A Conceptual Model for Pervasive Computing

Laurent Ciarletta, Alden Dima

A Conceptual Model for Pervasive Computing

Laurent Ciarletta, Alden Dima
Information Technology Laboratory
National Institute of Standards and Technology, Gaithersburg, MD, USA
{laurent.ciarletta,alden.dima}@nist.gov

Abstract

As an emerging field, pervasive computing has not had the opportunity to develop a conceptual model similar to the OSI Reference Model used to describe computer networks. Such a model would be useful in properly classifying design issues and providing needed context. Inspired by the layers of abstraction provided by the OSI Reference Model, we present our Layered Pervasive Computing (LPC) model to facilitate discussion and analysis of pervasive computing systems by providing a much needed conceptual framework. A key feature of our model is its representation of the human user at each layer of abstraction of the model. We will then use our model to analyze a research prototype created as part of our Aroma pervasive computing project. This analysis is illustrative because it quickly reveals issues that must be addressed to realize our research prototype as a commercial product.

1. Introduction

Pervasive computing is the result of the convergence to three areas of traditional computing: personal computing, embedded systems, and computer networking. It can be distinguished from computing in general by its emphasis on: ubiquity, interconnectedness and dynamism. Pervasive computing aspires to be ubiquitous; it strives to be low-cost, embedded, distributed and non-intrusive. Networking via both traditional wired and newer wireless technologies also plays a central role. Its dynamic nature is a result of its mobile and adaptive applications that are able to automatically discover and use remote services. The notion of information appliances (small, specially-designed computing devices) is central to pervasive computing. The rapid rise of personal digital appliances (PDAs) is but one indication of this new computing paradigm.

As an emerging field, pervasive computing has not had the opportunity to develop a conceptual model similar to the OSI network model [1]. Such a model would be useful to properly classify issues raised during discussion and provide needed context.

As part of our Aroma project, we developed a conceptual model for pervasive computing inspired by the OSI Reference Model. The goal of this project is to explore key technical and standardization issues in pervasive computing and to work with the community to begin resolving these issues. A key tenet is our belief that within five years, systems on a chip (SOC) will cost approximately $10 and include a pico-cellular wireless transceiver and a sufficiently rich run-time environment capable of running sophisticated virtual machines. This belief led us to focus our research in the following areas:

- Connecting portable wireless devices to traditional networks;
- Service discovery, self-configuration, and dynamic resource sharing;
- Mobile code and data; and
- Software infrastructure needed to create and manage pervasive services and applications.

We built an adapter capable of emulating future SOCs and of providing a platform to examine pervasive computing challenges. This adapter is based on embedded PC technology and can run Java and Jini-based applications. It communicates via a 2.4 GHz wireless LAN PCMCIA card. We've used it to implement a pervasive projector that serves as a prototype information appliance for the project.

While developing the adapter and projector, we began to feel the need for a conceptual model that could help us sort through the issues that we were uncovering. We felt that a properly designed conceptual model could help us focus on all relevant issues and not just certain technical issues. One thing that we focused on was the fact that some pervasive computing technologies have succeeded where others have failed. If the model was to be useful, it would have to be able to serve as a framework for discussion about the success or failure of a particular pervasive technology. It became clear that human beings are an integral part of pervasive computing and could not just be abstracted away.

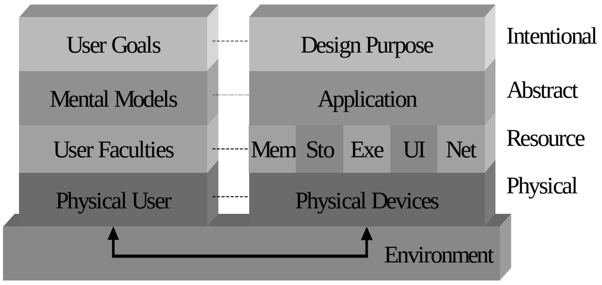

Figure 1. Aroma pervasive computing conceptual model

We were inspired by the OSI model but added a model of the user (Figure 1), the environment, and an intentional layer. Like the OSI model, our model is layered. While for devices, the higher layers represent increasing degrees of abstraction, for users, the higher layers represent increasing temporal specificity. This means that change occurs more slowly at the lower levels. For example, a user's goals in using a device may change by the minute, but his physical characteristics take much longer to change. User faculties, as described below, take time to develop, but through training and practice can be acquired in a reasonable amount of time. The remainder of this paper will describe the semantics of this conceptual model from the bottom up.

