User-centric Integration of Contexts for A Unified Context-aware Application Model

Yoosoo Oh, Sangho Lee, and Woontack Woo

***Abstract*— Context-aware application models can provide personalized services to users through user-centric integration of contexts. Recently, several research activities on context integration have been reported. However, the existing research activities don't consider much how contexts are integrated in a unified way. In this paper, we propose a unified method of user-centric integration of contexts for context-aware applications. The proposed method enables to extract the meaningful contexts based on each fusion procedure of 5W1H contexts. It integrates the formatted contexts through the user-centric classification. Also, it makes a decision by inferring user's explicit intention based on the integrated context.**

***Index Terms*— Context-aware, Context Fusion, Context Inference, User-centric integration**

I. INTRODUCTION

In order to create a meaningful context from heterogeneous sensors, it is efficient for context-aware application models to consider the integration method based on the characteristics of each context input. Such method can produce good results which show the proper services in a given situation. Additionally, it enables to provide personalized services to multiple users by integrating the inputted contexts for each user by exploiting user's profile.

Recently, several research activities on context integration have been reported. Context aggregator in Context Toolkit [1], Sensor Data Fusion method [2][3], Static/dynamic Context Integration [4], and Context Integrator in ubi-UCAM [5] are some of them. Context aggregator, which aggregates multiple pieces of context, is about a particular entity (person, place, or object) [1]. Sensor fusion method, used with dempster-shafer theory, can incorporate the quality of sensors and make decision [2]. Static or dynamic integration can describe the entities that are responsible for collection and production of context information [4]. The ubi-UCAM integrated the contexts obtained from sensors periodically [5].

Manuscript received Aug. 14th, 2005. This work was supported by Samsung Electronics Co., Ltd., in S.Korea.

Yoosoo Oh is with GIST U-VR Lab., Gwangju, 500-712 S.Korea (e-mail: yoh@ gist.ac.kr).

Sangho Lee is with Samsung Electronics Co., Ltd., Seoul, 135-524 S.Korea (e-mail: lsh7210@samsung.com).

Woontack Woo is with GIST U-VR Lab., Gwangju, 500-712 S.Korea (corresponding author to provide phone: +82-62-970-2226; fax: +82-62-970-2249; e-mail: wwoo@ gist.ac.kr).

However, the existing research activities do not consider much how contexts are integrated in a unified way. Thus, we are concerned with the following issues. First, context fusion should be specified in a unified way. Second, the proper fusion method according to the characteristics of contexts should be adapted. Finally, the fusion mechanism, which can extract user's intention, should be developed to provide personalized services suitable for users.

Therefore, we propose a unified method of user-centric integration of contexts for context-aware applications in ubiquitous computing environments. User-centric integration of contexts is classifying and integrating the inputted 5W1H contexts according to each user. 5W1H contexts are contexts which describe the situation as a form of Who, What, Where, When, How, and Why context. 5W1H context representation simplifies to extract characteristics of each user for user-centric integration.

The proposed method enables to extract the meaningful contexts based on each fusion procedure of 5W1H contexts. It integrates the formatted contexts through the user-centric classification. Also, it makes a decision by inferring user's explicit intention based on the integrated context. In addition, the proposed method can give following advantages. It can be helpful to present a way to integrate contexts from any heterogeneous sensors. It can extract semantics from contexts by context integration. Accordingly, it can provide intelligent services according to user's explicit intention.

This paper is organized as follows: The Chapter 2 explains Context Integrator in ubi-UCAM 2.0. The Chapter 3 describes 5W1H Context Fusion in detail. The Chapter 4 explains Context Inference. The experimental setup and experiments are explained in Chapter 5. Finally, conclusion and future works are presented in Chapter 6.

II. CONTEXT INTEGRATOR IN UBI-UCAM 2.0

The ubi-UCAM 2.0 is a unified context-aware application model for ubiquitous computing environments [6]. It consists of ubiSensor and ubiService. The ubiSensor consists of physical sensor, feature extraction module, preliminary context generator, and self configuration manager. The ubiService consists of Self Configuration Manager, Context Integrator, Context Manager, Interpreter, and Service Provider. Fig. 1 shows the architecture of ubi-UCAM 2.0. PC (preliminary context), IC (integrated context), and the others are defined as context type [6].

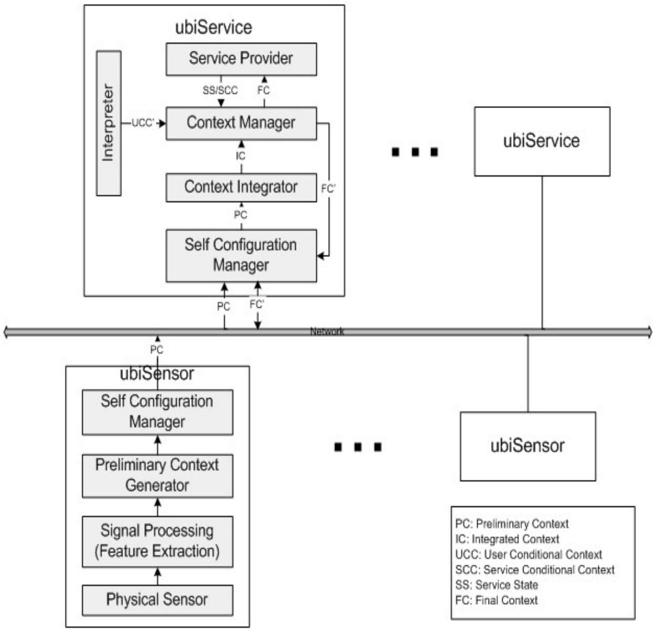


Fig. 1. The architecture of ubi-UCAM 2.0.

