SmartShadow: Modeling A User-centric Mobile Virtual Space

Li Zhang, Gang Pan*, Zhaohui Wu, Shijian Li
Department of Computer Science
Zhejiang University, China
Email: gpan@zju.edu.cn

Cho-li Wang
Department of Computer Science
The University of Hong Kong
Email: clwang@cs.hku.hk

***Abstract*—This paper attempts to model pervasive computing environments as a user-centric "*SmartShadow*" using the BDP (Belief-Desire-Plan) user model, which maps pervasive computing environments into a dynamic virtual user space. SmartShadow will follow the user to provide him with pervasive services, just like his shadow in the physical world. In the BDP model, desires of a user are inferred from his belief set, and plans are made to satisfy each desire. Pervasive service is introduced to describe computing resources in the cyberspace, which can be organized by the user's BDP to accomplish his desires. The composition process maps pervasive services into a user's SmartShadow. The model is logically natural and simple, and can flexibly model dynamics of pervasive computing spaces. In addition, we implement a simulation system to verify and evaluate the SmartShadow model.**

I. Introduction

Pervasive/ubiqitous computing has become an emerging field since Mark Weiser's pioneering article [1]. Various technologies have been developed for supporting pervasive computing, and many prototype systems have been built to convey and illustrate the paradigm of pervasive computing.

However, there is currently little work on modeling pervasive computing environments. Dima et al [2] proposed a layered conceptual model for pervasive computing, composed of environment layer, physical layer, resource layer, abstract layer and intentional layer. Similarly, Muhlhauser et al [3] divided a pervasive computing system into three layers: gadgets, integration and UC world. Another high-level conceptual model was presented by Henricksen et al [4], consisting of four components: devices, users, software components and user interfaces. Costa et al [5] made an effort to design a general software infrastructure for pervasive computing. They reviewed ten issues/challenges of pervasive computing and discussed each aspect across application life cycle [6]. The work described above is more from the layered view and the systematic view.

Blackstock et al [7] presented a Ubicomp Common Model (UCM) for interoperability in order to allow developers to innovate and evolve their platforms while allowing others to build interoperable applications. Ranganathan et al [8] used ambient calculus and ambient logic to formally verify that the pervasive computing environment satisfies certain desired properties. The work is not strong at interactions and dynamics of ubiquitous computing systems. A mechanism for modeling pervasive computing environments is still a challenging task.

TABLE I
Smart Space vs. SmartShadow

	Smart Space	**SmartShadow**
Substantiality	physical	virtual
Mobility	attached to a physical space	attached to a user like his shadow
Dynamics	low	high
User-centric	no	yes

This paper attempts to build a user-centric abstract model to characterize the properties and capabilities of pervasive computing environments. We name this model *SmartShadow*, with which the pervasive computing environment around a user could be modeled as a mobile virtual space always coupled with the user like his/her shadow. Actually, SmartShadow can be considered as a generalization of the concept of smart space [9] from the user-centric view. The comparison with smart space is shown in Tab.1.

II. SmartShadow Model

The goal of pervasive computing is to spontaneously provide users transparent services anytime and anywhere. In a pervasive computing environment, we can image that each user has a user-specific space always along with him/her wherever he/she moves. The virtual user-specific space can be considered as a service subspace that is projected from the global space of spontaneous and transparent services. The user-specific subspace is called SmartShadow, a virtual user-centric smart space, which mainly depends on the user.

A. Modeling Users by BDP

In a pervasive computing environment, one of the most important factors is the user. To formally define the SmartShadow, firstly we propose a Belief-Desire-Plan(BDP) model to describe users, inspired by BDI agent model [10]. The BDP model consists of three key components:

- **Belief**: user's beliefs describe the informational state of a user. Examples of beliefs might be: user's schedule, behaviorial characteristics, preference.
- **Desire**: desires represent the motivational state of a user. They represent objectives and situations that the user

978-1-4244-3304-9/09/$25.00 ©2009 IEEE

would like to bring about or finish, for example, *open the door*, *listen to music*, *book tickets*. User's desires may vary with user's beliefs and physical environmental context. Each desire has its dynamic priority. Desires with high priority would be satisfied at first.

