SENSORMASH: EXPLORING SYSTEM FIDELITY THROUGH SENSOR MASHUP

Steve Neely, Matthew Stabeler and Paddy Nixon[1]

Abstract
Context-aware services are driven by sensed data from both real and virtual worlds. Building effective pervasive systems involves administration and fine tuning of sensors toward optimal operation. Difficulties arise because sensors are prone to inaccuracies through miscalibration, malfunction and component limitations. Any incorrect values of sensed data need to be accounted for and dealt with appropriately otherwise the system as a whole may not behave in a useful manner. Trying to understand the limitations of sensors, service tolerances to inaccuracies, and the multiplicative effects of erroneous data on a given system can be an extremely complicated task. In this paper we present SensorMash, a tool for exploring sensor interactions as a mashup of inputs into a context-aware system. Built on top of a general model for sensor data, SensorMash allows developers to explore the effects of massaging tolerances to inaccuracy ratings and uncertainty. A small user trial is described with initial results driving research into an autonomic sensor management system.

1. Introduction

Pervasive systems observe characteristics of the physical and virtual environment. From these observations they create models of the world to support user tasks. As developers of these *context-aware* systems we install and carefully configure sensors to gather data and drive our pervasive services. Calibration of sensors is vitally important as inaccurate readings from misconfiguration is difficult to later trace. The bad data pervades through the system with effects that cannot be predetermined.

In a multi-sensor environment the effect of a single misbehaving sensor is not always easy to notice. The difficulty of tracing which sensor (or sensors) is causing the problem may be further compounded by layers of abstraction, merging of various data sources and processing by complex algorithms. As a community we expend considerable effort to combat these kinds of errors by engineering systems by considering sensor placement, introducing sensor redundancy, applying fusion and filtering algorithms [4], creating models (e.g. [3, 5]) and building self-organising or autonomic components into our systems [2].

We are working towards creating a service that monitors sensor outputs and automatically adjusts their influence on the construction of context models. It will tweak parameters for decay, precision, and confidence in an attempt to fine-tune sensor readings for a given environment. Automated solutions for this are the panacea but to achieve them we need to gain an insight into what the system is actually doing and why. In our experience, analysing raw sensor outputs in comprehension of a system is an intractable task. The systems are too complicated for even expert users to fully understand.

[1]{*firstname.lastname*}@*ucd.ie*, School of Computer Science and Informatics, Complex and Adaptive Systems Laboratory, University College Dublin, Belfield Dublin 4. Ireland. *This work is partially supported by Science Foundation Ireland under grant number 04/RPI/1544, "Secure and Predictable Pervasive Computing.", and Irish Research Council for Science, Engineering and Technology EMBARK Initiative.*

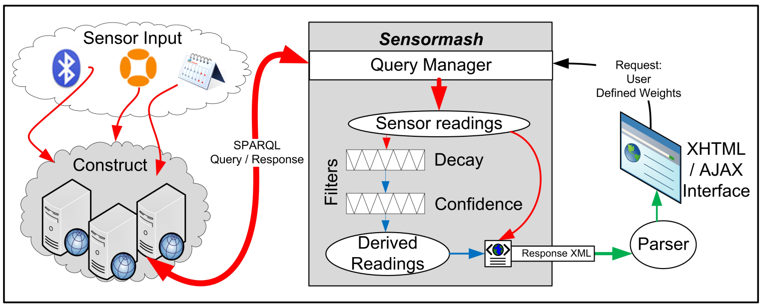

Figure 1. SensorMash Architecture.

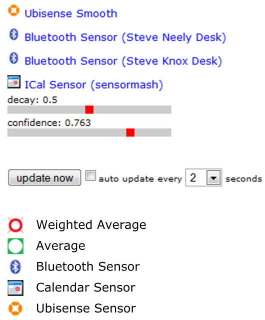

Figure 2. A screenshot of the SensorMash visualisation.

