PerAdaMagazine
towards pervasive adaptation
10.2417/2200809.1262

An open-source infrastructure for pervasive computing

Simon Dobson, Graeme Stevenson, Graham Williamson, Stephen Knox, Matthew Stabeler, Lorcan Coyle, Steve Neely, and Paddy Nixon

New software for gathering and disseminating information makes it easier to develop services for sensor-rich environments.

Pervasive computing provides a means of broadening and deepening the reach of information technology (IT) in society. It can be used to simplify interactions with Web sites, provide advanced location-specific services for people on the move, and support all aspects of citizens' life in the community. Integrating IT services into everyday life requires that we can sense the environment where services are offered, and tailor them as the environment changes. People are not automata, however, and will often perform the same activity in slightly different ways. Moreover, the methods used to sense a person's actions are inherently error-prone and imprecise, and the same events may be observed from different sensors or information sources. Users' support needs also evolve over time. These triple problems of *situation identification*, *context fusion*, and *behavioural evolution* constitute the major challenges to building robust pervasive applications or services.

Implementing individual pervasive applications, such as tour guides[1] or healthcare,[2] has been straightforward. But it has proved more difficult to build pervasive systems in which a dynamic population of services share infrastructure, sensing, and capabilities. Each new system requires a considerable investment of time to acquire expertise and money to create the necessary infrastructure. We aim to reduce these barriers and simplify the construction of extensible, long-lived pervasive systems.

We have developed our system, Construct, by identifying the best-of-breed techniques that have been successfully implemented for pervasive systems. We have collected these together into a middleware platform, an intermediary between sensors and services. Construct provides a uniform framework for situation identification and context fusion, while providing transparent data dissemination and node management.[3]

Construct's basic architecture (see Figure 1) relies on services and sensors that access a distributed collection of nodes, which are responsible for aggregating data from the sensors. Construct regards all data sources as sensors: for example, physical ones for temperature, pressure, and location are included along with virtual ones that access digital and Web resources.

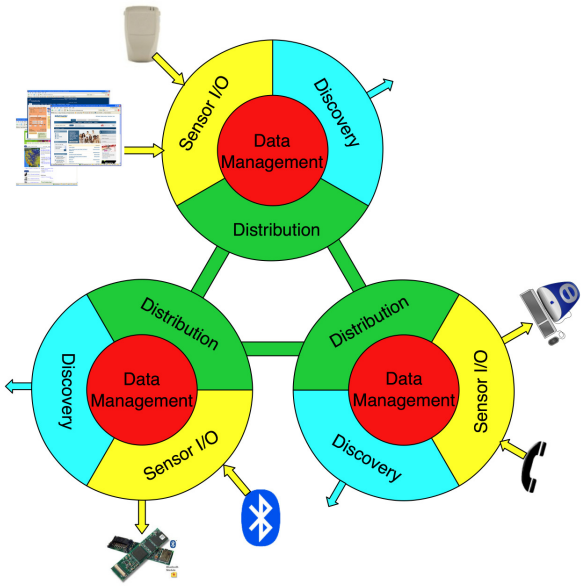

Figure 1. *Data from sensors like Bluetooth or RFID is aggregated by nodes, which then disseminate the information.*

A sensor injects information into Construct's resource description framework (RDF) triple-store[4] database. The triple store provides a set of common descriptions for concepts across domains.[5] This model means that different sensors can be used to detect the same information. Location may be sensed directly from RFID (radio frequency identification) or Ubisense, or inferred from diary or proximity information. Yet all this information can be accessed by services using a common data model. To request information from the database, applications query the triple store using the standard SPARQL language.

Construct does not provide remote access to sensors: instead, sensor data is transmitted around the network using the Zeroconf protocol[6] for node discovery and gossiping to exchange

Continued on next page

data.[7] Gossiping means that nodes randomly synchronise their triple stores. This can lead to substantial background communications traffic, but increases the robustness of the system, since a node failure will not cause sensed data to be lost. It can also improve responsiveness, since all requests for information from the database are performed locally. Taken together, these approaches mean that developers need never deal with distributed systems issues, since these are taken care of by the middleware. The Construct architecture is completely modular, allowing us to experiment with data dissemination protocols like adaptive gossiping[8] and piggybacking.

We used a suite of sensors including Ubisense, Bluetooth location, diary and web scraping for an example service called Basadaeir (the Irish word for matchmaker). This program aims to provide a context-sensitive display of information that interests a group of people. We use Bluetooth sensing of people's mobile phones to detect their presence near the display, and fuse this with information about their research interests, available papers and events, and joint projects. The resulting display shows information tied to the common research interests of the individuals present, by using a combination of location sensing, digital information and social networking.

Construct provides a useful platform for both researchers and practitioners. By providing a foundation for creating and exchanging components, we aim to simplify the development of systems, and speed up the convergence of best practices and design patterns. Our next steps include increasing the population of sensors and exploring alternative gossiping strategies that can reduce communications overhead. Beta-test releases of Construct and various sensors are available.[9]

This work is partially supported by Science Foundation Ireland under grants 04/RPI/1544, 'Secure and Predictable Pervasive Computing,' 03/CE2/I303-1, 'Lero: The Irish Software engineering Research Centre,' and 05/RFP/CMS0062, 'Towards a Semantics of Pervasive Computing.'

Author Information

Simon Dobson, Graeme Stevenson, Graham Williamson, Stephen Knox, Matthew Stabeler, Lorcan Coyle, Steve Neely, and Paddy Nixon
School of Computer Science and Informatics
University College Dublin
Dublin, Ireland

References

1. K. Cheverst, N. Davies, K. Mitchell, A. Friday, and C. Efstratiou, *Developing a context-aware electronic tourist guide: some issues and experiences*, **Proc. ACM Conf. Comput.-Hum. Interact.**, pp. 17–24, 2000.
2. G. Hayes, G. Abowd, J. Davis, M. Blount, M. Ebling, and E. Mynatt, *Opportunities for pervasive computing in chronic cancer care*, in **Pervasive Computing**, **LNCS 5013**, pp. 262–279, Springer, 2008.
3. L. Coyle, S. Neely, G. Stevenson, M. Sullivan, S. Dobson, and P. Nixon, *Sensor fusion-based middleware for smart homes*, **Int'l J. Assist. Robot. Mechatron. 8** (2), pp. 53–60, 2007.
4. O. Lassila and R. Swick, *Resource Description Framework model and syntax specification* tech. rep., World Wide Web Consortium, 1999. http://www.w3.org/TR/1999/REC-rdf-syntax-19990222/
5. A. Clear, S. Knox, J. Ye, L. Coyle, S. Dobson, and P. Nixon, *Integrating multiple contexts and ontologies in a pervasive computing framework*, **ECAI06**, pp. 20–25, 2006.
6. D. Steinberg and S. Cheshire, **Zero Configuration Networking: The Definitive Guide**, O'Reilly Media, Sebastopol, CA, 2005.
7. P. Eugster, R. Guerraoui, A. Kermarrec, and L. Massoulie, *Epidemic information dissemination in distributed systems*, **IEEE Comput. 37** (5), pp. 60–67, 2004.
8. S. Anawar, L. Coyle, S. Dobson, and P. Nixon, *Context delivery in ad hoc networks using enhanced gossiping algorithms*, **Proc. 1st Eur. Conf. Smart Sens. Context 4272**, 2006.
9. http://www.construct-infrastructure.org Accessed 18 September 2008.

© 2008 PerAda