An Application Model for Pervasive Computing

Sasikumar Gurumurthy, M. Narayanamoorthy, A. Valarmozhi

Abstract—**This paper challenges the mobile computing community by questioning the roles of devices, applications, and a user's environment. A vision of pervasive computing is described, along with attributes of a new application model that supports this vision to reality. Pervasive computing is more art than science. It will remain this way as long as people continue to view mobile computing devices as mini-desktops, applications as pro-grams that run on these devices, and the environment as a virtual space that a user enters to perform a task and leaves when the task is finished. This paper challenges the mobile computing community to adopt a new view of devices and its applications.**

Index Terms—**Mobile Computing, Pervasive Computing, Devices, desktops, virtual space.**

I. Introduction

Pervasive computing encompasses three things all of these areas: 1. It concerns the way people view mobile computing devices, and use them within their environments to perform tasks. 2. It concerns the way applications are created and deployed to enable such tasks to be performed.3. It concerns the environment and how it is enhanced by the emergence and ubiquity of new information and functionality. Pervasive computing is the trend towards increasingly ubiquitous (another name for the movement is ubiquitous computing), connected computing devices in the environment, a trend being brought about by a convergence of advanced electronic - and particularly, wireless - technologies and the Internet. Pervasive computing devices are not personal computers as we tend to think of them, but very tiny - even invisible - devices, either mobile or embedded in almost any type of object imaginable, including cars, tools, appliances, clothing and various consumer goods - all communicating through increasingly interconnected networks.

Sasikumar Gurumurthy is with the VIT University, Vellore, Tamil Nadu, India. He is now with the School of computing science and Engineering. He also persuing his Ph.D degree in Soft Computing in Vellor Institute of Technolology,Vellore.His current research focuses on Wireless Networks,Artificial Intelligence He is member of IAENG (corresponding author to provide phone: 0416-2202016; fax: 0416-2243092; e-mail: g.sasikumar@vit.ac.in).

M.Narayanamoorthy is with the VIT University, Vellore, Tamil Nadu, India. He is now with the School of computing science and Engineering (corresponding author to provide phone: 0416-2202016; fax: 0416-2243092; e-mail: mnarayanamoorthy@vit.ac.in).

Ms.A.Valarmozhi is with the Bannari Amman Institute of Technology, Sathyamangalam, Tamil Nadu, India. She is now with the Department of Information Technology (corresponding author to provide e-mail: valar_113@yahoo.co.in).

Today, pervasive computing is more art than science. It will remain this way as long as people continue to view mobile computing devices as mini-desktops, applications as programs that run on these devices, and the environment as a virtual space that a user enters to perform a task and leaves when the task is finished. This paper challenges the mobile computing community to adopt a new view of devices, applications and environment.

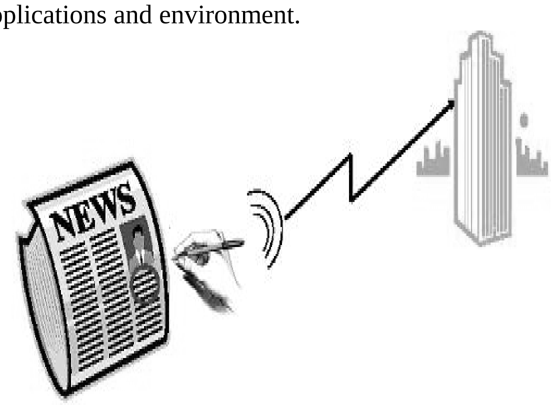


Fig. 1. Pervasive Computing Application.

Specially, our vision can be summarized in the following precepts: 1. A device is a portal into an application/data space. 2. An application is a means by which a user performs a task.3. The computing environment is the user's information enhanced physical surroundings. A new application model is needed to support this vision. This paper describes the attributes of such a model. Pervasive: Tending to pervade or spread throughout. Definition**:**" The most profound technologies are those that disappear. They weave themselves into the fabric of everyday life until they are indistinguishable from it"–Mark Weiser1991 Scientific America.

II. Wireless Network

Wireless is a term used to describe telecommunications in which electromagnetic waves (rather than some form of wire) carry the signal over part or the entire communication path. Some monitoring devices, such as intrusion alarms, employ acoustic waves at frequencies above the range of human hearing; these are also sometimes classified as wireless. The first wireless transmitters went on the air in the early 20th century using radiotelegraphy (Morse code).[4] Later, as modulation made it possible to transmit voices and music via wireless, the medium came to be called "radio." With the advent of television, fax, data communication, and the effective use of a larger portion of the spectrum, the term "wireless" has been resurrected.

