Pervasive Sensing: Addressing the Heterogeneity Problem

Michael J. O'Grady, Olga Murdoch, Barnard Kroon, David Lillis, Dominic Carr, Rem W. Collier and Gregory M. P. O'Hare

CLARITY: Centre for Sensor Web technologies, University College Dublin, Belfield, Dublin 4, Ireland.

E-mail: `michael.j.ogrady@ucd.ie`

Abstract. Pervasive sensing is characterized by heterogeneity across a number of dimensions. This raises significant problems for those designing, implementing and deploying sensor networks, irrespective of application domain. Such problems include for example, issues of data provenance and integrity, security, and privacy amongst others. Thus engineering a network that is *fit-for-purpose* represents a significant challenge. In this paper, the issue of heterogeneity is explored from the perspective of those who seek to harness a pervasive sensing element in their applications. A initial solution is proposed based on the middleware construct.

1. Introduction

Historically, sensor networks are tightly coupled to particular application domains, for example, environmental monitoring. In the case of a scientific experiment, a dedicated network would be designed such that the data required for the experiment in question would be collected within defined quality-of-service parameters. Pervasive sensing radically challenges the conventional perception of sensor networks in many dimensions. Specifically, a sensing infrastructure may be envisaged as multi-functional, fulfilling the requirements of a diverse end-user constituency. Conceptually, a standard security alarm system for a home might also serve as a basis for an Ambient Assisted Living (AAL) service. However, ensuring that the needs of both services are sufficiently addressed is a complex endeavour.

Mote platforms that encapsulate a range of sensing modalities, many in a *Plug-and-Play* manner, are increasingly available. The SunSPOT [1] and Tyndall [2] platforms are archetypical of platforms that incorporate a variety of physical sensors, and are augmented with sufficient computational resources on the platform to engage in relatively sophisticated pre-processing. Indeed, this development is a logical continuation of the process of moving intelligence to the fringe nodes of networks [3]. However, the inherent heterogeneity that arises from the variety of hardware and software platforms poses significant problems for those aspiring to design, implement and deploy sensor networks. In this paper, this issue of heterogeneity is explored more detail, and an initial solution based on the middleware paradigm is illustrated.

2. Heterogeneity in Wireless Sensor Networks

Heterogeneity amongst Wireless Sensor Networks (WSNs) is a significant barrier to integrating various types of sensor platforms into a common application. Utilising a number of disparate

sensor motes and platforms may require familiarity with multiple implementation programming languages, operating systems (OSs) and deployment mechanisms. Table 1 outlines a number of WSN products in terms of the languages and OSs they support.

Table 1. Sensor Motes Comparison

Namef	Programming Language	Operating System
EcoSpire	C	Enix
Namimote	C, C++	RTOS
Shimmer	nesC	TinyOS (default)
SunSPOT	Java ME	Java VM
TelosB	nesC	TinyOS
Tyndall	nesC, C	TinyOS, Contiki
Waspmote	C++	Embedded

In this table, the phrase "Operating System" is used loosely, as the Java Virtual Machine supported by SunSPOT motes and the Waspmote's approach of running compiled C++ code directly do not technically constitute OSs. However, the table indicates the range of technologies that an application developer must master in order to combine the capabilities of these platforms. More detailed comparisons of commonly-used WSN technologies can be found in [4] and [5].

This problem is exacerbated further if the definition of a "sensor" is expanded beyond these traditional WSNs to other data-producing entities. For instance, modern mobile smartphones typically include a range of sensors including accelerometers, GPS capabilities, digital compasses, etc. Additionally, both raw data and more complex information is available from internet sources. In many cases, it may be desirable to use this data to augment that gleaned from WSNs deployed directly by an application developer in a physical environment.

2.1. Standardization

The vast amount of different sensors generate a diverse set of issues when trying to integrate the heterogenuous data types and communication models into a sensor network. The Open GeoSpatial Consortium (OGC) have proposed the Sensor Web Enablement (SWE) Initiative to help alleviate these issues, especially when exposing these sensor networks on the internet [6]. The SWE proposes a standard data model, in the form of the Observations and Measurements XML Schema, to encode observations and measurements from sensors. Additionally a Sensor Model Language (SensorML) is also given, and this allows for describing the deployed sensor, including such things as the location of the sensor and its various tasking capabilities.

