

A Rich Servants Service Model for Pervasive Computing

Huai-dong Shi, Ming Cai, Jin-xiang Dong, and Peng Liu

College of Computer Science and Technology, ZheJiang University,
Hangzhou 310027, China
{shd,perryliu}@zju.edu.cn {cm,djx}@cs.zju.edu.cn

Abstract. Pervasive computing has received an intensive interest in the past years and will be a fertile source of challenging research problems in computer systems for many years to come, but there is still no good model to distinguish it from other technologies, such as distributed system, mobile computing, etc. This paper presents a Rich Servants Service (RSS) Model to help researchers understand pervasive computing, demonstrate the features and evaluate the implementations of pervasive computing systems. The model also exposes several research problems of pervasive computing for further research. A prototype based on the RSS Model is designed and demonstrated to prove the interpretation and evaluation capability of the model.

Keywords: pervasive computing, RSS Model, distributed system, mobile computing, operating system, context aware.

1 Introduction

“The most profound technologies are those that disappear. They weave themselves into the fabric of everyday life until they are indistinguishable from it.” Mark Weiser described his vision of ubiquitous computing in his seminal 1991 [1].

Pervasive computing involves many related technologies which include mobile computing, embedded computing and distributed system etc. Promiscuously each of these technologies has some features which are also the features of pervasive computing. Many research papers depict pervasive computing as calm computing technology [2], user distraction free computing [3] etc. This exposes the problem that the essence of pervasive computing is ambiguous, which is an obstacle for further research work.

This problem has brought the attention of some foreseeable researchers and in paper [4] a conceptual model for pervasive computing is presented to facilitate the discussion and analysis. However, it doesn't show the essence of pervasive computing and it is more of a logical layered architecture of pervasive computing systems.

In this paper, we present a Rich Servants Service (RSS) Model for pervasive computing to reveal the essence of pervasive computing, distinguish it from analogical research fields and evaluate the implementations of pervasive computing systems.

The rest of this paper is organized as follows. In section 2, current researches and the key characteristics of pervasive computing are studied. In section 3 the RSS

Model for pervasive computing is demonstrated and the related technologies are integrated into it. The interpretation and evaluation capability of the model and the problems exposed by it are described. In section 4 a prototype system spawned from the model and a demo system are presented. The paper is concluded in section 5 with the key contributions of this model emphasized.

2 Current Researches of Pervasive Computing

The key characteristics of pervasive computing include *Scalability, Heterogeneity, Integration, Invisibility and Perception, Context awareness Smartness and Context management* [5]. And it also has the advantages of *Effective Use of Smart Spaces, Invisibility, Localized Scalability, Masking Uneven Conditioning* [6]. The following is the summarization of key characteristics:

Pervasive. User has access to the same information anytime and anywhere by various methods.

Nomadic. User and computing can move everywhere when required.

Invisibility [5]. User's attention will not be distracted when he is working in the changing environment. The system deals with the changes silently and smartly.

Perception and smartness. This feature is mainly related to context-awareness computing. The pervasive computing system (including mobile devices) perceives context, makes timely and context-sensitive decisions and takes proactive actions to prepare for the user's future requirement which hides the computer system from user.

These are the typical features of pervasive computing. Other characteristics, such as *Heterogeneity, Scalability* etc are not the features that distinguish pervasive computing from other related computing technologies, but problems to be solved.

A number of projects have been launched for pervasive computing research. MIT's *Oxygen* [7], CMU's *Aura* [3] and Don Norman's *Invisible Computer* [8] express the idea of hiding computing around the user. Other projects focus on different characters of pervasive computing, such as *Endeavour* [9] on fluid software; *Portolano* [10] on smart data; *Sentient Computing* [11] presents some context-aware applications and smart posters; *Cooltown* [12] describes virtual bridge between mobile users and physical entities, electronic services.

