IEEE Pervasive Computing, 7(4),
pp. 20-27. (Draft version)

Object-Blog System for Environment-Generated Content

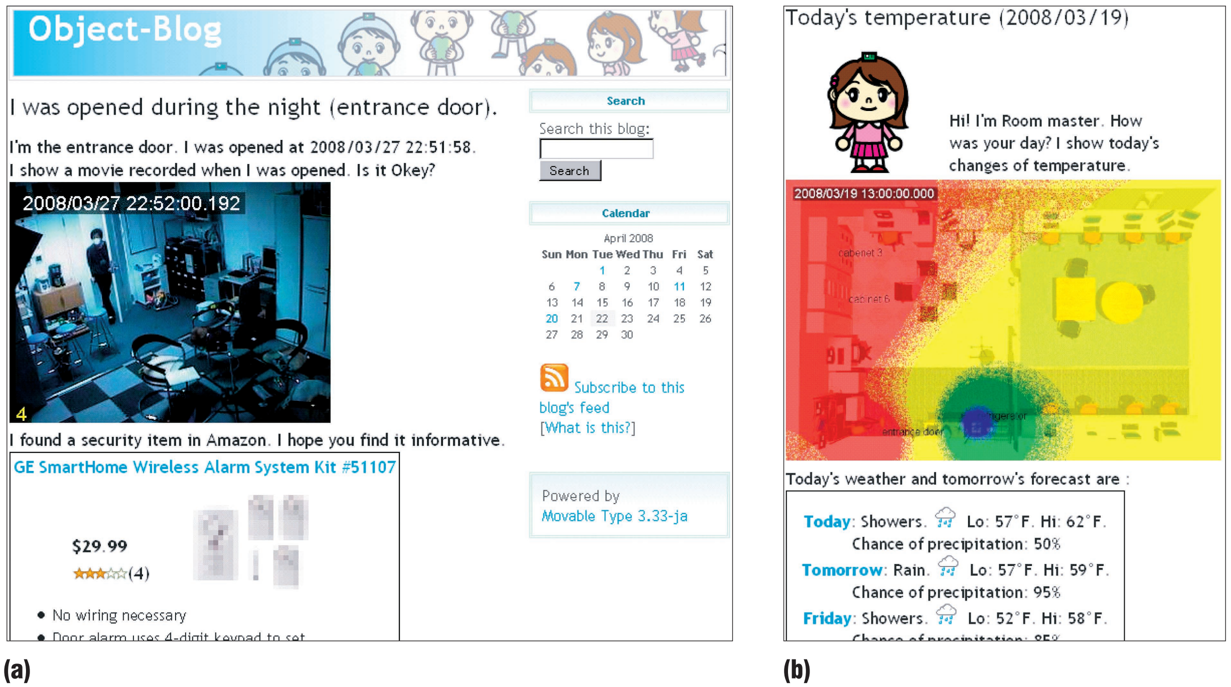
The Object-Blog system automatically converts raw sensor data into environment-generated content, facilitating data searching and browsing.

Takuya Maekawa,
Yutaka Yanagisawa,
Yasue Kishino, Koji Kamei,
Yasushi Sakurai,
and Takeshi Okadome
NTT Communication Science Laboratories

Vannevar Bush's Memex,[1] which stores everyday artifacts and links them for future access, has inspired many research projects. MyLifeBits,[2] for example, stores all types of captured digital media, including photos, videos, and sounds. The applications described by Khai Truong, Gregory Abowd, and Jason Brotherton store and support events such as lectures and meetings by using capture devices (including cameras, microphones, and electronic whiteboards) installed in indoor environments.[3] By attaching tiny ubiquitous sensors to various indoor objects and furniture, we can observe their use and phenomena occurring around them. These sensors include radio frequency identification (RFID) tags, accelerometers, thermometers, cameras, microphones, and object-specific sensors for doors, drawers, and cabinets that detect their opening and shutting.[4] Recent reductions in data storage costs make it possible to store real-world data related to normal daily living observed with ubiquitous sensors. However, Ramesh Jain suggests that many raw logs will be write-only logs; they will be written but never accessed.[5] Jain also recommends processing raw logs into meaningful data events and aggregating the events into a kind of chronicle. We agree with these suggestions. To move from raw logs to chronicles in the ubiquitous-sensor environment, our approach involves automatic content generation from real-world phenomena.