ISBN: 978-988-18210-3-4
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

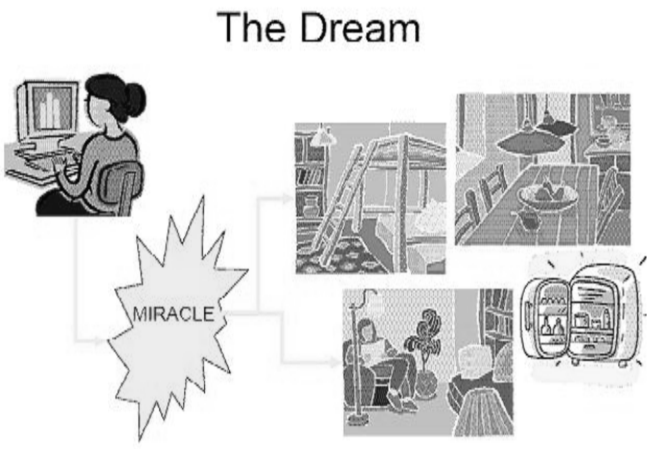


Fig. 2. Wireless equipments with different Application

Common examples of wireless equipment in use today include:

Cellular phones and pagers -- provide connectivity for portable and mobile applications, both personal and business

Global Positioning System (GPS) -- allows drivers of cars and trucks, captains of boats and ships, and pilots of aircraft to ascertain their location anywhere on earth

Cordless computer peripherals -- the cordless mouse is a common example; keyboards and printers can also be linked to a computer via wireless

Cordless telephone sets -- these are limited-range devices, not to be confused with cell phones

Home-entertainment-system control boxes -- the VCR control and the TV channel control are the most common examples; some hi-fi sound systems and FM broadcast receivers also use this technology

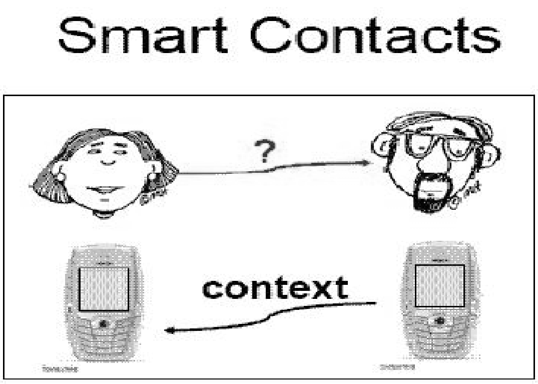
Remote garage-door openers -- one of the oldest wireless devices in common use by consumers; usually operates at radio frequencies

Two-way radios -- this includes Amateur and Citizens Radio Service, as well as business, marine, and military communications

Baby monitors -- these devices are simplified radio transmitter/receiver units with limited range

Satellite television -- allows viewers in almost any location to select from hundreds of channels

Wireless LANs or local area networks -- provide flexibility and reliability for business computer users.



. Fig. 3. Wireless Devices with different application

Wireless technology is rapidly evolving, and is playing an increasing role in the lives of people throughout the world. In addition, ever-larger numbers of people are relying on the technology directly or indirectly. (It has been suggested that wireless is overused in some situations, creating a social nuisance.) More specialized and exotic examples of wireless communications and control include:

Global System for Mobile Communication (GSM) -- a digital mobile telephone system used in Europe and other parts of the world; the de facto wireless telephone standard in Europe

General Packet Radio Service (GPRS) -- a packet-based wireless communication service that provides continuous connection to the Internet for mobile phone and computer users

Enhanced Data GSM Environment (EDGE) -- a faster version of the Global System for Mobile (GSM) wireless service

Universal Mobile Telecommunications System (UMTS) -- a broadband, packet-based system offering a consistent set of services to mobile computer and phone users no matter where they are located in the world

Wireless Application Protocol (WAP) -- a set of communication protocols to standardize the way that wireless devices, such as cellular telephones and radio transceivers, can be used for Internet access

i- Mode -- the world's first "smart phone" for Web browsing, first introduced in Japan; provides color and video over telephone sets.

