Wireless Sensor Network, 2010, 2, 74-84
doi:10.4236/wsn.2010.21011 Published Online January 2010 (http://www.SciRP.org/journal/wsn/).

Agent-Oriented Architecture for Ubiquitous Computing in Smart Hyperspace

Yichao JIN[1], Ruchuan WANG [1,2,3], Haiping HUANG[1], Lijuan SUN [1,3]
[1]*College of Computer, Nanjing University of Posts and Telecommunications, Nanjing, China*
[2]*State Key Laboratory of Information Security, Graduate School of Chinese Academy of Sciences, Beijing, China*
[3]*Institute of Computer Technology, Nanjing University of Posts and Telecommunications, Nanjing, China*
Email: tcjyc@yahoo.cn, {wangrc, hhp, sunlj}@njupt.edu.cn

Received September 3, 2009; *revised September* 23, 2009; *accepted September* 24, 2009

Abstract

Agent-oriented approach is increasingly showing its magic power in a diversity of fields, specifically, ubiquitous computing and smart environment. Meanwhile, it is considered the next creative issue is to interconnect and integrate isolated smart spaces in real world together into a higher level space known as a hyperspace. In this paper, an agent-oriented architecture, which involves the techniques of mobile agents, middleware, and embedded artificial intelligence, is proposed. Detailed implementations describe our efforts on the design of terminal device, user interface, agents, and AI computing module to combine two single smart spaces, UbiLab and UbiDorm, into a practical smart hyperspace.

Keywords: Agent-Oriented, Ubiquitous Computing, Smart Hyperspace, Middleware, Embedded Artificial Intelligence

1. Introduction

The progressive advances in computer system together with the simultaneous improvement in Wireless Sensor Networks (WSNs) [1] and other related fields, contribute to the ubiquitous computing era, when our physical space will be filled with different kinds of smart devices, which possess the capability of computing and communication. At the same time, pervasive network, which are formed by ubiquitous devices interconnected with each other through wireless communications, internet and other medium, supply services and useful information to us instantly and constantly? Thus, Mark Weiser's ubiquitous computing, "services can be provided to users anytime and anywhere with any devices" [2], which was first propounded in 1991, is approaching. And we name this kind of physical space as smart environment, where the knowledge about its users and the surroundings could all be acquired and applied based on the ubiquitous computing, in order to adapt to users and meet the goals of convenience and efficiency.

However, the tendency that a variety of smart devices such as laptops, handhold PDAs, tiny sensor nodes etc. are becoming common facilities in people's daily life, not only offers us with the fundamental platforms, but also raises issues on how to take advantage of these heterogeneous devices to realize ubiquitous computing. Besides, in order to deploy such smart environment, context, which is defined as "any information that can be used to characterize the situation of entities (i.e. whether a person, place or object) that are considered relevant to the interaction between a user and an application, including the user and applications themselves" [3], is of paramount importance. Meanwhile, with the emergence of a number of context-aware systems and smart environment projects, the increasing natural demand on interconnecting and integrating isolated smart spaces in real world together into a higher level space known as a hyperspace, also raises essential difficulties on how to implement context-aware systems in large-scale intelligent environment and how to expand context-awareness to dynamic, open systems.

As a feasible solution to those problems, agent-oriented approach provides our research with the key feature of autonomy, collaboration and especially intelligence. Furthermore, a novel architecture with multi-agents to realize a smart hyperspace (including two single physical intelligent environments, UbiLab and UbiDorm) is adopted by us.

The rest of this paper is organized as follows. Section

2 contains relevant researches on the fields of ubiquitous computing, smart environment and agent-based approaches. Section 3 explains the overview of our paradigm's framework and two essential components, namely, middleware and AI computing model. Section 4 details the design and implementation of our experimental smart hyperspace. Section 5 discusses the results and the analysis of this work. And finally the conclusion and future works are given in Section 6.

2. Related Works

2.1. Ubiquitous Computing

Industries as well as academia have advanced lots of ubiquitous projects, over the last two decades. SAFE-RD [4], MARKS [5] and ETS [6] characterize the works carried out by M. Sharmin, S. Ahmed, and S. I. Ahamed in the field of ubiquitous computing, in which security, adaptability, efficiency, middleware, resource discovery and self-healing are mainly investigated. MIT's Oxygen project [7], Carnegie Mellon University's Aura project [8], UC. Berkeley's Endeavour Project [9] also devote their efforts to a number of different aspects in realizing ubiquitous computing (i.e. large-scale computing, QoS, task scheduling, context awareness) in some particular conditions, significantly accelerating the growth of smart spaces.

2.2. Smart Environment

So far, in spite of no existing explicit definition of what smart environment exactly is, massive endeavors towards proposing such prototypes are already ongoing. According to Mark Weiser, smart environment is “a physical world that is richly and invisibly interwoven with sensors, actuators, displays, and computational elements, embedded seamlessly in the everyday objects of our lives, and connected through a continuous network” [10]. The Aware Home Research Initiative at Georgia Tech, which is viewed as one of the first living laboratories, aimed at multidisciplinary exploration of emerging technologies and services based in smart home [11]. Another relevant research launched by The University of Texas at Arlington which was known as MavHome (Managing an Adaptive Versatile Home) project, focus on the creation of an intelligent and versatile home environment with state-of-the-art algorithms and protocols used to provide a customized, personal environment to the users of this space [12]. Besides, the increased interests of industrial labs in constructing smart environments are evidenced by Microsoft's Easy Living project [13], IBM's BlueEyes project [14], and the Speakeasy project at Xerox PARC [15], etc.