We automatically generate understandable, user-friendly formatted content such as text reports of real-world events and of statistics calculated from sensor data. This environment-generated content (EGC) includes reports and diaries created by users with reference to other EGC or to events detected in the environment. To establish design guidelines for an EGC system in daily-living environments, we first use a questionnaire survey to clarify the kinds of data that users want to record in their daily lives and the characteristics of that data. The survey results indicate that large quantities of EGC would be generated every day. Also, the recorded data can serve many purposes, including aiding memory, improving daily life, and sharing information. To achieve these purposes while dealing with large quantities of EGC, we established six requirements for storing, managing, and providing EGC. We designed the Object-Blog system (www.kecl.ntt.co.jp/csl/sirg/people/maekawa/egc) as an EGC service application that meets these six requirements and incorporates our survey's findings. We assume an indoor environment in which wireless sensor nodes equipped with sensors such as accelerometers and thermometers are attached to objects. In the system, personified objects automatically post entries to a weblog about events those objects experience and their use. To our knowledge, ours is the first

Published by the IEEE CS ■ 1536-1268/08/$25.00 © 2008 IEEE



I was opened during the night (entrance door).

I'm the entrance door. I was opened at 2008/03/27 22:51:58.
I show a movie recorded when I was opened. Is it Okey?

2008/03/27 22:52:00.192

I found a security item in Amazon. I hope you find it informative.

GE SmartHome Wireless Alarm System Kit #51107

$29.99
☆☆☆☆☆(4)

- No wiring necessary
- Door alarm uses 4-digit keypad to set

Search

Search this blog:

Search

Calendar

April 2008

Sun	Mon	Tue	Wed	Thu	Fri	Sat
		1	2	3	4	5
6	7	8	9	10	11	12
13	14	15	16	17	18	19
20	21	22	23	24	25	26
27	28	29	30			

Subscribe to this blog's feed
[What is this?]

Powered by
Movable Type 3.33-ja

Today's temperature (2008/03/19)

Hi! I'm Room master. How was your day? I show today's changes of temperature.

2008/03/19 13:00:00.000

Today's weather and tomorrow's forecast are :

Today: Showers. Lo: 57°F. Hi: 62°F.
Chance of precipitation: 50%
Tomorrow: Rain. Lo: 57°F. Hi: 59°F.
Chance of precipitation: 95%
Friday: Showers. Lo: 52°F. Hi: 58°F.
Chance of precipitation: 85%

Figure 1. Weblog, with entries describing (a) a door and (b) temperature changes during one day.

comprehensive system that ranges from generating Web content about normal daily lives to providing users with methods for accessing this content.

Toward EGC services

Researchers have developed many sensor-data-processing methods to detect activities of daily living (ADLs) such as making tea, vacuuming, and sleeping,[6] and object-specific events such as opening and shutting doors and drawers.[4] We employ these methods of detecting meaningful events to generate content. Our goal is to summarize the requirements for generating, storing, and providing EGC, and to design and implement an EGC service that satisfies these requirements. We try to store everything automatically through ubiquitous sensors and provide the stored sensor data by automatically converting it into content. Also, because our approach generates large quantities of EGC, we provide users with effective EGC access methods.

Many studies have implemented recording and access applications for particular circumstances such as meetings, lectures, academic conferences, battlefields, travel, Web browsing, and online shopping.[2,3] There are also several studies that detect daily activities by employing wearable sensors, including cameras and accelerometers, to make media clips when users perform specific activities.[7] Few studies, however, have dealt with a comprehensive system that ranges from recording various normal daily events by using environment-embedded sensors to providing users with methods for accessing those events. Gamhewage De Silva, Toshihiko Yamasaki, and Kiyoharu Aizawa have implemented a lifelogging system with a GUI that permits easy retrieval of multimedia data such as video, audio, and floor sensor data, all captured from a home-like environment.[8] Users can retrieve video recorded during the occurrence of typical events, such as a change in the house-lighting level. The system isn't a content generation system but a multimedia data provision system designed to be a memory aid. Mik Lamming and Mike Flynn originally implemented systems that generated episodes about human activities.[9] PEPYS generates an episode based on user location data and sends the episode to a user by email at the end of each day.[10] The episode consists of text-formatted summaries of a day's activities. Forget-Me-Not provides daily summaries for mobile devices such as PDAs.[9] Because mobile devices have small screens, Forget-Me-Not provides these summaries as icons. These systems are designed as memory aids. Our lifelog system, on the other hand, provides daily-life management and information sharing, as well as memory aids.

Figure 1a shows a weblog entry that an office entrance door posted when it was opened during the night. Figure 1b shows a weblog entry of temperature changes. This entry includes a movie that shows the temperature changes in a room during one day created by using thermometers attached to objects. (A virtual object posted this entry.) Our EGC system automatically generates such entries in the weblog.

IEEE Pervasive Computing, 7(4), pp. 20-27. (Draft version)

Questionnaire survey

Khai Truong, Elaine Huang, and Gregory Abowd investigated desired capture applications through questionnaires.[11] In their investigation, however, the participants proposed only about a dozen applications. In addition, although the survey included some active recording applications such as voice memo pads, it didn't address the requirement of realizing automatic (passive) recording by using environment-embedded sensors. We used a 90-minute free-form questionnaire to ask five participants (not researchers) what kinds of data they wanted to store and how they wanted to reuse the stored data. Before giving the participants the questionnaire, we explained sensing and activity recognition technologies and asked them to assume an indoor environment such as a future home or office in which events and activities could be detected.