The unified context expressed with 5W1H ensures independence between sensors and services. It also has an advantage of being re-used by other services. In addition, it can reduce additional management to form the context according to an individual service.

A. *Context Processing in ubi-UCAM 2.0*

The ubiSensor plays a role in forming preliminary context (PC) by perceiving a change about a user and his environment. The ubiSensor transfers part or all of 5W1H context into ubiService according to a sensor type. Preliminary Context Generator plays a role in converting feature extracted from a physical sensor into the formatted 5W1H context. The Self Configuration Manager of ubiSensor multicasts PC to ubiService which are dynamically connected to ubiSensor.

The ubiService plays a role in providing the application service that a user wants by recognizing contexts. Self Configuration Manager of ubiService receives contexts through forming a multicasting group dynamically. It supports ad-hoc networking which all ubiSensors and ubiServices can share context in the same active range through forming a multicasting group. Context Integrator collects Preliminary Contexts (PCs) in a periodic interval from various kinds of ubiSensors in the same active range with ubiService, and classifies the context as each item of 5W1H particularly. Context Manager takes charge of searching the condition of context which corresponds to integrated context (IC) in Hash table, and executes the appropriate service. Service Provider manages the implemented code of service module that ubiService provides, and operates service directly after receiving necessary information about service execution. Interpreter provides the environment where a user can designate context condition for service execution.

B. *Context Integrator*

Context Integrator creates an IC from various kinds of context input, which can be PCs from sensors or final contexts (FCs) from other services. Context integration reconstructs a meaningful integrated context. It is a kind of decision making process by user-centric integration methods. User-centric integration is performed by each user's identity. It is helpful to provide personalized services based on characteristics of each user.

In order to build Context Integrator, the following constraints should be considered:

1) To be context input, context type should be specified (PC or FC).
2) To make user-centric integration, "Who" context must be decided at least once in 5W1H fusion.
3) To infer "Why" context (intention or emotion), the integrated 4W1H context should be decided in advance.

Context Integrator collects preliminary contexts periodically from various kinds of ubiSensor which is placed in same active area with ubiService. Then, it classifies the contexts as each element of 5W1H. It creates integrated context by applying the proper fusion method that reflects characteristics of each element. Fig. 2 shows the architecture of Context Integrator.

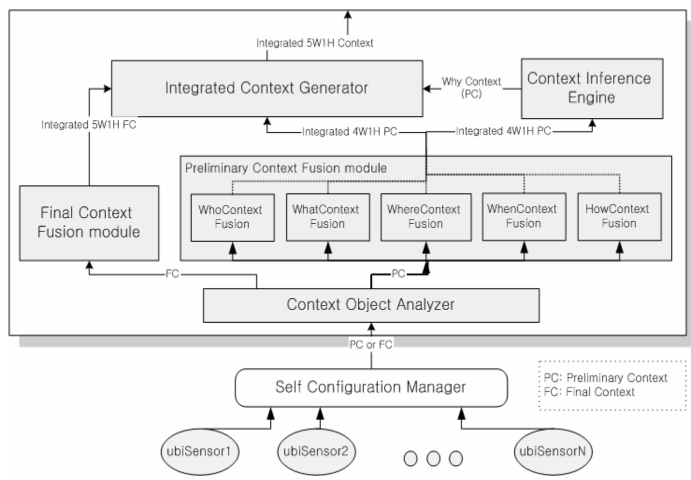


Fig. 2. The architecture of Context Integrator.

Context Integrator is composed of Context Object Analyzer, Preliminary Context Fusion module, Final Context Fusion module, Context Inference Engine, and Integrated Context Generator. Context Object Analyzer collects contexts in user-centered view, and classifies the contexts as PCs and FCs. Preliminary Context Fusion module integrates the inputted PCs as a integrated 4W1H according to characteristics of each sub-context of 4W1H (Who, What, Where, When, and How). It is divided into 5 fusion modules, as shown in Fig. 2. Final Context Fusion module simply integrates the inputted FCs according to "Who" context. Context Inference Engine plays a role in inferring "Why" context by using the result of Context Fusion module. It infers user's explicit intention by the integrated 4W1H PC. Finally, Integrated Context Generator makes an IC which contains information, such as user's identity, location, activities, behavior, patterns, and explicit intention.

III. 5W1H Context Fusion

5W1H context fusion is Who, What, Where, When, How, and Why context fusion. Each context fusion has the specific fusion method according to its sub-contexts. Sub-contexts express characteristics of each 5W1H context in more detail. 5W1H context fusion is a process to reduce uncertainty of each sub-context. The followings are the description about the fusion method based on characteristics of each sub-context.