2.3. Agent-Oriented Solution

An agent is a software entity that has some properties of a human such as autonomy, reasoning, learning, and knowledge level communication, etc. [16]. Agent-oriented approach, which is highly expected to play a vital role in achieving smart space in high level, has already expressed energetic effects on ubiquitous computing and smart environments in numerous applications. Essex's iDorm project targets to realize the vision of ambient intelligence in health care environments by combining the use of unobtrusive sensors and effectors with intelligent embedded-agents [17]. A novel type-2 fuzzy systems based adaptive architecture for agents embedded in ambient intelligent environments, a hierarchical fuzzy genetic multi-agent architecture for building learning mechanism, together with another novel life-long learning approach based on intelligent agents are addressed by Essex's group in [18–20], respectively. Other Contemporary researches also cover the following areas: using a neural networks agent based approach to recognize different high level activities [21], systematic and useful methodology to develop agent-based system [22], intelligent agents involving case-based reasoning (CBR) and Bayesian Network [23] etc.

Based on these aforementioned technologies and theories, a foundation for our work had been laid. However, dissimilar with the related projects, we introduce the mobile agent to our intelligent hyperspace, in which improved operation efficiency, minimal manual interaction, seamless context exchange between multiple smart spaces, optimal control strategy for a large and dynamic environment are all ensured by us.

3. Framework

3.1. System Architecture

Figure 1 shows the configuration of our experimental smart hyperspace, which involves the UbiLab and UbiDorm. We select laboratory and dormitory to build our prototype, because they are the most common places a researcher's daily activities may cover. In our research, both of them are furnished with many various devices ranging from small ones like RFID and embedded nodes suited wireless sensor networks (WSNs), to middle ones such as PDAs, mobile phones and gateway, and to large ones represented by laptops and PCs. These either mobile or immobile devices are distributed over their local physical space, constituting a smart network, which is capable of obtaining context, computing, and providing services to users. These two physical separate spaces connect with each other by telecommunication net such as GPRS and GSM, Internet, etc. So they can be regarded as the conceptual smart large environment as a whole.

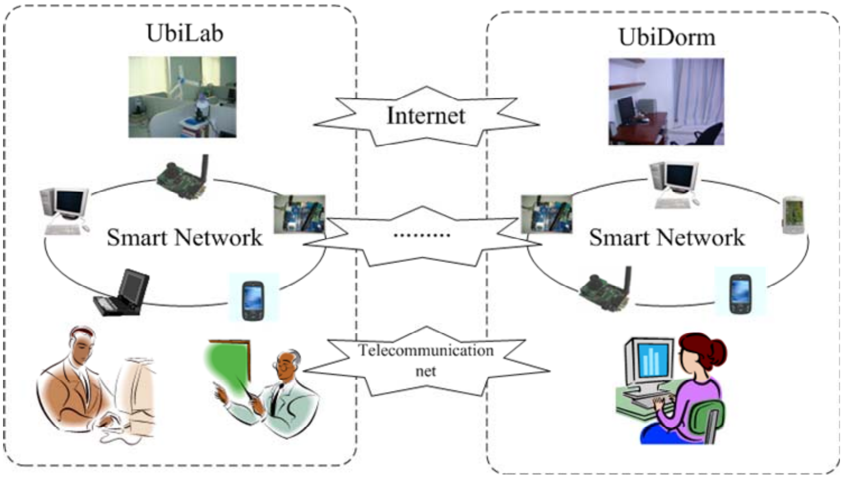


Figure 1. Smart hyperspace configuration.

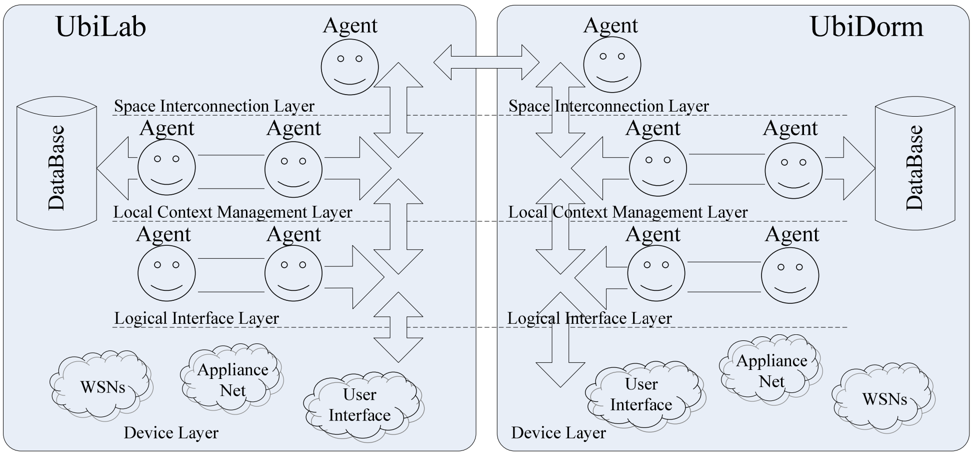


Figure 2. Agent-based architecture for smart hyperspace.

As depicted in Figure 2, the architecture for smart hyperspace is a hierarchy of rational agents, which are able to accomplish their specific tasks to meet the overall goal, and a set of concrete functional layers, which co-operate to realize our system. The following details the interpretation of each layer.

a) *Device Layer* (*DL*)*:* The Device Layer contains the very basic hardware including daily appliances (i.e. fan, light, alarm, etc.), wireless sensor nodes, network hardware such as gateway, and user interface devices involving PCs, PDAs, and mobile phones. This layer takes charge of gathering context directly from surroundings through sensors installed in tiny wireless nodes or from user interface devices and other sources, controlling appliances according to the upper layer's commands, exchanging physical bytes throughout heterogeneous networks.

b) *Logical Interface Layer* (*LIL*)*:* The main task of this Layer, whose crucial part is middleware, is formatting and extracting the data either from the lower layer, DL, or the higher layer, LCM. It is also responsible for managing some necessary information between the agents and users.

c) *Local Context Management Layer* (*LCM*)*:* This layer takes the responsibility of gathering, storing, and generating useful knowledge to make our experimental prototype operating as a smart entity. Besides, its duties also consist of integrating both contextual information and the message came from Space Interconnection Layer, mining data, making proper decisions, and coordinating the tasks allocated to different agents. In particular, a database and a computing module with sufficient artificial intelligence play the most important role in this layer.

d) *Space Interconnection Layer* (*SIL*)*:* In order to realize a hyperspace, we lay this layer in the top of our hierarchical framework. Internet, GSM, and some other widespread nets provide basic platforms to this layer.

