

Research Article

A Three-Tier Architecture for User-Centric Ubiquitous Networked Sensing

Jin Nakazawa and Hideyuki Tokuda

Faculty of Environment and Information Studies, Keio University, 5322 Endo, Fujisawa-shi, Kanagawa 252-0882, Japan

Correspondence should be addressed to Hideyuki Tokuda, hxt@ht.sfc.keio.ac.jp

Received 13 February 2012; Accepted 18 March 2012

Academic Editors: D. Cassioli, A. Maaref, and Y. M. Tseng

Copyright © 2012 J. Nakazawa and H. Tokuda. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

In a sensor network, sensor data are usually forwarded from sensor nodes to a database. This tight coupling between the nodes and the database has been complicating user-centric applications that traverse multiple different sensor networks. To break this coupling, thus enabling user-centric applications, we propose a three-tier architecture for ubiquitous networked sensing. Its major feature is that it contains the “core” device, which is assumed to be a terminal held by users between sensor nodes and sensor databases. This architecture supports the sensor data directly transmitted to and consumed by the core device, in addition to the classic ones that are transmitted to the sensor database first, and downloaded to the core. The major contribution of this paper are the following three-fold. First, we clarify the architecture itself. Researchers can leverage the architecture as the baseline of their development. Second, we show two types of prototype implementations of the core device. Industry is allowed to develop a new product for practical use of ambient sensing. Finally, we show a range of applications that are enabled by the architecture and indicate issues that need to be addressed for further investigation.

1. Introduction

The recent research and productization of wireless sensor nodes have been enabling ubiquitous networked sensing environment where sensor nodes are densely embedded around users in homes, offices, parks, roads, and so forth. For example, home owners would manage their own sensor networks [1]. A university campus can install its own campus sensing network [2]. Opposing to the node-side, technologies towards sophisticated sensor database have been deeply investigated. Usually, sensor data are directly forwarded from sensor nodes to a database. In the above examples, there would be a sensor database at the home and the campus to store the data captured there. These two sides, sensor nodes and sensor databases, thus form a tightly coupled networked sensing architecture.

This tight coupling has been complicating user-centric applications that traverse multiple different sensor networks. For example, suppose an application that records aerial pollution in the places where its user visits. Since aerial sensors are not small, it is not practical to assume that the user carries

the sensors. This application thus needs to acquire data from the aerial sensors around the user. With the classic tightly coupled architecture, the application is required to identify the databases where the aerial pollution data are stored and query for the particular pieces of data based on the time and the user's location. Observing that the user and the sensor nodes are close to each other, the above indirect data acquisition is inefficient. Therefore, we need a new architecture that support users leveraging in situ sensor applications more efficiently.

To break the tight coupling in the existing architecture, thus enabling user-centric in situ applications, we propose a three-tier architecture for ubiquitous networked sensing. The major feature of the architecture is that it contains the “core” device, which is assumed to be a terminal held by users, between sensor nodes and sensor databases. This architecture support the sensor data directly transmitted to and consumed by the core device, in addition to the classic ones that are transmitted to the sensor database first, and downloaded to the core. This paper contributes to the field of ubiquitous networked sensing by the following three-fold.

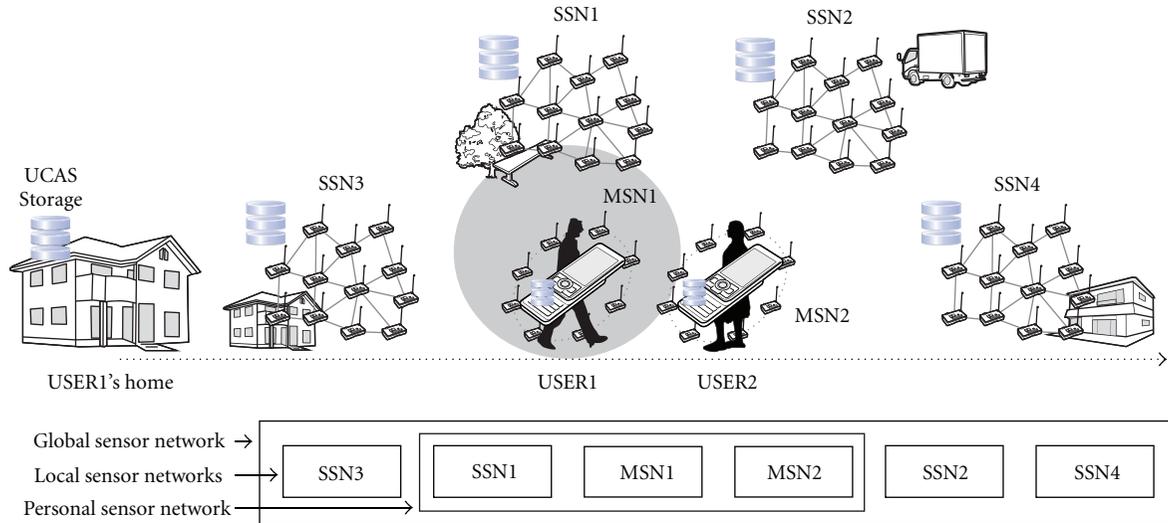


FIGURE 1: UNETS environment: federated sensor networks. Sensor networks are installed everywhere, and users go through them with their own body-worn sensor network and the core device. Each sensor network has its own sensor database to which the data are accumulated. This picture contains four networks of static sensor nodes (SSNs) and two of mobile sensor nodes (MSNs). The user *USER1* walks through this environment collecting the sensor data from surrounding nodes in his handheld device. After a series of activities, he goes back home where his permanent storage is located. The data are transferred from the core to the storage and accumulated there.

First, we clarify the architecture itself. Researchers can leverage the architecture as the baseline of their development. Second, we show two types of prototype implementations of the core device. Industry is allowed to develop a new product for practical use of ubiquitous networked sensing. Finally, we show a range of applications that are enabled by the architecture and indicate issues that needs to be addressed for further investigation.

The rest of this paper is organized as follows. The next section shows the architecture. Section 3 describes the prototyping studies of the core devices and their applications, and Section 4 discusses lessons learnt from the prototyping studies indicating open issues for further investigation. Section 5 surveys related work, and finally Section 6 concludes the paper.