In this paper we describe SensorMash, an application that enables developers of pervasive environments to *explore* the fidelity of their systems through a *sensor mashup* visualisation. It allows the contribution of sensor readings to be varied in real-time by adjusting a confidence weighting on them. We describe the application of SensorMash to Construct [1], a context-aware platform deployed in University College Dublin. The UCD deployment of Construct includes a wide variety of sensors that are blended into a common context model based on RDF and governed by shared ontologies. We report user trials with SensorMash which show that our location tracking system could be improved as a direct result of developer interactions through our tool.

2. SensorMash

The SensorMash application[2] (see figure 1) merges sensor data and displays it as a visualisation. Sliders on the visualisation allow the user to adjust weightings on individual data sources whilst applying filters to affect the generation of the underlying context model.

The internal functionality of SensorMash is shown in figure 1. The system is based on two filters for *fixed* decay and *user-defined* confidence. Initially a sensor reading is given the maximum overall weight value of 1. The filters are applied in series; decay followed by confidence to give a final weighting for each reading. This weight is used to calculate the derived value. The average centroid fusion (the geometric centre of all sensor readings) value is also displayed.

Decay Filter – affects the length of time that a sensor reading is valid; each sensor type is given a default time-to-live value. This allows the readings to become stale and disappear over time e.g. calendar entries expire at their given end time.

Confidence filter - defines how much confidence to put on the reading of a particular sensor; for example a sensor with a confidence value of 0.3 has less weight than a sensor with a value of 0.7. The overall weight for a sensor is simply multiplied by the given value to derive the level of confidence.

The resulting overall weighting for each sensor is used in the weighted centroid fusion algorithm for each dimension (x, y and z) where D is the resulting coordinate, d is the set of readings coordinates for a dimension, w is the set of overall weights normalised to 1 for readings and where the size of d and w are equal, as follows: $D = \frac{1}{sum(w)} \sum_{i=0}^{n} d_i w_i$

[2]Available at http://sensormash.ucd.ie/ with special thanks to Thomas Holland for valuable help during development

Location	Active sensors	Centroid dist.	Weighted centroid
Location 1 (cube 4)	Bluetooth, Calendar	4.79	3.39
Location 2 (cube 2)	Bluetooth, Ubisense, Calendar	3.04	2.36
Location 3 (meeting area)	Ubisense, Calendar	7.84	2.07
Location 4 (water fountain)	Bluetooth, Ubisense	6.33	3.39
Location 5 (at lifts)	Ubisense, Bluetooth	8.75	2.08

Table 1. Location of office worker and average accuracy of basic centroid algorithm and weighted centroid algorithm (distances in metres)

The current implementation of SensorMash (see figure 2) includes inputs from Ubisense, Bluetooth, and Google Calendars. FOAF sensors provide information that identifies the current user, e.g. their mobile phone's Bluetooth MAC address or their calendar URL. Other metadata is used to describe the layout of the coordinate space, building floor plans, sensor locations and sensor characteristics.

The web-based interface is configured using the metadata for a particular user, and displays sliders for each sensor type. Every few seconds data from each sensor is retrieved from Construct and the filters applied based on the slider input. The derived information and original readings are communicated to the front end interface, which refreshes the display.

3. Tasting the Mash and Reflections on Initial Results

Five developers from the Construct project helped evaluate the efficacy of SensorMash by using it to explore the pervasive computing infrastructure at University College Dublin. We chose five areas on one floor of a building and monitored the location of an "office worker" using Bluetooth spotters, Ubisense and calendar entries as sensor inputs. The locations were chosen to give different combinations of sensors in action across the floor of the building. Whilst the office worker was at each location, the evaluators tracked the sensor activity through SensorMash and adjusted the sliders to alter the results of the sensor fusion algorithm. Their task was to manipulate the confidence values in an attempt to have the derived location of the office worker match his real locations. We recorded the sensor readings, output from a basic centroid fusion algorithm and the user weighted readings. The locations and results are presented in table 1.