A. Who Context Fusion

The “Who” context has sub-context, such as identity, priority, sex, weight, and height. Fig. 3 explains the “Who” context fusion. Identity can be decided by using a weighted voting method which elects a leader among votes with weights (in Voter, Fig. 3). The identity can be made, even though it doesn't contain any information from ubiSensor. That means the identity can be verified by deciding an uncertain context. Preliminary Context Fusion module can build identity information to an unknown user by comparing the number of persons in the environment with the number of the inputted identity. The remained sub-contexts are updated by the latest information (in Modifier, Fig. 3).

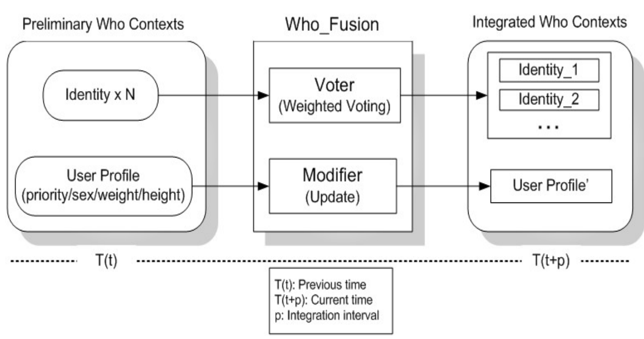


Fig. 3. The Who Context Fusion process.

B. What Context Fusion

The “What” context of ubiSensor consists of sensor ID, sensor type, and accuracy. Sensor ID and sensor type express the unique information that each sensor has, and they can describe characteristics of the sensors. For example, if they are filled with information about location or tracking sensors, Context Integrator can know that the delivered PC includes position information. In addition, accuracy shows reliability of PC generated from a sensor, and this can be used as basic information to integrate “How” or “Why” context. Also, accuracy can be dynamically adjusted according to the situation of context input. Fig. 4 explains the “What” context fusion. Feature Extractor gets the characteristics about a sensor.

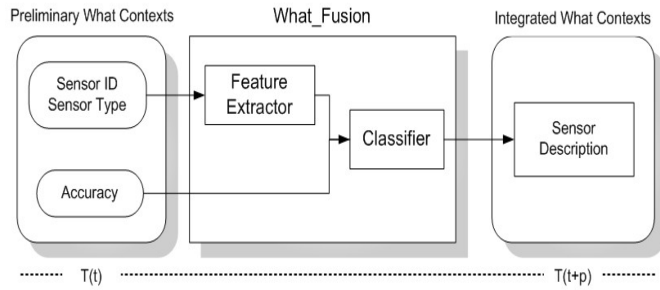


Fig. 4. The What Context Fusion process.

C. Where Context Fusion

The “Where” context has sub-context, such as absolute location and symbolic location. The fusion of “Where” context is used to analyze behavior patterns of a user by using position information expressed with coordinates or symbols. Fig. 5 describes the “Where” context fusion. Context Integrator can know that the user is passing in front of a specific device, by monitoring a change of location information during the given duration (in Location Tracker, Fig. 5). Absolute location can give a clue for user’s attention by using user’s trace, orientation, and the adjacent object’s area (in Location Tracker & Location Calculator, Fig. 5). For example, by observing a change of absolute location, Context Integrator can know user’s attention is changed from “TV” to “Audio”. That fact can be inferred by having coordinate information of the user and the surrounding objects. Symbolic location is information which is obtained, when a user moves by the side of an object. For instance, it means information such as “A TV is located in front of a sofa, and the sofa is located in the center of a living room” (in Symbol Extractor, Fig. 5).

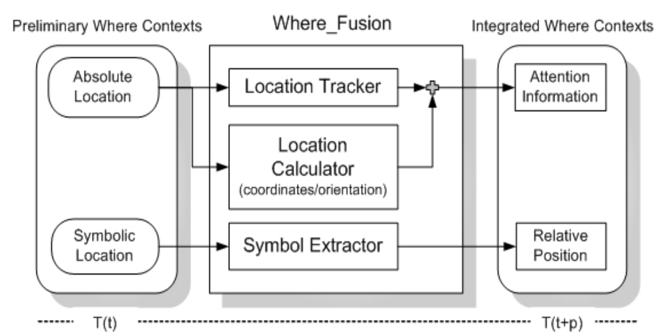


Fig. 5. The Where Context Fusion process.

D. When Context Fusion

The fusion of “When” context decides absolute time and symbolic time. Fig. 6 explains the “When” context fusion. The fusion of “When” context imprints time-stamp on every inputted PC (in Time Stamper, Fig. 6). This fusion obtains the efficient results by flexibly varying time of integration. Also, this fusion plays a role in imprinting time-stamp at the time when IC is generated. Furthermore, it could manage user’s history based on the record of the timestamp (in Context Recorder, Fig. 6).