2. Ubiquitous Networked Sensing Architecture

The purpose of defining the architecture is to make a common view of ubiquitous networked sensing, to which one can ground his/her technology. The architecture thus should be common enough to accommodate different aspects of ubiquitous networked sensing. In this section, we first observe the activities of users and the environment we assume, depict the architecture overarching those aspects, then describe components included in the architecture.

2.1. Federated Sensor Networks. We assume in this paper federated sensor networks, depicted in Figure 1, where users are surrounded by static sensor networks installed in many places and the mobile sensor network they carry. The total network view is a three-tier one. First, static and mobile sensor networks form their own local network where a sensor database resides to store the data. Second, the nodes

around a user form a personal sensor network from the user's perspective, whose sink node is the handheld device carried by the user. This network is created in an ad hoc manner. Third, the global sensor network is the collection of sensor databases in the world. They are accessed from applications (not depicted) via the Internet.

Advancement of the research on sensor networks has been enabling such an environment. Urban sensing [3, 4] has been investigated aiming at fine-grained environment monitoring in cities. Sensor networks for generic environment monitoring is also widely researched. AiryNotes [5] is our past project, where we installed more than 200 sensor nodes in Shinjuku Gyoen Park. SenseWeb [6] is aiming at sharing sensor data captured by the sensor nodes embedded in cities to monitor aerial pollution, car traffic, and so forth. Habitat monitoring [7] is used to monitor animals invading farms. Complementing these outdoor sensor networks are indoor sensor networks. SensorAndrew [2] aims at creating a sensed university campus. It monitors buildings, social infrastructures, and students to provide context-aware ubiquitous computing services. AwareHome [8] and PlaceLab [1] are the two major examples of building context-aware homes [9] using sensor networks. Their prototype homes are filled with various types of sensors that are used to capture user activity there. Structural health monitoring [10, 11] protects buildings from being left unwarned when they are damaged due to, for example, an earthquake. Many other places are also targeted such as mountain [12], under water [13], and in a volcano [14]. Observing those places enhanced with a rich sensing capability, users will be surrounded by sensor networks in the future in almost all the places they visit.

Users themselves are also enhanced with sensing capability. Mobile phone sensing [15] leverages the phone carried by users to provide context-aware services on the phone. For

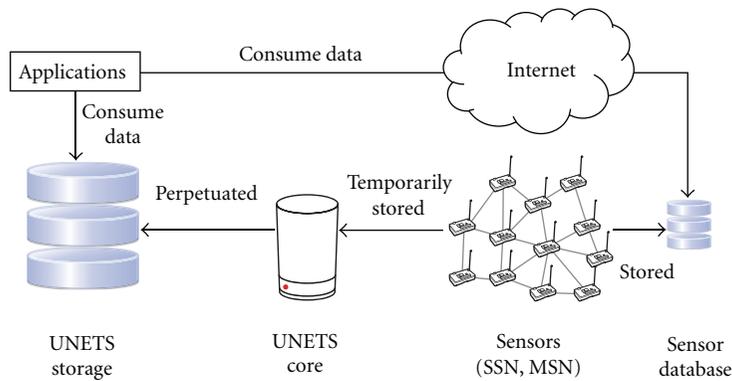


FIGURE 2: UNETS architecture components and data flow.

example, the phone can recognize whether its user is walking, running, sitting, and so forth using an embedded accelerometer. The phone can provide a pedometer capability using this context. Body-area sensor networks [16] is used typically to monitor users' personal health. Thermometer, heart rate meter, and/or tonometer would be able to be networked to the user's cellular phone to record their data, while the user is in an activity. These mobile sensors are also the constituents of the federated sensor networks.

2.2. UNETS Architecture Components. Based on this observation, we describe the user-centric ubiquitous networked sensing (UNETS) architecture from the perspectives of components, interactions, and applications. Figure 2 overviews the components in the architecture and the data flow among them. The users in the UNETS architecture are involved with mobile sensor nodes that are carried by the users, and static ones around them. The data generated by these nodes are first received by the mobile device (UNETS core) carried by the users for temporal storage, and then perpetuated into the user's sensor database (UNETS storage). This sensors—temporal storage—permanent storage configuration constitutes the three-tier architecture.

2.2.1. UNETS Core. A UNETS core is the device that provides sensing (optional), processing, and actuation (optional) capabilities. Readers can suppose, as a virtual example of UNETS core, a cellular phone with an additional network interface (e.g., ZigBee) to communicate with sensor nodes as depicted in Figure 1. It receives the sensor data from the mobile and static nodes and processes them. Processing includes calculating statistics of data and storing the raw and/or calculated data to its internal storage. As the sensing capability, UNETS core can contain its own sensor device like an accelerometer and a GPS receiver on which an application depends to recognize user behavior. It can also contain an LCD, a speaker, and so forth as the actuation capability. Applications on the UNETS core with the actuation capability can provide context-aware digital services using this capability.

2.2.2. Mobile Sensor Nodes (MSNs). MSNs are carried by the user, forming a body-area sensor network with the UNETS

core. Biometric sensors like body temperature sensors and heart rate sensors are their typical examples. Sensors like an accelerometer and a GPS receiver are classified into this class when a user carries them individually instead of having them in his/her UNETS core. A user's mobile sensor nodes can be public or private. Public ones are the sensor nodes that are allowed by the user to serve the data for other users, while private ones are those not. This means a user's UNETS core (e.g., USER1's UNETS core in Figure 1) would be able to receive the data from the public mobile sensor nodes (e.g., MSN2) owned by other users within the UNETS core's radio vicinity (depicted as a gray circle in the figure).

2.2.3. Static Sensor Nodes (SSNs). Immobile sensor nodes, such as those included in sensor networks for campus sensing, environment monitoring at a park, and urban sensing, are classified into this class. Usually, sensor networks are installed and managed by different parties other than the users themselves. This means that the data captured in a sensor network are stored in an individual sensor database in the corresponding facility (campus, park, and city). The immediate SSNs are within a user's vicinity in that the user's UNETS core can receive data via single- or multi-hop communications assuming that they are public. The remote SSNs are outside it; the user's UNETS core would receive the data by referring to the sensor database where the data captured by the remote SSN are stored. Therefore, a sensor node is an immediate for a user close to it, and at the same time a remote one for users far from them. In Figure 1, SSN3 are immediate for USER1 when he is at location 2 and remote for USER2.

2.2.4. UNETS Storage. A UNETS storage serves as a user's personal sensor database where the data collected by his/her UNETS core are transferred and kept. We assume that the storage in a UNETS core is relatively smaller than that in a UNETS storage, since the former is a mobile device with limited computational resource while the latter is static and resource-rich.

