Towards Initiative Smart Space model

Yue SUO[1,*], Yuanchun SHI[1,+]
[1]*Department of Computer Science and Technology, Tsinghua University, Beijing, China*
*suoy@mails.tsinghua.edu.cn
+shiyc@tsinghua.edu.cn

Abstract

This paper presents an initiative model for building Open Smart Space (OpenSS) environment. OpenSS is the middle stage of Smart Space identified as the test-bed for pervasive computing research. The model is regarded as a three-layered model: 1) Physical Space Layer, introducing an Ontology-based model to represent the entities, the boundary and the fusion of physical space and cyber space; 2) Agent Layer, adopting the multi-agent architecture to specify the relationships between the entities in the Physical Space Layer; 3) Mechanism Layer, aiming to manage the agents in Agent Layer to provide enhanced mechanism. This model is motivated and partially validated in the previous and ongoing scenarios, hopefully facilitating building the future OpenSS related projects.

Keywords: Pervasive Computing, Open Smart Space, Ontology-based Model, Multi-agent System.

1. Introduction

From the end of the last century, the revolution of ubiquitous/pervasive computing emerges, that various computers with different size located or embedded everywhere facilitate people's tasks, based on Weiser's vision [17]. This pervasive vision stimulates the researches on building a more natural and more efficient working/living environment for people. As the first step, Smart Space presented by NIST [1] is this kind of environment focusing the view from the "everywhere" into some "spaces". NIST emphasizes the Smart Space are work environments with embedded computing devices and multi-modal sensors, offering initiative assistance and unprecedented levels of access to the information. Later, Sailesh from Nokia complements the definition of Smart Space on W3C [18] that "Smart space is a multi-user multi-device dynamic interaction environment that is aware of its physical environment working on top of heterogeneous radio technologies/software distribution platforms". This definition comparing with the former shows the development of Smart Space from the initial to more an open environment where the multi-user multi-device dynamic interaction occurs.

Several "Smart Spaces" are deployed in universities or research institutions, validating the effectiveness of Smart Space as the test-bed for pervasive computing. As these projects on Smart Space goes future on, it is found that three successive stages exist for Smart Space research: Individual Smart Space, Open Smart Space and Smart Community, whose view from the limited-space closed area to the whole world. And currently, the Open Smart Space is the stage on which most of the Smart Space researches focuses.

Although various promising features have been achieved and presented in these Smart Spaces, there is still little research on some of the essential issues for Smart Spaces: what the model of Smart Space is and what its constitution is, etc. Having several years' experiences on Smart Spaces researches, we are trying to explore these issues in this paper. Being an initiative work, this paper mainly focuses on building the model for Open Smart Space (OpenSS), trying to answer some essential questions and hopefully presenting the model as the guidelines for building future OpenSS related projects.

This paper is organized as follows: Section 2 presents the development of the Smart Space and analysis of the requirements of OpenSS model. Section 3 gives the overview of the initiative three-layer-OpenSS-model and Section 4 goes further to illustrate the details of each layer. Section 5 and Section 6 introduces the related ongoing scenarios for supporting the model and the comparison with the related work. And finally we draw the conclusion in Section 7.

2. Open Smart Space

2.1. Three stages for Smart Space

Researches on Smart Space keep active in recent years, as many related Smart Space-based projects has been achieved or is still ongoing in both of the industry and academia [1][2][3][4][5][6][9][11]. Summarizing the current Smart Space-based projects and its developing history, we divided the whole Smart Space evolution into a three-stage developing process: Individual Smart Space, Open Smart Space and Smart Community. From Individual Smart Space to Smart Community, the Smart Space is developing from a limited-space, isolated environment to becoming multiple open, interrelated and interactive working spaces.

Individual Smart Space focuses on building a smart

978-1-4244-2020-9/08/$25.00 ©2008 IEEE

human-computer interactive environment in a room-limited space: embedding the awareness and communication into the spaces, creating natural human-computer interface and providing eternity, robustness and invisible services for the users. In this stage, the constitution of the Smart Space is stable, certain and lack of openness. The interactions between devices and users are limited in one "space" (e.g. classroom, meeting room).