SWE also describes a number of service protocols for accessing the sensors, and these include the Sensor Observation Service (SOS), Sensor Planning Service (SPS), Sensor Alert Service (SAS) and Web Notification Services (WNS). SOS and SPS provide access mechanisms for requesting, filtering and retrieving sensor observations and information. The SAS provides mechanisms for subscribing and publishing sensor alerts, and WNS provides mechanisms for asynchronous access to SAS and SPS web services.

2.2. Ontological issues

The OGC SWE Initiative is one effort to move sensors to wider audience; however the SWE standards are quite verbose and challenging to implement [7]. Another more generic effort to enable wider interaction and better understanding between different systems is the emerging semantic web. There have been also efforts to semantically annotate the SWE standards and encode them using the semantic notation [8].

Other attempts at creating formal Ontologies to represent sensors and their data have also emerged. These include the Sensor Data Ontology (SDO) [9], OntoSensor [10], Ontonym [11] and the CSIRO Sensor Ontology [12].

All these diverse attempts at creating sensor Ontologies have motivated the World Wide Web Consortium (W3C) to create a working group with the express purpose of creating a standardized sensor ontology, namely the W3C Semantic Sensor Network (SSN) Ontology [13]. This uses the CSIRO Sensor Ontology as a base, incorporating elements from various other existing Ontologies. It should be noted that the W3C SSN uses entities defined in the Dolce Ultralight (DUL) Ontology, and has defined all sensors to be physical entities. This means that any sensors that are not also physical entities cannot be described using the W3C SSN without causing contradictions and inconsistencies in the resulting Ontology.

3. Towards an Adaptive Middleware for Heterogeneous Sensors

In pervasive systems, "context" comprises any information that characterises the situation of any person, place, or object that is considered relevant to the interaction between a user and an application. Context data may be gathered from many sources including physical sensors deployed in the environment (e.g. sensing light or sound levels) and non-physical sources such as calendar information and data accessible via the World Wide Web (WWW). As illustrated in Figure 1 and Table 2, the current WWW encompasses diverse documents and services that produce and/or maintain real-time and static information relating to a wide range of users, places, topics, and concepts. With the movement towards the vision of the Internet of Things (IoT) and Web of Things (WoT), the availability of web-accessible data providing resources is expected to increase dramatically in the coming years.

The term *cyber sensor* describes software that accesses data from non-physical sources. Specifically, a cyber sensor programmatically monitors a non-physical environment, in contrast with physical sensors that can only monitor the physical environment in which they are situated.
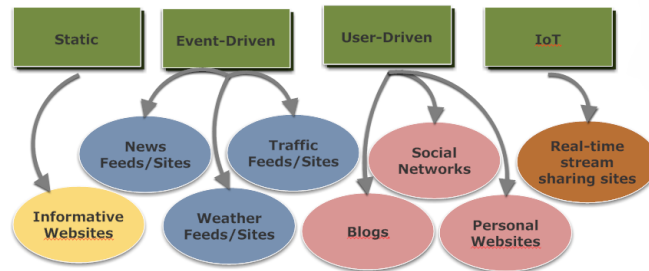

Figure 1. Taxonomy of Web-based Sources of Context

The fusion of web-based sources of context data with that emitted from physical sensors will bridge the divide between the physical and cyber worlds, enabling a wealth of web-based and immersive complex context-driven applications. As a simple example, in the context of a building subject to real-time fluctuating energy prices, physical sensors monitoring electricity usage can be combined with cyber sensors that capture real-time price information as part of a smart energy management system.

Development of systems driven by WSNs is complicated by the dynamic and uncertain nature of the physical devices and the data they produce. Programming with data from cyber sensors poses alternative challenges to those inherent with physical sensors. Many web-based environments produce data that is manually entered by users (e.g. social networks, news websites). Manually entered data is mostly unstructured, prone to human error and can often be

Table 2. Classification of Web-based Context Sources

Type	Definition	Example
Static	Knowledge or fact	Documents, Web1.0
User Driven	Data availability relies on user updates	Blogging, social networks
Event Driven	Scheduled or manual reporting	Weather site/API, newsfeed
IoT	Live stream from remote networks/devices	COSM, Thing Speak
Service	Provision of information via data processing	Klout, Peer Index

subjective and dependent on the provider's perspective. These factors not only impact on how reliable the data is but also make managing and interpreting the data a difficult task. Situation identification and uncertainty management techniques in pervasive computing typically aim to manage uncertain data from physical sensors. For systems comprising both physical and cyber sensor data to become a reality it is imperative that these techniques begin to take into account diverse sources of data.

