Common models, patterns and best practices for pervasive systems design, implementation and evaluation

Roy H. Campbell
Sohaib and Sara Abbasi Professor of Computer Science
Information Trust Institute, University of Illinois at Urbana-Champaign
Urbana, IL 61820 email rhc@uiuc.edu

Practical experience building Gaia (Active Spaces), TEEVE (Tele-immersive Environment for Everybody), and Matrix (A Samsung sponsored project examining Active Spaces on PDAs) suggest certain common models, patterns and best practices aid the construction of pervasive and ubiquitous systems. While sharing many concerns with other software systems, these kinds of systems have particular problems. For example, these systems must be robust and provide highly available and reliable service without intervention of the user or manufacturer. These concerns form the focus of this position paper.

The Gaia system depended on a distributed object, client/server architecture. Pervasive computing required the use of discovery, authentication, events and notification, repositories, location, and trading. Since Gaia, TEEVE, a tele-immersive environment and Matrix, a Gaia-like application for mobiles, produced research prototypes with many similar architectural similarities. Availability of these services requires fault-tolerance and robustness. To create active spaces, Gaia coordinated the actions of tens of distributed systems and involved many services. Initialization required orchestrating many different software components implemented on many different machines. A heart-beat mechanism detected failures and scripting allowed many different configurations of the system be brought up in rapid succession should failures occur. Likewise in TEEVE, a room had from four to over forty cameras feeding 3-D video into a collaborative, tele-immersive environment. Four cameras constitute a 3-D camera with a particular view and a room has multiple views. All the cameras have synchronized shutters using serial lines and the output for each 3-D camera is processed by an independent computer. Again, the system was implemented as a client-server architecture and the availability of the system depended on making the components fault-tolerant and robust. Initialization and camera calibration posed problems similar to Gaia. The Matrix project attempts to build user-centric active spaces on mobile platforms. Groups of devices can share information and participate in mobile active spaces. Interconnectivity is accomplished over various changing networking media: Bluetooth, WiFi, IR, or mobile communications. Robustness for these active spaces is a research concern.

Our research and that of others suggests that ubiquitous systems must be based on good software engineering practices to achieve reliability. Although many systems have been built by researchers and developers, our research suggests these systems are best developed as frameworks of modular software with well-specified interfaces and standardized distributed object protocols that allow loose-coupling between components. The loose-coupling of components allows robust, reconfigurable implementations and initialization. Further, in many distributed ubiquitous systems the systems must either be constructed to be self-stabilizing or

should incorporate “self-aware” control mechanisms that monitor the state of the system and adapt the available components and hardware into working implementations, accommodating software and hardware failures. Many pervasive and ubiquitous systems aim at interactions with people requiring an accurate user model.

Benchmarks for ubiquitous systems are difficult to devise. Ultimately, most systems do not need extremely accurate timing models but do need to act within the acceptable limits of users and their ability to perceive latency. Consistency of behavior and timing is very important in this regard. Multi-stream multimedia presentations are particularly sensitive to timing inconsistencies. Benchmarks of common ubiquitous computing mechanisms would be appropriate. Examples would include discovery, location, authentication, access, joining and leaving a group, and querying the configuration of composition of a ubiquitous system. Perhaps a good benchmark to devise would be user notification systems with well-known timing and performance characteristics for different kinds of user interaction since most ubiquitous systems must notify users of errors, failures, and results.

References

Towards Fault Tolerant Pervasive Computing. Shiva Chetan, Anand Ranganathan, and Roy H. Campbell. In *IEEE Technology and Society* . Volume: 24, No. 1, pp 38-44, Spring 2005.

An infrastructure for context-awareness based on first order logic. Anand Ranganathan, and Roy H. Campbell. *Personal and Ubiquitous Computing* . (2003) 7:353-364.

Use of Ontologies in a Pervasive Computing Environment. Anand Ranganathan, Robert E. McGrath, Roy H. Campbell, M. Dennis Mickunas. *The Knowledge Engineering Review,* Vol 18:3, 209-220, 2004, Cambridge University Press.

Gaia: A Middleware Infrastructure to Enable Active Spaces. Manuel Román, Christopher K. Hess, Renato Cerqueira, Anand Ranganathan, Roy H. Campbell, and Klara Nahrstedt, In *IEEE Pervasive Computing*, pp. 74-83, Oct-Dec 2002.

Providing Middleware Support for Active Space Applications. Manuel Roman and Roy H. Campbell, In *Proc. of ACM/IFIP/USENIX International Middleware Conference (Middleware 2003)*, Rio de Janeiro, Brazil, 2003.

Towards a Pervasive Computing Benchmark. Anand Ranganathan, Jalal Al-Muhtadi, Jacob Biehl, Brian Ziebart, Roy Campbell, and Brian Bailey, In *PerWare '05 (Workshop on Middleware Support for Pervasive Computing) at the IEEE International Conference on Pervasive Computing and Communications (PerCom 2005)* ,Kauai Island, Hawaii, March 8-12, 2005.

Zhenyu Yang, Yi Cui, Zahid Anwar, Robert Bocchino, Nadir Kiyanclar, Klara Nahrstedt, Roy H. Campbell, William Yurcik, **Real-Time 3D Video Compression for Tele-Immersive Environments**, in *Proc. of SPIE/ACM Multimedia Computing and Networking (MMCN'06)*, San Jose, CA, 2006.