- **Plan**: a plan is a sequence of actions to achieve one or more of user's desires. Each desire has at least one plan. Which plan is the best for a desire depends on the context.

Thus, each user has three sets for beliefs, desires and plans respectively: B, D, P.

In the BDP model, we assume that user's desires could be inferred from the user's beliefs and the context. The suitable plan for a desire will vary with the context. The BDP-driven computing is illustrated as:

Step 1: $B + Context \Rightarrow D$
Step 2: $D + Context \Rightarrow P$
Step 3: Build the Instance of P
Step 4: Run the Instance

B. SmartShadow: Description and Definition

An entire pervasive computing environment could be regarded as a space full of various *pervasive services*. Here, a *pervasive service* means a functional element provided mutually among applications, devices, and users. The assumption is reasonable since some work on wrapping devices as services is available, for example, Service Oriented Device Architecture (SODA) [11].

Assume that $S_{available}$ is the available services in the global space, U_{bdp} is a user, and $S_{desired}$ is the services that satisfy the user's desires. Similar to generation of a shadow by casting light onto a object, $S_{desired}$ is a result of casting $S_{available}$ on U_{bdp}, i.e. $S_{desired}$ is the dynamically mapping of $S_{available}$ upon U_{bdp}. A shadow is always movings along with the object, equally the $S_{desired}$ varies with U_{bdp}. We call $S_{desired}$ as *SmartShadow*. It is a subspace of $S_{available}$, shown in Fig. 1.

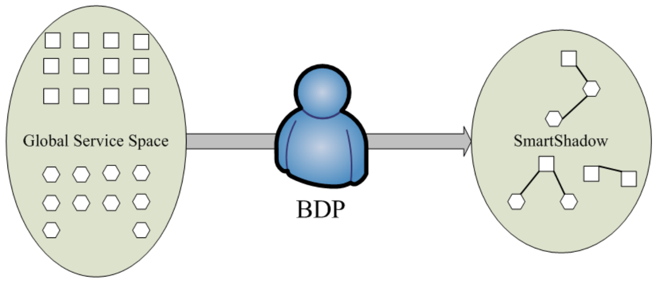

Fig. 1. Illustration of Building SmartShadow

Assume that the global space of pervasive services is:

$$PerSS = \{S_i\}$$

Thus, SmartShadow is defined by a quadruplet:

$$SmartShadow = \{PerSS, U_{BDP}, \Psi(\bullet,\bullet), S_{desired}\}$$

Where,

$$S_{desired} = \{\{S^i_{i_j}, j = 1, ..., N_i\}, i = 1, ..., T\}$$

$\{S^i_{i_j}, j = 1, ..., N_i\}$ is associated with a plan P_i in U_{bdp}. $\Psi(\bullet,\bullet)$ maps $PerSS$ to $S_{desired}$ based on U_{bdp}, that is:

$$S_{desired} = \Psi(PerSS, U_{bdp})$$

It provides dynamical generation of a virtual user space upon user's BDP.

C. Building SmartShadow

The pervasive services are organized in a SOA-like architecture [12] for flexibility and agility. They are a world wide mesh of collaborating services, which are published and available for invocation on Service Bus. We can benefit from ubiquity and compositing mechanism of services to realize a SmartShadow in an open framework. We augmented OWL-S [13], [14] to describe pervasive services. Besides the existing functional description specification, two new fields are added: *Composition Conditions* is for defining constraint conditions for composition and algorithms, and *Service Context* denotes the context of a service.

A generic framework for SmartShadow building algorithm Ψ is described as follows.

Generic framework of SmartShadow building algorithm

```
1   Get a set of services $S_p = \{S_i\}$ from $PerSS$ by
    constraint filtering subject to the current context;
2   Select services from $S_p$ according to $U_{bdp}$, a common
    way is ontology-based service matching;
3   For each plan $P_i$ of the user, do service composition
    $\{S^i_{i_j}, j = 1, ..., N_i\}$ ;
4   Return the service composition result, noted as $S_{desired}$.
```

Firstly a typical set of services is obtained via service discovery protocols. Those services that do not meet the contextual constraints of service are filtered out. Prior to service composition, the service dependency is examined. After service composition, the services will run in the order defined for each plan, like the orchestration manner in service composition: utilize a high-level scripting language to control the sequence and flow of service execution.