2.3. Interactions in UNETS Architecture. The above components interact with one another to exchange sensor data. We analyze the interaction focusing on data flow from the UNETS core's perspective.

2.3.1. Core—MSN. A UNETS core is the sink node for its user's MSN. It receives the current sensor data from his/her own MSN via single- or multi-hop communications. It may also receive the current sensor data from the public ones among the MSN owned by other users when applications running on the UNETS core require to. For example, suppose a toy application that calculates average heart rate of the people around the user. To do this, the UNETS core needs to find the MSN to receive the data from. This can be achieved by either or both of the following two schemes. In both cases, the MSN must be publicized by their owner. First is search basis. This proactive scheme requires the UNETS core to inquire within the network for the MSN that can serve the sensor data of the UNETS core's interest (current heart rate). If there are such MSN, they respond to the inquiry with the data. Second is broadcast basis. Publicized MSN may transmit sensor data periodically via a broadcast channel. The UNETS core receives these broadcast data reactively, filter them, and consume the data of its interest. Typically, the sensor data from publicized MSN are not secured by, for example, encryption, thus can be used by any UNETS core.

2.3.2. Core—SSN. Similar to the above case is the interaction between a UNETS core and immediate SSN. Based on the interest to particular type of sensor data (e.g., temperature and humidity of surrounding environment), a UNETS core locates the nodes that satisfy it. Proactive and reactive schemes can again be used to fulfill this requirement. The architecture assumes that a sensor data dissemination path from SSN to a UNETS core are established by a routing protocol that supports mobile sink nodes, such as [17, 18].

On the other hand, there is no single- or multi-hop communication path between a UNETS core and remote SSN. Therefore, the UNETS core needs to acquire the current sensor data generated by remote SSN indirectly via the Internet. The data are first stored in the sensor database associated to the remote SSN network (e.g., a campus sensing database). The UNETS core then queries for the data to the database via the Internet by using mechanisms for large-scale or global sensor network, such as [19], can be used.

2.3.3. Core—Storage. A user's UNETS core should hold a reasonable amount of sensor data after the user's activities in the UNETS environment. Since the computational resource of the UNETS core is limited, the data should be externalized for permanent storage. Therefore, the UNETS core transfers the data to the user's UNETS storage, and the data are accumulated there to enable applications to refer to the history of sensor data.

While the core—MSN and core—SSN interactions are for collecting the current sensor data, the past data are also important for applications. UNETS core, similarly to the indirect interaction with remote SSN, receives past data of any sensor node from the database where those data are stored. If the UNETS core needs to consume the data generated by the MSN of another user, it queries for the data to the UNETS storage owned by the user. If it needs data generated by SSN, it retrieves them from their sensor database. The

architecture assumes that there exists a mechanism to share sensor data via the Internet such as [6, 20].

2.4. UNETS Applications. We classify the user applications in the architecture from the orthogonal perspectives of location (*here or there: HoT*) and time (*current or past: CoP*). The HoT basis means where the data consumed by the users come from. *Here* indicates that the users acquire the data from the sensor nodes where they reside(d), and *there* indicates that they do from those from remote nodes. For example, sensors here (SH) would be the source for the users to know the degree of aerial pollution around them, while sensors there (ST) would serve as the source for the road traffic information. The CoP basis on the other hand means when the data consumed by the users are captured. *Current* represents that the users acquire the latest data from sensor nodes, and *past* indicates that they use the data captured at some point(s) in the past. This means that all the *here—past* data are stored UNETS Core temporarily, and finally accumulated in the user's UNETS storage. Applications that consumes the past data in a UNETS storage or a UNETS Core are thus classified to *here—past*, and those consuming the data in a sensor database via the Internet are classified to *there—current* or *there—past*. The other class of applications, *here—current*, consumes the current data running in a UNETS Core. To clarify more practically how users behave in the environment, let us discuss how sensor network applications seen in 2.1 can be extended by UNETS architecture in each quadrant of *HoT* and *CoP* space.

Here—Current. Personal health monitoring can be extended by using the data captured by environment monitoring sensors. Users, for example, can correlate their heart rate data with environmental information such as temperature and humidity to calculate the risk of heat stroke at the place where they reside.

Here—Past. Users can know the risk of the building in terms of its structural health by referring to the series of the past data captured by structure monitoring sensors. For example, they can use accelerometer data to calculate the accumulated damage of the building, since the vibration pattern of a building changes when it is damaged.

There—Current. Sensors for Urban monitoring and environment monitoring can be used for users deciding which way to go to when the risk of the heat stroke is high at their place. Instead of immediate sensors, the users consume data from the remote sensors beyond the immediate ones.

There—Past. Users can know the risk of the buildings in the city where they reside in terms of their structural health by referring to the series of the past data captured by structure monitoring sensors. They can decide, using this information, the safest building to go for a certain purpose (e.g., lunch).



FIGURE 3: uCore Recorder.



FIGURE 4: uCore Dock.

3. Prototyping Studies

The master piece in the UNETS architecture is the UNETS core, which collects the data from the surrounding sensor nodes, store the data temporarily, and perpetuate them into UNETS storage. However, no such device is available currently, which makes it difficult to experiment on the effectiveness of the architecture. We thus prototyped two types of UNETS core: *uCore Reader* and *Sense Phone*. *uCore Recorder* is a unique device with our own design, and *Sense Phone* is a ZigBee interface for Japanese cellular phone. This section first overviews these prototypes and their applications, then the next section discusses lessons learnt.

3.1. Prototyping Study 1: *uCore Recorder*. The first prototyping study is with *uCore Recorder* (Figure 3), and its cradle *uCore Dock* (Figure 4).

3.1.1. Device Configuration. *uCore Recorder* is a device with great portability embedding a range of sensors capable of communicating with external wireless sensor nodes via a ZigBee network. Figure 5 and Table 1 show its hardware specifications. A user can record three different digital information into the 512KB Flash ROM storage in the *uCore Recorder*. First, the user can record the interest to a real-world object by reading the RFID tag on the object. We suppose that digital information about the object such as its name and its description can be retrieved by an RFID tag pasted on the object, so the *uCore Recorder* records the information when it reads the RFID tag. Second, the user can record sensor data from the sensors embedded in the *uCore Recorder*. For example, the *uCore Recorder* embeds an accelerometer, whose data can be used to extract the user's status of movement such as walking, running, or sitting. Third, the user can record sensor data transmitted from MSN and SSN through a wireless network. Users of *uCore Recorder* can perpetuate the stored data to UNETS storage for further analysis, visualization or sharing simply by putting the device on a cradle, called *uCore Dock*.