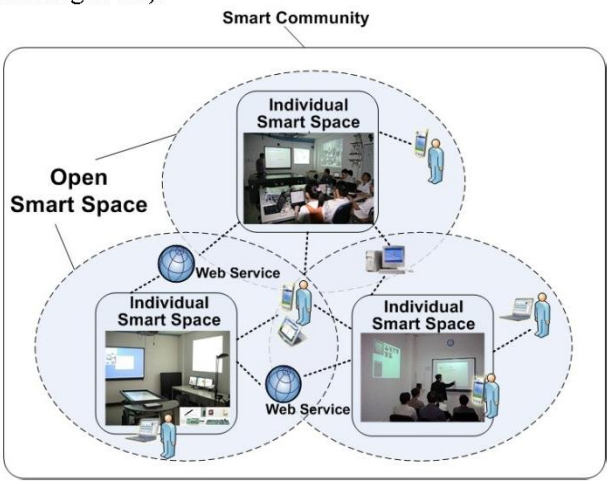

Figure 1. Open Smart Space Sketch Map

Open Smart Space aims to bring more openness into Smart Space, which enables mobile and handheld devices roaming with users entering or leaving. The devices, services and interfaces are no longer only limited in the "space". Identifying its dynamical constitution, facilitating the interoperation between services and devices and supporting multi-users simultaneously interaction and collaboration, are the three main research issues of Open Smart Space.

Smart Community tries to connect and organize multiple or even large scale of Open Smart Spaces, where the services and devices are shared and the users are able to seamless move from one Smart Space to another.

Nowadays, majority of the Smart Space-based projects has made some extent achievements in the first stage, and mainly focusing on the second stage. Even a few projects [3] [7] [15] has started to explore the third stage.

2.2. Open Smart Space

Discussed in Section 2.1, Open Smart Space (OpenSS) is the middle stage of the Smart Space evolution. Figure 1 illustrates the overview of the OpenSS. OpenSS extends the boundary of Smart Space and allows people and resources (computing devices and services) which do not belong to Smart Space to dynamically join in and leave. Any devices brought by the users could interact with each other when they first "belong" to the environment. Also as Figure 1 shows, Open Smart Space acts as a mediator between multiple Individual Smart Spaces, enabling them to communicate and collaborate with each other.

The key feature of OpenSS is its openness, which is mainly relies on three main characteristics: 1) mobility, dynamism and open boundary. Mobility requires the spaces should aware the movements of the users and their devices and adapt itself to these movements, especially when they enter or leave the space; 2) dynamism shows that the relationships and interactions between the entities are established and destroyed and dynamically in terms of the dynamical changing status of OSS; 3) open boundary means the boundary of the Smart Space is no longer limited to the Space boundary, but largely extended by the pervasive technologies.

Understanding the three main characteristics, we encounter interesting questions which are essential to OpenSS:

- How to describe or formalize the constitution of OpenSS?
- Is it possible to define the boundary of the OpenSS?
- How to aware the movement of the entities?
- How to understand the dynamically built interactions between the user and the devices?

In the following section, we will try to answer this question by leveraging our presented OpenSS model.