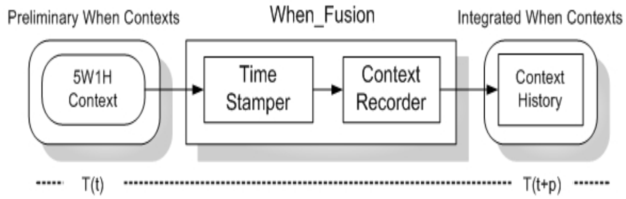


Fig. 6. The When context fusion process.

E. *How Context Fusion*

The fusion of “How” context integrates bio-signal, control information, and others. Fig. 7 explains the “How” context fusion. Bio-signal is detected by bio-sensors attached in human body. PPG (photoplethysmogram) for detecting heart rate, GSR (galvanic skin response) for detecting skin conductance, and SKT (skin temperature) for detecting temperature are examples of them. In case of bio-signal, this fusion filters only keep the proper information by using threshold values, such as mean/variance/power of PPG, GSR, and SKT (in Threshold Measurement, Fig. 7). This fusion integrates sub-contexts of “How” context by selecting dominance among the current input and the previous input (in Voter & Selector, Fig. 7). In case of control, this fusion can extract information which is related to user’s gesture or activity (in Behavior Extractor, Fig. 7).

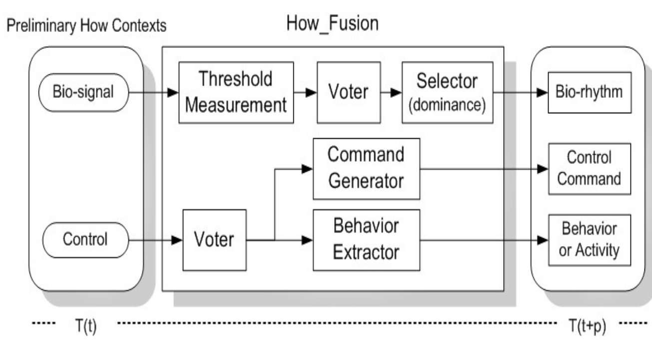


Fig. 7. The How Context Fusion process.

F. *Why Context Fusion*

The fusion of “Why” context integrates sub-contexts, such as attention, intention, and emotion of users. This fusion module is in Context Inference Engine. Fig. 8 explains the “Why” context fusion which contains Context Transition Analyzer and Context Pattern Analyzer. Context Transition Analyzer observes changes of contexts and Context Pattern Analyzer monitors patterns of contexts by comparing 4W1H contexts. By combining the results from Context Transition Analyzer and Context Pattern Analyzer, this fusion module infers higher-level contexts, such as attention, intention, and emotion.

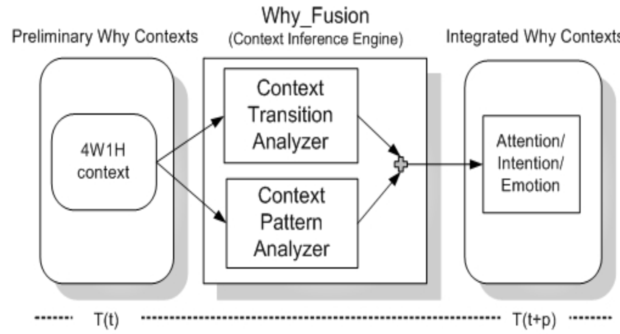


Fig. 8. The Why Context Fusion process.

IV. Context Inference

Context Inference infers uncertain contexts or gets new reasoned contexts. It determines which device or service a user is currently interested in and what his intention may be. It is used for generating Why context. It extracts user’s attention, intention, or emotion by observing a change of sub-contexts. Context Inference is based on context transition [2] and complex fusion. Context transition is a method which infers IC by observing a change of context. Complex fusion uses two more fusion methods along with 4W1H (Who, What, Where, When and How) context fusion.

A. *Context Transition*

Context transition can get a new reasoned context by observing a change of the other contexts. For instance, there are location-change, proximity-change and function-time change. A change of “Where”/“When”/“How” context can extract user’s action. It means a change of a region where the user moves. In addition, it obtains the reasoned information such as a user currently walks or runs by calculating a speed. Moreover, a change in “What”/“Where”/“When” context can infer user’s attention. It shows that the device which adjoins the user is continuously changing. This means a change of available devices in a present place which the user exists at present time. Context Inference Engine can infer what device or service currently a user has an interest in. A change of absolute time of the “When” context expresses a change of the expected activity time. It infers whether it is time to have lunch or to work. Namely, it is based on user’s profile. If this inference extends, it can deduct information of history, schedule, and expectation of users.

B. *Complex Fusion for Identity*

Complex Fusion is the combination of two additional fusion methods. Complex Fusion for identity can be performed by “Who” and “Where” context. Especially, Symbolic Location in “Where” context is important to extract identity. The Symbolic Location is expressed as Object Name, Object Region, Sensor ID, Sensor Region and User’s Orientation (Radian). In smart home, sensors are embedded in an object. Fig. 9 explains how to get Symbolic Location from a couch. Three couch sensors

[7] on a couch object are registered in a PDA. When a user approaches to the couch, the user can get Symbolic Location information which has a region of each sensor registered in the couch. Context Integrator attaches the user's identity from the user's PDA to the couch sensor as PC input. Thus, Context Integrator can infer the user's identity on a couch sensor, even though the couch sensor can't create "Who" context.

