# Dynamic Contents Provision for aPost-it[1]

**Yoosoo Oh*, Minkyung Lee*, Seokmin Jung*, Woontack Woo*, and Sanggook Lee****

*GIST U-VR Lab, Gwangju 500-712, S. Korea,

**SAIT Interaction Lab, Yongin 449-712, S.Korea

*{yoh,mlee,sjung,wwoo}@gist.ac.kr, sglee@samsung.com*

## Abstract

We propose a Dynamic Contents Provision method for aPost-it that allows to augment, and to share realistic media according to a user's context. The proposed system provides dynamic augmentation of information, dynamic provision of realistic media, and natural exchange of information. In augmentation, realistic media are augmented on a marker of aPost-it Object. This is done by exploiting ARToolKit on PDA. For dynamic provision of information, aPost-it Server reconstructs information by exploiting contexts of user and object. And exchange of information is based on the context generation and reconstruction in aPsot-it Server. Users can share information through their aPost-it Clients. Thus, the proposed system dynamically provides realistic media which are essential and appropriate to users. Therefore, the proposed system can be considered as an effective way of information representation in ubiquitous computing environments.

**Key words**: Context-aware, Ubiquitous Computing, Information augmentation & Sharing

## 1. Introduction

In Ubiquitous Computing environment, users can interact with computing resources anywhere at any time. In addition, users can utilize intelligent services based on personalized information that is created or extracted according to user context, such as identification, location, time, intention and emotion.

KARMA [1], NaviCam [2], CyberGuide [3], Stick-e Note [4], and ColorCode [5] augment information by searching information mapped to each marker. These previous works provide all users with the same media contents regardless of users' tastes. In addition, it is difficult for users to add, delete, or modify information. The cPost-it [6] provides an easy modification of information by group members. However, it just provides simple two-dimensional information. Therefore, it is not easy to provide information dynamically, and to give immersion to each user [6].

The existing researches have limitations in terms of manipulation and effective representation of information. Thus, we propose a Dynamic Contents Provision method for aPost-it which augments and shares realistic media according to user context.

The proposed system provides dynamic augmentation of information, dynamic provision of realistic media, and natural exchange of information. The dynamic augmentation is a feature that naturally presents information to a user after shift and modification of realistic media. The proposed system obtains the location of information based on context. Thus, the modified information is immediately augmented. With dynamic provision of realistic media, the proposed system offers personalized services which interpret the situation by exploiting context of user and object. In addition, the information is dynamically reconstructed according to the context of each user. The proposed system provides bidirectional manipulation and exchange of information based on context generation and reconstruction. The use of context makes information exchange easily. Therefore, the system dynamically provides contents to users.

## 2. aPost-it : Context-based Information Augmentation & Sharing System

The proposed aPost-it consists of (i) aPost-it Object, (ii) aPost-it Client, and (iii) aPost-it Server. First, aPost-it Object contains information related to the object. It augments information into user's environment. Second, aPost-it Client displays the augmented media according to user's context. Finally, aPost-it Server is a management medium which dynamically offers user's context and information on the selected object. The augmented information is supplied as a personalized service based on user's context. Also, information can be shared by inferring user's intention based on his context in aPost-it Server.

Figure 1 shows a conceptual operation of aPost-it. The proposed aPost-it augments relevant information on aPost-it Object by exploiting context in forms of 5W1H (Who, What, Where, When, Why, and How). Also, the proposed system dynamically offers realistic media

[1] This work is supported by SAIT in Korea.

contents which are based on user's taste. The proposed system is based on ubiSensor and ubiService of ubi-UCAM, which is a context-based application service model for providing context-based services [7]. Table 1 shows an example of context usage in aPost-it.