Due to its existence, it is possible to build up the effective and efficient links between two physical separate spaces (or further, among more than two spaces).

3.2. Middleware

Figure 3 shows the architecture of DisWare developed by us [24]. This mobile agent-based middleware, which chiefly contains interface for instructions, agent management module, system management module, and interface for network communication, mediates between mobile agents and OS layer (and device layer). It separates applications from the specific design of system. Consequently, the flexibility and availability of the whole system are greatly improved. There are four major parts composing our DisWare.

a) *Instructions Interface* (*II*)*:* This interface provides the instruction set for mobile agent, so that we could realize different types of agents by simply modifying the particular code of agent.

b) *Agent Management* (*AM*)*:* Agent Resource Management, Agent Transfer Control, and Execution Management constitute the Agent Management module. This module views agent as a certain class, management of agent's resource, dispatch and retraction of agent, management of execution queue etc. as functions in this class. Some characteristics such as modular, encapsulation etc., which exist in object-oriented programming, are also emphasized here to benefit the design of agent.

c) *System Management* (*SM*)*:* This System Management comprises Network Management, Device Management, and Memory Management. Information management of neighbor related with nodes and sensors, allocation of both local and remote memory are involved in this module.

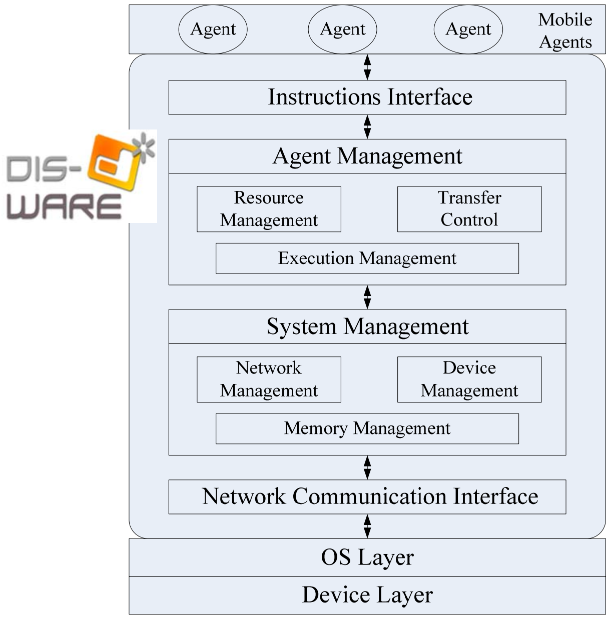


Figure 3. Architecture of DisWare.

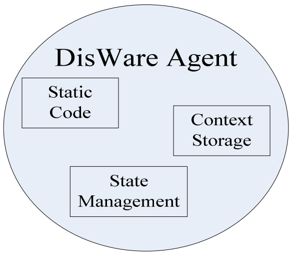


Figure 4. The construction of DisWare agent.

d) *Network Communication Interface* (*NCI*)*:* This interface takes charge of the contact with OS layer, device layer and SM. The differences among heterogeneous operating systems (we choose TinyOS and MantisOS for WSNs, WinCE for PDA and smart phone, and WinXP for PCs in our DisWare) are eliminated thanks to this interface as well. As a result, System Management module could call some corresponding functions directly to configure the wired or wireless communication.

3.3. Mobile Agent

Mobile agents are programs that can migrate from host to host in a network, at times and to places of their own choosing. The state of the running program is saved, transported to the new host, and restored, allowing the program to continue where it left off [25].

To meet this requirement, our DisWare agent possesses these three components: static code, context storage, and state management as demonstrated in Figure 4. Our DisWare agent runs static code which contains some static functions to fulfill the particular missions. The code size is usually much smaller than data size in mobile agent. Context storage takes charge of storing all kinds of useful context information including sensor data gathered from physical environment, agent ID, program counter, operation pointer and the address of different kinds of static codes etc. Recording the agent's state (running states, migration states etc.) is achieved by state management.

3.4. Embedded Artificial Intelligence

Automation was initially targeted as soon as the concept of pervasive computing and smart spaces first came into our view. It is widely accepted that the development of artificial intelligence (AI) is the key solution to this problem. In our research, we combine the benefits of diverse AI algorithms into our UbiLab and UbiDorm environment. Owing to the DisWare, real-time context gained by hardware devices can be easily obtained by agents. Besides the communication between different agents become convenient, and other functions like sensor reconfiguration, dynamic reprogramming, and remote task assignment which are difficult to achieve by some traditional methods are all available. Based on these advantages, we import a data mining module to clarify the useful information from historical records, and a fuzzy module [20] to fuzz the real-time context to reduce the computing complexity. The neural network [21], which has the ability of learning, adapting, predicting, making suitable decisions, is considered as the essential part in AI module, and genetic algorithm [19] is used to tune the neural networks. A rule management is set here to store both the general knowledge and the correct rules. Eventually, a fault tolerance module filters the actions generated by the Decision Module. And then the behaviors which are carried out finally, feed back to the History Module. Figure 5 illustrates how these components collaborate with each other to improve the overall performance in our smart hyperspace.

The fuzzy module extracts information by categorizing the real-time context into a set of fuzzy membership functions, so that a simple but effective approach is formed to build models at a certain level of information granularity. Once the agent has extracted the membership functions and the set of rules from the user input data, the fuzzy module has learnt how to fuzz those contexts. The sample rule is as follows:

IF *Temperature* is $X_1(I_t)$ and *Light* is $X_2(I_l)$ and

Humidity is $X_3(I_h)$ THEN *O* is $Y(CF)$. (1)

Where X_1, X_2, X_3 are conditions of fuzzy logic memberships function. I_t, I_l, I_h, are the sense data, representing the exact values of temperature, light and humidity respectively. Y is the fuzzy output, and CF is the confidence factor attached to consequent part of this rule.