2. The Environment

In traditional computing, the physical environment is regarded as irrelevant unless it becomes incompatible with the operation of the computer, in which case it is a nuisance to be engineered away. The computer is logically self-contained and users can be viewed as simply a source of input for a process, not unlike a file. Pervasive computing is ubiquitous, interconnected and dynamic which implies that it is embedded, distributed and mobile.

The environment cannot be ignored, it must be factored into the conceptual model. The embedded aspects of pervasive computing suggest that sensors and actuators will play a significant role in many pervasive applications. This implies that there will be issues that exist "beneath" the physical layer of the computer, beyond the sensors and actuators. Interconnectedness, especially via wireless technologies, also implies again that environmental issues will play a significant role. Finally, but most significantly, the mobile nature of many pervasive computing systems ensures that the environment's presence will determine the "semantics" of pervasive computing – the very meaning of the term "pervasive" will depend on whether the device can cope with a wide variation in its surrounding environment while performing its intended function. In the past, environmental issues were simply relegated to the physical layer, but it is more than just convenient to create a separate environment layer. The focus of the physical layer is typically on engineered devices; in the climate controlled conditions of traditional computing, the environment is just another engineered component. The mobile nature of many pervasive computing applications ensures that the environment cannot just be engineered into submission. A new set of issues arises which are best served by having a separate layer to contain them.

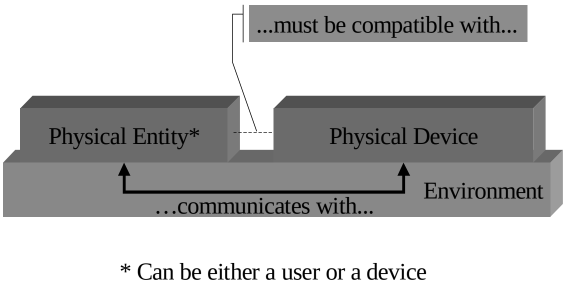
...must be compatible with...
Physical Entity*
Physical Device
Environment
…communicates with...
* Can be either a user or a device

Figure 2. Environment and physical layers represent the physical environment, the user physiology and the physical devices of a pervasive computing system.

3. The Physical Layer

The physical layer is typically thought of as consisting of the hardware aspects of the computing system. We believe that for pervasive computing, the physical user must also be included. By this we mean the user's body and the signals it is capable of sending and receiving. Many pervasive computing applications involve speech recognition and user biometric identification for security purposes – the flow of control in such an application depends on the signal received from the user’s body.

But there are other issues as well. A PDA that does not properly consider human physical characteristics in its design is doomed to failure even though it may have a brilliant software architecture. Unlike a desktop personal computer where an offending monitor or mouse can be replaced, many information appliances will not be modular in nature. Failure to meet user physical requirements means the device itself has failed to fulfill its intended purpose.

A key issue in the physical layer will be to ensure that its entities are physically compatible with one another. Some of the compatibility issues are electrical or mechanical in nature, others will be physiological. Since the functioning of physical devices and human physical responses depend on the surrounding environment, the physical layer can be thought of as being “above” the environmental one.

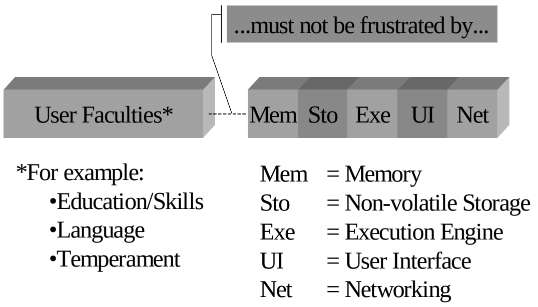

Figure 3. The resource layer serves as the foundation for the application software. It contains the available computational resources and assumed user skills and abilities that developers can count on being present.