3. Overview of the initiative OpenSS model

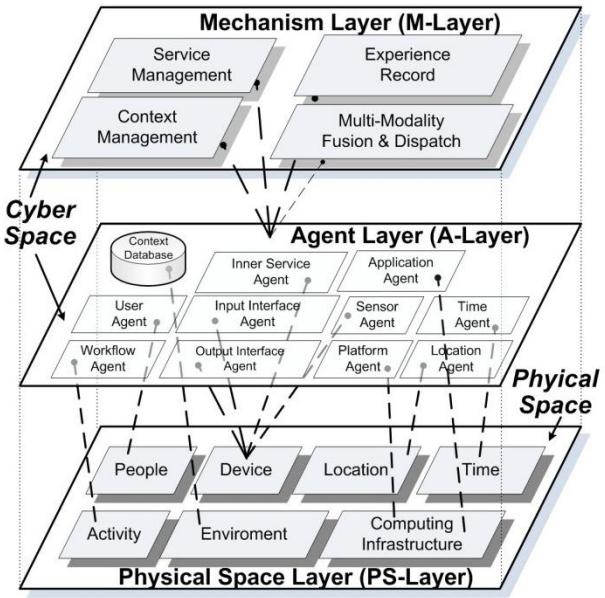

Figure 2. Overview of the Open Smart Space Model

OpenSS is a Smart Space, which fuses the physical space and the cyber space. In the physical space, it is intuitive and clear to define the constitution of OpenSS, while in the cyber space; it is more suitable to identify their interactions. Hence, we present our OpenSS model as a three-layered model: Physical Space Layer (PS-Layer), Agent Layer (A-Layer) and Mechanism Layer (M-Layer); the first layer denotes the physical space and the other two denote the cyber space, as Figure 2 shown below. Here we only give the overview of the model, leaving the detailed explanation is the next

Section.

To embody the fusion between physical and cyber space, there is mapping relationship between the layers, that all the entities in the physical space have their corresponding counterparts in cyber space (refer to the parts connected by dash line between A-Layer and PS-Layer in Figure 2). The fusion between physical space and cyber space is regarded as the entities in physical space mapping from the physical space into the counterparts in cyber space, interacting between each other and also with other entities that only belongs to cyber space inside the cyber space.

4. Layers of the model

4.1. The Physical Space Layer

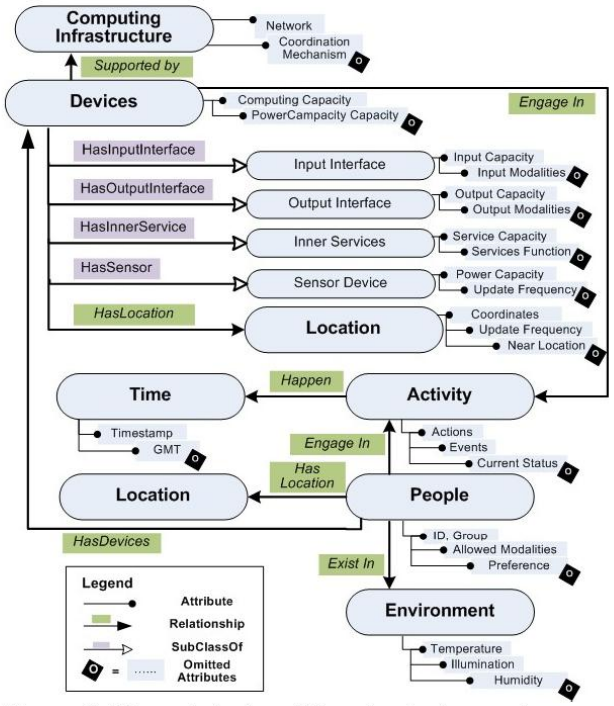

Figure 3. The ontologies of the physical space layer.

The ontology-based modeling is adopted to model the physical space layer, in order to clearly describe the constitution of OpenSS from the physical space view. Figure 3 shows the ontologies in a general view.

The OpenSS consists of seven entities, which can be formalized as follows:

$$OpenSS = \langle P, A, D, I, E, L, T \rangle$$

, where $\langle P, A, D, I, E, L, T \rangle$ denotes ***P***eople, ***A***ctivity, ***D***evices, software ***I***nfrastructure, ***E***nvironment, ***L***ocation and ***T***ime respectively, and the relationship between these entities can be indentified in the Figure 3.

- ***P***eople: denoting the users in OpenSS;
- ***A***ctivity: denoting the tasks in OpenSS. The task may involved by the People or devices or both of the two;
- ***D***evice: denoting the computing devices in OpenSS. It encompasses three kinds: the input/output devices that allows the users to input or receive output; the service devices that does not have any input/output interface but only execute computing tasks; sensor device that senses the status of the other entities in OpenSS;
- software ***I***nfrastructure: denoting the computing platform for supporting distributed computing modules to communicate and coordinate with each other.
- ***E***nvironment: denoting the environmental status of OpenSS, such as the temperature;
- ***L***ocation: denoting the location information of the related devices.
- ***T***ime: denoting the timestamp information of the related devices.

In OpenSS, there are two kinds of entities having mobility that can join in and leave OpenSS over time: Devices and People, which both have the relationship with the Location entity and Activity entity. Hence we could define whether or not the devices or people belong to the OpenSS.