ii-

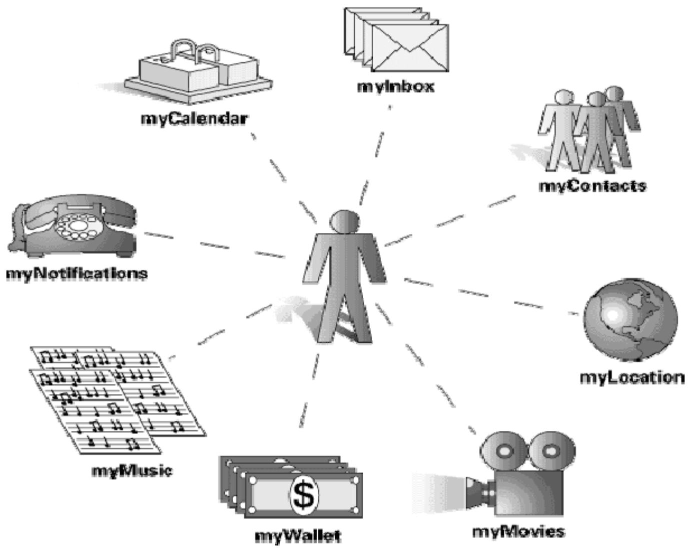


Fig. 4. Wireless Technology Evolution

Wireless can be divided into:

Fixed wireless -- the operation of wireless devices or systems in homes and offices, and in particular, equipment connected to the Internet via specialized modems.

Mobile wireless -- the use of wireless devices or systems aboard motorized, moving vehicles; examples include the automotive cell phone and PCS (personal communications services).

Portable wireless -- the operation of autonomous, battery-powered wireless devices or systems outside the office, home, or vehicle; examples include handheld cell phones and PCS units.

ISBN: 978-988-18210-3-4
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

IR wireless -- the use of devices that convey data via IR (infrared) radiation; employed in certain limited-range communications and control systems.

Fig. 5. Environmental Sensors

Pervasive Computing allows you to mix your interest in technology with your other passions: Healthcare, Sports, Entertainment, Sciences (e.g. Chemistry, Biology), Design/Art–Business, and Personal Sensors, Behavioral Sciences (e.g. Psychology, Sociology, and Environmental)

Fig. 5. Personal Sensors

III. A NEW APPLICATION MODEL

The applications that we are envisioning, it is necessary to consider the life cycle of an application. This life cycle can be divided into three parts: 1. Design-time is when the developer creates, maintains and enhances the application. 2. At load-time, the system composes, adapts and loads the application components into an application instance on particular hardware devices. 3. At run-time, the end-user invokes the application and uses its functionality.

The system provides an environment in which the application can run, and adapts the application to variations in this environment. We show how the attributes of the new model support the precepts “device as portal", “application as task" and “physical surroundings as computing environment".

A. Design Time

If devices are portals then the application should not be written with a specific device in mind. The developer should not make any assumptions about the screen size or device capabilities, or even that there is a screen at all (for example, an application may be run using a voice synthesizer and a phone). If applications are to be device-neutral, then the developer should not start with the presentation and then fill in the underlying logic. The task logic should not be secondary to the user interaction. The user interface definition should not include a rigid decomposition of the interaction.

Rather, the decomposition of the user interaction should be driven by the definition and structure of the tasks. The application description should capture the purpose of the user interaction at a high level. Services that the application needs in order to run should not be explicitly named, but rather specified in an abstract manner. Furthermore, there may be services available to the application at run-time that are not known or available to the developer at design-time, but may be useful for the task. Applications should be able to use such services. When appropriate, the designer can abstractly specify optional services that, if present at run- time, enhance the application.

Programming Model

The programming model must allow for the description of abstract user interfaces and abstract services. The structure of the program should be described in terms of tasks and subtasks. The relationship among the tasks must be rich enough that the user interface can be actualized at the various granularities.[1][3] We call this relationship navigation, as it specifies how the user will navigate the sub-tasks that make up the application. The challenges for this programming model are: For example: An application running on a device with a GUI -It may offer a button for the user to perform some action; on a voice-activated device the same action may be performed via a spoken command. Creating a task-based model for program structure - The application should be delineated into tasks and subtasks. A task includes the abstract interaction and the application logic, including the use of the services.[2] The structure is used by the system to generate device specific “presentation units"; e.g., screens. Creating a navigation model - The navigation specifies what causes a task to begin and end (e.g., a user action), and what tasks precede and follow it. This information is complementary to the task structure, and is used by the system to automate the flow of the “presentation units" when the application is running.