> The main idea is to make the game persistently available to the players anytime. ALIGN THE TOP BAR WITH BASELINE. NEED PULL QUOTE.

In our surveys, we obtained 66 answers that included the data and associated services the participants wanted to store and reuse. We categorized this data into three types: object related (75.8 percent), person related (21.2 percent), and location related (3.0 percent). Object-related data concerns an object's states, uses, and related activities—for example, "Record which shoes to wear to my office. I want to monitor this because wearing the same shoes for several consecutive days is not good for the shoes." Person-related data concerns a person's states—for example, "If my sleep time is recorded, I can manage it and check my biorhythms." Location-related data concerns a location's states—for example, "If the temperature in a room is recorded, I can know which part of the room is too cool." Most data is object related.

Anind Dey and his colleagues described desired context-aware applications.[12] Because object-related applications accounted for the majority in the survey, they stated that users focus on objects in their environment.[12] This statement accords with our result.

Classifying the data according to the participants' intended purpose lets us categorize this data as memory aids (51.2 percent), daily-life management (83.3 percent), and information sharing (22.7 percent). Users can employ the daily-life management data to improve their lives by exploiting stored data and monitoring daily activities and environments. Most data is multipurpose—for example, "Record what we eat. Such information is useful for maintaining a well-balanced diet." This example has two purposes: as a memory aid and for daily-life management. Another example is "Monitor CD players. We can know who likes what kind of music." This can serve as a memory aid and for information sharing. "Record flower watering to control the amount of water provided because in an office with multiple persons there may be too much or too little watering" has daily-life management and information-sharing roles.

We also found that most data in the answers should be translated to content periodically (87.9 percent) or when a user requests it (93.4 percent)—for example, "Record the number of visits to the restroom per day. Then I can take care of my health and monitor my digestive system." By computing sensor-data statistics, EGC systems can issue restroom reports every week or for a given number of days to meet the user's request. From this result, we see that the systems continually generate a considerable amount of content. Thus, users need effective content-access methods. The systems translate some data to content when particular events occur (21.2 percent)—for example, "Monitor an entrance because recordings of comings and goings at unusual times are useful for security." This entrance example is environment dependent, because what constitutes an unusual time is different for homes than for offices. Of the data we obtained, 60.6 percent of services required domain-specific information.

Dey and his colleagues obtained some context-aware applications via questionnaire surveys[12]—for example, "When close friends are over, they know a lot of my music, so I'd like to expose them to some new quirky stuff" that reflects the participants' preferences. Unlike the case in context-aware applications, all services obtained in our survey—such as temperature recording, sleep time recording, and meal recording—would be useful to any user. Also, 72.8 percent of all data (including duplication) overlapped with other data, and the services desired by users were very similar. This is because our survey focused on logging systems for environments—that is, for everyone rather than for specific individuals, which would reflect the participants' preferences.

Six requirements for storing, managing, and providing EGC

Our survey indicates that a lot of EGC is generated automatically. Here, we focus particularly on how to achieve the three purposes of memory aid, daily-life management, and information sharing, while dealing with large quantities of EGC. We established the following six requirements for storing, managing, and providing EGC by referring to studies that also deal with massive amounts of content such as lifetime data and experience:

- *Generating content.* We generate EGC with, for example, texts, graphs, and figures. Figure 2 conceptualizes the process from EGC generation to EGC use. The upper portion shows content