The results show that in every scenario the evaluators increased the accuracy of the location value derived from the fused sensor readings by adjusting the sliders. The improvements varied from 70cm to 6.5m. Each evaluator was given a maximum of five minutes to do this for each scenario. In reality, the evaluators took much less than this at about 1-2 minutes on average.

The results suggest that that we can rapidly improve upon a base fusion centroid algorithm with our system. After the evaluators explored SensorMash they all agreed that it was a useful tool for gaining insight into their pervasive deployment. One user reported: *"the use of drag bars is an intuitive and fast way to get users to experiment with different confidence values"*.

Using SensorMash we surveyed the impact of three sensor types across a single floor in about thirty minutes. This allows a very rapid insight into the pervasive environment and the behaviour of deployed sensors. Furthermore, the impact of introducing new sensors into the infrastructure can be quickly and easily assessed through our visualisation.

During the trials we noticed that multiple combinations of slider values can lead to the same derived location being calculated. This is analogous to the problem of calculating a position of an object that is a known distance from only two points. Drawing arcs from those fixed points will intersect in two places. A solution to this is to introduce more fixed points (sensors) and employ a voting style system to derive the actual location. SensorMash can assist developers in discovering these ambiguous locations in a context-aware environment and address it.

An interesting observation from one of the participants was that in some cases a geographically close sensor should have its confidence value reduced. This happens when a user is not between sensors but on the outside edge of coverage. In this situation only the closest sensor reading should be taken (known as the "point" reading).

4. Conclusions and Future Directions

In this paper we presented SensorMash, an application for exploring the behaviour of pervasive systems with a sensor mashup visualisation. SensorMash allows real-time manipulation of individual sensor inputs to the system as it builds internal context models. It allows developers to rapidly gain an intuition into what the system is doing in terms of reacting to sensed data. We discussed our experiences applying SensorMash to a deployed system at University College Dublin resulting in an improvement in its fidelity.

Using the data derived from the user interaction allows us to model cases for weighting sensor readings in various locations. We are currently writing a extension to SensorMash which introduces a precision value. In this version a ground truth value is injected and the system *automatically* adjusts the sliders to have the fusion algorithm generate the correct value. The target is to minimise the precision value and maximise the confidence. This will allow rapid configuration of weightings for sensors towards their optimal. We will feed this back through Construct in an attempt to have the system self-optimise whilst applying a particle (Bayesian) filter.

References

[1] Simon Dobson, Paddy Nixon, Lorcan Coyle, Steve Neely, Graeme Stevenson, and Graham Williamson. Construct: An open source pervasive systems platform. *Consumer Communications and Networking Conference, 2007. CCNC 2007. 4th IEEE*, pages 1203–1204, 2007.

[2] Deborah Estrin, David Culler, Kris Pister, and Gaurav Sukhatme. Connecting the physical world with pervasive networks. *IEEE Pervasive Computing*, 1(1):59–69, 2002.

[3] Karen Henricksen, Jadwiga Indulska, and Andry Rakotonirainy. Modeling context information in pervasive computing systems. In *Pervasive '02: Proceedings of the First International Conference on Pervasive Computing*, pages 167–180, London, UK, 2002. Springer-Verlag.

[4] Jeffrey Hightower and Gaetano Borriello. Particle filters for location estimation in ubiquitous computing: A case study. In Nigel Davies, Elizabeth Mynatt, and Itiro Siio, editors, *Proceedings of the Sixth International Conference on Ubiquitous Computing (Ubicomp 2004)*, volume 3205 of *Lecture Notes in Computer Science*, pages 88–106. Springer-Verlag, September 2004.

[5] Thomas Strang and Claudia Popien. A context modeling survey. In *Workshop on Advanced Context Modelling, Reasoning and Management as part of UbiComp 2004 - The Sixth International Conference on Ubiquitous Computing*, September 2004.