beach? what is the water algae level?

• Getting around
  – Where is the nearest available parking space? What is the traffic like on the bridge? How long is the queue at the gas station? Where is the bus?
SenseWeb

- Goal: Simplifying publishing live data on Web and querying that data (from sensors)

SenseWeb
(Indexing, query processing, caching, scalability, etc.)

- Generic functionalities implemented
  - Programmable

MSRSense

Query response

Indexing live data
Processing queries on the data

http://research.microsoft.com/senseweb
Monday, May 88, 2006

Microsoft's Plan to Map the World in Real Time
Researchers are working on a system that allows sensors to track information and create up-to-date, searchable online maps.

By Kate Greene

Researchers at Microsoft are working on technology that they hope will someday enable people to browse online maps for up-to-the-minute information about local gas prices, traffic flows, restaurant wait times, and more. Eventually, says Sumant NATH, a Microsoft researcher who works on the project, which is called SenseWeb, they would like to incorporate the technology into Windows Live Local (formerly Microsoft Virtual Earth), the company's online mapping platform.

By tracking real-time conditions, which are supplied directly by people or automated sensor equipment, and correlating that data with a searchable map, people could have a better idea of the activities going on in their local areas, says Nath, and make more informed decisions about, for instance, what driving route to take.

"The value that you get out of [real-time data] ranges from finding a parking spot in a cavernous parking garage to finding out if a restaurant has a long wait," Nath says.

News articles on our SenseWeb:

• Microsoft's Plan to Map the World in Real Time, Technology Review, 5/8/06
• Adding Real-Time Info to Local Searchers, InternewNews.com, 5/11/06
• Microsoft Research showcases future technology, EETimes, InformationWeek, CRN, 5/3/06
• Microsoft Keeping Web Arsenal Well-Stocked, TheStreet.com, 5/3/06
• 135,000 hits on “SenseWeb” and “Microsoft” with Google search
Multi-tier net of sensornets, coupled with Internet

Characteristics:
- Rich data sources: sensor, image, GIS, …
- Steaming data, live, historical or processed
- Concurrent, uncoordinated queries
- Uncertainty/noise in data
- Semantic interpretation often needed
Three Challenge Problems

• Need to standardize **interfaces** between apps/db/nw/dsp/hw (e.g., SP link layer abstraction, SensorML)

• Need to treat **uncertainty** (in both data and systems) as first-class entity, for reasoning and system mgmt

• Need (simple) **tools** for sys config/mgmt, for data collection and vis, for in-situ debugging
But the problem is made harder by …

• A great deal of variability
  – Variability in app requirements
    • Use scenarios (data collection, control in the loop, or in-network compression), data rate, hw
  – System dynamism and unreliability
    • Nodes/links come and go
    • Parts of system may be deployed over time by different vendors, using different technologies (e.g., network protocols)
    • Possibly non-replenishable resources
  – Variability in data
    • Uncertain data due to sensor noise, packet loss
    • Incomplete information due to partial observability of the world

• User tasks often specified in high-level semantic queries
  – E.g., “Tell me if you see a red car”, or, “Doing this with that data”.

• As a result, systems are often built vertically
  – With own system abstractions and data models
  – Make sense from efficiency POV, but with relatively small code reuse
  – Phil Buonadonna: “Every sensor net experiment needs 2 PhDs and 5 grad students”, or something like that
A brief history of sensornet

- 1980: DARPA DSN (PM: Bob Kahn) – sensors are truck-sized, connected via Ethernet using microwave radios.
- 1994: Smart dust (Kris Pister)
- 1994-98: UCLA WINS, Xerox PARC Smart Matter
- 1999-2004: DARPA SensIT (USC/ISI, Cornell, Xerox PARC, BBN, BAE, UCLA, Penn State, Wisconsin, UIUC, MIT, Berkeley, LSU/Tennessee)
- Berkeley motes; Intel Berkeley Lablet
- Startups: Crossbow, Ember, Dust, Sensicast …
- 2001-2005: DARPA NEST (Berkeley TinyOS, Ohio State, UVA, …)
- 2003-Now: NSF NETS/NOSS
- Industrial R&D: Agilent, HP, Intel, IBM, Microsoft, Motorola, Nokia, Sun, Xerox PARC, …
Many systems got built over the past few years …

Where is the narrow waist?
From printer land, Mojave Desert, to parking garage … our own experiences

MSR parking garage testbed, June 2004

Distributed Attention

SONGS

PARC camera testbed, 2003

Collaboration groups

MSR in-building testbed, 2005

Machine diagnostics, 1997

DARPA SensIT experiments, 29 Palms, 2000-2001

Outdoor experiment, 2002
What we’ve learned …

• Proposed tracking as a canonical problem for sensornet problem (circa 2000)
  – Dynamic data, close the loop between sensing & decision making (not just data collection)
  – Created SensIT data sets (with ground truth!), app scenarios, experimental testbed (used by others)
  – Many projects since then: SensIT 29P demo, NEST demo, EnviroTrack, …
  – Lesson: a successful deployment is a function of many pieces that work together - dsp, networking, db, debugging tools, …

• Developed the IDSQ (Information Driven Sensor Querying) framework/Sensor tasking (circa 2001)
  – Pay more attention to data: value of information
  – Interface btw app ↔ routing: value of info and routing decision at each node
  – Move beyond individual nodes: group/neighborhood management
    • Many other work in this space: Hood, Regions, …
  – But ours (and others) were built with its own interface assumptions and data representation
More on Lessons Learned

OS/Network-centric view

Designing component-level abstractions

Information-centric view

Designing application-level abstractions

- algorithms
- levels of refinement

common interfaces

Majority of the code is in glue logic

Macroprogramming (TinyDB, Regions, Kairos, State-space, EnviroTrack …)
A Scenario: What’s happening in the garage?