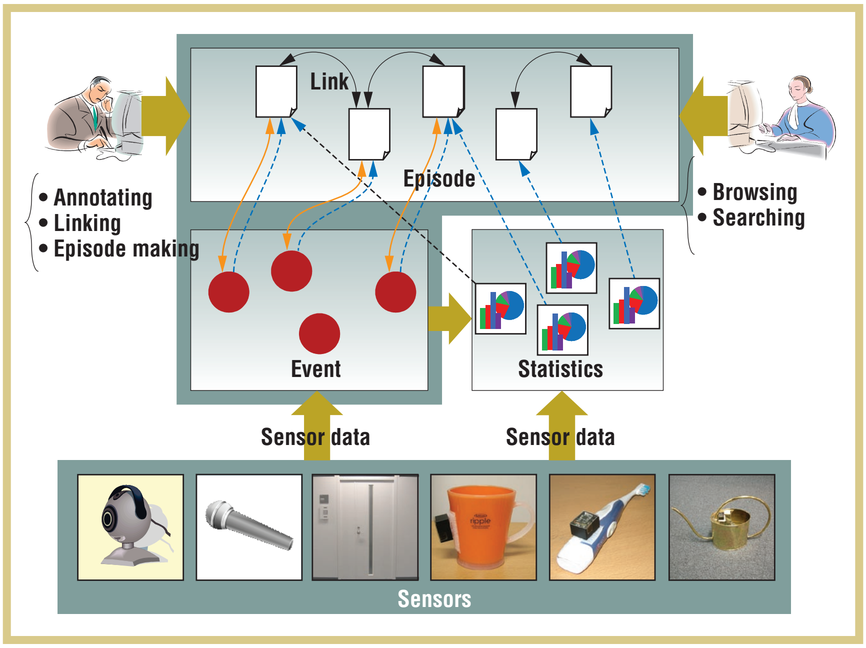


Figure 2. Concept of environment-generated content service. The EGC system calculates events and statistics by processing sensor data in real time. The system generates episodes from the events and statistics. Users can browse and search for the generated episodes. They can also annotate the episodes, link them, or make an episode such as a diary.

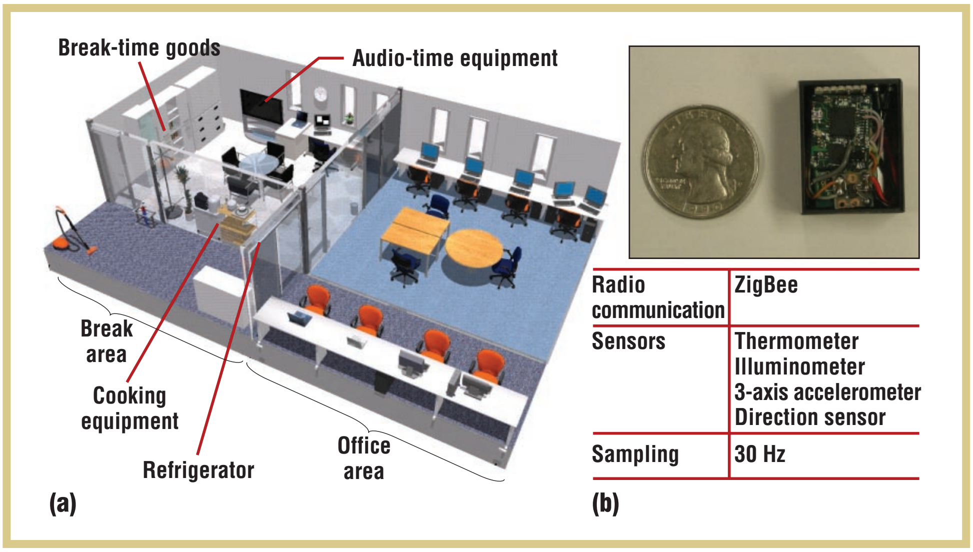
automatically generated from events and statistics obtained through sensor-data processing. We call such content an *episode*. Examples of events are "a door is opened" detected by a door sensor, and an ADL of making tea inferred by using sensors attached to a kettle and cups. Examples of statistics are the temperature changes exhibited by an object during a week, and the number of times plants are watered each day. Generated content such as a text-formatted report of an observed event or a graph depicted according to calculated statistics is useful for daily-life management because users can easily grasp characteristics of the event and the statistics of their lives at a glance. We can also leverage generated content such as text for text searches.

- *Search and collection*. The management of massive amounts of data is shifting from a hierarchical structure to search and collection—for example, Gmail, Macintosh Spotlight, and Windows Desktop Search. We deal with a large number of episodes and events to aid memory and achieve daily-life management. Thus, our search interface must enable searching for episodes and events using simple keywords. Incorporating multiple ways of looking at things helps foster user understanding and insight in relation to those things.[2] We must also provide various visualization methods, such as sorting search results for an episode in terms of time and categorizing the search results for an event. We can improve episode accessibility by categorizing a massive number of episodes similar to the way Gmail uses labels.
- *Sharing*. Many studies aim to share recorded experiences, memories, and knowledge.[3] Our survey obtained some information-sharing services. Information sharing among users in an environment could encourage communication and mutual understanding. To achieve information sharing, we need to provide generated episodes through easily accessible media.
- *Summarization*. In environments in which events and changes occur frequently, users can miss some of them. Ramesh Jain suggested that chronicling systems could provide summaries of important events.[5] We must provide summaries of events (episodes) such as the index pages of news sites. We can improve these summaries by highlighting important episodes. This function is useful for daily-life management.
- *Annotation*. Nontext media such as those representing an episode containing graphs and videos require annotations for a keyword search.[2] Even for a text episode, if users can annotate the episode with what they think about it, the annotations will help them to reuse it.
- *Linking*. MyLifeBits links media and their annotations, as well as the original media and other media made from them, so that users can find context and commentary related to the media.[2] In general, links enable easy retrieval of related information. In our case, if we connect one episode with another that's written with reference to the former via a two-way link, we can easily trace the ancestry and posterity of episodes. An event should also be connected with an episode that refers to the event. Ali Mazalek, Glorianna Davenport, and Hiroshi Ishii have implemented a system that provides multiviewpoint stories of characters (storytellers) by using a tangible interface.[13] Providing multiple viewpoints of an event lets users grasp a holistic image of the event. Assume many users write episodes about an event. If the event connects the episodes, we can easily provide the users with various views of the event.