Our neural network is a multi-input multi-output connectionist feed forward architecture with two hidden layers. The conditions about surroundings, current time, and the state of various kinds of home appliances are all considered as inputs, while the outputs are the related commands taking charge of controlling corresponding effectors. As soon as the users change the context, the neural network is triggered to enter a new training. Iterations of each training are set at 2000.

And the genetic algorithm's experimental setup is as follows:

Probability of crossovers: 90%; Probability of mutation: 0.1%; Population: 20; Generation: 2000.

Fitness function: Fitness =

$$\sum_n [desired\ output - actual\ output]^2 \qquad (2)$$

Where n represents the total amount of outputs.

4. Implementation

4.1. Terminal Device

To implement our experimental smart hyperspace, designing terminal devices, which have the first contact with users or surroundings, are expected as the primary step. Figure 6(a) exhibits the UbiCell node with a sensor board installed on it, which was designed by us before especially for such applications to perceive humidity, temperature, and luminance information at one time. Figure 6(b) displays the UbiDot node equipped with pulse, body temperature and blood oxygen sensors. Similarly, it can acquire all the three types of physiological information simultaneously [26]. Figure 6(c) expresses the gateway, whose mission is exchanging messages throughout wireless sensor networks, GSM/ GPRS, and Internet. The wireless multimedia sensor node used to capture sound and images is described in Figure 6(d).

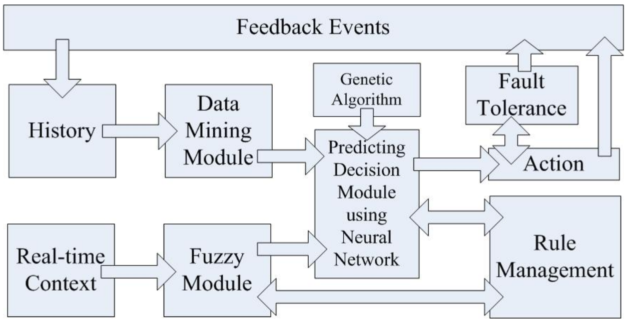


Figure 5. Hybrid AI techniques into smart hyperspace.

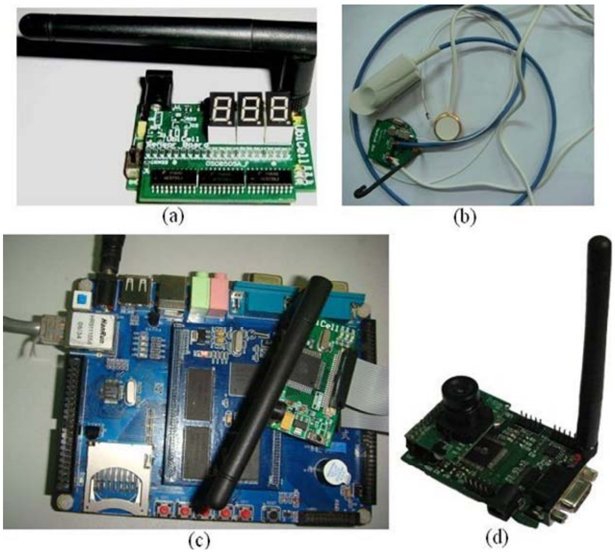
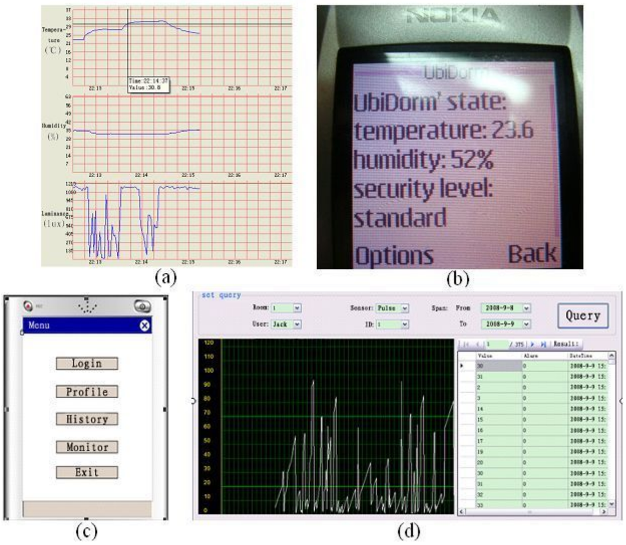


Figure 6. Various terminal devices.



UbiDorm' state:
temperature: 23.6
humidity: 52%
security level:
standard

Options Back

set query

Room: 1 Sensor: Pulse Span: From 2008-9-8
User: Jack ID: 1 To 2008-9-9

Query

1 / 375 Result:

Value	Alarm	DateTime
30	0	2008-9-9 15:
31	0	2008-9-9 15:
2	0	2008-9-9 15:
3	0	2008-9-9 15:
14	0	2008-9-9 15:
15	0	2008-9-9 15:
16	0	2008-9-9 15:
17	0	2008-9-9 15:
19	0	2008-9-9 15:
20	0	2008-9-9 15:
30	0	2008-9-9 15:
31	0	2008-9-9 15:
32	0	2008-9-9 15:
33	0	2008-9-9 15:

Figure 7. Snapshots of GUI in various devices.

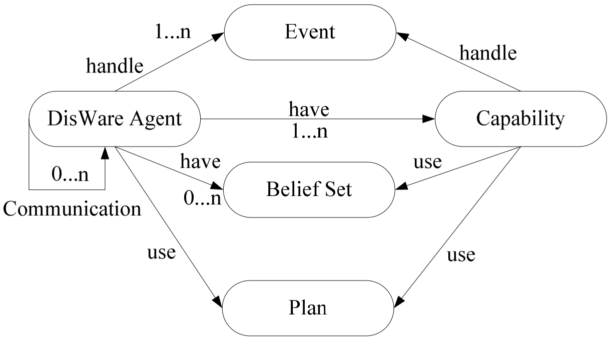


Figure 8. The architecture of DisWare agent.