Each non-physical environment has different access policies and restrictions that impact how often and how much data can be accessed. Application Programming Interfaces (APIs) most often have a limit on the number of requests that can be made per minute/hour. Especially for applications that require near real-time data, fusing related context from multiple sources can be problematic when considering the different rates at which data will reach the system.

Cyber sensors are more complex than physical sensors in terms of location and time. Physical sensors have a single location at any given point in time and as such data recorded at a particular time is related to a specific location. Web-based sources often provide information relating to many geographical locations and the information may be (near) real time or historical/futuristic to some degree. For example, web sites providing weather information provide data relating to many locations and while the data provided can be considered real time the information doesn't always relate to 'now', i.e., weather forecasts and historical weather data. These differences offer alternative perspectives on data other than that available through use of physical sensors. While such diversity of location and time offers a richer contextual basis for applications it also makes the task of managing and incorporating these diverse data sources more difficult.
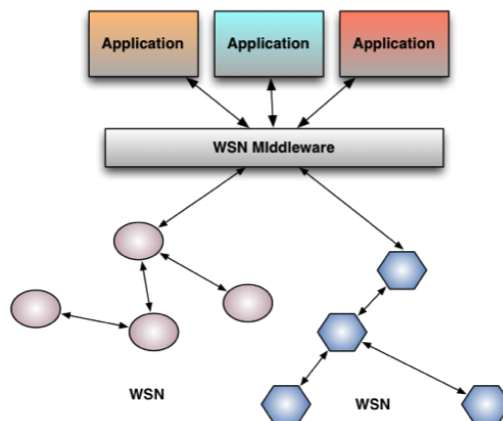

Figure 2. Wireless Sensor Network middleware

In order to address the challenges of pervasive sensing outlined above, a sensor middleware solution is proposed. This acts as an intermediary between an array of sensors that capture data on one hand and applications that make use of this sensor data on the other. By addressing the

low-level challenges, it empowers the developer to rapidly develop complex real-time applications. The role of a sensor middleware system in providing access to WSNs is illustrated in Figure 2.

In this configuration, application developers are not required to become familiar with the specific implementation details of each of the sensor types with which they wish to interact. This is a particularly important concern when heterogeneous sensors support a variety of operating systems, programming languages and deployment strategies. Instead, the middleware provides a uniform interface whereby applications can gain access to sensor data and event information in addition to being able to re-task individual sensor networks. The incorporation of cyber sensors adds substantially to the complexity of such a system.

4. SIXTH: A case study

The SIXTH sensor middleware has been created to address the challenges outlined in the previous section. In addition to the basic WSN middleware model shown in Figure 2, SIXTH also provides support for other heterogeneous sensing resources that can provide context, particularly cyber sensors. The key features of the SIXTH middleware is as follows:

- **Uniform adaptor interface:** Integrating a particular sensor type requires an *adaptor* to be written for it. This allows the sensor's data to be used by any SIXTH application in a consistent way. An example of the structure of an adaptor created for a SunSpot WSN is shown in Figure 3. Here, sensor data generated by the WSN is routed through the adaptor to the middleware, which can then distribute this data to appropriate applications. The applications need not be concerned with the mechanism by which this data was sourced. As adaptors can be created for both physical and cyber sensors, this is the core enabler for heterogeneous sensor support.
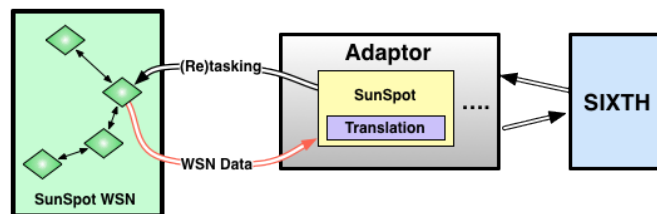

Figure 3. Anatomy of a SIXTH Sensor Adaptor

- **Retasking capabilities:** The interaction between applications and sensors is not one-way. In addition to gaining access to sensor data, it is also important for certain applications to be capable of retasking the sensors that they use. An example of this is when a change to the sampling frequency of the sensor is required. Again, the application can create a retasking message in a uniform way, which is passed to the sensor by the middleware using the adaptor. Here, the adaptor's role is to translate the SIXTH retasking message into the native instructions required by the sensor itself.