The *uCore Dock* (Figure 4) plays two roles in the system. First, it is the battery charger of the *uCore Recorder*. It fully recharges the *uCore Recorder* approximately in four hours. Second, which is more important, it is the entrance to the UNETS storage. When a user finishes a series of activities in a day, he can move the data recorded during the day to the database by putting his *uCore Recorder* on the *uCore Dock*. To simplify this data perpetuation, the *uCore Dock* contains

TABLE 1: Specification of *uCore Recorder*.

Wireless sensor node	MICA2DOT
Processor	Atmel ATmega128
Wireless	315 MHz ISM Band
Flash ROM	512 K bytes
Rotary encoder	ALPS EC12E24244A3
Resolution	24 pulse
Switch	Yes (Act.Force 6N)
2-axis accelerometer	Analog Devices ADXL311
Range	± 2 g
RFID reader	SkyeTek SkyeRead M1-Mini
LED	Full-color LED $\times 2$
Battery	Korea PowerCell PD2450 (Li-Ion)
Size (mm)	42(W) \times 42(H) \times 77(D)
Weight	63 g

an RFID tag, which is registered to the *uCore Recorder* as a special tag representing the entrance to the database. The *uCore Recorder*, when reading this registered tag, flushes the recorded data out to the database through the wireless network interface of the containing MICA2DOT sensor node. The data, then, is received by the storage host, to which another MICA2 node is connected, via the same wireless network. To this extent, *uCore Recorder* itself does not require users to manipulate computers using classic interface devices like keyboards and mice. Users can record their activities and put the recorded data to the storage through simple interactions with the *uCore Recorder*, namely, putting it on an RFID tag, and putting it on a *uCore Dock*.

3.1.2. Applications. *uCore Recorder* enables various applications that utilizes data accumulated in UNETS storage. The followings are a few examples of the applications that we have implemented.

Click Catalog. Click Catalog [21] is a *here—past* application for mobile memory aid tool using *uCore Recorder*. In our daily lives, we find many interesting items during shopping, in exposition to events or in conferences. We often try to remember them for later recall by writing memos, taking pictures, and so forth. Those hints (notes and pictures) are, however, easy to be lost. Click Catalog, instead of requiring

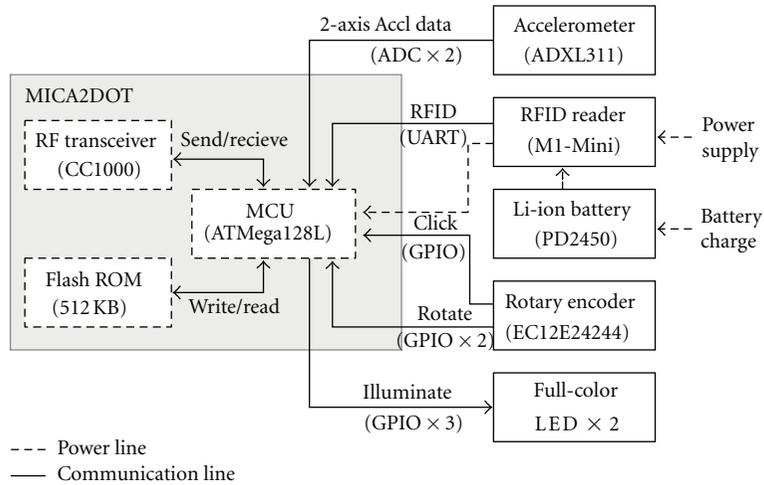
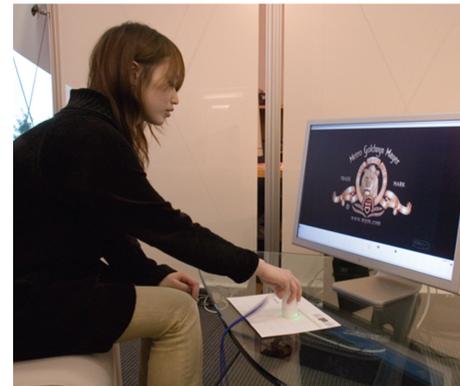


FIGURE 5: Block diagram of uCore Recorder.

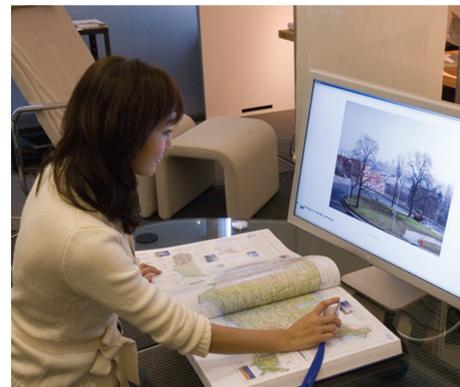
the users to do these proactive memory, automatically records the situations of surrounding environment by receiving sensor data from the user's MSN and/or SSN. The user can also record the items, in which he/she was interested, using the uCore Recorder's RFID reading capability.

The recorded data are perpetuated to the user's UNETS storage using the uCore Dock's perpetuation mechanism. The unique feature of Click Catalog is that it enables users to recall the recorded data using the following two types of paper interfaces. One is an interface using a paper calendar (shown in Figure 6(a)), and the other is an interface using a paper atlas (shown in Figure 6(b)). Those paper-based interfaces are pasted RFID tags. A user can easily refer to the past data recorded on a date by putting his/her uCore Recorder on the calendar interface to read the RFID tag pasted on the calendar at the particular date. They can also refer to the past data recorded at a location similarly by putting it on the atlas interface to read the tag pasted on the atlas at the particular location.