III. Prototype and Evaluation

A. Prototype

In order to verify the SmartShadow model, we built a prototype, which consists of two parts, shown in Fig.2:

- **BDP management module**. For simplicity, we consider three kinds of user beliefs in this implementation, they are user's schedule, preference, and behavioral characteristics. The implementation of BDP model is mainly based on Jadex[15], a BDI agent construction and reasoning tool. A few modifications is made to meet special requirements of the BDP model. The user's beliefs, desires and plans are represented in Jadex ADF files in the BDP

TABLE II
EXPERIMENT CONFIGURATION

Category	Parameter	Value scope
Service	Execution time	200(±50)-2000(±500)
	Execution timeout	5000
	Discovery timeout	5000
	Concurrency limits	$N(3,1)$
Space	Service resident duration	Infinity
	User resident duration	2000-20000
	Average of service number	30-120
User	User number	1-30

base. A desire is associated with one plan. The reasoner is in charge of the BDP evolution. With Jadex BDI Reasoning Engine [16], the prototype supports several different methods for reasoning and selecting of plans. The planner, built based on ScudCORE [17], is used to generate user's plans with constraints of the current contexts. It also is responsible for replanning once the associated contexts changed; the executor manages the life cycle of user's plans;

- **Pervasive service framework**: Service discovery employs the UDDI protocol [18]. In the prototype, scenarios and activities in physical spaces are simulated with MASON [19], a multi-agent simulation framework, which supports spatial relation and debug mechanism of agents, and enables development of graphical interface to observe simulation results.

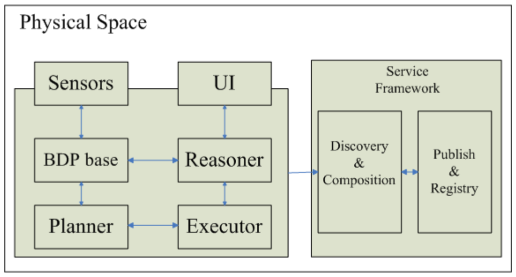

Fig. 2. The prototype framework

B. Evaluation configuration

We set up a simulation of three physical regions, shown in Fig.3, where a green circle represents a pervasive service, a white rectangle represents a user, and a line segment between them shows their relation to a SmartShadow. A user may switch from a region to another. The contexts are also always varying, which may make some new desires occur and some old ones canceled. A plan served for a desire is carried out by searching and compositing pervasive services in the available service space. In order to simulate real situation better, we let parallelization number of each pervasive service subject to the Gaussian distribution of $N(3,1)$. The general configuration for experiments are shown in Tab.II.

We employ *success rate* of execution of user plans to measure the system. There are four results for each execution of a plan: 1) success, 2) failure due to some services unavailable,

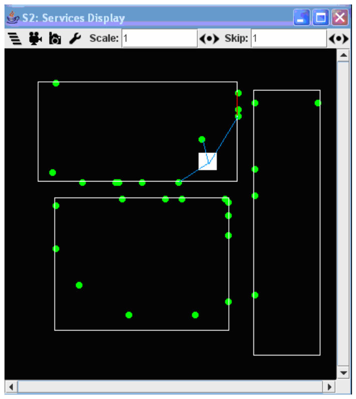

Fig. 3. Evaluation environment using MASON

denoted as "*Failure 1*", 3) failure due to timeout of execution, denoted as "*Failure 2*", 4) failure due to context changed, denoted as "*Failure 3*". We design four experiments to assess how the factors of users, services, mobility affect success rate of plans' completion. Since many parameters are randomly generated, all results are the average of many simulations.

C. Results

Experiment 1: Single user. The simulation with a single user is carried out with 30 services distributed in the three isolate regions, to roughly verify the proposed SmartShadow model. Results demonstrate that, for a user with average of 31 desires, 23 desires are completed smoothly, shown in Fig.4.