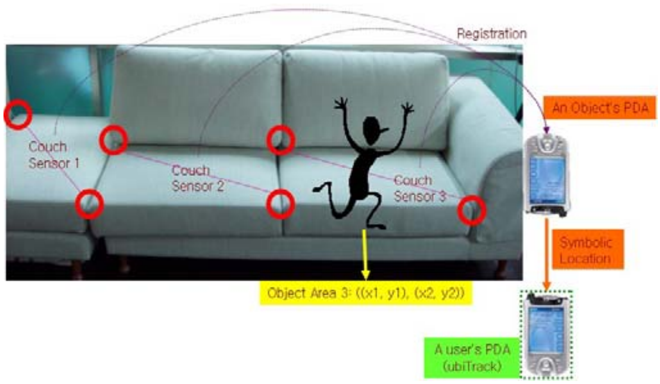
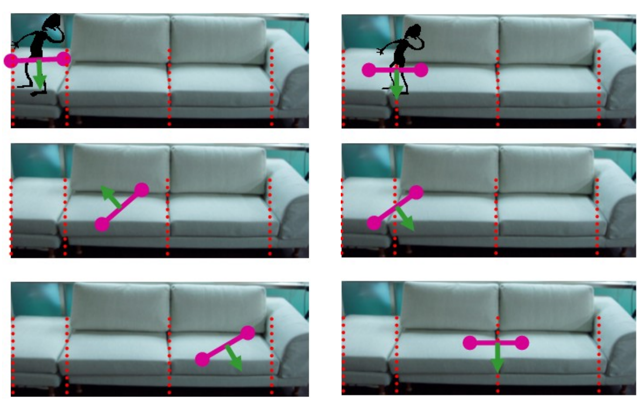


Fig. 9. Symbolic Location acquirement from a couch..

C. *Behavior Inference*

Our Context Integrator infers user's behavior or gesture. At this point, previous contexts are an important clue. Thus, context history is used to evaluate user's behavior. Fig. 11 represents an example of user's posture on a couch. The user's posture on a couch contains a wide variety. Fig. 10 just shows two cases. First case is when a user sits in a seat (sensor) on a couch (Fig. 10(a)). Second case is when a user sits in two seats (sensors) on a couch (Fig. 10(b)).



(a) (b)

Fig. 10. An example of user's posture on a couch.

Our Context Integrator can get coordinates on both shoulders of a user. Left shoulder has the coordinate (x1, y1) and right shoulder has the coordinate (x2, y2). By using those coordinates, user's orientation can be calculated. Both coordinates and the orientation are used to infer user's posture. If (x1, y1) and (x2, y2) are included in a sensor region, Context Integrator can infer that a user sits in a seat in the obtained direction. If (x1, y1) and (x2, y2) are included in the obtained different region, Context Integrator infers that the user sits in two more seats in the direction. Thus, this inference is used to extract user' attention. Additionally, our Context Integrator supports to extract users' postures on a couch, like Fig. 11. In Fig. 11, three users sit in three seats. The light service is automatically triggered as the proper level.