ActiBlog and ActiMap. ActiBlog is a *here—past* application that allows users to publish their activities on their weblogs. Figure 7(a) shows its screen image. It visualizes the digital information of the real-world object, and accompanying sensor data. It is implemented as a plug-in module to a weblog system, called Nucleus (<http://nucleuscms.org/>), thereby enabling users to comment, link, and track back to others' activities with the same manner as ordinary weblogs. Similar, but visually different, application is ActiMap, which visualizes the user activities on GoogleMaps. Figure 7(b) shows its screen image. User activities are marked on the map with icons. The points, on which the activities are placed, are achieved from GPS receivers either public or private. A GPS receiver can be carried with the uCore Recorder as an MSN, or it can be an SSN. The uCore Recorder does not constrain any of these, since it can receive data from those sensors through the same wireless network interface. When a user clicks on an icon, its digital information and the sensor data are shown similarly to the ActiBlog. If multiple users'



(a)



(b)

FIGURE 6: Browsing by a Paper Calendar and by a Paper Map.

activities are published on a same digital map, they can know the geographical relation of their activities on the map.

3.2. Prototyping Study 2: Sense Phone. The second prototyping study is with Sense Phone (Figure 8). The major difference from uCore Recorder is that Sense Phone perpetuates

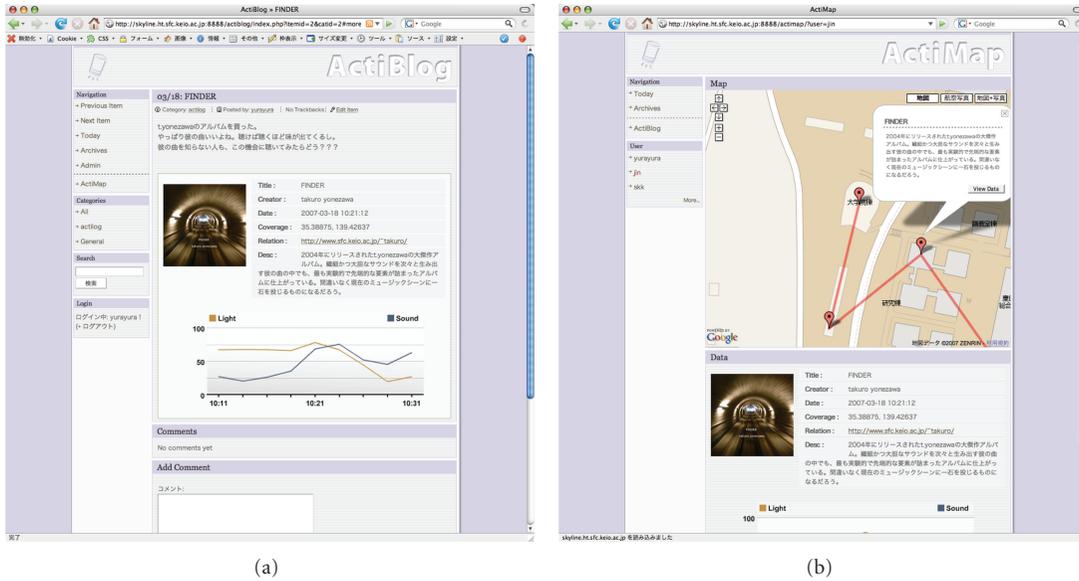


FIGURE 7: Screen shots of ActiBlog and ActiMap.

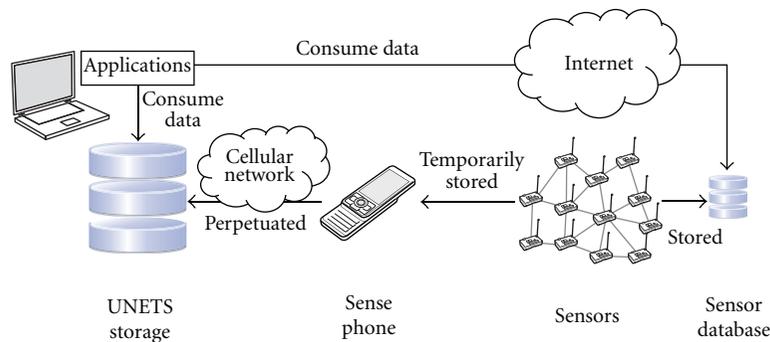


FIGURE 8: Architectural placement of sense phone.

the data via the cellular network. Also, since it contains actuation capability, it can run in situ *here—current* and other classes of applications consuming sensor data captured by MSN and surrounding SSN.

3.2.1. Device Configuration. Sense Phone [22] consists of a sensor data transceiver and the software running on the phone. The transceiver operates as a transport-level bridge between the cellular network and the Internet, and the software as a device-level bridge [23].

This transceiver contains Crossbow MPR2600CA, an OEM module of MICAz Mote, and can operate approximately 8 hours when it receives 3 sensor data packets per second. It connects to a cellular phones via UART as shown in Figure 9. The current prototype can communicate with the series of MICAz Mote sensor nodes and other ZigBee-based nodes if they share the same transport and routing protocols. To achieve this condition, the prototype provides static extensibility; the transceiver embeds one protocol stack that is capable of communicating with one of the different types of nodes.

The software in our technique consists of two layers: middleware and applications. They are implemented with C language on BREW [24]. The middleware operates the transceiver directly and receives the sensor data. The received raw data is stored in the database in the cellular phone and also exported to the applications via APIs. We provide the applications with the following two kinds of APIs. One is the event-driven interface that allows the applications to hook particular range of values, values from particular sensor nodes, and both of them. This enables applications to consume the current data effectively. A special application on the phone uses this interface to perpetuate the received data to the user’s UNETS storage over the cellular network. The other is query interface, with which the applications can collect the past data of their interest from the database. Using these APIs, the applications process the sensor data for their purposes, such as abstracting them into contexts, transmitting them to a web service or exporting them to devices on other communication platforms such as nearby Bluetooth devices. Applications can be downloaded to the cellular phone from their carrier’s networks (in our case, KDDI),



WSN module	Crossbow MPR2600CA
Wireless	ZigBee (IEEE802.15.4)
External IO	UART, JTAG
Power	DC 3 V
Battery	Li-ion PD3048 (300mAh)
Operating time	8 hours (for 3 packets/s)
Dimension	42 × 80 × 10 mm

FIGURE 9: Specifications of Sense Phone transceiver.

sharing the same middleware instance. This feature enables dynamic extensibility in terms of device-level bridge. For example, if a user wants to export the data to another communication platform, he/she can do so just by downloading the application for this purpose. Users can do so from the carrier's application catalogue, such as that for Apple's iPhone, with several clicks or taps of buttons.

