Ubiquitous Computing & The Role of Geometry

Barry Brumitt, John Krumm, Brian Meyers, Steven Shafer
{barry, jckrumm, brianme, stevensh}@microsoft.com
Microsoft Research

Abstract

Mark Weiser described ubiquitous computing as, "invisible, everywhere computing that does not live on a personal device of any sort, but is in the woodwork everywhere."[11] The EasyLiving project at Microsoft Research is focused on those aspects of ubiquitous computing relevant to smart environments, including work in distributed computing, geometric world modeling, computer vision, and user interfaces. Though the need for research in distributed computing, perception, and interfaces is widely recognized, the importance of an explicit geometric world model for ubiquitous computing has not been well-articulated. This paper elucidates the role of geometry in ubiquitous computing, offering example scenarios which require or benefit greatly from geometric knowledge, and describing four primary benefits of a geometric model.

Introduction

The goal of EasyLiving research project[8] is to develop a prototype architecture and necessary technologies for intelligent environments. EasyLiving concentrates on applications where interactions with computing can be extended beyond the confines of the current desktop model. In future homes and offices, access to computing should be as natural as access to lighting. Such a computing system should maintain an awareness of its occupants, understand their physical and functional relationship to I/O devices, respond to voice and gesture commands, and be easily extended. This technology will, for instance, enable a home's resident to make a phone call by simply speaking his intentions from where ever he happens to be. It will allow a user to move from room to room while still maintaining an interactive session with the computer. All these tasks require the coordination of many devices for computational activities involving both perception (Where is the user now?) and interaction ("Call Bob.")

In such a space (one populated by many small, networked computing devices) several devices will typically have to work together to perform a particular task. Dynamically collecting a group of smart devices to enable an interaction or to perform a perceptual task requires a shared computational substrate that allows the devices to communicate bits which represent concepts in a shared ontology. For example, three devices might announce to each other over a wireless network: "I am a display device", "I am a DVD Player", and "I am an acoustic speaker." Once these capabilities are known to exist in the same place, the ability for a user to play a movie should be enabled. If a pair of headphones with appropriate capabilities were to enter the fray ("I'm a pair of headphones"), redirecting the sound output to them should become an available option. A formalized protocol for this type of distributed computing, including both communications and capability self-description, is essential to dynamic device amalgamation for ubiquitous computing.

Beyond this requisite distributed computational substrate, a fundamental unifying aspect of a ubiquitous computing environment is a representation of the physical relationships between people, things, and devices -- in other words, a geometric model. While the need for an appropriate distributed computing infrastructure has been widely recognized, both in general (Sun's Jini[6], IBM's T-Spaces[12], Microsoft's DCOM/Universal Plug-and-Play[9]) and specifically for intelligent environments (Metaglue[4]), the importance of physical geometry has been largely overlooked.

This paper discusses the importance of geometry and geometric models for ubiquitous computing by illustrating their necessity with respect to two essential capabilities of ubiquitous computing systems. The paper also describes three insufficient approaches for incorporating that knowledge in smart environment systems.

Capability 1: Casual Access to Computing

Users should not be required to go to a special place (i.e. the desktop) to interact with the computer. Nor should they be required to wear special devices[10] or markers to have the computer know where they are. This goal of "casual access to computing" means that the computer should always be available anywhere in a suitably-equipped space. Through cameras, microphones, and other input devices, the user will always be able to signal the computer. Conversely, through displays, speakers, and specialized output devices, the computer will always be able to signal the user. With appropriate sensors, (cameras[8], pressure sensors[1], active badges[10], etc.), the computer will know the location and identity of users as well as the context[5] in a given space, thus allowing devices to signal the users in a suitable unobtrusive way.

Signaling requires knowledge of the physical space beyond merely the proximity of the user to appropriate devices. There are several ways in which the simple task of notifying the user of a phone call or E-mail

From: IEEE Personal Communications, Vol. 7, No. 5, Oct. 2000, pp. 41-43.

could fail without a suitably rich geometric model:

- The screen used for display (though inches away) is behind the user, so the displayed message cannot be seen.
- The speaker used for sounding a ringing signal is on the other side of a wall from the user, muffling the sound.
- Two people are in a room; one is typing on a keyboard. Which application receives the keystrokes?