Object-Blog system

We implemented the Object-Blog system in our experimental environment, which Figure 3a shows. Ten workers (not researchers) undertook their

IEEE Pervasive Computing, 7(4),
pp. 20-27. (Draft version)



Radio communication	ZigBee
Sensors	Thermometer Illuminometer 3-axis accelerometer Direction sensor
Sampling	30 Hz

Figure 3. Our experimental settings: (a) experimental environment and (b) our implemented sensor node and its specifications (radio communication protocol, installed sensors, and sampling rate of sensors).

ordinary work from 9 a.m. to 5 p.m. every weekday. We installed eight video cameras and two microphones in the room and taped our implemented sensor nodes to 50 daily objects such as doors, drawers, cups, a coffee maker, a kettle, a watering can, slippers, and a toothbrush. Figure 3b shows a sensor node and its specifications. We used simple generic sensors to detect object usage. Emmanuel Tapia, Stephen Intille, and Kent Larson employ a low-power consumption sensor node especially designed to detect object use.[14] The node wakes up and senses its acceleration only when a piezoelectric-film sensor detects motion.

Design guidelines

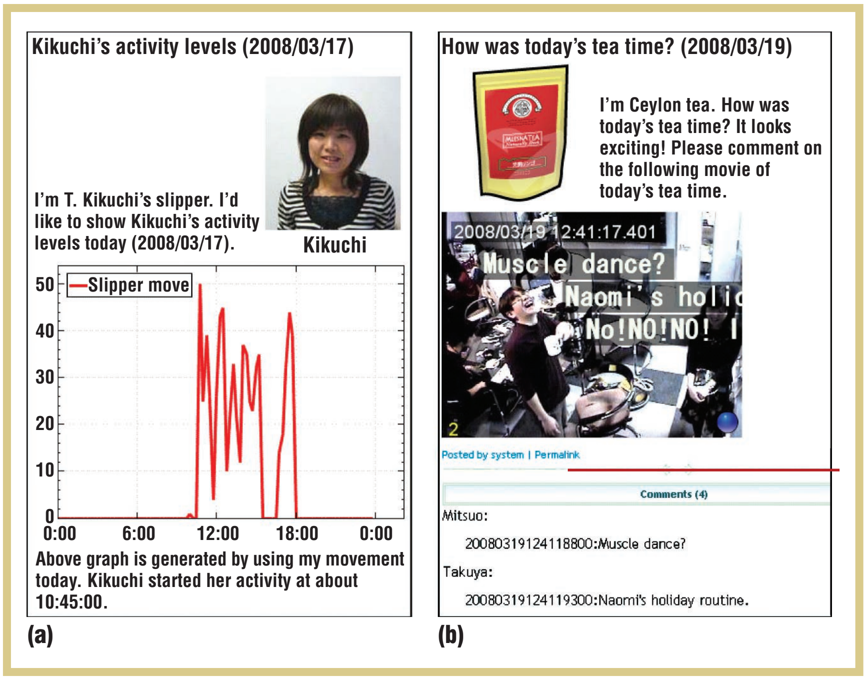
We established design guidelines for the EGC provision service to meet our six requirements and incorporate the knowledge obtained from the questionnaire survey. These guidelines address four essential elements: Web publishing, object centricity, rule-based system, and easy setting. Although we describe these essentials in relation to the Object-Blog system, the basic idea behind them can apply to any other EGC provision service.

Web publishing. Many logging applications use the Web to share experiences and events.[3] Because the responses to the survey also mentioned information-sharing services, we published episodes as Web content with high accessibility. On the other hand, the logging applications on the Web involve privacy issues. However, according to David Brin, the major issue is not whether activities should be recorded but rather who should have access to the data.[15] We can control user access by employing various types of Web security technology, including basic access authentication and electronic signatures. We selected a weblog, one of the most popular tools for experience archiving, as the medium for publishing episodes. Thus, we publish episodes as weblog entries.

Weblogs already have some of the elements necessary for providing EGC. Most weblog systems have collection functions, such as monthly and daily entry archives and entry categorization. However, the entry search functions installed in weblog systems aren't powerful. Thus, we must prepare another search function with various visualizations. Also, weblog systems provide a summary: the index page of the weblog and the resource description framework (RDF) site summary. Moreover, because users can leave comments on weblog entries, they can communicate with one another and share their experiences via a weblog. In addition, users can easily attach annotations to entries (episodes) by leaving comments on the entries. Finally, all weblog systems have a track-back function that links one episode to another that refers to the former.