Definition 1: *Device/People belongs to the OpenSS if and only if the Device/People has relationship with one location entity or one activity entity which belongs to the OpenSS.*

The boundary of the OpenSS is no longer limited by the space (meeting room or classroom) but regarded as the limitation of the network access. In the OpenSS, if the system is able to be aware of the location of the devices or people, we think they are reachable and partially accessible; or if the devices or people have engaged in one of the activities in OpenSS, it is intuitive that they belong to the OpenSS; otherwise, they do not belong to OpenSS.

Defining whether or not the two mobility factors (People and Devices) belong to OpenSS helps to specify the constitution of the OpenSS, which also answers one of its key essential questions.

The ontology-based physical space layer specifies the constitution and the relationship between the entities of OpenSS, and thus is the basic and fundamental layer for building OpenSS environment.

4.2. Multi-agent layer

Above PS-Layer in cyber space, Multi-agent layer aims to specify the relationship and the data stream of OpenSS, illustrated in Figure 4.

The cyber space of OpenSS is defined as a multi-agent system, where agents communicate and coordinate with each other to facilitate users' task. Brought the idea of W3C Multimodal Interaction Framework [10], the M-layer emphasizes the multi-modalities data stream from User to the cyber space and returns back. Also, in order to maintain the mobility, the location agent will be attached to every userAgent, InputAgent and OutputAgent as long as their location could be aware by the system. The Location Agent not only updates its attached agents' location information

automatically, but shares it to the attached agent as well. The advantages of involving Location Agent are: 1) it saves the attached agents' work to maintain location information; 2) multiple location Agents could work together to update the location information, enabling the location awareness from the center mode only into a center-and-distributed way.

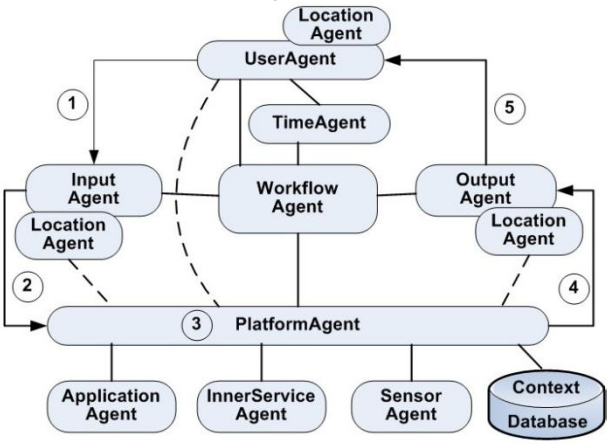

Figure 4. The multi-agent layer (M-layer).

Moreover, note that the activity entities are mapped to Workflow Agents, which is because we use the workflow to present and manage the activities in OpenSS.

All the agents are inherited from the Agent class, which provides the interfaces of communication with other agents and maintains current status. Due to the limited space in this paper, we omit the formalized expression of the Agent and interaction primitives, and focus on the multi-modalities interaction between the User and the OpenSS.

There are totally five steps (refer to Figure 4: 1-5)

1) Multi-Modalities Input:

In physical space, user generates multi-modalities input through the input interface of the devices, such as using laser pen to input a track of laser points, or using vocal to input a voice stream. However, in cyber space, these inputs are received by Input Agent, not sent by User Agent. Therefore, in this step, the Input Agent needs to identify which User Agent corresponds to the user who makes the input, and establish the relationship between the User Agent and this multi-modalities input. The identification is mainly done by considering the distance between the Input Agent and the User Agent and the relationship saved in the system database.

2) Modalities-relevant Input

In this step, the Input Agent sends the modalities-relevant input as a six-tuple data to the Platform Agent:

$$MInput := \langle U_{id}, IN_{id}, L, Type, Data, Time \rangle$$

, where U_{id} and IN_{id} denotes the identifier of the User Agent and Input Agent; L denotes the location information; *Type* and *Data* present the modalities-relevant input, such as *Type* is laser pointer and *Data* is the track of the laser points; *Time* records the timestamp of the occurred input.