With a geometric model., appropriate decisions can be made to select one device over another, based on their physical locations, e.g.: a large display far away might be more appropriate than a small display close up. It further enables the user interface designer to properly select and optimize an interaction based on the dynamic configuration of the I/O devices. While geometric knowledge alone is insufficient to select the best device for a given interaction (context and other world knowledge are helpful), it is necessary component for reaching the ideal decision.

Currently, computing and home electronic systems have an extremely impoverished world model. They lack significant knowledge about the layout of the physical space, the presence of other devices, or the current physical context. The addition of basic geometric knowledge to the ubiquitous computing system greatly increases the shared understanding between user and system. If the user requests “light”, he can reasonably expect that a light nearby will come on. Without this shared knowledge, the user would need to specify precisely which light (e.g. by name, ID number, network address, etc.), rather than being able to assume the system will interpret his request correctly. For computing to move off of the desktop and be accepted, it must have a comprehension of physical space which is related to that of the user, else the proliferation of smart devices will only increase the complexity of the user’s experience, instead of simplifying it.

Fundamentally, a geometric model can greatly improve the user’s experience in a system where devices and interactions are dynamic. This occurs because appropriate devices can be automatically selected, and because the system’s and the user’s knowledge of the physical world has been moved one step closer together.

Capability 2: Extensible Computing

Similar to the concept of “plug and play”, the capabilities of an intelligent environment should grow automatically as more hardware is added. One aspect of such extensibility is “resource extensibility”; new devices become new resources that the system can use at will. If, for instance, a CRT is added to the kitchen, it becomes a new way of presenting information in that space, and the system should automatically take advantage of it. Another aspect is “physical extensibility.” If a new camera is added, it not only extends the system's resources as a new device, but also extends the system's physical coverage.

The complexity introduced by physical extensibility can be simplified through the use of geometry. When trying to marshal an interaction, investigating the physical relationship of every possible combination of devices to the target user (or users) is a computationally expensive proposition, which grows combinatorially with the number of devices. If that lookup can be bounded by requesting only devices in the region surrounding the user, every device on the network need not be independently examined. Similarly, when one device is trying to communicate location-sensitive context, geometry gives a mechanism to avoid system-wide broadcast. For example, if the system wants to notify someone in a space via an acoustic signal, it need only check nearby audio-related devices to determine if that signal is likely to be heard. In other words, by utilizing information about the physical world, device lookups can be reliably and accurately scoped.

Resource extensibility is greatly eased by the abstraction layer that a geometric model provides. Consider a phone system which routes incoming calls to the physical phone nearest the intended recipient. Such a system could obtain person-location information directly from an active badge system[10]. However, what if different parts of the building utilize different localization technologies? The routing system would have to know explicitly about all different perception methodologies, and be able to speak to and understand them all. If, instead, a geometric model is used, the sensing components can provide information to this model, and the telephone system can query the model; neither information provider nor information consumer need know about each other. Additionally, this abstraction layer simplifies sensor fusion. Sensed geometric information will always have uncertainty associated with it. Worse yet, multiple sensors may perceive the world in inconsistent ways. A geometric model can hide this complexity, allowing applications which use geometric knowledge to be independent from a particular perception technology.

Existing Approaches

Many intelligent environment systems[2][3][7] have avoided using an explicit geometric model by either manually entering the physical configuration of devices, assuming network-presence translates to co-location, hand-coding the available interactions, or otherwise implicitly representing the requisite geometric information.

Fixed Configuration

While adequate for demonstration purposes, it is unreasonable to expect that future ubiquitous computing systems will be comprised only of static elements. Obviously, PDA's, cell phones and laptops will change locations frequently, and will be expected to work appropriately with an understanding of their locale. Another type of dynamicism occurs when a consumer purchases a new device for their home, car or office. To integrate this new device into the system, it is preferable to avoid requiring the user to take any action beyond merely plugging the device in, or otherwise giving it power and data connectivity.

Implicit Geometry

Beyond assuming a fixed configuration of devices, a straightforward next-approach is to use implicit geometry, assuming that network or data connectivity is equivalent to co-location. One might assume that if two devices can communicate directly (by RF, IR or other "local" transmission method), they are co-located. However, RF transmission (not to mention physical network protocols) can easily span rooms, floors or even buildings. Without some more precise model of geometry, making this type of assumption will result in a excessively large set of potentially available devices, many of which may not actually be available or usable for any particular task. For single-room demonstration systems, again, this assumption is reasonable -- it stops being viable when the system must scale to larger spaces.