Object centricity. The participants in our survey were interested in objects and those objects' states and changes. Also, an earlier survey on context-aware applications showed that participants would normally interact with devices (objects) in their home.[12] Thus, in our application, personified objects tell stories in the first person about their experiences by posting episodes about their events to a weblog. Personified objects can provide familiar, more natural logs for family members than impersonal logs. Furthermore, because we assume that each object has its own personality, we let objects communicate with one another via the weblog—that is, an object can post a comment to an entry that another object posted. Assume, for example, that a door posts an entry about a detected anomaly. Then, other objects can post comments to the entry about their use around the time of the anomaly. To accomplish this, we can naturally present information aggregated from multiple objects, and present additional information if needed.

Rule-based system. Object-Blog is a rule-based system, meaning it has a set of condition-action rules. Each rule is associated with a physical object. When a rule's condition is satisfied, the system posts an episode about the object associated with the rule. The rules' conditions



Kikuchi's activity levels (2008/03/17)

I'm T. Kikuchi's slipper. I'd like to show Kikuchi's activity levels today (2008/03/17).

Above graph is generated by using my movement today. Kikuchi started her activity at about 10:45:00.

(a)

How was today's tea time? (2008/03/19)

I'm Ceylon tea. How was today's tea time? It looks exciting! Please comment on the following movie of today's tea time.

Posted by system | Permalink

Comments (4)

Mitsuo:

20080319124118800:Muscle dance?

Takuya:

20080319124119300:Naomi's holiday routine.

(b)

Figure 4. Weblog, with entries indicating (a) activity levels during one day and (b) a comment-able movie.

fall into three categories: periodic (for example, every Wednesday at 12 midnight), when an event occurs, and when a user makes a request that triggers episode generation in our survey. Users can request episode generation by posting an entry to Object-Blog. (For example, they can request an episode that includes a digest of the clothes they've worn over the past 15 days.) Thus, if a user posts an entry that requests episode generation, Object-Blog posts the requested episode as an entry comment. (The present system requires the user to give a command for episode generation.)

Easy setting. Many rule-based context-aware systems use rules that users create because their applications are environment specific, and these rules reflect user preferences.[12] However, we don't have to use rules created by users, because the services obtained in our survey should be useful to any user, and the desired services are similar. We combine prepared rules in advance in a domain-independent format with domain-specific information that users input, because many services obtained in our survey use domain-specific information. For example, consider a service consisting of the following: "Monitor an entrance. Recordings of comings and goings at unusual times are useful for security." In this example, users input the definition of "unusual times" into a database in advance. Prepared rules make it possible to reduce user effort because users don't have to construct rules.

Also, we introduce the concept of a *character package* into the system. A character package is a collection of rules that define an object's personality. It has its own name—for example, a voluble cabinet, a concerned entrance door, or a proud kettle. Users can breathe life into an object by simply associating a character package with an object that has a sensor node, on their PCs. If required, a user can input domain-specific information. Then, the object automatically generates episodes by using its sensor data according to rules in the character package.

An important issue concerning rule-based content generation systems is user tiredness. The simplest solution to this issue is to provide a wide variety of rules. The way to realize the solution is by disclosing the rule description specifications to encourage rule development according to the history of software plug-in development.

Implementation

We implemented an Object-Blog system that follows the design guidelines. We used Movable Type as the weblog platform and the C# language to implement sensor-data processing, event detection, and automatic episode making. This implementation has three key components: event detection, episode making, and episode and event search.

Event detection. We implemented four event detection plug-ins in the present system: The *conversation detector* detects conversations from microphone signals. The *physical-phenomena detector* detects roughly 10 different kinds of physical phenomena (such as the movement, dropping, and rotation of objects) by using acceleration signals. The *object event detector* detects events that are specific to approximately 10 kinds of objects, such as doors, cabinets, and drawers. For example, it detects the opening and shutting of the objects by using signals from accelerometers and direction sensors. The *ADL detector* uses machine learning to detect roughly 20 kinds of ADLs, such as make tea, cook pasta, cook rice, make juice, brush teeth, listen to music, and practice aromatherapy. We based our implementation of the ADL detector on the ADL inference algorithm presented by Tapia, Intille, and Larson.[6]

Episode making. To provide content effectively, we implemented plug-ins for episode making such as the movie plug-in, graph plug-in, and superimposed-picture plug-in. We used the movie and superimposed-picture plug-ins to create the movie in Figure 1b representing temperature changes. Figure 4a shows an entry that includes a graph of a user's

IEEE Pervasive Computing, 7(4),
pp. 20-27. (Draft version)