B. Load Time

To realize the concepts of “application as task" and “physical surroundings as computing environment", the system must be dynamic at load-time. That is, the tasks that a user wishes to perform may depend on the physical surroundings. Such tasks are enabled by contextual services. The system must, therefore, be able to discover and compose the services that are available in the physical environment, in order to perform desired tasks.[6] This is in contrast to today's model, where applications are loaded onto a device manually from a CD or other storage medium, and managed by the user rather than the system.

Dynamic Discovery

Applications and services live in the surrounding physical distributed environment. Discovery mechanisms allow a mobile device to dynamically identify and enumerate the applications and services in its local vicinity. The major challenge posed by dynamic discovery is the definition of a service adaptation layer.[7] A standard definition is needed, to both hide the differences between heterogeneous service frameworks and to maximize the use of legacy code.

Requirements and Capability Negotiation

At load-time, a device needs to negotiate with a server that hosts applications and services for several reasons. First, the device may not have all of the resources needed to run some of the applications and services. The set of available software needs to be pruned so that only the hostable functions are presented to the user. Second, application performance is a concern, so it may be desirable to split the execution burden between the device and available servers. This split, which we call apportioning, uses information about the currently available resources and the resource demands of the application. .

Presentation Selection, Adaptation and Composition

A good user interface must exhibit qualities such as consistency and style, which are difficult to quantify and synthesize. Indeed these qualities are subject to human taste. These qualities are embodied differently on devices with different interface modalities and form factors (e.g., a graphical input device versus one with a speech interface). Thus, it may be desirable to have multiple abstract representations of the application interface, one for each combination of interface modality and form factor.

C. Runtime

To understand the principle the device as portal, the run-time must monitor the resources for the currently active portal, or portals set, and appropriately adapt the application to those re- sources. In addition, the run-time must respond to changes initiated by the user. For example, the user may choose a different set of portal devices.

To understand the application as task," the run-time must allow a user to initiate and perform a task in an uninterrupted manner, despite changes in the environment and portal devices. [9][10]The run-time should support handoff of task context from one environment (e.g., office) to another (e.g., car), possibly through a disconnected state. The key to supporting a task-oriented application is that a user's access to the task be continuous.

To understand the “physical surroundings as computing environment," the run-time must be able to take advantage of services provided by the environment and the physical resources available within it. The run-time must handle unexpected failures, such as exhausting batteries or a service crash. Existing failure detection and recovery mechanisms may need to be re-examined for their applicability in this new paradigm.

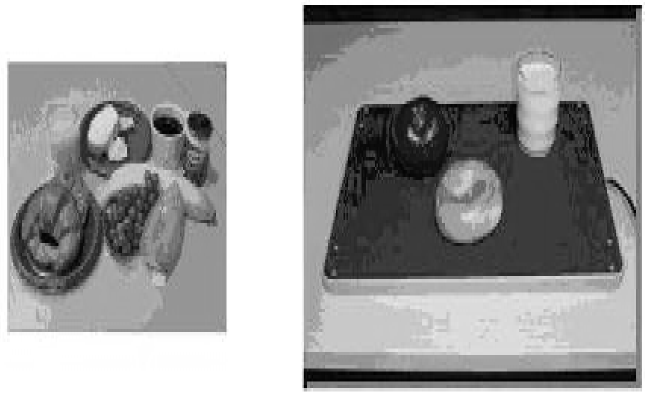


Fig. 6. Physical surrounding computing environment: Make a Meal –Put food item, queries USDA database about nutrition content, display graphics .

D. Client Server Model

The client-server model was introduced as a way to share applications and data within an organization. Applications were generally developed for a specific set of platforms, but they were divided to better accommodate resource consumption.[12] Those parts of the application that were best supported by centralized servers (e.g., data access and memory/compute intensive portions) were included in the server piece of the application. Those that more closely interacted with users were included in the client piece. Standard protocols, such as sockets and RPC, were developed for communication among the pieces.