4.2. User Interface

In our smart hyperspace, services should be supplied to users conveniently through efficient interfaces on the human interaction devices. Hence, graphical user interfaces (GUIs) are designed to hit this target. Figure 7(a) presents the GUI of real-time monitor on temperature, humidity and luminance [27] in PC. Their values would be updated periodically according to the information the agents transmits, so the changes could be identified vividly. Unlike the PC, even a notebook PC device, devoid of mobility is no longer the problem of some PDA based or smart phone based solutions. Figure 7(b) indicates the mobile phone wireless application interface. Figure 7(c) shows the initial welcome screen with five choices, while one instance of history query interface is portrayed by Figure 7(d). In our experiment, the GUIs in PC are achieved using Visual C# in Visual Studio 2005. And the GUIs in PDA or smart phone are implemented in windows CE.net framework 2.0.

4.3. Design of Agent

The programming framework and strategies involved in JACK Intelligent Agent [28] are consulted here in order to carry out the agent-based solution. The advantages of BDI [28] and feedback structure are mixed with each other in the agent based on our DisWare, which could also be called as DisWare agent. Hence, DisWare agent could perform either some event-driven reactions or aim-oriented processes on its initiative. In detailed implementation, we compile our DisWare agent language code into pure nesC (a programming language for deeply networked systems), and calls the nesC compiler to generate executions for the nodes in wireless sensor networks. And we use C# language to realize the same function in PCs, PDAs or smart phones. Figure 8 describes the components of our DisWare agent [24].

4.4. Learning Phase

In our work, the learning is achieved through interaction with the actual environment. During the learning phase, every request offered by users, together with the corresponding environment states and other related information captured by the mobile agents will be viewed as an input sample. And whenever the request is received, this neural network with the assistance of other AI techniques discussed before, are trained based on the new sample set. In our implementation, the summation of samples is no more than 2000 due to the limitation of computational resource. Case there are already 2000 samples, which are stored in our system, when the new sample is added, the earliest sample would be deleted from the database simultaneously. Thus, an incremental and lifelong learning phase is formed.

4.5. Experimental Scenario

All of the components mentioned above are involved in our experimental test bed, the UbiLab workplace environment as shown in Figure 9(a) and the UbiDorm as represented in Figure 9(b) and Figure 9(c). Custom power line control automates all the lights, fans, air-conditioner, and other appliances such as fire alarm, humidifier, etc. Perception of light, humidity, and temperature, smoke, motion, and switch settings is performed through wireless sensor networks or wireless multimedia sensor networks. Identity check-up was accomplished by RFID techniques and motion sensors as soon as the users entered either UbiLab or UbiDorm, not only to prevent illegal persons from our smart hyperspace, but also to record the context about when the every single valid user reached there and how many authorized experimenters in the UbiLab in any specific time etc. The security of our smart hyperspace is also insured by the smoke sensors and the fire alarm. A gateway fulfills the target of interconnection among isolated physical spaces. And Figure 9(d) demonstrates a base node also can integrate the information, and communicate with PCs and other terminal devices.

Figure 10 and Figure 11 show both the sensor layout and the actuator layout in UbiLab and UbiDorm, where six kinds of sensors or devices, three sorts of physiological sensors, and a variety of effectors are involved. Some UbiCell nodes are pre-installed to monitor the environment, while others, which are connected with effectors, take charge of controlling them. DisWare is installed in every single terminal device in order to manage mobile agents. Volunteers in UbiDorm are additional required to wear the UbiDot node to sensor the physiological index.

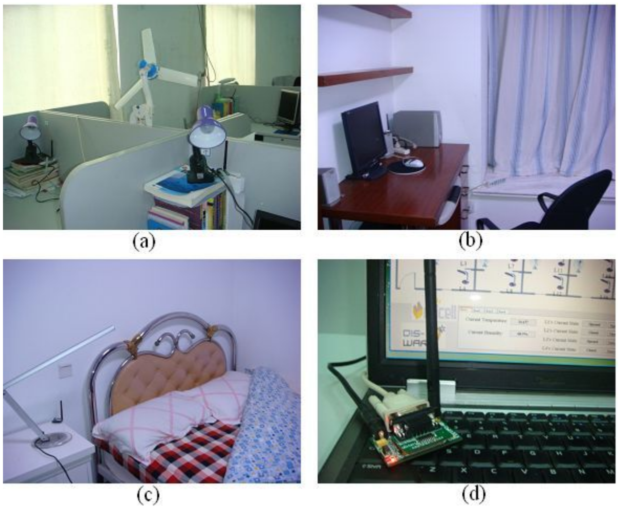

(a) (b) (c) (d)

Figure 9. The UbiLab and UbiDorm.

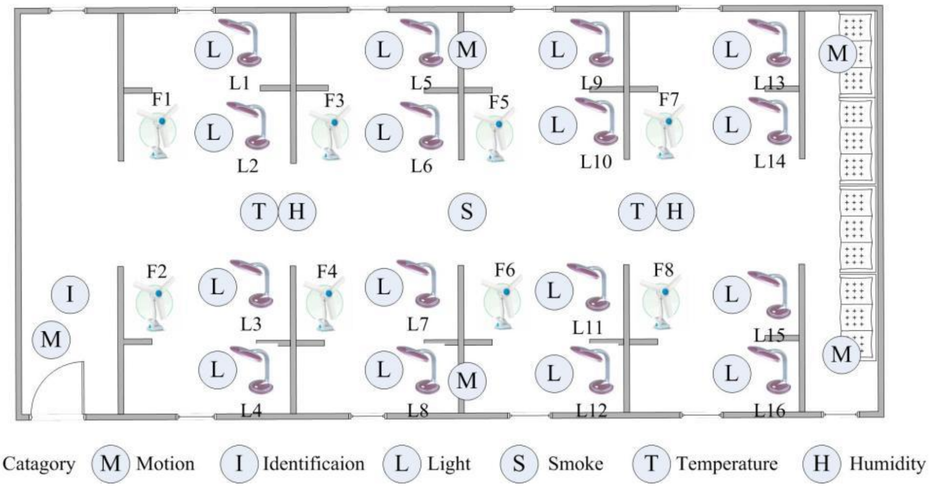


Figure 10. The UbiLab layout.