In pervasive computing area, the whole world is a pervasive computing society integrated through Internet. People live in and move across smart spaces. Satyanarayanan [6] discussed some typical problems of pervasive computing that can be shown in Fig.1. What *policy* is needed to inform the user and provide data in advance, how data and process *moving* across the infrastructure and mobile device. At door (5, 7), *privacy policy* and *authentication* are needed. The required data that is left behind at leaving time can be sent by Internet at 15 which leads to the research of the management of user's private data in Internet. Is it feasible to cache data in some public sites? How do these sites expose their services and what is the caching *policy*? How do mobile devices express their data request to the infrastructure? When the user travels across smart spaces, his smart phone *self-tunes* to adapt to the local environment and finds the potential local *services* for the user (7). It leads to *adaptation strategy* research. And the smart phone must have *Energy Management policy* optimized (6).The research problems found in Fig.1 are italicized and still many others implied.

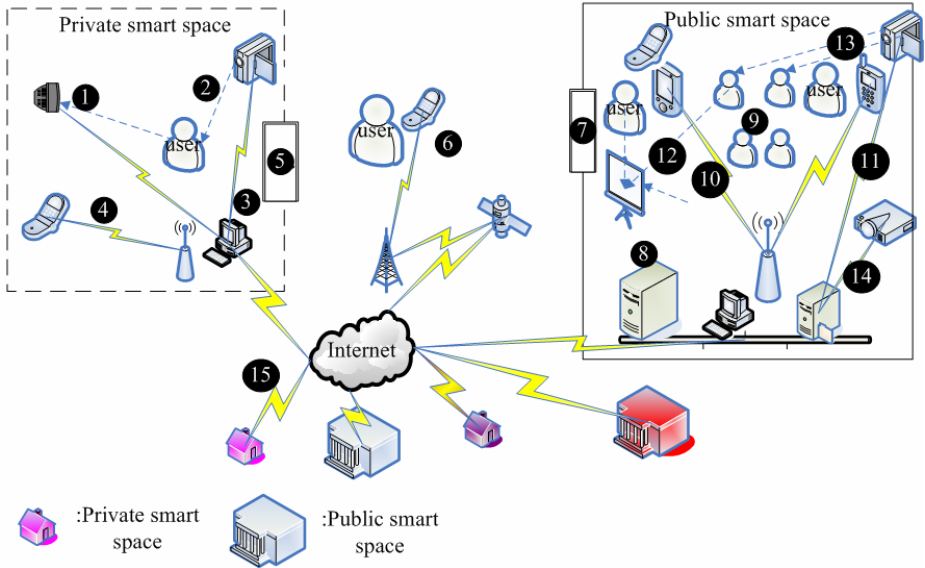


Fig. 1. Physical architecture of pervasive computing

3 RSS Model

The concept of pervasive computing sounds graceful and the features such as invisibility and pervasive have been explicitly stated. But many features can also be found in other technologies such as mobile computing and distributed system etc. These technologies are good ways to solve the problems rather than the leading actor after the following discussion. Pervasive computing includes these fields but it is not the simple addition of them. As Artificial Intelligent classifies itself in one sentence: “To build a system that has Intelligence like human”, the perfect model for AI is human’s intelligence. There’s an absence of the counterpoint for pervasive computing which can help us comprehend the essence of pervasive computing. Here we present such an ideal model – Rich Servants Service Model for pervasive computing.

The model presents an ideal pervasive computing world described by Mark Weiser [1] at conceptual level. Correspondingly, Fig.1 presents the physical model of pervasive computing. People live in such environment are like ancient rich men. A lot of digital servants (hollow man-flag) reside everywhere around the user (black-solid man-flag), listen, look, learn and analyze the master’s context and take action proactively.

We define Rich Servants Service Model as: “The user in Rich Servants Service Model is called master. Digital entity that provides services for a master is called servant. Each master is surrounded by a dynamic set of servants. Servants must be *smart, careful, proactive, inexhaustible, cooperative, faithful, invisible at most time and deployed all around the environment.* The services cover from master’s daily life