3.2.2. Applications. Leveraging the actuation capability of Sense Phone, we have created a here—current application called Infaucet that enables users to acquire digital information being aware of their current context. The application continuously receives sensor data from surrounding SSN and calculates its user's context such that it is raining in the place where the user resides. Meanwhile, the party that installed the SSN can register to the Infaucet server the information that the party wants to disseminate to users of particular context around their SSN. The Infaucet application on the user's cellular phone retrieves the information that conforms to his/her current context.

For example, suppose an urban sensing facility in a city where SSN with weather sensors are installed along the streets. The facility is installed and managed by the city's local government, and the government wants users not to use a slippery street when it is raining. In this case, the government registers to the Infaucet server the recommended road usage information associated with the SSN installed at the intersections to the street. Infaucet users can record the sensor data received from the SSN in the urban sensing facility and simultaneously receive the road usage information.

The application can notify users with this information using the cellular phone's LCD, its speaker set, and so forth.

Readers are reminded that the recorded sensor data would be accumulated in the users' UNETS storage and can be used in *here—past* applications like ActiBlog, ActiMap, and Click Catalog.

4. Lessons Learnt

The aforementioned prototyping studies have pointed out the effectiveness of the architecture and challenges inherent in it. We discuss lessons learnt in the following.

Effectiveness of the Architecture. Researchers are eager to collect sensor data to a centralized computer and export them to the Internet [20, 25]. With such mechanisms, users can consume sensor data on their hand-held device like a cellular phone. However, even if the users want to get the temperature data from the sensor node in front of them, they need to query the data to the computer via the Internet. The two prototypes of UNETS core shown above achieve more efficient utilization of sensor data by enabling users to communicate with sensor nodes directly using uCore Recorder and Sense Phone. Each user having this mechanisms with him/her establishes a distributed, robust, and scalable infrastructure to federate ubiquitous applications to our daily lives. Some existing work shares the same goal [26, 27]. However, their work requires embedding Bluetooth interface to sensor nodes, which is not efficient in energy, and not flexible in network topology.

The applications shown above are enabled by the three-tiered UNETS architecture in that here—past or there—past applications like ActiBlog, ActiMap, and Click Catalog do not need to access multiple sensor databases located with SSN. The past data related to a user are accumulated in his/her UNETS storage (TIER 3), after temporarily stored in a UNETS core (TIER 2) that receives the raw sensor data from MSN and SSN (TIER 1). Therefore, the applications can acquire all the data related to the series of the user activities from his/her UNETS storage. Without having UNETS core in the architecture, the data are spread over the distributed sensor databases associated with SSN, and the applications need to collect the data from all of them. This scheme is inefficient; many transactions required to collect the data would consume network bandwidth and time.

Open Sensor Networks. UNETS core is assumed to communicate with SSN included in sensor networks established for certain purposes by different parties. Usual sensor networks are closed in that the data are collected to a central database, and alien devices like UNETS core are not assumed. The sensor networks, however, need to be open in our architecture; SSN in the networks are required to transmit sensor data to alien UNETS cores. We found the following two issues to achieve this.

First, the communication protocol used between UNETS cores and SSN needs to be standardized. UNETS cores traverse different sensor networks collecting sensor data. If the sensor networks utilize different protocols for sensor data

transmission, the UNETS cores required a priori knowledge about all of those protocols. Assuming existence of numerous sensor networks, thus protocols, such UNETS cores having such knowledge is impractical. In addition, the computational resources embedded in UNETS cores are limited based on our prototyping studies of uCore Recorder and Sense Phone. This resource limitation makes it difficult for UNETS cores to have a number of protocol stacks in them. Therefore, we need a standardized communication protocol for sensor data transmissions. Standardization in physical and MAC layers have been addressed by IEEE, resulting in promotion of IEEE802.15.4 (ZigBee). That in network and above layers needs to be addressed.

Second, security would be of a great concern for the parties that open their sensor network to the UNETS architecture. Unlike the current practice of sensor networks where the parties can close their network by using their own protocols and security mechanisms, use of a standard protocol would disclose SSN to malicious users. They might intrude to SSN and disable them. Therefore, a new security mechanisms with assumption of open sensor networks need to be investigated.

Synchronization. We have learnt that synchronization is important between UNETS core and SSN in the following two reasons. Both are for achieving the architecture's requirement that UNETS core is mobile and assumed to communicate with SSN in an ad hoc manner.

First, the amount of energy is limited in SSN and UNETS core, since they usually operate with battery power. Outdoor sensor nodes including Field Server [28] and eKo Mote (<http://www.memsic.com/products/wireless-sensor-networks/environmental-systems.html>) are operated with solar reactor. This infers that such nodes are operated with an intermittent mode such that a node wakes up to sense and transmit the data once 10 minutes. On the other hand, the mobility pattern of a UNETS core is random. Since the UNETS core's energy is also limited, it is also operated with an intermittent mode. These conclude that without any synchronization mechanism the UNETS core cannot collect sensor data.

Second, if a UNETS core is migrating with a very high speed (e.g., in a running car or a running train with 100 km/h), it may overpass a radio coverage of SSN instantly. Multihop sensor data transmission would take reasonable amount of time to reach the destination. Therefore, the data needs to be routed synchronously to the mobility of the destined UNETS core. This issue is investigated by researchers of mobile sink mechanisms such as [17].

Multiuser Extension. The UNETS architecture described above is focused on a single user's perspective; however, it needs to be extended to support mutual exchange and fusion of sensor data. The data accumulated in UNETS storage would contain personal behavioral information, such as distance of activity of the users and environmental context (e.g., temperature and humidity) where they visited. Such information is valuable for many purposes including dynamic demography and urban strategy. Therefore, the data

in UNETS storage should be able to be reused by third parties.

Privacy is, however, a major issue to achieve this. Since UNETS storage contains the users' private information, the data should not be disseminated without the users' permission. This infers that the users need to (1) grant particular permissions to the third parties, (2) mosaic the data to export if needed, and (3) confirm that the data are deleted by the third parties after they finished using them. Security is also a major issue, since the third parties need to avoid spoofing. When the sensor data are exported from their owner to a third party, the data need to be guaranteed that they actually contain the owner's information.

5. Related Work

There are several classes of research towards effective use of user-side terminals for sensing. One common approach to this direction is the use of cellular phone as a sensor or as a sink node. Use of the phones as a sensor is known as the mobile phone sensing research [15]. Recent cellular phones contain a range of sensors including ubiquitous networked light, proximity, cameras, GPS, accelerometer, compass, and so forth. Due to their richer computational capability, phones can be used to capture context of users in the ubiquitous/pervasive computing perspective. Based on the captured context, they can provide user-aware digital services with them. This approach is limited compared to our architecture. As shown in Figure 10 as a dotted rectangle A, mobile phone sensing can be observed as the model that solely rely on the UNETS core. Applications are limited only to *here—current* and *here—past* that use sensors in the phone. All other applications that consume data from SSN are disabled.