4. The Resource Layer

The resource layer can be defined by the question: “ What can we count on being available? ”. On the device side of the model, the resource layer sits at a higher level of abstraction than the physical layer and therefore defines the set of logical resources available for the device. In traditional terms, this includes items such as the operating system (and the logical devices that it presents to the user), the user interface (e.g., the window system), system APIs, and the network protocol stack. It is on this foundation, that a pervasive computing software application is created.

For the user, this layer represents the user faculties that an application developer can assume are present. The term “faculty” here means a developed skill or ability such as a user’s ability to speak a particular language, the user’s education or even the user’s temperament (for example, the ability to tolerate frustration). User faculties will vary widely depending on the user physical condition and thus are supported by the physical layer.

Many developers make implicit assumptions about a user faculties – they are in essence, counting on the user to have certain skills and abilities. Being able to expect that all users will speak the same language is fundamentally a resource that the developer can count on and that will impact the design of system software. In more traditional computing systems, erroneous assumptions about the user can usually be corrected with the next release. In an information appliance that has its operating software burned into ROM, faulty assumptions are costly. Our model places human faculties in the resource layer to ensure that the appropriate decisions are made at an economically appropriate time.

A fundamental issue in this layer is to ensure that user faculties are not frustrated by the logical resources of the device. In general, system resources must be matched to user goals and needs. Assuming that all users are similar to fellow developers is a sure-fire way to discover that they don’t have the same ability to deal with the arcane “features” imposed by the selection of system software. Users typically have immediate tasks to perform, so user interfaces must be intuitive and consistent, and must accommodate user’s language and physical limitations. Users are not system administrators, so networking features should be automatically available, self-configuring and compatible with existing technologies. The device’s mass storage must support the user’s need to access and retrieve information. This is not just an issue of capacity and speed, but of allowing users to flexibly organize information in a manner that suits their purposes. A device’s execution environment and volatile memory must be sufficiently responsive and yet use other resources economically. Again, this is not just an issue of speed, but also of responsiveness and control: a single-threaded system that does not allow a user to abort a task causes needless frustration and will ultimately alter the patterns of usage.

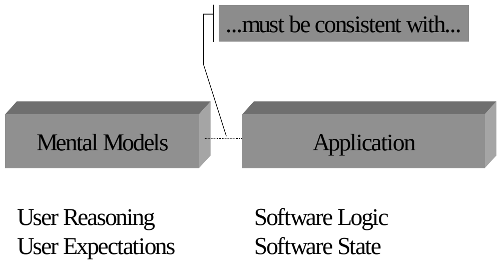

Figure 4. The abstract layer corresponds to the user mental representations and the application software.

5. The Abstract Layer

The abstract layer is similar to the application layer in the OSI model in that it contains the pervasive computing application software. We have however added the notion of user mental models to this layer. The key issue that must be addressed in this layer is maintaining consistency between the user's reasoning and expectations and the logic and state of the application. Anyone who has written and debugged software will understand the interplay between mental models and software behavior. Unfortunately, for too many users, using software becomes a mental exercise similar to debugging. Desktop users have the luxury of on-line help systems, Internet access, and bookshelves. These items will not always be available to pervasive computing users, but more important is the fact that pervasive computing activity will tend to be more immediate and focused. A desktop user may spend hours writing a conference article and will spend five minutes reading about a needed word processor feature. A PDA user trying to quickly schedule an appointment will not have the patience to spend five minutes using on-line help to relearn how to use a seldom used feature.

The mental models that a user can create will depend greatly on his faculties, which are in turn dependent on the physical condition of the user and the environment. Concepts that seem intuitive to a computer scientist sitting in a quiet office, may be difficult to grasp for a non-technical user riding the subway with a headache. Traditional computing has experienced these issues. In many cases, PC developers have used common metaphors, application frameworks and toolkits to present a consistent interaction model that goes beyond mere look-and-feel by eliminating unnecessary surprises. Most users know what to expect when they ask an application to perform a given task and their mental models tend to be portable across existing applications.

The nature of pervasive computing will place restrictions on the complexity that users will be able to cope with and will also tend to limit the techniques that developers can use to communicate the state of the application. As a result, pervasive developers will be forced to consider the user's point-of-view much more than developers in traditional environments. We've placed the user's mental model in the abstract layer to help ensure that application developers consider it during development.