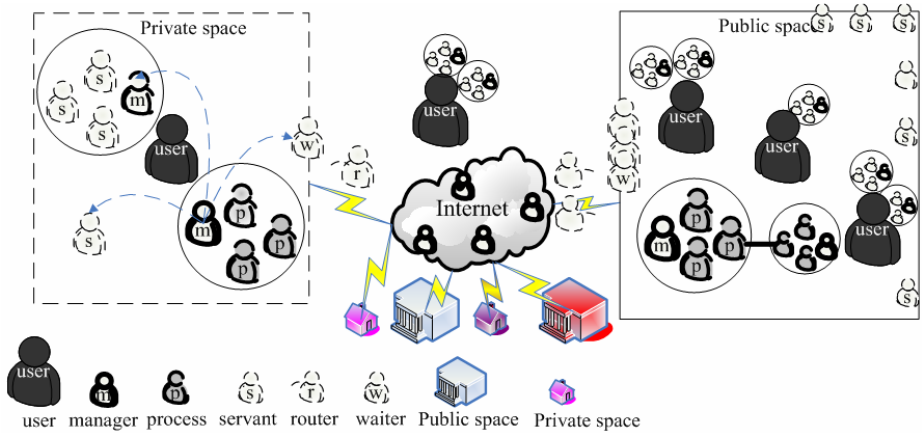


Fig. 2. Rich Servants Service Model

to working. In one word, the model is ‘a satisfied rich man serviced by a world of smart digital servants.’ This is what pervasive computing is more than mobile computing or distributed system. The essence of pervasive computing can be presented clearly by this model. It has the following features:

- The model expresses all the features under consideration.
- The result model is able to accommodate the other necessary technologies.
- One can distinguish pervasive computing from analogical fields by the model.
- One can use this model to explore new research fields, correct the ongoing research and evaluate an implementation.
- The model is addressed at conceptual level and free from any concrete designs or implementations and can be taken as a solution.

In Fig 2 there are eight kinds of elements. *User* stands for the person who lives in the pervasive computing system. *Public space* / *private space* represent the smart spaces shared by a dynamic /static set of people respectively. *Servant* is the digital element which executes the housekeeping stuff such as tracking the user’s location, tracing the room temperature and controlling the poster. They compose the user interfaces and stays all around smart spaces and user’s mobile devices. *Manager* is part of the infrastructure and it maintains the servants set and life cycle, does most high level analysis, makes decisions, supervises the whole smart space data, organizes and delivers the context data. It also maintains the set of users and makes decisions on whether to receive or refuse a foreign user. *Process* helps the manager to complete its task and sometimes act as “manager” when it communicates with the lowest servant. *Waiter* is a special kind of servant who welcomes the coming foreign user and serves the leaving user. *Router* carries the data throughout the spaces when user travels across smart spaces. The manager traveling with the user may call the router to fetch or carry data to the user’s private space. Router may be issued by the mobile device

manager or some smart space manager with demands by the user. All these elements communicate with each other through some kind of servant-language. We have designed a context exchange language, Context-C for it which will be demonstrated in another paper.

RSS Model has strong interpretation power to show the essence of pervasive computing clearly and thoroughly. RSS Model for pervasive computing is like human intelligence for AI research. The emphasis in the model is *servant* that is something “human-being”. RSS Model uses the real smart human servant system to analogize pervasive computing which clearly illustrates the characteristics of the pervasive computing and distinguishes it from distributed system, mobile computing etc. and also clarifies their relationships.

Pervasive. As mentioned before, users can get into the information world anywhere and anytime by various methods. RSS Model contains many smart spaces and servants deployed everywhere the user might visit. The valets (servants with the master-body) in the mobile device carrying with the master are aware of the local environment (communicate with waiter) and provide the most proper services. To avoid getting into the design level of RSS Model here we will not discuss how valets accomplish this task in detail. But a lot of communication protocols have already been invented to do things like this. For example [13] has discussed how foreign devices can express the data requirement to a local environment through a profile. In a word, computing is pervasive in RSS Model.

Nomadic. User and computing can move anywhere when required. The user has a set of valets moving with the master. They cooperate to log out from the old environment, enter a new one and provide continuous services. Waiters at the door help user’s movement. For example, environment and mobile device communicate by service discovery mechanism such as JINI or UPNP at implementation level.

Invisibility. User has no sense of the servants system around him because servants are hidden in physical spaces such as coffee cup, TV set, camera and even the wall. This is one aspect of invisibility. What’s more, all elements are intelligent. Servants are able to perceive the master’s emotion status and the intention expressed by his action. All the smart targets are reached by the “human servants”.

Perception and smartness. RSS Model is inherently smart and sensitive. The digital servants are designed to have some intelligence of human.

When using this model to evaluate the implementations of pervasive computing systems, the more similar it is to RSS Model, the better it is a pervasive computing system. A good servant must perceive master’s status and try to provide services proactively [3].