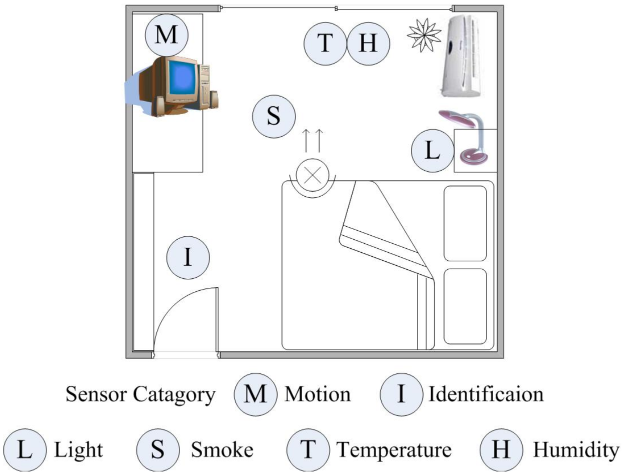


Figure 11. The UbiDorm layout.

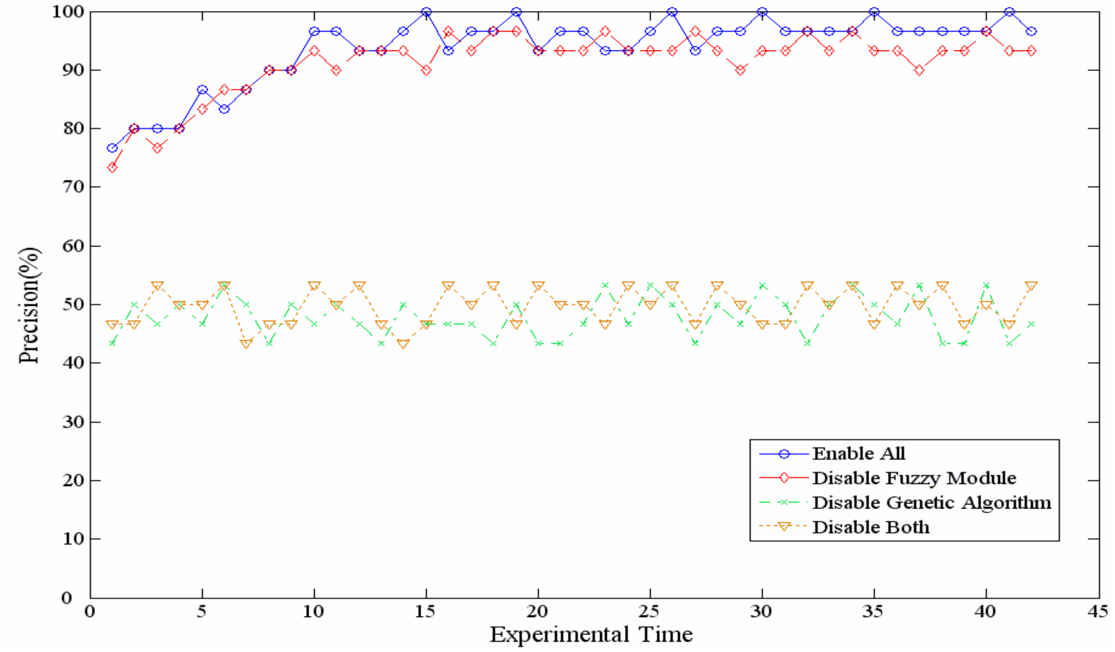


Figure 12. The precision on test set against experimental time.

Finally, as an illustration of techniques used in our research, a continuous evaluation on this prototype system, which lasted three weeks, has been conducted by us. During this period, volunteers who didn't take part in the development of our system executed their daily work in our UbiLab in the daytime, and one of them occupied the UbiDorm in the night time. Both the lights, fans, others effectors in laboratory and the home appliance such as air-conditioner were the pre-configured instruments, which could be operated by users in accordance with their own feeling via the interface we designed in all kinds of devices. For instance, when the user felt hot, he would probably turn on the air-conditioner. To achieve a higher level goal, the condition that the user certainly wanted to enter his cool dorm during harsh summer, is under consideration too. So he could turn on the UbiDorm's air-conditioner when he was about to leave for UbiDorm a few minutes later, but still stayed in UbiLab. A more complex situation would be as follow, if the user felt nervous partly because of the high temperature and low level of humidity, consequently, his pulse would become higher than usual. At this time, the user would turn on the air-conditioner and humidifier, and be likely to enjoy some bright music. Subsequently, after the initial monitor phase, our system would try to predict user's action based on the trained embedded artificial intelligent module, and then automate corresponding actuators. Meantime, for users, they were hardly aware of the cycle that new introduced samples brought modification and adaptation to our system every now and then.

5. Results and Analysis

Each day in this experimental period, we estimated the performance of this system twice per day (at 9:00AM and 9:00PM), which used our agent-based solution, by inputting 30 simulated data (not including multimedia information), and then identified whether the outputs are correct by human. Meanwhile, in order to prove the availability of learning mechanism using hybrid AI techniques, we disable only Fuzzy Module, only Genetic Algorithm, both Fuzzy Module and Genetic Algorithm, respectively, under the same samples.

Figure 12 summarizes the experiment's consequence. The Y-value in Figure 12 presents the quotient which was obtained by the number of outputs match the simulative environmental condition to user's demand dividing by the sum of input data, while the X-value illustrates the experiment time. It is obvious that because of insufficient samples, the first three days' (that is, the initial monitor phase's) execution seems not so ideal. However, with the increment of samples, the precision grows. Besides, Figure 12 also addresses that the AI computing module without Fuzzy module act a little inferior to that of hybrid AI. And if there's no Genetic Algorithm adding intelligence to our system, there are no apparent relationships between the amount of samples and its precision, under what condition, the prototype performed much worse.