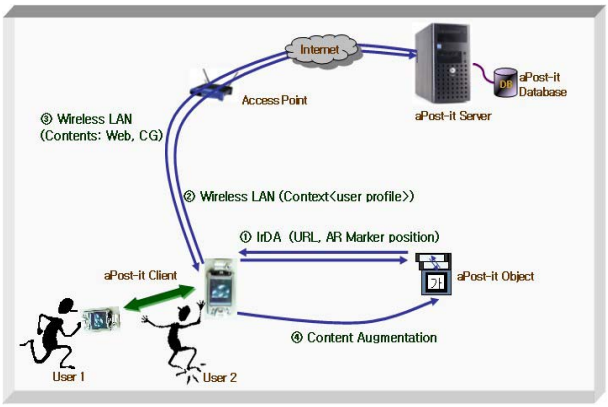


Fig. 1 The conceptual diagram of aPost-it

Table 1: An example of context usage

| 5W1H | Definition | Example |
|---|---|---|
| Who | User Profile | Y.Oh, ubiAgent |
| What | Object Information or Contents | obj_Pavv, obj_Album1 or Album01.avi |
| Where | Location | '-' |
| When | Time | 200405040830 |
| How | Object Function | AR, Text, Web |
| Why | Intention / Behavior | Copy/Paste/Delete |

The aPost-it Client automatically generates context according to user's information, and creates action and different menu according to each object. Since aPost-it Client dynamically acquires object information, relevant information can be augmented on relative position obtained by recognizing an AR marker after the realistic media are moved or modified. Also, we augment realistic media that a user needs at a certain place by considering a user's environmental information. The aPost-it Server manages a user's information and dynamically offers realistic media. In other words, aPost-it Server reconstitutes personalized information according to each user's history, and updates it when the user modifies his information. Moreover, it can dynamically offer realistic media by exploiting context, such as identification, object, location, time. In addition, aPost-it can exchange information between aPost-it Clients (or aPost-it Objects) by adopting a user's context.

### 2.1. Realistic media Augmentation

Realistic media, which are obtained from aPost-it Server, are augmented on the marker of aPost-it Object. This is done by exploiting ARToolKit on PDA [8]. Most ARToolKit-based systems only augment information mapped to each marker. Thus, it is hard to modify information or to change the location where information is stored. The proposed system, however, dynamically obtains the location where information is stored. AR markers only give position to augment realistic media. The modified information or stored location is immediately reflected. Thus, the inadequacy of existing AR systems can be compensated. The concept is shown in Figure 2.

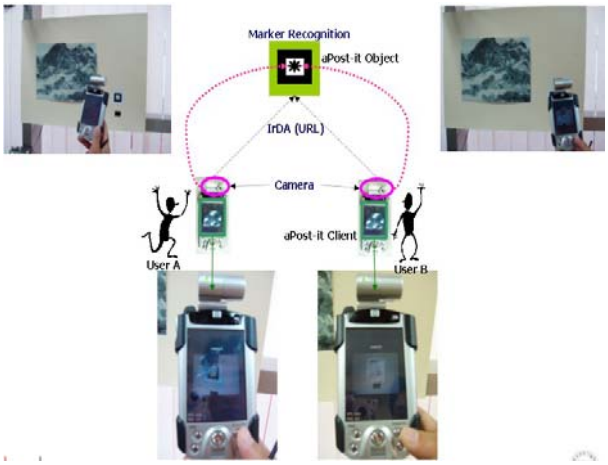


Fig. 2 Personalized Information Retrieval

### 2.2. Dynamic Contents Provision

The aPost-it Server provides realistic media dynamically. It analyzes context input from aPost-it Client, and handles database query. In other words, aPost-it Server analyzes individual context, and augments appropriate information according to the situation. For instance, if a user wants to get information of aPost-it Object, aPost-it Server interprets user context, such as the information use time (When), action (How), and intention (Why). Then, the server selects the most suitable information dynamically. Figure 3 shows the structure of implemented aPost-it Server. Context Integrator collects contexts received from aPost-it Client. Also, it analyzes individual elements of the pushed 5W1H context, and fuses them. Thus, realistic media are dynamically provided according to a user's context. Additionally, Context Manager refers to context condition of Interpreter. It evaluates whether aPost-it Client should augment realistic media or not. Service Provider conveys realistic media to aPost-it Client by searching contents in database.