Due to the strong interpretation power of the model, it distinguishes pervasive computing from other research fields. Paper [6] discusses the relationship of pervasive computing and distributed system, mobile computing. It compares the distributed system with pervasive computing at *remote communication*, *fault tolerance*, *high availability*, *remote information access*, and *security* which are indirect. However, when investigating these problems through RSS Model, the difference can be told more easily. Pervasive computing has particular consideration at these problems themselves. For example, remote communication in distributed system is to aid two computer systems working together at the system level. In pervasive computing the communication takes place between the servants to serve their master better. The

former focused on computer system's interaction and the latter must take place at a higher level. It focuses on the master — "The computing is around the user". The latter may be implemented based on the former technology but it is not pervasive computing itself.

Likewise we could use RSS Model to distinguish pervasive computing from mobile computing. Paper [6] also compares the two research fields. Mobile computing focused on *mobile networking, mobile information access, support for adaptive applications, system-level energy saving techniques, location sensitivity*. In pervasive computing we must have mobile computing support for the user wandering around. Furthermore for *mobile information access*, RSS Model can exploit new problems that need to be solved rather than mobile computing. RSS Model shows that the digital servant must utilize its intelligence to automatically accomplish the master's data requirement by discovering and analyzing to satisfy the master. There must be research fields on how to exchange mobile information based on the user's context.

In a word, RSS Model takes great advantage over other models to grasp the essence of pervasive computing. It can tell the difference between pervasive computing and other research fields.

There are several research problems revealed by RSS Model.

Pervasive Computing involves all aspects of computer systems from operating systems to applications as many research papers have described. The tri-layer model of Mühlhauser [14] covers from Gadgets, integration to UC world. RSS Model implies these layers. Thinking deeply into RSS Model, we can find many research problems across these layers spontaneously. Below we'll discuss some of those problems.

Impact on Operating System. Servants in RSS Model are smart for we define it to be context-aware, self-tuning, etc. which impacts underlying operating system deeply. Traditional operating system theory has many conclusions that are inferred from the assumption "application process is not able to predict its resource requirement and action" which was true in old days. For example, Banker-algorithm for deadlock-avoid needs the resource requirement in advance to calculate the deadlock risk. Virtual memory maintenance predicts the coming page-using-sequence with LRU algorithm etc. While for servant process in RSS Model, it takes action according to the context-based prediction of user and does act in a predictable manner—they have changed their behavior. There must be changes for operating system accordingly.

The integration of AI and Pervasive Computing. Digital servants are "human being". The AI technology will play an important role in pervasive computing.

Intention-oriented Schedule. When master moves his focus, smart servants will evaluate the master's intention and change state to meet the master's demands. It schedules the action according to the master's intention even though the whole smart space behaves in the same way. The problems on how to quantify user's intention and what schedule policy is proper remain to be studied.

There're a lot of other problems that can be found by RSS Model. For example, the software manufacturer is most likely to provide servant that is trained to a particular customer, the language used by different "servant country" is need to be defined.

4 Prototype Based on RSS Model

Although RSS Model is a target model to clarify the essence of pervasive computing, it is useful in implementation. In this prototype the primary element is servant (which belongs to the infrastructure) and valet (belongs to the mobile device). To manage the servants we place Servant/Valet manger as well as user manager to organize users in a smart space.

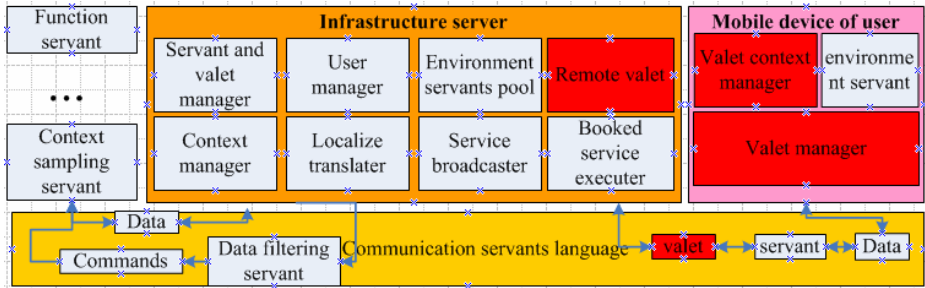


Fig. 3. A pervasive computing prototype designed based on RSS Model with the concepts of RSS Model mapped to a design model directly and perfectly

We choose mobile agents system to allow the valet to move into infrastructure and vice versa. To demonstrate the power of the model, a demo system with simulated environment is developed for the life and work in the university.