- **Distribution and Discovery:** SIXTH can run on a variety of computer types including servers, desktop computers, laptops and mobile devices (currently Android devices are supported). Because each individual device may have direct access to a different set of sensors, it is important that the sharing of sensor data between devices be supported. Direct access to a sensor or sensor network may come through software hosting of cyber sensors, bluetooth connections, USB connections to base-stations of WSNs and in the case of mobile devices a range of sensors is typically built in to the device itself. In addition to the possibility of distributed deployment of SIXTH, a discovery service is also provided whereby SIXTH deployments can be identified and their capabilities shared.

- **Querying:** In a system where multiple applications are utilising a range of sensor types, it is necessary to avoid flooding the communication channels with unwanted sensor data. For this reason, a querying system is available whereby applications can describe the type of data in which they are interested either programmatically or using a specialised query language. Only sensor data matching the query is forwarded to the interested application.

- **Data Management:** In some instances, sensor data may require aggregation or contextualisation using data from other sources. SIXTH provides a framework to support the management of data through the system prior to consumption by applications.

- **Event Notifications:** As sensor data not is all an application may require, SIXTH also notifies interested components about events that affect the middleware. This includes details of new sensors or sensor networks being added to the system, removal or failure of existing sensor capabilities and any other events in which an application may be interested.

- **Standards-based:** SIXTH uses a range of existing standards. The Java programming language facilitates the use of the Open Services Gateway Initiative framework (OSGi) to allow for the dynamic addition and removal of components at runtime. RESTful transfer of messages and data is inbuilt and JmDNS is used for service registration and discovery.

5. Conclusion

Heterogeneity increases complexity; thus engineers will find it increasingly difficult to design sensor networks that serve multiple purposes while remaining fit-for-purpose. In this paper, some of the key sources of heterogeneity were discussed. The middleware construct offers one viable approach for mitigating the complexity inherent in heterogeneous sensor networks.

References

[1] Smith R 2007 *Information Processing in Sensor Networks, 2007. IPSN 2007. 6th International Symposium on* pp 565 –566

[2] O'Flynn B, Lynch A, Aherne K, Hare P, Barton J, Harte S, O'Mathuna C, Diamond D and Regan F 2006 *Adjunct Proceedings of the 4th International Conference on Pervasive Computing* pp 21–26

[3] O'Grady M J, O'Hare G M P, Chen J and Phelan D 2009 *Information Systems Frontiers* **11**(1) 61–73 ISSN 1387-3326

[4] Johnson M, Healy M, van de Ven P, Hayes M, Nelson J, Newe T and Lewis E 2009 *IEEE Sensors* pp 1439–1442 ISSN 1930-0395

[5] Madan V and Reddy S N R 2012 *VSRD International Journal of Electrical, Electronics and Communication Engineering (VSRD-IJEECE)* **2** 50–55

[6] Botts M, Percivall G, Reed C and Davidson J 2008 *GeoSensor Networks* (*Lecture Notes in Computer Science* vol 4540) ed Nittel S, Labrinidis A and Stefanidis A (Springer Berlin / Heidelberg) pp 175–190

[7] Rouached M, Baccar S and Abid M 2012 *Services (SERVICES), 2012 IEEE Eighth World Congress on* pp 65–72

[8] Broring A, Foerster T and Jirka S 2010 *Pervasive Computing and Communications Workshops (PERCOM Workshops), 2010 8th IEEE International Conference on* pp 732 –737

[9] Eid M, Liscano R and El Saddik A 2007 *Computational Intelligence for Measurement Systems and Applications, 2007. CIMSA 2007. IEEE International Conference on* pp 59 –62

[10] Goodwin C An ontology-based sensor network prototype environment

[11] Stevenson G, Knox S, Dobson S and Nixon P 2009 *Proceedings of the 1st Workshop on Context, Information and Ontologies* CIAO '09 (New York, NY, USA: ACM) pp 9:1–9:8

[12] Michael Compton Holger Neuhaus K T and Tran K N 2009 *2nd International Workshop on Semantic Sensor Networks, CEUR-WS* pp 33–48

[13] Semantic sensor network xg final report `http://www.w3.org/2005/Incubator/ssn/XGR-ssn/` last visited 2013/05/13