In addition, we also established the solution with no fuzzy module under the same situation, and then we

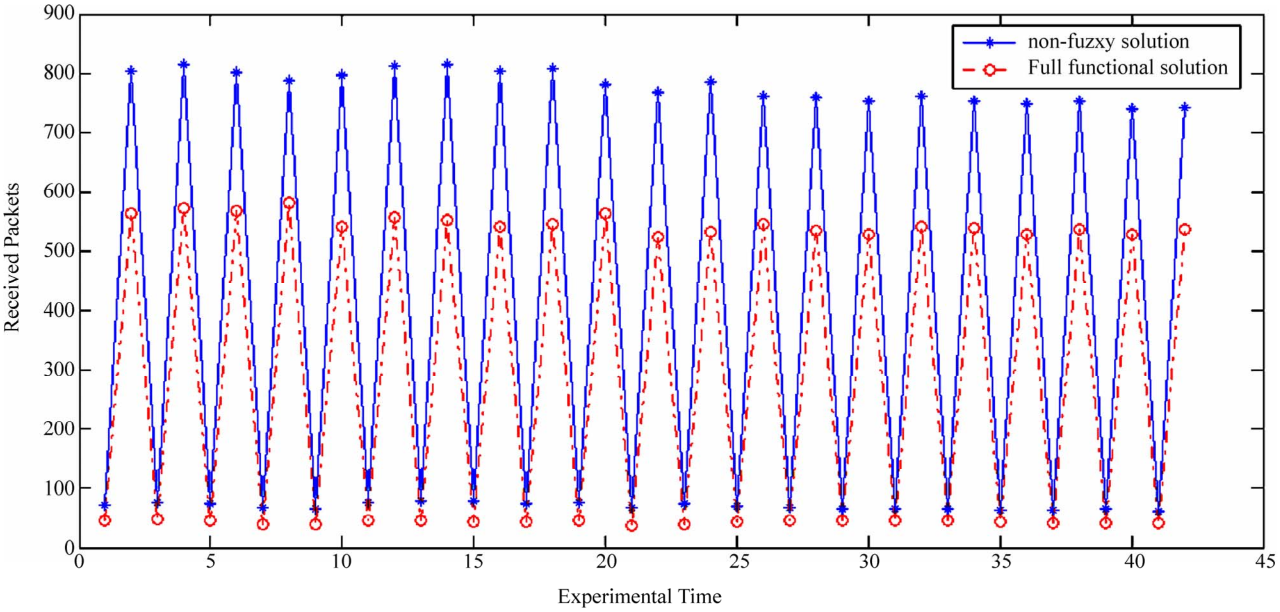


Figure 13. The received packets against experimental time.

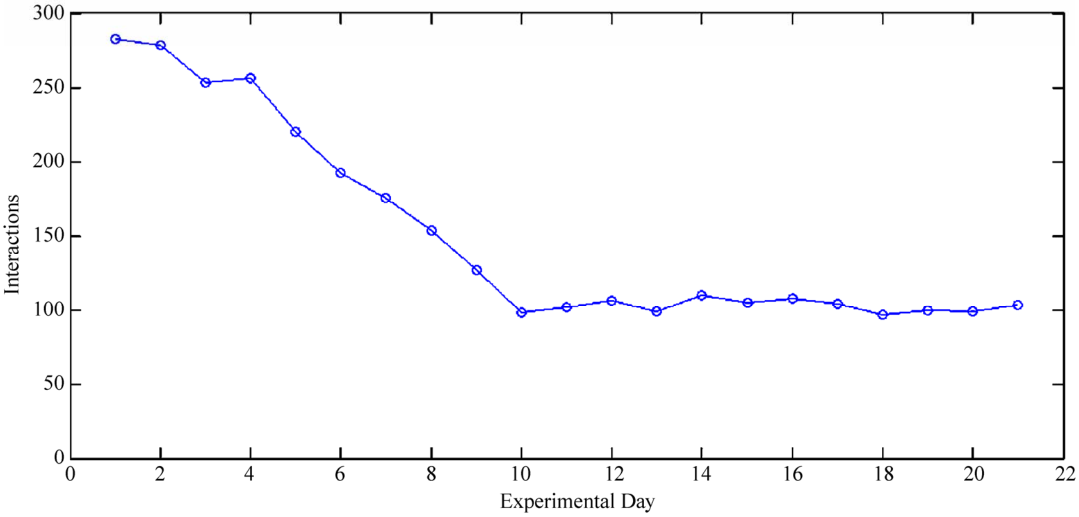


Figure 14. The number of manual interaction with our system.

compared our system with that by counting the overall received packets in every 12 hours. Figure 13 clarifies the result of this comparison. Obviously, the non-fuzzy solution's overload is much higher than that of our full functional system. And during our experimental time both solutions perform stably in the every certain interval.

Furthermore, the number of manual interaction with our proposed system was recorded by us each day. Owing to the existence of the predicting and decision module, the interactions were reduced apparently according to Figure 14.

6. Conclusions and Future Work

In this paper, we have proposed an agent-oriented architecture for ubiquitous computing, as well as an actual paradigm of smart hyperspace based on this novel structure, which is context-aware, capable of monitoring and providing automation to researchers. It is proved that this novel architecture involving DisWare, AI computing module, multi-agents, a diversity of terminal devices and user interfaces can be put together harmoniously and successfully into practice. And the results also address our propounded hybrid AI techniques perform availably, effectively and efficiently.

Our future work experimental program includes the plans about adding more physical spaces which can cover almost each aspect of a person's daily activities such as classroom and car to our paradigm under this architecture. In addition, we also aim to extend the types of both sensors and actuators to complete a full functional smart hyperspace, which really acts a unique test bed for relevant further researches, and provides potential value to commercial activities.