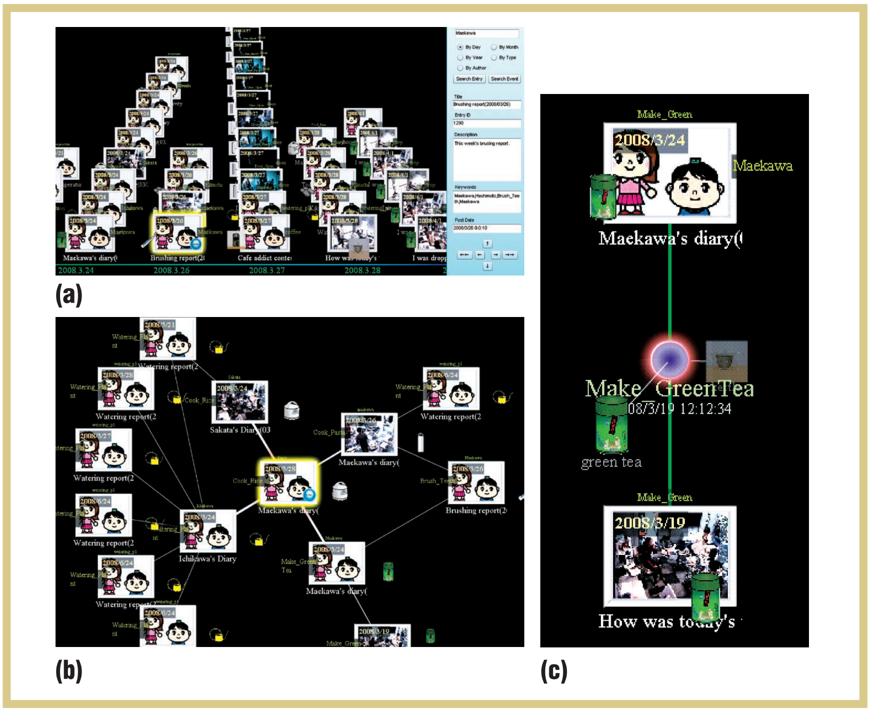


Figure 5. Screenshots of the event-episode explorer, which provides a keyword search function for entries and events and provides various views of search results: (a) timeline, (b) graph of similar episodes, and (c) event-centric modes. (Each rectangle with a white-colored frame represents a weblog entry. The purple-colored circle in (c) represents a physical event.)

activity levels during one day. It's necessary to provide graphs and charts so that users can more easily understand a holistic image and find new characteristics about their lives. Annotating is a troublesome chore. So, we implemented the comment-able movie plug-in so that users could annotate easily and enjoyably. Figure 4b shows an entry that encourages us to annotate a conversation during tea time. The entry includes a movie with a voice recorded during tea time. If a user leaves a comment consisting of time and text on the entry, the movie shows the text at that time in the Times Square format, which scrolls the text on one horizontal line across the screen. This function was inspired by the system used by the video-sharing site Nico Nico Video (www.nicovideo.jp). Using this function, we can annotate episodes by sharing and exchanging opinions and experiences.

Episode and event search. The event-episode explorer is a search interface for episodes and events that provides various visualizations. Users can launch it by selecting a hyperlink on the Object-Blog page. Figure 5a shows keyword search results of episodes sorted by day. The keywords extracted from an episode and images of objects related to the episode surround one of the images representing the episode. The keywords and images help explain what the episode is about. Also, by selecting an episode, users can browse the corresponding entry page. Figure 5b shows an episode, along with other similar episodes, represented in a graph structure. To remind users of past experiences by association, we provide a similarity search. Users can also browse multiple viewpoints of an event by selecting the event ("Make GreenTea"), causing episodes that refer to the event to appear around it (Figure 5c).

Participants working in our experimental environment have provided feedback about Object-Blog. Some said that they could more easily record the daily events of their lives than they could in a paper-based diary because objects automatically record logs on their behalf, and participants simply commented on the logs. Also, participants could input casual comments because the objects generate enjoyable, conversation-like logs. By providing EGC on the Web, we could implement some services that can't be achieved solely using sensor data. P