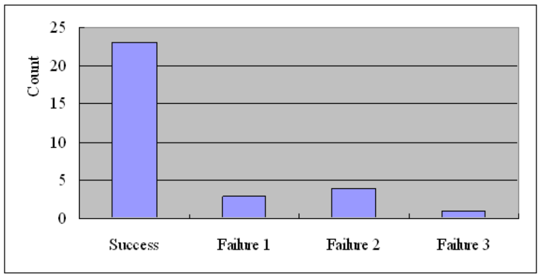

Fig. 4. Result of the experiment 1

Experiment 2: Effect of users. This experiment is to evaluate the effect of user number. We keep 120 services for each region. Figure 5 shows that the success rate decreases when the user number increases, which could be explained that the pervasive services are not always available due to their concurrency limits.

Experiment 3: Effect of services. Another key factor for success rate is the number of services. Figure 6 demonstrates that, the success rate will rise with increase of service number, since users can obtain more services for satisfying their desires. In some scopes an abnormal phenomenon exists, where the success rate declines locally with the increasing service quantity. This is because in special cases increase of services causes extra cost on service management, discovery and execution.

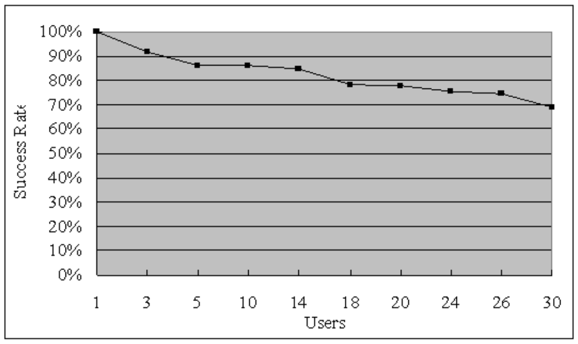

Fig. 5. Result of the experiment 2

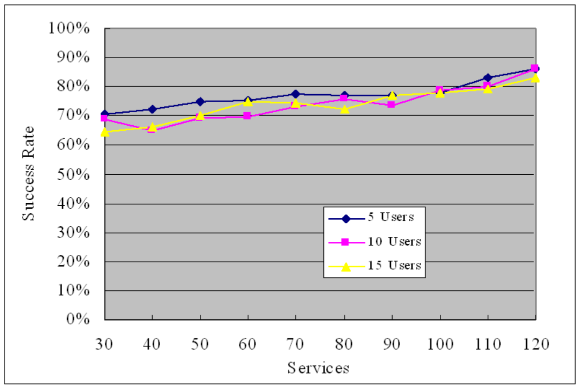

Fig. 6. Result of the experiment 3

Experiment 4: Effect of users' mobility. This experiment is to assess the effect of frequency of a user switching from one region to another within 100000 time unit. The number of services in each region is set to 120, which is nearly sufficient. The result is shown in Fig.7, where the success rate decreases as the switch frequency increases. With the user frequent mobility, services may be interrupted, or the change contexts may cancel execution of a plan.

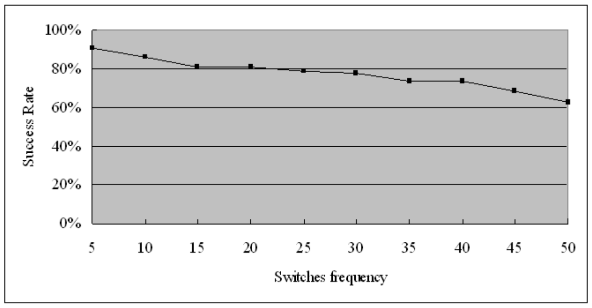

Fig. 7. Result of the experiment 4

IV. Conclusions

We have preliminarily built *SmartShadow*, a user-centric model, to characterize the properties and capabilities of a mobile virtual user space. By modeling service space and user's belief, desire and plan, pervasive services could be dynamically filtered and mapped into the virtual user space.