3) System Processing

When the Platform Agent receives the *MInput*, it first transforms the *MInput* into a modalities-irrelevant data *NmInput*:, that

$$f_{in} : MInput \rightarrow NmInput := \langle U_{id}, L, Obj_{id}, Action, Act_{Data}, Time \rangle$$

The U_{id}, *Time*, L keep equal in *MInput* and *NmInput*, while the f_{in} interprets the modalities-relevant input $\langle IN_{id}, Type, Data \rangle$ into modalities-irrelevant commands $\langle Obj_{id}, Action, Data \rangle$, which is understandable and general input command between the agents in the OpenSS. For example, the former *MInput* may be interpreted as:

$Obj_{id} := Video\_Window$

$Action := MovingObject$

$Act_{Data} := \{Point_{from}, Point_{end}\}$.

The Platform Agent will send *NmInput* to the corresponding agents, such as Application Agent or InnerService Agent to let them handle these input commands.

Should the agents have some feedbacks that need showing to the users, the agents gives a similar *NmOutput* command, which is also modalities-irrelevant, to the related Output Agent in the following format:

$$NmOutput := \langle U_{id}, L, Obj_{id}, Action, Act_{Data}, Time \rangle .$$

4) Modalities-relevant Output

In this step, Platform Agent generates the *MOutput*, which is multi-modalities, from the NmOutput:

$$f_{out} : NmOutput \rightarrow MOutput := \langle U_{id}, OU_{id}, L, Type, Data, Time \rangle$$

The process is similar as the 2) step. Function f_{out} dispatches the Output information to the appropriate OU_{id} Output devices, and transforms the modalities-irrelevant output command into modalities-relevant output information $\langle OU_{id}, Type, Data \rangle$. For example, for the voice-out output, its information could be as follows:

$OU_{id} := Speaker1$

$Type := VoiceStream$

$Data :=$ The voice stream of the Object (Obj_{id}), such as a text file.

5) Multi-modalities Output

Finally, the Output Agent gives the output to the user through the output interface in the physical spaces, which is from the cyber space view, the Output Agent sending the *MOutput* to the User Agent.

Note that the relationships between input, output and the user are established dynamically in terms of the current status of the OpenSS instead of the static pre-defined one, the multi-modalities interaction addresses the issues of dynamically interaction in OpenSS.

4.3. Mechanism layer

The Mechanism Layer aims to provide enhanced mechanisms for the agents in Agent Layer. Actually, these mechanisms are implemented by one or several special Platform Agents.

Currently, as Figure 2 shows, there are four

mechanisms in this layer:

- Service Management [19]:

All the interfaces of the agents in Agent Layer are regarded as the "service" in OpenSS. The Service Management facilitates the management of all these services, including *service discovery* when one agents wants to find an appropriate service to finish their task; *service composition* when the query services does not exist, requiring multiple services to work together to satisfy with the query; *service arbitration* when multiple agents query for the same service whose maximum capacity is exceeded at the same time.

- Context Management [14]

Context Management aims to provide the augmented features for OpenSS to maintain and utilize the context information. It tries to organize the entire context in Agent Layer (including the context in every agents and the core database), to extract the high-level semantic from the context by reasoning methods, and to facilitate the query interface for other agents to obtain the context & semantic information.

- Multi-modalities Fusion & Dispatch:

Multi-modalities fusion and dispatch implements the interpreting from the multi-modalities input into the modalities-irrelevant input, processing the input to achieve the modalities-irrelevant output, and transforming it to the multi-modalities output. As introduced in the Section4.2, the f_{in} and f_{out} are actually supported by this mechanism.

- Experience Record [13]:

The whole OpenSS keeps evolving as time goes on, and it is significant to capture the experience of the OpenSS for reviewing and scene re-creation. It collects all the necessary information of all the agents and the database, stores them with indexing and optimizing, and provides the interface or application for later accessing the history of the OpenSS.

5. Validating and ongoing projects

The presented Open Smart Space model is partially motivated by reviewing the existing Smart Space related projects mainly in our research group from earliest to the latest: 1) Smart Classroom [2]; 2) Open Smart Classroom [3]; 3) Smart Environment for Multi-user Interactive Collaboration(SEMIC) [11] ; 4) UTable [12] .