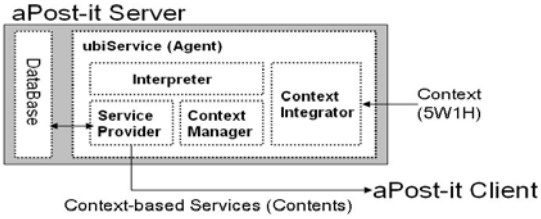


Fig. 3 The Architecture of aPost-it server

### 2.3. Context-based Information Sharing

Realistic media acquired from an aPost-it Object are different for each user. If users, having the same interests, want to share realistic media with each other, the exchange of information is essential. For this purpose, the channel formation of context information is required to share the same content between users. It, however, may raise privacy concerns. Thus, aPost-it Client offers sharing function, by reflecting the user's intention, with intuitive control mechanism. Also, information can be shared by inferring a user's intention based on his context in aPost-it Server. User context is dynamically reconstructed in aPost-it Server. Therefore, the proposed system provides exchange of information based on the context generation and reconstruction.

## 3. Implementation

The aPost-it Object consists of real object, AR marker, and ubiSensor. In order to be combined with an object that exists in a user's environment, aPost-it Object is attached to a real object, such as TV or projector, of smart space. AR marker defines location coordinates in which realistic media will be augmented. In other words, it supplies only location information to augment multimedia information, e.g. text, graphics model, movie file, etc. ubiSensor senses a signal through IrDA communication, and transfers aPost-it Server's URL information.

The aPost-it Client consists of interface, camera module, and ubiSensor. Interface is implemented on HP iPAQ5450 by using EVC3.0, which is based on PocketPC2002. The PDA supports IrDA communication and Wireless LAN 802.11b communication. Camera module augments realistic media on a AR marker by using Hanbit IT's SD type camera (HVC 480) and ARToolkit. ubiSensor creates 5W1H context by integrating URL information with user's profile information. This context is transferred to aPost-it Server through Wireless LAN.

The aPost-it Server consists of database and ubiService. Database manages various kinds of digital contents, specific objects, shared information, and user context. It is implemented by using MS-SQL Server2000. ubiService consists of Context Integrator, Context Manager, Interpreter, and Service Provider [7]. The ubiService efficiently manages context according to each user. Also, it dynamically provides various kinds of contents to each user by handling database query based on the input context. It records user's history, and provides new information which is presented to the user according to the situation. That is, the same aPost-it Object will have different information in different situations.

## 4. Experimental Results

Every experiment is performed in mobile environment (PDA, Wireless LAN, and Internet explorer).

### 4.1. Experiment Ⅰ

Table 2 shows experimental results of dynamic contents provision. If aPost-it Client generates 5W1H context, a user can utilize different kinds of (m x n) contents (m: No. of functions, n: No. of uses per time division).

Table 2: Results of Dynamic Contents Provision

| | With Context | Without Context |
|---|---|---|
| Contents Providing | automatic | manual |
| No. of input procedure | only 1 | more than 3 |
| No. of the provided content | 6 | 3 |

We set up the experiment with 'm=3' and 'n=2'. We experimented for minimum input procedure with 'IR contact', 'menu selection', and 'setting or saving'. 'm' denotes the number of functions, such as AR, Text, and Web menu. 'n' denotes the number of uses per time division, such as a.m. and p.m. We observed that 'without context', the number of provided contents was the same as 'm=3', because the 'when' context was not used. From the result of table 2, it is evident that the method of 'with context' is more efficient.

### 4.2. Experiment Ⅱ

We also experimented with an arrangement of personalized information when information is modified. We selected three people as volunteers. The user modified information according to his taste when he approaches an object for the first time. We evaluated how much the modified information helped him when he approached the object for the second time. The menu was the same as Experiment Ⅰ, and consisted of AR, Text, and Web. Each user selected the preferred menu among the menus for augmenting dynamic information. Information access frequency represents the number of trials, when each user accesses any of three menus. It was measured for 3 minutes when the user used 3 objects. Information access rate was calculated considering the preferred menu and the information access frequency. The formula is shown in (1).