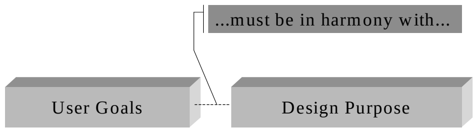

Figure 5. The intentional layer deals with issues related to the user's intentions and the purpose of a computing system.

6. The Intentional Layer

Models such as the OSI model usually stop at the application layer. We believe that there is a higher layer, which we call the intentional layer. This layer represents the purpose of an application or device and the goals of the user. Let us first consider the user. Stopping at the abstract layer leaves us with the user's mental representations but without any idea of why they were created in the first place. We can make reasonable guesses as to the user's goals and intentions but we have no way of representing them.

The design of pervasive applications and devices reflects either stated or implicit requirements addressed by their designers. These requirements reflect the device's purpose – the reason it was created and the needs it attempts to fulfill. While requirements are an important part of development there is a tendency for them to be relegated to a secondary role as the development process takes on a life of its own.

The impersonal focus of many technical fields often gives us the illusion that we can safely ignore these concerns in a technical analysis. But the history of computing is replete with failures of technically “superior” products and the success of “inferior” ones. An impersonal analysis of these events leads many to confusion, frustration and cynicism without offering tools to address the future. We believe that the probability of success is greatly enhanced when a system’s design is in harmony with the user’s goals. By truly understanding a user’s goals, designers can develop a system that works towards fulfilling them.

7. Analysis of a Pervasive Computing System

The central focus of our Aroma project is to address technical, standardization and measurement issues related to the networking and software aspects of pervasive computing. We’ve decided to simulate the capabilities of the future Information Appliances via our Aroma Adapter, an embedded PC capable of running pervasive computing software [2], and by developing some “challenge applications”.

These challenge applications will serve as a test bed for emerging wireless LAN technologies and service discovery protocols. Our first application is the Smart Projector, which consists of a commercially available digital projector, the Aroma Adapter, and the Java™/Jini™-based services and clients that allow this projector to export two services:

- Projection of a remote laptop display; and
- Remote control of the projector.

We can partition this pervasive computing application using the five layers of our model. For the sake of this discussion, we will not give a detailed analysis, but will use the model only to categorize concerns into the appropriate layer of our model. There are four major physical and logical entities in our example:

- A user wishing to make a presentation to an audience;
- The laptop used to make the presentation;
- The smart projector consisting of the projector, the Aroma Adapter and related software; and
- The Jini™ Lookup Service.

From a conceptual level, however, we will focus on two: the presenter and the remote projection service. We will not concern ourselves with the audience.

7.1. The Intentional Layer

It is important to ensure that a pervasive computing systems design be in harmony with the needs of its users. This statement seems so obvious that it is easy to dismiss it. At first glance, the Smart Projector’s intentional layer appears fairly simple. Let’s assume that a user wants to make a presentation, but does not necessarily want to perform unnecessary system interconnection and configuration. Then we could say that the Smart Projector is designed to provide a remote projection service that meets these needs. But this is not an honest answer.

Our intended audience is a group of computer scientists performing pervasive computing research. Our Smart Projector is designed as a vehicle to research, measure, and demonstrate service discovery and other pervasive computing infrastructure issues. Our research-oriented design goals mean that in its present form, our Smart Projector will not necessarily be in harmony with the needs of a casual user expecting a commercial-grade product, but it does satisfy the needs of its intended users.

7.2. The Abstract Layer

In its current implementation, the Smart Projector has two separate services, one to project the user’s presentation and one to control the projector. AT&T’s Virtual Network Computer (VNC) is used to make the laptop display available to the Aroma adapter which in turn displays it via the projector. Session objects are used to ensure that another user cannot inadvertently “hijack” either the use or control of the projector.

In order to successfully use the Smart Projector, the user must form a mental model that is consistent with the design of the system. The user must understand that both clients must be started in order to project and control the Smart Projector from a single laptop. Likewise, when finished, the user must stop both clients. The VNC server must also be started on the laptop for projection to succeed. The user must realize that only one person can use either service at a time. Since the Smart Projector is a research prototype, its operation is more complex than would be tolerated for a commercial product with similar functionality. However, it serves to illustrate that even relatively simple applications can place a conceptual burden on its users. If this burden is greater than what users are willing to bear in meeting their goals, then the system will not be used.