They all adopt the multi-agent system architecture to facilitate the communication between the distributed modules, which validates multi-agents system architecture is effective and successful in building cyber space. Moreover, layered-based design is implemented in all these projects that providing enhanced mechanism in the high layer, such as context management, service management, etc., and addressing the basic functions in the low layer.

In addition, these four projects shares four entities: People, Devices, Environment and Time. The earliest Smart Classroom does not take Location and Activity into consideration. Open Smart Classroom considers Activity that implemented by Workflow and SEMIC augments the Smart Space by involving location-aware system. Involving Activity and Location entities really enhances the functions of Smart Space, and more importantly, making Smart Space constitution integrated to some extent.

UTable in fact is only an interactive device in Smart Space, while it is used to research on multi-users multi-modalities interaction. In uTable, the devices are separated into three parts: input interface, output interface and computing modules. Input/Output interface is responsible for receiving/giving modalities-relevant input/output. And computing modules takes charge of processing modalities-irrelevant input, which is obtained by interpreting and fusing multiple modalities-relevant inputs, and sending modalities-irrelevant output, which will be transformed into modalities-relevant output. uTable motivates the Open Smart Space model to address the multi-users multi-modalities interaction, which has been elaborated in Section 4.2.

To sum up, OpenSS inherits the advantages of the existing projects, enlarging the entities in physical layer, organizing the agent layer to address multi-user multi-modalities interaction, and providing the enhanced agent management mechanism in the mechanism layer.

Currently, there are also ongoing projects that are going to adopt the whole or partial the proposed OpenSS model to design their systems. For example, we are now trying to integrate the uTable into the OpenSS-based meeting room or classroom, where the OpenSS model plays the fundamentally important role.

6. Comparison with related works

As the objective of building OpenSS model is trying to guide the future OpenSS projects, we first compare our model with several (Open) Smart Space projects, whose explicit or implicit models are identified. NIST Smart Space [1] clarifies the research issues and presents merging multiple technologies into Smart Space. Interactive Workspaces [4][8] focuses on the interaction between multiple users and devices, presenting a service framework for building Smart Space. Intelligent Room [6][16] or Smart Classroom [2], present a multi-layered framework, including computing platform, resource management, context management, user interface (GUI or multi-modality) and applications. The above five projects implicitly built the Smart Space from the view of software infrastructure or cyber space, while none of these projects specifies the constitution of Smart Space. Active Space [9] presents an explicit model for Smart Space, which consists of several entities: physical space, software infrastructure, context, applications and user. Active Space generally identifies the relationship between these entities，however, fails to explore a further step to make it detailed.

There are also many works trying to model the pervasive computing environment by presenting a context model [5][14][22] or location model [20][21]. These works focus more on the context, location, and the whole pervasive computing environment, while our model focuses on Open Smart Space. Using their general model could not answer the questions for OpenSS, such as the boundary and the constitutions of OpenSS. As a specific model for OpenSS, our model fills the gap between general model for pervasive computing environment and the OpenSS.

6. Conclusion and Future Work

To best knowledge of the authors, this is the first work trying to establish the Open Smart Space model, which answers the essential question of the Smart Space presented in Section 2.2. In this paper, we present a three-layer-model to understand the constitutions and the relationships between each part of OpenSS. The presented ontologies in Physical Layer generally address the formalization of the constitution and the definition of boundary for OpenSS. Agent Layers not only adopts the location agent to aware the movement of the entities, but also introduces the mechanism of addressing multi-modalities interaction to understand the dynamically built interactions between user and multi-devices. Mechanism Layer presents several key high-level mechanisms for building OpenSS projects.

Although the presented model is able to address many essential questions, its performance still needs more practical evaluation to make it more convincing. Currently we are designing evaluation criterion and implementing the application to evaluate and polish our OpenSS model. Hopefully the introduced model or the revised one could be the guidelines for building future OpenSS project, and also facilitate the researcher or even normal person to understand OpenSS better.

Acknowledgement

This work is supported by National High-Tech Research and Development Plan of China under Grant No. 2006AA01Z131 and National High-Tech Research and Development Plan of China under Grant No. 2008AA01Z132.