Multiple, concurrent queries:

- Traffic engineering: where can one find a parking spot?
- Security: what is going on down there?
- Corporate health services: when should the air exhaust fan run?
A Parking Garage Example

**Sense**
- Speed, length, and direction of vehicles
- Magnetic signature of vehicles
- Image of vehicles

- Break beam sensors
- Camera
- Magnetometer
- Micro-server
- User interface
Tasks are sent to microservers at uncoordinated times, running for unpredictable duration. Tasks may partially overlap.
Interfaces Between Apps and Run-Time

- **Declarative queries**
- **planner**
- **Excel**
- **config scripting**
- **service lib**
- **Task Pool**
- **sensor tasking**
- **optimizer 1**
- **optimizer 2**
- **run time monitoring**
- **System Run Time Support**
- **Tasking ML**
- **concrete services**

**abstract services, dependencies, priorities, E-E constraints**

**link reliability, net latency, data fidelity**

**Sensor Net Deployment**
Example of services and their composition

**Counting vehicles with a sensor array**

- Extract edges from break beam detections
- Sort edges into consecutive detections
- Detect vehicles based on timing relations among detections
- Count vehicles
- Generate an arrival histogram report
Another Example: where is the elevator?
SONGS: Service Oriented Networked ProGramming of Sensors

Service Abstraction and Interface

Service Planning

histogram(S) → Car(X) → speed(X,S)

Service Embedding

histogram(S) → Car(X) → speed(X,S)

Service Scheduling and Execution

car detection service

Car

Sensor 1

Sensor 2

Histogram

Speed

{x, y, z}

{S>30}

{S>30}
SenseWeb Architecture

- Fine-grain, modular services as building blocks
  - Library for common tasks: data cleaning, storage, aggregation, query processing, …
- Aggregated services built upon these fine-grain services
  - Simple, unified programming abstraction
- Extensible, friendly to 3rd party services
Sensor Ontology

- OWL sensor ontology exposes the semantics associated with sensor types, groups and relationships
- The ontology provides high-level interoperability at syntactic and semantic levels
GeoDB

• Processes queries on geographic data
  – E.g., cameras inside a region, near a route
• Sensors register themselves
  – Single rendezvous point for sensors and users
• Data indexed with Hierarchical Triangular Mesh (HTM)
  – Proximity-preserving 1D mapping of Lat/Long
  – Implemented as table valued functions in SQL Server 2005 +
    external library
• Simple APIs and Web Service interface to store, modify, and query the database
Aggregator (IconD)

- Aggregates sensor data, produces icons
- Aggregation at different granularities
  - Dynamically, e.g., based on zoom levels
Client-side GUI library

Java-script library to develop web-based GUI with more controls to interact with data
MSRSense: A toolkit for data publishing, archiving, visualization

- **User Interface / Data Processing** (Excel viz)
- **Sensor Net** (Tmote Sky)
- **Gateway** (MicroServer data publishing web service)
- **Archiving Events**
  - **Database** (SQL Server 2005 or Express)
  - **SQL Query / Report**
  - **Raw data** (XML packets)
- **Publishing Data**
  - **Packet Stream Player**
  - **Transformed XML**
- **Visualize Events/ Process Data**
Software release

- MSRSense toolkit available (in source code) from http://research.microsoft.com/nec/msrsense
  - Over 5,000 downloads since the 12/14/05 announcement on TinyOS mailing list

- SenseWeb portal will be publicly available in July, 2006
  - http://research.microsoft.com/senseweb
  - With webcam, traffic, weather, city street parking, soil, lake data being added right now
To probe further

• Papers from our group:
  • “Semantic Streams: a Framework for Composable Semantic Interpretation of Sensor Data.” EWSN’06.
  • "A Spreadsheet Approach to Programming and Managing Sensor Networks." IPSN SPOTS’06.
  • “Kinetically Stable Task Assignment for Networks of Microservers.” IPSN’06.

• A few related conferences/Journals
  • ACM Sensys06, Boulder, Nov 2006
  • ACM/IEEE IPSN07, MIT/Boston, April 2007
  • EmNets, DCOSS, EWSN, SECON, …

• MSR Project: http://research.microsoft.com/nec
Wrap up: Challenge Problems

• As a community, we need grand challenge problems

• Three focus areas:
  – A community effort to standardize interfaces between various sensornet components/layers
  – Build models for uncertainty that can be utilized by data and system management
  – Create more tools, for sys config/mgmt, data collection and vis, debugging
Acknowledgment

• Benefited from discussions with my colleagues at MSR

• Joint work with Jie Liu, Suman Nath, Bodhi Priyantha, Michel Goraczko, Alec Woo, Tim Olson, Leo Guibas, Jeremy Elson, Kamin Whitehouse, Elaine Cheong, Prabal Dutta, Zoe Abrams, Siddharth Seth, Ryan Newton, Andrew Parker