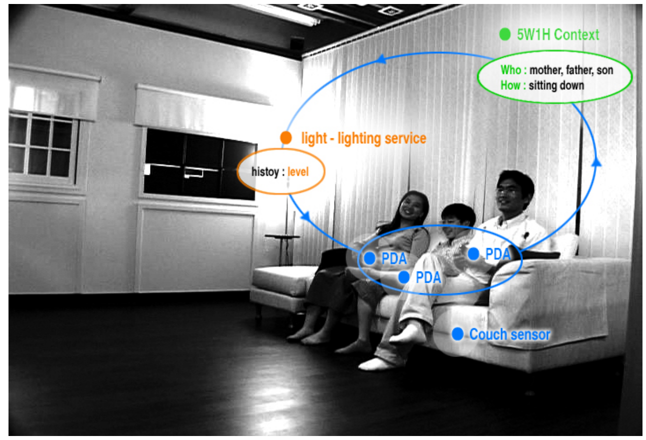


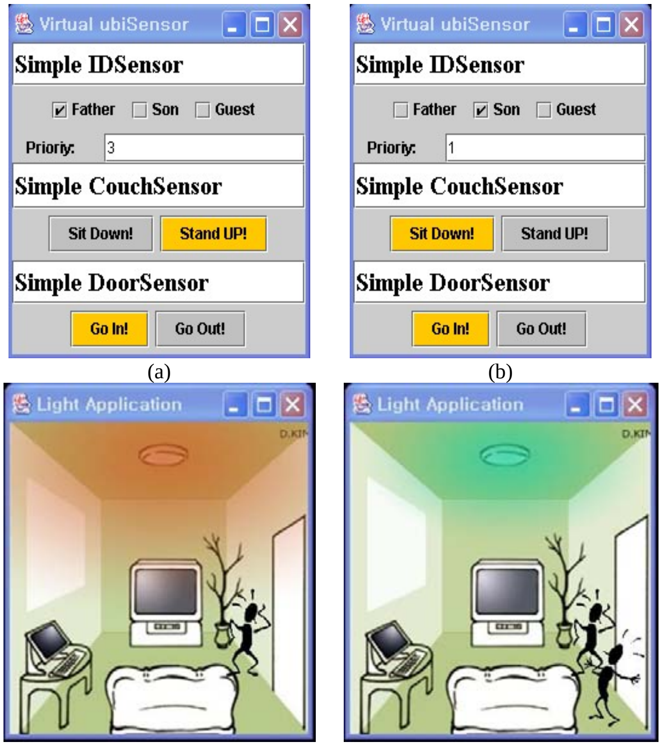
Fig. 11. Three users' postures on a couch. Three users sit in three seats.

V. Experimental Setup and Experiments

To verify our method, we simulated situations where Context Integrator integrates contexts from various kinds of sensor, and makes a decision. Thus, we built the simulation environment and the smart home test-bed, ubiHome [7][8]. Fig. 12 shows ubiHome test-bed. Context Integrator was implemented with J2SDK 1.4, in order to support various service platforms.

Fig. 12. ubiHome test-bed.

First, we established the simulation environment, which is composed of Virtual Light Application and Virtual ubiSensor. Fig. 13 shows the implemented simulation environment. Virtual ubiSensor consists of "Simple IDSensor", "Simple CouchSensor" and "Simple DoorSensor". "Simple IDSensor" decides on identity and priority of the "Who" context. "Simple CouchSensor" detects user's behavior which consists of sitting down and standing up. Finally, "Simple DoorSensor" perceives entering and exiting of virtual ubiHome environment. Virtual Light Application shows how virtual lamp is controlled when a user enters the virtual ubiHome.



(c) (d)

Fig. 13. Virtual Sensor & Virtual Light Service
(a) Virtual Sensor Information for a son
(b) Virtual Sensor Information for a father
(c) Virtual Light Service Status for a son
(d) Virtual Light Service Status for a father and a son..

In the simulation, we tested that electric lamp level becomes 5 when a son entered in virtual ubiHome, and then electric lamp level automatically changed to 3 when a father entered by integrating contexts obtained from a father and a son. This simulation shows that Context Integrator can efficiently create the integrated context from sensors when multi-user tries to use the same service simultaneously. For example, Virtual Sensor can create PCs from two users, a father and a son. At this point, Context Integrator integrates those PCs and infers users' intention as in Table 1. As shown in Table 1, Context Integrator infers that a father wants to move to other area by observing a father's location and a son wanting to watch a service, such as TV, Movie, etc. As the result, Context Integrator in Virtual Lighting Service decides to provide a light service with a green color and a level-3 (max. level: 5). It is the result for a son on a couch based on his preference. In real situation (social protocols), this result can be changed by discussion between two users. However, Context Integrator can flexibly decide IC by integrating command context after their discussion.

TABLE I
CONTEXT INTEGRATION IN VIRTUAL LIGHT SERVICE

Who	What	Where	When	How	Why
a father	Lighting Service	On a couch	In the morning	Standing Up	Intention (To move)
a son	Lighting Service	On a couch	In the morning	Sitting Down	Intention (To watch)

As the experiment, we simulated it in a real test-bed, ubiHome. Many sensors and context-based services have been embedded in ubiHome. The context is created by various kinds of sensors in 5W1H form. To integrate and manage user-centric contexts in an application, we applied ubi-UCAM 2.0. As shown in Fig. 14, various kinds of sensors such as ubiKey [9], Couch sensor, IR sensor, USB camera, web camera, PDA [7], space sensor [10], ubiFloor [11], ubiTrack [12] RF tag etc. are deployed in ubiHome, the smart home test-bed at GIST U-VR Lab.

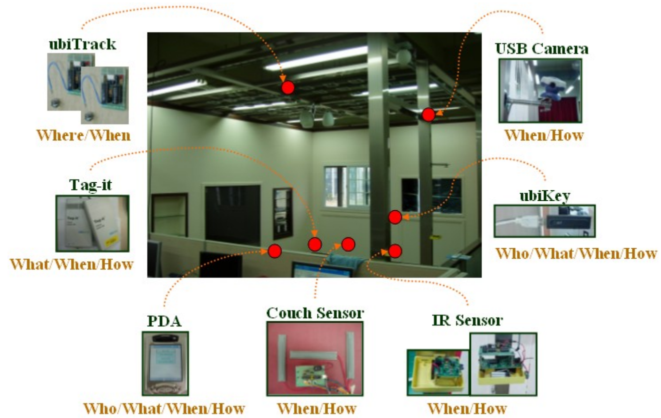


Fig. 14. ubiHome test-bed with various sensors.

For this experiment, we implemented a TV application (ubiTV) in ubiHome. The ubiTV is a context-based TV application for multiple users in smart home environments [6][8]. It is an efficient multi-media service to increase communications between members of a family. It is implemented to interact with various sensors and services in ubiHome. It provides media service, such as music and movie service as well as traditional TV service. The ubiTrack [12] which tracks user's location, and CouchSensor [7] which detects user's action, were utilized together as ubiSensor with this service.