Sensor network deployment projects, such as campus sensing [2], home sensing [1], structure monitoring [10], and environment monitoring [7], focus on the use of SSN. They are seen in the UNETS architecture as the dotted rectangle C in Figure 10. Each deployed sensor network collects data into its associated sensor database. The collected data can then be used by applications that run locally in the network, or remotely via the Internet. This classic model is data-centric instead of user-centric in that it is optimized for efficient data collection into the centralized database, and users are always required to query in the database even when they are directly facing a sensor node.

The model of participatory sensing partly shares user-centric view with us. Some urban sensing projects [3, 4, 29] utilize user-side terminals that correspond to UNETS core in our architecture. Aquiba [3] is a system for achieving human probe capability in urban sensing. They leverage our Sense Phone for their purpose. MetroSense [4] defines a similar architecture that includes the notions of mobile sensors and static sensors. However, their goal is to share the data among users who participate in a sensing project. Their focus is thus closed in the interaction between the user-side terminals and sensors, which is depicted as the dotted circle B in Figure 10.

Besides the sensing models, the following two issues are investigated recently aiming at enhancing interaction between the architectural components. First issue is global

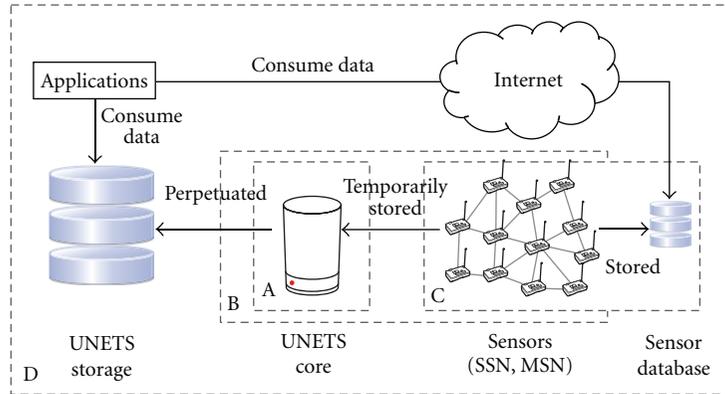


FIGURE 10: Relation of sensing models.

sensor networks as the means to enable for applications to acquire sensor data via the Internet. IrisNet [20] provides a middleware system to share sensors among multiple computers within a city-scale. SenseWeb [6] aims at sharing sensor nodes worldwide. It proposes a software architecture for global sensor sharing assuming sensor, data, and application heterogeneity. GSN [30] also aims at the worldwide sensor sharing. Our architecture does not specify how applications acquire sensor data via the Internet, since sensors may be exported to the Internet through different mechanisms described above. Applications can use the mechanism that exports the sensors needed by the applications. Second, technologies for mobile sink nodes have been investigated. Those technologies can be used to achieve more efficient interaction between UNETS core and SSN in the architecture. Several researchers propose a routing protocol to use mobile nodes as a sink node of a sensor network effectively [17, 18].

6. Conclusions

Human-centric ubiquitous networked sensing is still in its infancy. There is little or no consensus on the sensing, processing, and actuation architecture for the involved devices. Our work contributes to this field in that it gives a generic and total view of such an architecture with component, data flow, and application models. The major feature of our three-tier architecture is that it contains the “core” device, called UNETS Core, which is assumed to be a terminal held by users. The sensor data are directly transmitted to UNETS Core from surrounding sensor nodes including static and mobile nodes. The data are then perpetuated into UNETS Storage. This three-tier sensor—UNETS Core—UNETS Storage enables effective use of sensor data by applications. We classified UNETS applications with location (here or there) and time (current or past) basis. This classification and its associated examples gives insights to researchers of practical sensing about the users’ expectation to sensor networks. Our prototype implementations of UNETS Core device can be leveraged by industry to develop a new product for practical use of ubiquitous networked sensing. Finally, we have clarified lessons learnt in which several open issues are indicated for further study.

Acknowledgments

The authors would like to express our gratitude to Dr. Junichi Yura at Fujitsu Laboratories and all members of our laboratory.

References

- [1] S. S. Intille, K. Larson, J. S. Beaudin, J. Nawyn, E. M. Tapia, and P. Kaushik, “A living laboratory for the design and evaluation of ubiquitous computing technologies,” in *Proceedings of the Conference on extended Abstracts on Human Factors in Computing Systems (CHI ’05)*, pp. 1941–1944, ACM Press, New York, NY, USA, 2005.
- [2] A. Rowe, M. E. Berges, G. Bhatia et al., “Sensor andrew: large-scale campus-wide sensing and actuation,” *IBM Journal of Research and Development*, vol. 55, no. 1, article 6, 2011.
- [3] N. Thepvilojanapong, S. Konomi, and Y. Tobe, “A study of cooperative human probes in urban sensing environments,” *IEICE Transactions on Communications*, vol. 93, no. 11, pp. 2868–2878, 2010.
- [4] A. T. Campbell, S. B. Eisenman, N. D. Lane, E. Miluzzo, and R. A. Peterson, “People-centric urban sensing,” in *Proceedings of the 2nd Annual International Workshop on Wireless Internet (WICON ’06)*, ACM Press, New York, NY, USA, 2006.
- [5] M. Ito, Y. Katagiri, M. Ishikawa, and H. Tokuda, “Airy notes: an experiment of microclimate monitoring in shinjuku gyoen garden,” in *Proceedings of the 4th International Conference on Networked Sensing Systems (INSS ’07)*, pp. 260–266, June 2007.
- [6] A. Kansal, S. Nath, J. Liu, and F. Zhao, “SenseWeb: an infrastructure for shared sensing,” *IEEE Multimedia*, vol. 14, no. 4, pp. 8–13, 2007.
- [7] R. Szweczyk, E. Osterweil, J. Polastre, M. Hamilton, A. Mainwaring, and D. Estrin, “Habitat monitoring with sensor networks,” *Communications of the ACM*, vol. 47, no. 6, pp. 34–40, 2004.
- [8] G. D. Abowd, C. G. Atkeson, A. F. Bobick et al., “Living laboratories: the future computing environments group at the georgia institute of technology,” in *Proceedings of the Conference on Extended Abstracts on Human Factors in Computing Systems (CHI ’00)*, ACM Press, New York, NY, USA, 2000.
- [9] S. Meyer and A. Rakotonirainy, “A survey of research on context-aware homes,” in *Proceedings of the Australasian Information Security Workshop (AISW ’03)*, vol. 21, pp. 159–168, Australian Computer Society, Darlinghurst, Australia, 2003.