Dedicated Devices

Another alternative to having a geometric model is to directly couple particular devices to particular tasks. However, this scales poorly, both in terms of system complexity and extensibility. For example, an IR motion sensor can be tied to the lights in the room. To make the IR sensor (which is really only for guessing room occupancy) useful for other tasks, it's necessary to explicitly hook up this sensor to the desired effect, (e.g. redirecting phone calls to that room.) As the number of I/O devices grows, the complexity of defining system behavior grows exponentially, as it is necessary to define all appropriate connections between devices.

Future Research

The above capabilities suggest several fruitful areas of research.

- **Geometric representation:** Given that multiple sensors will perceive possibly contradictory facts about the physical word, how can this information be combined reliably to support ubiquitous computing tasks? Given that these sensors may provide geometric information in different formats, how can they be integrated?
- **Geometric discovery:** How should different sensors be used to discover the physical relationship between entities in the world (i.e. people, devices and things). How can this process be made as automatic as possible?
- **Scalability:** As the number of users and devices in a location and/or on the network grows, how can the appropriate device be selected? How can the computational burden of maintaining a geometric representation be managed? How do devices come and go?
- **Movable interactions:** Given a changing space of devices available to a mobile user and a geometric representation of these entities, how should interactions move fluidly with the user?
- **User experience:** With the PC broken up into separate components, how can a consistent natural interface be presented?

Conclusions

This paper has described two important capabilities for ubiquitous computing systems, "Casual Access to Computing" and "Extensible Computing." Through those scenarios, four primary benefits of geometric models have been introduced:

- **Physical Parameters for UI's:** Device selection and control is performed with knowledge of physical context.
- **Shared Metaphor:** User experience is simplified through a common understanding of the physical world shared by system and user
- **Physically-scoped Lookup and Broadcast:** Geometric information limits the scope of lookup/broad-

cast services.

- **Sensor Abstraction Layer:** Applications are independent from perception hardware and methods.

While geometric knowledge remains challenging to gather, represent, and provide, its inclusion will significantly both improve the performance and speed the acceptance of ubiquitous computing systems

References

1. Addlesee, M.D. et al. "ORL Active Floor", IEEE Personal Communications, Vol.4, No.5, October 1997, pp. 35-41.
2. Bobick, A. et al. "Design Decisions for Interactive Environments: Evaluating the KidsRoom", Intelligent Environments, Papers from the 1998 AAAI Spring Symposium, March 23-25, 1998, Technical Report SS-98-02, AAAI Press.
3. Coen, M. et al, "Design Principals for Intelligent Environments", Intelligent Environments, Papers from the 1998 AAAI Spring Symposium, March 23-25, 1998, Technical Report SS-98-02, AAAI Press.
4. Coen, M. et al, "Meeting the Computation Needs of Intelligent Environments: The Metaglue Environment", Managing Interactions in Smart Environments, Springer-Verlag, 1999, pp. 201-213.
5. Dey, A. et al. "A Context-Based Infrastructure for Smart Environments", Managing Interactions in Smart Environments, Springer-Verlag, 1999, pp. 114-128.
6. Jini Overview", http://www.sun.com/jini/faqs/index.html
7. Mozer, M. "The Neural Network House: An Environment that Adapts to its Inhabitants", Intelligent Environments, Papers from the 1998 AAAI Spring Symposium, March 23-25, 1998, Technical Report SS-98-02, AAAI Press.
8. Shafer, S. et al."The New EasyLiving Project at Microsoft Research", *Proceedings of the 1998 DARPA / NIST Smart Spaces Workshop*, July 1998, pp.127-130.
9. "Universal Plug'n'Play", "http://www.microsoft.com/hwdev/upnp/"
10. Ward, A. et al. "A New Location Technique for the Active Office", IEEE Personal Communications, Vol. 4, No. 5, October 1997, pp. 42-47
11. Weiser, M. "Ubiquitous Computing", http://www.ubiq.com/hypertext/weiser/UbiHome.html
12. Wyckoff, P. et al. "T Spaces", http://www.ibm.com/java/education/t-spaces/t-spaces.html