Future research and development in the abstract layer should focus on improving user access to these pervasive services by minimizing the conceptual burden placed on the user. Service discovery technologies can be better integrated into the window system used by the OS to better determine the user’s context and intentions. For example, if the Smart Projector’s services are currently not available, the icons on the user’s desktop should change their appearance accordingly. Mechanisms must be developed to manage interrelated services to gracefully resolve issues related to attempts by multiple users to access the services in different orders with minimal user intervention. Other mechanisms should be developed to deal with users who forget to relinquish control of the projector without relying on a system administrator to intervene.

7.3. The Resource Layer

The resource layer supports the abstract layer by providing the foundation on which applications can be built. The presence of logical entities in the resource layer, such as the network, ultimately leads to application code that makes use of it. This in turn affects the user mental models required to understand and use the system.

The application layer also makes demands of the resource layer. Java^{TM}-based technologies and VNC are used to implement the Smart Projector software, so we expect that they will be available on the user’s laptop. The ability to automatically discover the projector service is implemented using Jini and relies on having a Jini lookup service present.

User faculties are an important part of the resource layer. As the developers of the projector, we assumed that the user has a basic understanding of graphical user interfaces and is able to speak English. We have also assumed that users have a basic understanding of projectors and their operation, and that they are capable of fixing whatever problems may arise with the wireless network, the Linux-based adapter, and the lookup service. These expectations are not unreasonable since they describe the situation found in our laboratory. A number of these expectations, however, are unreasonable if the Smart Projector is used outside our laboratory. To move beyond our current implementation, we will have to research pervasive computing application deployment, automated diagnostics, fault tolerance and recovery, internationalization and accessibility issues. By using our conceptual model, we are better able to see our underlying assumptions and direct our future research and development.

7.4. The Physical Layer

The Smart Projector physical layer consists of a number of physical entities and their interactions. Among these are physical devices such as the wireless network adapters, the digital projector and the Aroma Adapter hardware. One physical layer issue that we have encountered is the relatively low bandwidth of current wireless networking adapters. Their use in our application prevents us from displaying rapid animation.

The user as a physical entity also belongs in this layer, although in this case, there are few physical human-computer interaction issues. One user-related physical issue centers on the use of a personal computer (the user’s laptop) for controlling the projector. This directly constrains the presenter by requiring physical proximity to the laptop. A truly pervasive system should place minimal physical constraints on its users. A future version of the Smart Projector could conceivably offer voice control, in which case human physical characteristics will play a greater role in the physical layer.

7.5. The Environment

While working with the Smart Projector, we have identified several environmental issues. For example, we are using wireless networking technologies with ranging, radio interference and scaling constraints. There are many wireless devices operating in the 2.4GHz radio band, and the effect of a high concentration of these devices needs to be studied.

Environmental issues can also affect users. Background noise, that is currently acceptable, may become objectionable if voice recognition is used in a pervasive computing system. If the noise is due to background conversation, then restrictions may have to be placed on the social interactions allowed while using the system. Conversely, the use of voice-based devices may be socially inappropriate in a cramped office environment with cubicles, regardless of the abilities of the voice recognition system.

8. Conclusion

We have presented a conceptual model useful for properly classifying issues raised during discussion and analysis of pervasive computing systems. Like the OSI Reference Model, it offers multiple layers of abstraction and serves as an abstract framework that allows its users to place issues in their appropriate context. A key feature of our model is its representation of the human user at each layer of abstraction of a pervasive computing system.

We then used our model to analyze a prototype Smart Projector created as part of our Aroma pervasive computing project. This analysis was illustrative because it quickly revealed several issues that must be addressed, particularly at the higher levels of abstraction, to realize our prototype as a commercial product.

9. References

[1] J. D. Day and H. Zimmerman, The OSI Reference Model, Proceedings of the IEEE, vol. 71, pp. 1334-1340, December 1983.

[2] K. L. Mills, AirJava: Networking for Smart Spaces, Usenix Embedded Systems Workshop, March 1999.