- [10] N. Kurata, B. F. Spencer, and M. Ruiz-Sandoval, "Risk monitoring of buildings with wireless sensor networks," *Structural Control and Health Monitoring*, vol. 12, no. 3-4, pp. 315–327, 2005.
- [11] S. Kim, S. Pakzad, D. Culler et al., "Health monitoring of civil infrastructures using wireless sensor networks," in *Proceedings of the 6th International Symposium on Information Processing in Sensor Networks (IPSN '07)*, pp. 254–263, ACM Press, New York, NY, USA, April 2007.
- [12] I. Talzi, A. Hasler, S. Gruber, and C. Tschudin, "PermaSense: investigating permafrost with a WSN in the Swiss Alps," in *Proceedings of the 4th Workshop on Embedded Networked Sensors (EmNets '07)*, pp. 8–12, ACM Press, New York, NY, USA, June 2007.
- [13] I. Vasilescu, K. Kotay, D. Rus, M. Dunbabin, and P. Corke, "Data collection, storage, and retrieval with an underwater sensor network," in *Proceedings of the 3rd International Conference on Embedded Networked Sensor Systems (SenSys '05)*, pp. 154–165, ACM Press, New York, NY, USA, 2005.
- [14] G. Werner-Allen, K. Lorincz, M. Welsh et al., "Deploying a wireless sensor network on an active volcano," *IEEE Internet Computing*, vol. 10, no. 2, pp. 18–25, 2006.
- [15] N. D. Lane, E. Miluzzo, H. Lu, D. Peebles, T. Choudhury, and A. T. Campbell, "A survey of mobile phone sensing," *IEEE Communications Magazine*, vol. 48, no. 9, pp. 140–150, 2010.
- [16] A. Natarajan, M. Motani, B. De Silva, K. K. Yap, and K. C. Chua, "Investigating network architectures for body sensor networks," in *Proceedings of the 5th International Conference on Mobile Systems, Applications and Services*, pp. 19–24, ACM Press, New York, NY, USA, June 2007.
- [17] J. Luo, J. Panchar, M. Piórkowski, M. Grossglauser, and J.-P. Hubaux, "MobiRoute: routing towards a mobile sink for improving lifetime in sensor networks," in *Distributed Computing in Sensor Systems*, P. Gibbons, T. Abdelzaher, J. Aspnes, and R. Rao, Eds., vol. 4026 of *Lecture Notes in Computer Science*, pp. 480–497, Springer, Berlin, Germany, 2006.
- [18] K. Fodor and A. Vidács, "Efficient routing to mobile sinks in wireless sensor networks," in *Proceedings of the 3rd International Conference on Wireless Internet (WICON '07)*, pp. 1–7, ICST, Brussels, Belgium, 2007.
- [19] F. Ye, H. Luo, J. Cheng, S. Lu, and L. Zhang, "A two-tier data dissemination model for large-scale wireless sensor networks," in *Proceedings of The 8th Annual International Conference on Mobile Computing and Networking*, pp. 148–159, ACM Press, New York, NY, USA, September 2002.
- [20] P. B. Gibbons, B. Karp, Y. Ke, S. Nath, and S. Seshan, "IrisNet: an architecture for a worldwide sensor web," *IEEE Pervasive Computing*, vol. 2, no. 4, pp. 22–33, 2003.
- [21] A. Komaki, J. Yura, M. Iwai, J. Nakazawa, K. Takashio, and H. Tokuda, "Clickcatalog: retracing precious memory using paper-based media controller," in *Proceedings of the International Conference on Machine Learning and Cybernetics*, vol. 4, pp. 2077–2082, August 2007.
- [22] J. Nakazawa, J. Yura, T. Iwamoto, H. Yokoyama, and H. Tokuda, "Bridging sensor networks and the internet on cellular phones: towards richer and easier-to-use sensor network applications for end-users," in *Proceedings of the 6th International Conference on Networked Sensing Systems (INSS '09)*, pp. 94–97, June 2009.
- [23] J. Nakazawa, H. Tokuda, W. K. Edwards, and U. Ramachandran, "A bridging framework for universal interoperability in pervasive systems," in *Proceedings of the 26th IEEE International Conference on Distributed Computing Systems (ICDCS '06)*, July 2006.
- [24] Qualcomm Inc. Qualcomm brew.
- [25] M. Balazinska, A. Deshpande, M. J. Franklin et al., "Data management in the worldwide sensor web," *IEEE Pervasive Computing*, vol. 6, no. 2, pp. 30–40, 2007.
- [26] M. Leopold, M. B. Dydensborg, and P. Bonnet, "Bluetooth and sensor networks: a reality check," in *Proceedings of the 1st International Conference on Embedded Networked Sensor Systems (SenSys '03)*, pp. 103–113, November 2003.
- [27] L. Zhong, M. Sinclair, and R. Bittner, "A phone-centered body sensor network platform: cost, energy efficiency & user interface," in *Proceedings of the International Workshop on Wearable and Implantable Body Sensor Networks (BSN '06)*, pp. 179–182, April 2006.
- [28] M. Hirafuji, H. Yoichi, and M. Wada, "Field server: multifunctional wireless sensor network node for earth observation," in *Proceedings of the 3rd International Conference on Embedded Networked Sensor Systems (SenSys '05)*, p. 304, ACM Press, New York, NY, USA, 2005.
- [29] B. Hull, V. Bychkovsky, Y. Zhang et al., "CarTel: a distributed mobile sensor computing system," in *Proceedings of the 4th International Conference on Embedded Networked Sensor Systems (SenSys '06)*, pp. 125–138, Boulder, Colo, USA, November 2006.
- [30] K. Aberer, M. Hauswirth, and A. Salehi, "Infrastructure for data processing in large-scale interconnected sensor networks," in *Proceedings of the International Conference on Mobile Data Management*, pp. 198–205, Washington, DC, USA, 2007.