First, we experimented how our Context Integrator performed by user-centric integration. Table 2 shows the performance of Context Integrator. *Integration Interval* means the time period that Context Integrator decides IC. *CPU occupying ratio* represents the usage of CPU when Context Integrator integrates contexts. *User-centric Integration* is a measurement of how "Who" context fusion affects the integration. It is a ratio between the number of the generated IC (G) with user's identity and total number of input (T) in a given interval. Its result expresses user's identity is important because "Who" context fusion classifies the context input by user's identity.

TABLE Ⅱ
THE PERFORMANCE OF CONTEXT INTEGRATOR
(PC ENV.- CPU: PⅢ 800M, RAM: 512M)

Integration Interval	CPU occupying ratio	User-centric Integration (G/T)
0.1 sec	48 %	4/20
0.5 sec	32 %	1/20
3 sec	45 %	1/20

G: the number of the generated IC with user's identity
T: total number of input

Accordingly, we could verify that our Context Integrator can also be performed in user-centric view. Additionally, we could notice our Context Integrator has good performance when *Integration Interval* is 0.5 second. Our Context Integrator can support user-centric services based on user's behavior because it creates a meaningful context by user-centric classification.

Second, we tested how our Context Integrator influenced each service. Table 3 shows comparison results between Context Integrator in ubi-UCAM 1.0 [5] and our Context Integrator. *Service Execution* means a procedure that manipulates a channel or sound volume, including the execution of ubiTV service. *Multi-service support* means simultaneously providing various services to a user. The services are electric lamp service, music service, and movie service (cMP [7]). Finally, *Multi-user support* is the analysis about whether Context Integrator can support multiple users simultaneously.

TABLE Ⅲ
THE COMPARISON ABOUT MULTI-USER/SERVICE SUPPORT

Context Integrator	Service Execution	Multi-service support	Multi-user support
ubi-UCAM's	Good	One service at once	Single user
Ours	Good	Multiple services at once	Multiple users

As the results, the proposed method supports multi-service to a user by context integration and inference at the same time. It is precisely done by Context Integrator integrating contexts, and inferring user's intention. Also, the proposed method supports multi-user by user-centric classification according to each user. Additionally, Context Integrator can make a suitable decision for a user, by considering personal characteristics and the priority as sub-context fusion.

Third, we tested our method by using the ubiTV scenario [6]. The ubiTV scenario is tested among 3 users in ubiHome. It shows the usage of the ubiTV service by exploiting our Context Integrator. It also shows how the ubiTV provides media services to multiple users. Table 4 describes 5W1H contexts in the ubiTV scenario.

TABLE Ⅳ
5W1H CONTEXTS IN UBITV SCENARIO

5W1H Context	Description
Who	a father (age 37), a mother (age 34), a son (age 7)
What	services or contents
Where	Somewhere in a living room (ubiHome)
When	time (in the morning/afternoon/evening, at night), history
How	a resident's gesture, movement, activities, behavior, patterns, etc.
Why	a resident's attention or intention

In the scenario, the ubiTV service executes the proper services that a user wants by obtaining context inputs from various sensors. Moreover, Context Integrator in the ubiTV infers users' intention about display device. In ubiHome, two displays are at right angles to each other. Those are a TV screen and window monitors. Fig. 15 represents users' attention to the tiled display (MRWindow) based on context inference from the users' orientation. The orientation can be calculated in ubiTrack [12]. Therefore, the tiled display (MRWindow) can show the proper information to users.

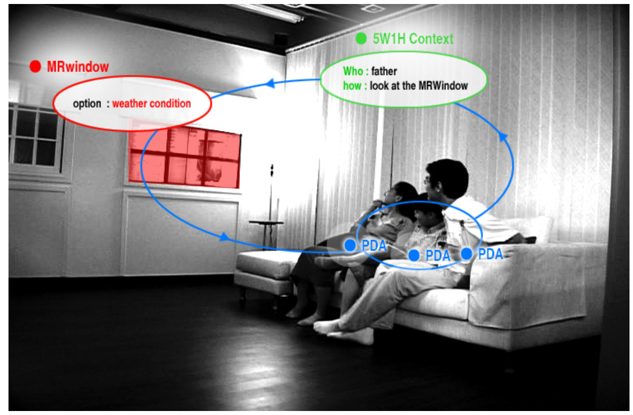


Fig. 15. User's attention on the tiled display (MR Window).

Lastly, we made up questions concerning users' satisfaction about the ubiTV service to see the efficiency of context inference. This questionnaire is performed after users repeatedly using the ubiTV for a quarter of a day in ubiHome. As the result of the degree of satisfaction about 20 volunteers (Fig. 16), we could conclude our Context Integrator gives enough satisfaction to users through the inference about users' behavior for user-centered personalized services.