7. Acknowledgements

We are grateful that the subject is sponsored by the National Natural Science Foundation of P.R. China (No. 60773041), the Natural Science Foundation of Jiangsu Province(BK2008451), National 863 High Technology Research Program of P.R. China (No. 2006AA01Z219), High Technology Research Program of Nanjing City (No. 2007RZ106, 2007RZ127), Foundation of National Laboratory for Modern Communications (No. 9140C11050 40805), Jiangsu Provincial Research Program on Natural Science for Higher Education Institutions (No. 07KJB 520083) and Special Fund for Software Technology of Jiangsu Province. Postdoctoral Foundation of Jiangsu Province(0801019C), Science & Technology Innovation Fund for higher education institutions of Jiangsu Province(CX08B-085Z、CX08B-086Z), and the six kinds of Top Talent of Jiangsu Province.

8. References

[1] F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, "Wireless sensor networks: A survey," Comput. Networks (Elsevier), Vol. 38–4, pp. 393–422, 2002.

[2] M. Weise, "The computer for the 21st century," Scientific American, pp. 94–104, September 1991.

[3] A. K. Dey, D. Salber, and G. D. Abowd, "A conceptual framework and a toolkit for supporting the rapid prototyping of context-aware applications," Human-Computer Interaction, Vol. 16, No. 2–4, pp. 97–166, 2001.

[4] M. Sharmin, S. Ahmed, and S. I. Ahamed, "SAFE-RD (secure, adaptive, fault tolerant, and efficient resource discovery) in pervasive computing environments," IEEE international Conference on Information Technology (ITCC 2005), Las Vegas, NV, pp. 271–276, April 2005.

[5] M. Sharmin, S. Ahmed, and S. I. Ahamed. "MARKS (middleware adaptability for resource discovery, knowledge usability and self-healing) for mobile devices of pervasive computing environments," In Proceedings of the Third International Conference on Information Technology: New Generations (ITNG 2006), Las Vegas, Nevada, April 2006.

[6] S. Ahmed, M. Sharmin, and S. I. Ahamed. "ETS (efficient, transparent, and secured) self-healing service for pervasive computing applications," International Journal of Network Security, Vol. 4, pp. 271–281, 2007.

[7] The Oxygen Project, http://oxygen.lcs.mit.edu/overvview. html.

[8] Project Aura, http://www.cs.cmu.edu/~aura/.

[9] The Endeavour Project, http://endeavour.cs.berkeley.edu/.

[10] C. Diane and D. Sajal, "Smart environments: Technology, protocols and applications," Wiley-Interscience, ISBN 0-471-54448-5, 2004.

[11] Essa, "Ubiquitous sensing for smart and aware environments: Technologies towards the building of an aware home," Position Paper for the DARPA/NSF/NIST Workshop on Smart Environments, July 1999.

[12] S. K. Das and D. J. Cook, "Designing smart home environments: A paradigm based on learning and prediction," In: Mobile, Wireless, and Sensor Networks: Technology, Applications and Future Directions (Eds. R. Shorey, *et al.*), Chapter 13, pp. 337–356, John Wiley, 2006.

[13] The Easy Living Project, http://research.microsoft.com/ easyliving/.

[14] The BlueEyes Project, http://www.almaden.ibm.com/cs/ BlueEyes/.

[15] W. K. Edwards, M. Newman, and J. Z. Sedivy, "The case for recombinant networking," Xerox PARC Technical Report, 2001.

[16] M. Wooldridge, "Agent-based computing," Interoperable Comm. Networks, Vol. 1, No. 1, pp. 71–97, 1998.

[17] H. Hagras, V. Callaghan, M. Colley, G. Clarke, A. Pounds-Cornish, and H. Duman, "Creating an ambient-intelligence environment using embedded agents," IEEE Intelligent Systems, Vol. 19, pp. 12–20, 2004.

[18] F. Doctor, H. Hagras, and V. Callaghan, "A fuzzy embedded agent-based approach for realizing ambient intelligence in intelligent inhabited environments," IEEE Transactions on Systems, Man and Cybernetics, Part A, Vol. 35, No. 1, pp. 55–65, 2005.

[19] H. Hagras, *et al.*, "A hierarchical fuzzy genetic multi-agent architecture for intelligent buildings learning, adaptation and control," International Journal of Information Sciences, Vol. 150, pp. 33-54, March 2003.

[20] H. Hagras, F. Doctor, V. Callaghan, and A. Lopez, "An incremental adaptive life long learning approach for type-2 fuzzy embedded agent in ambient intelligent environments," IEEE Transactions on Fuzzy Systems, Vol. 15, No. 1, pp. 41–55, 2007.

[21] F. R. Illingworth, V. Callaghan, and H. Hagras, "A neural network agent based approach to activity detection in AmI environments," In Proceedings of IEEE International Workshop on Intelligent Environments (IE05), 2005.

[22] S. Lee and T. Kim, "Generic engineering approach for agent-based system development," In Proceedings of MATA 2002, LNCS, Vol. 2521, pp. 58–71, 2002.

[23] N. Kushwaha, M. Kim, D. Y. Kim, and W. D. Cho, “An intelligent agent for ubiquitous computing environments: smart home UT-AGENT,” In Proceedings of IEEE Workshop on Software Technologies for Future Embedded and Ubiquitous Systems, pp.157–159, 2004.

[24] W. F. Li, “Research on agent-oriented middleware key technologies of wireless sensor networks,” Ph.D. dissertation, Nanjing University of Posts and Telecommunications, pp. 113–139, 2008.

[25] D. Kotz and R. Gray, “Mobile agents and the future of the internet,” ACM SIGOPS Operating Systems Review, Vol. 33, No. 3, pp. 7–13, July 1999.

[26] The UbiCell, UbiDot, http://www.wsns.net.cn/.

[27] R. C. Wang, L. J. Sun, *et al.*, “Mobile agent-based wireless sensor network middleware,” Technical Report for National 863 High Technology Research Program of China, pp.188–249, December 2008.

[28] JACK Intelligent Agents,
http://www.agent-software. com/.