The scenario is that Prof. Li will give a C language class to the students as scheduled from 2:00pm to 4:00pm, Dec 20th, 2006 in room #306 of the 10th teaching building.

At 1:45pm, Prof. Li finishes editing the presentation for the class and leaves his office for the 10th building. The office is a private space for Prof. Li. The edited presentation will be routed to the target space automatically by his valet. Probably Prof. Li will cross several public/private spaces before he reaches the building.

When he enters the building’s public space, the servant which detects Prof. Li’s location will send the information to the manager, and his valet moves to the building’s infrastructure after proper conversation. The valet works with the local manager to find the scheduled class information for Prof. Li. Accordingly it asks another type of valet to move into the infrastructure of room #306, download the presentation and warm up the projector. These valets are trained well and perform the work proactively. Private information of Prof. Li is concealed against other software entities.

When Prof. Li enters the room, everything is ready for him.

If Prof. Li feels sick after entering the building, he goes to the hospital directly. The valet that resides in the room will get this information from Prof. Li’s manager. It will print a line in the projector so that all the students know it. The system handles all the situations smartly and silently.

5 Conclusions

In this paper we present a target model for a better understanding in pervasive computing. The key proposition is that pervasive computing essentially aims at building a virtual servants service system for users. The widely deployed digital smart servants wait in coffee-cup, TV set and even the wall, carefully perceive the master's intention and emotion to provide the services in advance based on the information they get or infer. The master usually is unaware of the existence of digital servants.

We can use RSS Model to distinguish pervasive computing from other relevant research fields such as distributed system and mobile computing etc. It accommodates these fields into pervasive computing. RSS Model can evaluate an implementation of pervasive computing to see how smart it achieves. It can correct the current research's direction, find new research problems and provide a primary solution.

The future work includes finding more problems about pervasive computing with this model and bringing more intelligence into the prototype.

References

1. Weiser, M.: The Computer for the 21st Century. *Scientific American*, September, 1991.
2. Rekimoto J et al. Adding another communication channel to reality : Experience with a chat augmented conference. In : Proceedings of ACM SIGCHI Conference on Human Factors in Computing Systems, Los Angeles, U SA , 1998. 384–391
3. Garlan D, Siewiorek D P, Smailagic A, Steenkiste P. Project aura: Toward distraction-free pervasive computing. *IEEE Pervasive Computing*, 2002, 1(4–6) : 22– 31
4. Ciarletta L t Dima A: A Conceptual Model for Pervasive Computing. *IEEE*. 2000
5. Debashis Saha ,Amitava Mukherjee: Pervasive Computing: A Paradigm for the 21st Century. *IEEE*. 2003
6. Satyanarayanan M. Pervasive Computing: Vision and Challenges. *IEEE Personal Communications*, 2001, 8(4): 10–17
7. Dertouzos M . The future of computing. *Scientific American* 1999, 282(3) : 52–63
8. Norman D.The Invisible Computer. Cambridge, Mass : MIT Press. 1999
9. Brumitt B, Meyers B, Krumm J, Kern A, Shafer s. EasyLiving: Technologies for intelligent environments. *Handheld and Ubiquitous Computing*, Bristol, UK , 2000. 30– 36
10. Esler, M., Hightower, J., Anderson, T., and Borriello, G. Next Century Challenges: Data-Centric Networking for Invisible Computing: The Portolano Project at the University of Washington Mobicom 99.
11. Hopper. A. "Sentient Computing", The Royal Society Clifford Paterson Lecture, AT&T Laboratories Cambridge, Technical Report 1999.12, 1999
12. Kindberg, T., Barton, J.: A Web-based nomadic computing system,*Computer Networks* 35(4), pp 443–456, March 2001.
13. Cherniack M, Franklin M J. Zdonik S. Expressing user profiles for data recharging. *IEEE Personal Communications : Special Issue on Pervasive Computing*, 2001, 8(4) : 32–38
14. Mühlhauser M.:Ubiquitous Computing and Its Influence on MSE. *IEEE*, 2000