References

[1] Lynne R., Vincent S..: NIST Smart Space: Pervasive Computer Initiative. In Proc. of IEEE 9th International Workshop on Enabling Technologies: Infrastructure for Collaborative Enterprises (WETICE), p6-11, 2000.

[2] YC SHI, W Xie, GY Xu, et al.: The smart classroom: merging technologies for seamless tele-education. IEEE Pervasive Computing, Vol 2, No 2, 2003, pp. 47-55.

[3] Yue Suo, Naoki Miyata, Toru Ishida, Yuanchun Shi: Open Smart Classroom: Extensible and Scalable Smart Space Using Web Service Technology. In the Proceedings of ICWL 2007, pp. 428-439, 2008.

[4] Brad Johanson, Armando Fox, Terry Winograd: The Interactive Workspaces Project: Experiences with Ubiquitous Computing Rooms. IEEE Pervasive Computing, 2002.1(2): 67~74.

[5] Harry Chen, Tim Finin, Anupam Joshi, et al.: Intelligent Agents Meet the Semantic Web in Smart Spaces. IEEE Internet Computing, 8(6), pp.69-79, 2004.

[6] Phillips, B. Metaglue: A Programming Language for Multi-Agent Systems: [M.Eng Thesis]. Massachusetts, USA: Massachusetts Institute of Techonlogy, 1999

[7] Gajos K. Rascal - a Resource Manager for Multi Agent Systems In Smart Spaces. In: Proceedings of the Second International Workshop of Central and Eastern Europe onMulti-Agent Systems (CEEMAS 2001), Kraków, Poland, 2001. 111-120

[8] Ponekanti S R, Lee B, Fox A, etal. ICrafter: A Service Framework for Ubiquitous Computing Environments. In: Proceedings of the 3rd International Conference on Ubiquitous Computing, Atlanta, 2001. 56~75

[9] Román M, Hess CK, Cerqueira R, Ranganathan A, Campbell RH, Nahrstedt K. Gaia: A middleware infrastructure to enable active spaces. IEEE Pervasive Computing, 2002,Oct-Dec:74-83.

[10] W3C Multimodal Interaction Framework. http://www.w3.org/TR/mmi-framework/

[11] SEMIC: Smart Environment for Multi-user Interactive Collaboration.http://media.cs.tsinghua.edu.cn/~pervasive/#projects_semic

[12] Yue SHI, Chun Yu, Yuanchun SHI. Finger Gesture Interaction on Large Tabletop for Sharing Digital Documents among Multiple Users. In Proceedings of the First IEEE International Conference on Ubi-media Computing and Workshops. (to appear)

[13] WS He, YC Shi, WJ Qin. INFERS: An Infrastructure for Experience Record in Smart Spaces. In Proceedings of ICWL 2006). July 2006. Penang, Malaysia.

[14] Qin, W., Suo, Y., Shi, Y.: Camps: A middleware for providing context-aware services for smart space. In: Proc. GPC. (2006) 644–653

[15] Peters S, Look G, Quigley K, et al. Hyperglue: Designing High-Level Agent Communication for Distributed Applications. In Proc. of AAMAS 03'.

[16] Nicholas Hanssens, Ajay Kulkarni, Rattapoom Tuchinda and Tyler Horton. Building Agent-Based Intelligent Workspaces. In ABA Conference Proceedings. (2002)

[17] Weiser, M. The computer for the twenty-first century. Scientific American, 94-100, September 1991.

[18] Sailesh Sathish. Using Declarative Models in Multi-device Smart Space Environments. http://www.w3.org/2007/02/dmdwa-ws/talks/sathish.pdf

[19] Jingyu Li, Yuanchun Shi: Baton: A Service Management System for Coordinating Smart Things in Smart Spaces. In Proceedings of EUC 2005, 11–20.

[20] René M., et.al.: A Spatial Programming Model for Real Global Smart Space Applications. In Proc. of 6th IFIP WG 6.1 Intl' Conf., pp. 16-31, 2006.

[21] Satoh, I.: A location model for pervasive computing environments. In Proc. of PerCom 2005, pp. 215 – 224

[22] Qin, W., Daqing, Z., Mounir, M., et al.: Combining User Profiles and Situation Context for Spontaneous Service Provision in Smart Assistive Environments. In: Proc. UIC. (2008).