REFERENCES

1. V. Bush, "As We May Think," *The Atlantic Monthly*, vol. 176, no. 1, 1945, pp. 101–108.
2. J. Gemmell et al., "MyLifeBits: Fulfilling the Memex Vision," *Proc. ACM Int'l Conf. Multimedia*, ACM Press, 2002, pp. 235–238.
3. K.N. Truong, G.D. Abowd, and J.A. Brotherton, "Who, What, When, Where, How: Design Issues of Capture and Access Applications," *Proc. Int'l Symp. Ubiquitous Computing* (UbiComp 01), LNCS 2201, Springer, 2001, pp. 209–224.
4. S.S. Intille et al., "Tools for Studying Behavior and Technology in Natural Settings," *Proc. Int'l Symp. Ubiquitous Computing* (UbiComp 03), LNCS 2864, Springer, 2003, pp. 157–174.
5. R. Jain, "Multimedia Electronic Chronicles," *IEEE Multimedia*, vol. 10, no. 3, 2003, pp. 112–111.
6. E.M. Tapia, S.S. Intille, and K. Larson, "Activity Recognition in the Home Using Simple and Ubiquitous Sensors," *Proc. Pervasive Computing*, LNCS 3001, Springer, 2004, pp. 158–175.
7. M. Blum, A.S. Pentland, and G. Troster, "InSense: Interest-Based Life Logging," *IEEE Multimedia*, vol. 13, no. 4, 2006, pp. 40–48.
8. G.C. De Silva, T. Yamasaki, and K. Aizawa, "An Interactive Multimedia

Diary for the Home," *Computer*, vol. 40, no. 5, 2007, pp. 52–59.

9. M. Lamming and M. Flynn, "Forget-Me-Not: Intimate Computing in Support of Human Memory," *Proc. FRIEND21 Int'l Symp. Next Generation Human Interface*, Inst. for Personalized Information Environment, 1994, pp. 125–128.

10. W. Newman, M. Eldridge, and M. Lamming, "PEPYS: Generating Autobiographies by Automatic Tracking," *Proc. 2nd European Conf. Computer-Supported Cooperative Work* (ECSCW 91), Kluwer Academic Publishers, 1991, pp. 175–188.

11. K.N. Truong, E.M. Huang, and G.D. Abowd, "CAMP: A Magnetic Poetry Interface for End-User Programming of Capture Applications for the Home," *Proc. Int'l Symp. Ubiquitous Computing* (UbiComp 04), LNCS 3205, Springer, 2004, pp. 143–160.

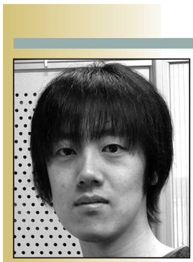
12. A.D. Dey et al., "iCAP: Interactive Prototyping of Context-Aware Applications," *Proc. Pervasive Computing*, LNCS 3968, Springer, 2006, pp. 254–271.

13. A. Mazalek, G. Davenport, and H. Ishii, "Tangible Viewpoints: A Physical Approach to Multimedia Stories," *Proc. ACM Int'l Conf. Multimedia*, ACM Press, 2002, pp. 153–160.

14. E.M. Tapia, S.S. Intille, and K. Larson, "Portable Wireless Sensors for Object Usage Sensing in the Home: Challenges and Practicalities," *Proc. European Conf. Ambient Intelligence* (AmI 07), LNCS 4794, Springer, 2007, pp. 19–37.

15. D. Brin, *The Transparent Society*, Perseus Books, 1998.

the AUTHORS



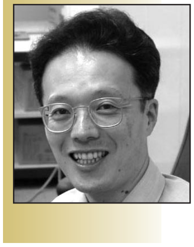
Takuya Maekawa is a researcher at NTT Communication Science Laboratories in Kyoto, Japan. His research interests include Web content engineering in mobile and ubiquitous environments. Maekawa received his PhD in information science and technology from Osaka University. Contact him at maekawa@cslab.kecl.ntt.co.jp.

Yutaka Yanagisawa is a researcher at NTT Communication Science Laboratories, in Kyoto, Japan. His research interests include localization systems for mobile computing environments, programming languages to develop middleware and operating systems for small mobile devices, and augmented-reality systems to support mobile device users. Yanagisawa received his PhD in engineering from Osaka University. Contact him at yutaka@cslab.kecl.ntt.co.jp.

Yasue Kishino is a researcher at NTT Communication Science Laboratories in Kyoto, Japan. Her research interests include ubiquitous computing and human interfaces. Kishino received her PhD in information science and technology from Osaka University. Contact her at yasue@cslab.kecl.ntt.co.jp.

Koji Kamei is a researcher at NTT Communication Science Laboratories in Kyoto, Japan. His research interests focus on sharing of real-world events, including event extraction from sensors and ontology construction for real-world events. Kamei received his ME in electronics and communication engineering from Kyoto University. Contact him at kamei@cslab.kecl.ntt.co.jp.

Yasushi Sakurai is a senior research scientist at NTT Communication Science Laboratories in Kyoto, Japan. His research interests include indexing and search algorithms, data mining, and sensor data processing. Sakurai received his PhD in engineering from the Nara Institute of Science and Technology. He is a member of the ACM. Contact him at yasushi@cslab.kecl.ntt.co.jp.



Takeshi Okadome is a group leader at NTT Communication Science Laboratories in Kyoto, Japan. His research interests focus on creating content for real-world events using sensors. Okadome received his PhD in computer science from the University of Tokyo. He is a member of the ACM. Contact him at houmi@idea.brl.ntt.co.jp.