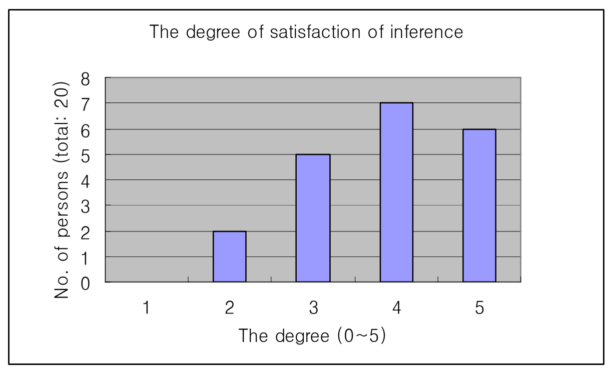


Fig. 16. The degree of satisfaction.

VI. CONCLUSIONS AND FUTURE WORK

In this paper, we propose the User-centric Integration of 5W1H Contexts for ubi-UCAM 2.0. The proposed method can ensure a seamless integration of contexts obtained from various kinds of sensors. Also, it can provide intelligent services in smart home environments. In near future, we will resolve uncertain contexts more accurately and verify the usability of context fusion.

ACKNOWLEDGMENT

We'd like to thank to Dahee Kim (Virtual Simulator Design) and Wonwoo Lee (Technical support for MRWindow).

REFERENCES

[1] D. Salber, A.K. Dey and G.D. Abowd, "The Context Toolkit: Aiding the Development of Context-Aware Applications," *In the Workshop on*

Software Engineering for Wearable and Pervasive Computing (Limerick, Ireland), Jun. 2000.

[2] H. Wu. "Sensor Data Fusion for Context-Aware Computing Using Dempster-Shafer Theory," *PhD thesis, Carnegie Mellon University, Pittsburgh, Pennsylvania 15213*, December, 2003.

[3] Honle, Nicola, et al., "Benefits Of Integrating Meta Data Into A Context Model," *Proceedings of CoMoRea (at PerCom'05)*, March, 2005

[4] Jan Van den Bergh, Karin Coninx, "Towards Integrated Design of Context-Sensitive Interactive Systems," *Proceedings of CoMoRea (at PerCom'05)*, March, 2005.

[5] S.Jang, W.Woo, "ubi-UCAM: A Unified Context-Aware Application Model," *LNAI (Context03)*, Vol.2680, pp. 178-189, 2003.

[6] Y.Oh, C.Shin, S.Jang and W.Woo, "ubi-UCAM 2.0: A Unified Context-aware Application Model for Ubiquitous Computing Environments", *The first Korea/Japan Joint Workshop on Ubiquitous Computing & Networking Systems 2005 (ubiCNS2005)*, 2005.

[7] Y.Oh and W.Woo, "A unified Application Service Model for ubiHome by Exploiting Intelligent Context-Awareness," *UCS04*, pp. 117-122, 2004.

[8] S.Jang and W.Woo, "Introduction of "UbiHome" Testbed," The first Korea/Japan Joint Workshop on Ubiquitous Computing & Networking Systems 2005 (*ubiCNS2005*), 2005.

[9] Y.Oh, S.Jang, W.Woo, "User Authentication and Environment Control using Smart Key," *KSPC 2002*, vol. 15, No. 1, pp. 264, Sep. 2002.

[10] D.Hong and W.Woo, "A Vision-based 3D Space Sensor for Controlling ubiHome Environment," *KHCI2003*, vol. 12, No. 2, pp. 358-363, Feb. 2003.

[11] S. Lee and W. Woo, "Music Player with the ubiFloor," *KHCI2003*, pp. 154-159, Feb. 2003.

[12] S.Jung, W.Woo, "UbiTrack: Infrared-based user Tracking System for indoor environment," *ICAT'04*, 1, paper 1, pp. 181-184, 2004

Yoosoo Oh received his B.S. degree in EE from Kyungpook National University, Daegu, Korea, in 2002 and M.S. degree in Department of Information and Communications (DIC) from Gwangju Institute of Science and Technology (GIST), Gwangju, Korea, in 2003. Now he is a Ph.D. Candidate in U-VR Lab., DIC at GIST since 2003.
Research Interests: Context Integration, Context Inference and Context-aware for Ubiquitous Computing

Sangho Lee received the B.S degree in Electronic & Communication Engineering from Kwangwoon University in 2001. Now he works in SAMSUNG ELECTRONICS since 2001.
Research Interest: Context Awareness, Ubiquitous Computing, eXtensible Home Theater, etc.

Woontack Woo received his B.S. degree in EE from Kyungpook National University, Daegu, Korea, in 1989 and M.S. degree in EE from POSTECH, Pohang, Korea, in 1991. He received his Ph. D. in EE-Systems from University of Southern California, Los Angeles, USA. During 1999-2001, as an invited researcher, he worked for ATR, Kyoto, Japan. In 2001, as an Assistant Professor, he joined Gwangju Institute of Science and Technology (GIST), Gwangju, Korea and now at GIST he is leading U-VR Lab.
Research Interest: 3D computer vision and its applications including attentive AR and mediated reality, HCI, affective sensing and context-aware for Ubiquitous Computing, etc.