$$R = \frac{b}{a}$$

R: Information Access rate

a : Sum of the access frequency

( = m1 + m2 + m3 ) (1)

b: Access frequency of preference menu

m1: Access frequency of 'AR' menu

m2: Access frequency of 'Text' menu

m3: Access frequency of 'Web' menu

Table 3: Results of Information Access rate

| | The preferred menu | Information Access Frequency(a) | Information Access Rate(R) |
|---|---|---|---|
| User A | AR | 7(=5+1+1) | 0.72(=5/7) |
| User B | Text | 6(=2+4+0) | 0.67(=4/6) |
| User C | Web | 5(=1+1+3) | 0.6(=3/5) |

Table 3 represents the experimental results of Information Access rate (R). From the results, we can infer that information access rate is higher in the preferred menu. In other words, it is observed that each user frequently uses the preferred menu itself. Therefore, each user can conveniently use the menu when the information access rate is high (the preferred menu). In conclusion, we could obtain results for which each user can use the information easily and quickly, by modifying the information.

### 4.3. Experiment Ⅲ

At last, we set up the experiment so that dynamic reconstruction of information was performed according to each user context in aPost-it Server. We divided the users into two groups, each including five persons. 'Group A' includes the service users who are offered dynamic reconstruction of information. We experimented with 'Group A' by using the proposed system, when they approach a specific object. 'Group B' are the service users who are not offered dynamic reconstruction of information. We experimented with 'Group B' by executing Internet explorer on a PDA. Experiment evaluation consists of questionnaire (qualitative analysis) about the degree of satisfaction, and quantitative analysis including information usage and retrieval times. Information usage time is measured when a user accesses information about aPost-it Object once. Information retrieval time-1 is measured when a user searches the given content. Finally, information retrieval time-2 is measured when a user repeats the work of information retrieval time-1 after modifying the information at first. Table 4 shows the experimental results of dynamic content reconstruction.

Table 4: Results of Dynamic Content Reconstruction

| | Information Usage Time | Information Retrieval Time-1 | Information Retrieval Time-2 | Degree of satisfaction |
|---|---|---|---|---|
| Group A | 27 sec | 5.5 sec | 4.3 sec | 85 % |
| Group B | 37 sec | 22.9 sec | 17.4 sec | 65 % |

As experimental results, we observed that 'Group A' was more efficient in terms of both information retrieval time-1 and information retrieval time-2. 'Group B' took more time because general user must modify information again during second information retrieval time. Therefore, we can establish the fact that 'Group A' saves time when they search or retrieve information. By comparing degree of satisfaction, we can infer that 'Group A' is satisfied by efficient role of the reconstruction function based on context.

## 5. Conclusions & Future Works

In this paper, we propose a Dynamic Contents Provision method for aPost-it that allows to augment, and to share realistic media related to an object according to user context. When portable or wearable computers are widely adopted in the near future, the proposed system will be widely used in various application services using personalized information devices. Therefore, as a future work, we will consider intelligent agent which manages user context in order to efficiently express and share information.

## References


1. S.Feiner, B.MacIntyre, and D.Seligmann, "Knowledge-based augmented reality," *Communication of the ACM*, Vol.36, No. 7, pp.52-62, 1993.

2. J.Rekimoto, "NaviCam:A Magnifying Glass Approach to Augmented Reality," *MIT Presence*, Vol.6, No. 4, pp. 399-412, 1997.

3. S.Long, et al. "Rapid Prototyping of Mobile Context-aware Applications: The Cyberguide Case Study," *Proc.2nd ACM International Conference on Mobile Computing and Networking (MobiCom96)*, pp. 97-107, 1996.

4. J.Pascoe, "The Stick-e Note Architecture: Extending the Interface Beyond the User," *International Conference on Intelligent User Interfaces*, pp. 261-264, 1997.

5. ColorZip, http://www.colorzip.co.kr

6. S.Jang, S.Lee, and W.Woo, "cPost-it: Context-based Information Sharing System," *LNCS(MIPS03)*, Vol.2899, pp. 352-363, 2003

7. S.Jang and W.Woo, "ubi-UCAM: A Unified Context-Aware Application Model for ubiHome," *Fourth International and Interdisciplinary Conference on Modeling and Using Context (Context'03)*, pp. 178-189, 2003.

8. D.Wagner and D.Schmalstieg, "ARToolKit on the PocketPC Platform," TR-188-2-2003-23, 2003.