The client/server division is often statically decided at design time. Such a division may not yield the best performance over the full range of network conditions or the full range of client devices, which may vary in processing power. The client is typically assumed to be constantly connected to the server, especially for the “thin-client" variation of the model. Thus, in a mobile environment, the model must be enhanced to accommodate sporadic disconnections, and caching is needed to buffer this effect from applications (e.g.) Furthermore, the model supports heterogeneous platforms, especially by the acceptance of the standard protocols. However, the application must be recoded to each platform, making it extremely costly and complex to develop and maintain the application code base.

IV. Conclusion

For pervasive computing to meet the expectations of mobile users, fundamental changes need to occur in the way people perceive the roles of devices, applications and the environment. Again, devices need to be perceived as portals into the application/data space supported by the environment, rather than repositories of custom software. Applications need to be seen as tasks performed on behalf of a user, not as programs written to exploit the resources of a specific computer. And, the computing environment needs to be recognized as an extension of the user's surroundings, not a virtual space for hosting and running programs.

To realize this vision of devices, applications and environments, we believe a new application model is needed.

ISBN: 978-988-18210-3-4
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

The model is characterized by a device-independent application development process, which includes abstract specification of the application front-end and the application's resource and service requirements. The model includes a highly dynamic load-time system supporting application discovery, resource and capability negotiation, and application apportioning. The run-time system allows the resources to be dynamically shared among client devices and servers. It also includes monitoring and check pointing, and enables a running application to migrate from device to device or to simultaneously utilize the interface capabilities of multiple devices.

ACKNOWLEDGMENT

Sasikumar Gurumurthy thanks to the Department of computer science, VIT University and Special thanks to Dr.M.Khalid, Director, SCSE, for his kind financial support. This work has been (Partially) supported by the research program in SCSE, VIT University, India.

REFERENCES

[1] Alvin T.S. Chan, J. Cao, Henry Chan, and Gilbert Young, Web-Enabled Framework for Smart Card Application in Health Services, Communications of the ACM, Vol. 44, No. 9, September, 2001, pp.77-82.
[2] E.S. Hall, D.K.Vawdrey, C.D. Knutson, and K.Archibald, Enabling remote access to personal electronic medical records IEEE Engineering in Medicine and Biology Magazine, Vol. 22, No. 3 pp. 133–139, May-June 2003.
[3] J. Beck, A. Ge_aut, and N. Islam. MOCA: A Service Framework for Mobile Computing Devices. In Proceedings of the International Workshop on Data Engineering for Wireless and Mobile Access, pages 62-68, August 1999.
[4] Jason Reid, Ian Cheong, Matthew Henricksen, and Jason Smith, A Novel Use of RBAC to Protect Privacy in Distributed Health Care Information Systems, 8th Australasian Conference on Information Security and Privacy (ACISP 2003), pp. 403–415, 2003, Wollongong, Australia
[5] K Arnold and J. Gosling. The Java Programming Language, Second Edition. Addison Wesley, 1998.
[6] K. Hopson, S.Ingram and P.Chan. Developing Professional Java Applets. Sam’s publishing, 1996.
[7] M.Elser, J.Hightower, T.Anderson and G.Borriello. Next Century challenges: Data Centric networking for invisible Computing. The portolano_project at the University of Washington. In Proceeding of the Fifth ACM/IEEE International Conference on Moblile.
[8] R.Eckstein, M.Loy and D.Wood.Java Swing.O’Reilly and Associates, 1988.
[9] Shu-Di Bao and Yuang-Ting Zhang, A New Symmetric Cryptosystem of Body Area Sensor Networks for Telemedicine, 6th Asian-Pacific Conference on Medical and Biological Engineering, Japan, 2005.
[10] S.K.S.Gupta,T.Mukherjee,andK. enkatasubramanian,CriticalityAware Access Control Model for Pervasive Applications, 4th IEEE Conference on Pervasive Computing (PERCOM), Pisa, Italy, 2006.
[11] S. Czerwinski, B. Zhao, T. Hodes, A. Joseph, and R Katz. Architecture for a secure service discovery service. In Proceedings of the Fifth Annual ACM/IEEE International Conference on Mobile Computing and Networking, pages 24-35, August 1999.
[12] Upkar vashney, Pervasive Healthcare, IEEE Computer, Vol 36, No.12 (2003) pp 138-140.
[13] Yanjiang Yang, Xiaoxi Han, Feng Bao, R.H. Deng, Asmart-card-enabled Privacy preserving E-prescription system, IEEE Transactions Information Technology in Biomedicine, Vol. 8, No. 1, pp. 47–58, March 2004.