A good model should facilitate developers to construct more flexible pervasive computing systems. Our ongoing work is to develop a generic software infrastructure using the proposed SmartShadow model and to deploy some prototypes in the real world to make the model really useful.

Acknowledgements

This work was supported by the 863 Program of China (No. 2008AA01Z132), NSFC (No. 60525202, 60533040), and the National Key Technology R&D Program of China (No. 2006BAH02A01).

*Corresponding author: Gang Pan.

References

[1] M.Weiser, ”The Computer for the 21st Century”, *Scientific American*, pp.94-100, 1991.
[2] A.Dima et al, A conceptual model for pervasive computing, *Int'l Workshop on Parallel Processing*, 2000.
[3] M. Muhlhauser, Ubiquitous computing and its influence on MSE, *Int'l Symposium on Multimedia Software Engineering*, 2000.
[4] K.Henricksen et al, Infrastructure for Pervasive Computing: Challenges, *Informatik01: Workshop on Pervasive Computing*, 2001.
[5] C.Costa et al, Toward a General Software Infrastructure for Ubiquitious Computing, *IEEE Pervasive Computing*, 7(1):64-73, 2008.
[6] G.Banavar, J.Beck, E.Gluzberg et al, Challenges: an application model for pervasive computing, *Int'l Conf. on Mobile Computing and Networking (MobiCom'00)*, pp.266-274, 2000.
[7] Michael Blackstock, Rodger Lea, and Charles Krasic. Evaluation and Analysis of a Common Model for Ubiquitous Systems Interoperability. In Proc. the Sixth International Conference on Pervasive Computing(PERVASIVE 08), Sydney, Australia, Springer-Verlag, 2008.
[8] Anand RanganathanRoy H. Campbell, Provably Correct Pervasive Computing Environments, In Proc. the Sixth Annual IEEE International Conference on Pervasive Computing and Communications, Hong Kong, 2008.
[9] L.Rosenthal et al, NIST Smart Space: Pervasive Computing Initiative, *IEEE WET ICE 2000.*
[10] A.Rao et al, Modeling Rational Agents in a BDI-architecture, *2nd Int'l Conf. on Principles of Knowledge Representation and Reasoning*, 1991.
[11] S.Deugd, R.Carroll, K.E.Kelly, B.Millett, J.Ricker. SODA: Service Oriented Device Architecture. IEEE Pervasive Computing, 5(3):94-96, 2006.
[12] T. Erl, Service-Oriented Architecture: Concepts, Technology, and Design, Prentice Hall, 2005.
[13] David Martin, Mark Burstein, Jerry Hobbs, Ora Lassila,et al. OWL-S: Semantic Markup for Web Services. W3C Member Submission 22 November 2004. Available: http://www.w3.org/Submission/OWL-S/
[14] S. McIlraith, T. C. Son, and H. Zeng, Semantic Web Service, IEEE Intelligent Systems, vol.16, no.2: pp. 46-53. 2001.
[15] Lars Braubach, Alexander Pokahr, Winfried Lamersdorf. Jadex: A BDI Agent System Combining Middleware and Reasoning, Chapter of *Software Agent-Based Applications, Platforms and Development Kits*, Birkh, Springer, 2005.
[16] Alexander Pokahr, Lars Braubach, Winfried Lamersdorf. Jadex: A BDI Reasoning Engine, Chapter of Multi-Agent Programming, Editors: R. Bordini, M. Dastani, J. Dix and A. Seghrouchni, Springer, 2005.
[17] Tong Li, Gang Pan, Haoyi Ren, Shijian Li. ScudCORE: A Context-driven Reasoning Engine. Journal of Electornics. 2008. In press.
[18] T. Kawamura, J.-A. De Blasio, T. Hasegawa, M. Paolucci and K. Sycara, Preliminary Report of Public Experiment of Semantic Service Matchmaker with UDDI Business Registry, In Proc. the First International Conference on Service-Oriented Computing (ICSOC 2003), Trento, Italy ,Springer, 2003
[19] S.Luke, C.Cioffi-Revilla, L.Panait, K.Sullivan, G.Balan. MASON: A multiagent simulation environment. Simulation, 81:517-527. 2005.