Spacelog Data Acquisition, Management and Applications in a Gateway-based Smart Space*

JIANHUA MA, RUNHE HUANG, TOMOMI KAWASHIMA AND BERNADY O. APDUHAN†

Faculty of Computer and Information Sciences
Hosei University
Tokyo, 184-8584 Japan
Faculty of Information Science
Kyushu Sangyo University
Fukuoka, 813-8503 Japan

Spacelog is a continuous collection of existing states and activity experiences of various entities in a real spatial site or environment such as a laboratory and a home, and it can be regarded as a context history database or conceptually seen as a memory organ specifically for the physical environment. Spacelog can be exploited to enhance functions of the computerized physical environments, enabling users to know what happened and how they happened in the environments, and further make the environment smarter to improve users’ quality of life. However, the logs captured directly from various sensing devices are raw data, heterogeneous types, massive in data amount, and with illegible meanings. Thus, the challenges in developing spacelog applications are how to effectively acquire, administer and apply the heterogeneous, massive and illegible raw log data. In this study, we proposed a gateway-based spacelog system architecture to deal with various sensing devices, discuss techniques on log data acquisition and management, and demonstrate two spacelog applications called LabLog and HomeLog.

Keywords: spacelog, sensor, context, heterogeneity, database, gateway, smart space, lablog, homelog, service, retrieval, activity summarization

1. INTRODUCTION

Context-awareness has been generally regarded as one of the basic features in many ubiquitous systems and applications. Context is any information used to characterize the situation of an entity [1, 2]. In terms of time occurrence, the contexts can be the present ones currently existing or the past ones that previously happened. To further precisely characterize the situation of an entity, it is often necessary to use not only the present contexts but also the past contexts, called context history. So far, there are many studies on how to acquire and use contexts, but very less study has been done on how to keep, manage and utilize the context history although it has been realized as a key issue for better and more sophisticated ubiquitous services [3]. Therefore, we proposed the concept of “spacelog” [4], which can be seen as a kind of context history.

Spacelog is a continuous collection of digital records on existing states and activity experiences of multiple entities in a real space or physical environment, e.g. a laboratory, a classroom, a library, a home, a clinic, a shop, a restaurant, a farmland, a construction field, etc. A smart space, one of the major ubiquitous applications, is an intelligent real environment such as a room equipped with sensors, actuators and other devices, which
can provide automatic responses according to users’ needs and contextual information in the environment [5]. Hence, the spacelog can also be seen as a context history database in a smart space system, and functions as a “memory organ” of a real spatial environment. Spacelog can be exploited to enhance/extend functions of the computerized physical environments and provide many novel intelligent services such as (a) effective retrievals of past events that happened in a space; (b) systematic summaries of entities’ experiences which occurred in a specified period; and (c) automatic detections of important events or changes for prompt reactions.

Spacelog data is captured automatically via various ubiquitous devices such as cameras, microphones, sensors, RFID tags, etc., which are distributed in a physical spatial environment or carried by users/robots who are in this space. However, the logs acquired directly by the various sensing devices are raw data, and they are heterogeneous in types, massive in data amount, and illegible in meanings. Thus, the challenges in developing spacelog systems and applications are how to effectively capture, store, manage and utilize the heterogeneous, massive and illegible raw log data.

In what follows, we present our gateway-based system architecture to acquire and administer spacelog data as well as two representative application prototypes. The next section further clarifies the spacelog concept as compared with computer-log and lifelog, and discusses spacelog data characteristics. Section 3 describes a general spacelog system based on gateways, wherein each gateway can handle some type of sensing devices and the system as a whole can deal with heterogeneous devices and networks. Two spacelog applications called LabLog and HomeLog which demonstrate spacelog acquisitions and processing are given in sections 4 and 5, respectively. Conclusions and future work are addressed in the final section.

2. SPACELOG CHARACTERISTICS AND TECHNICAL ISSUES

Log is generally some kind of records about history or experience of an entity in a certain period. The log or history data does not only enables a user/system to know what happened in the past, but also knows what led to the present state and further predict what will occur in the future so as to provide proper proactive actions when necessary. Actually, the logs have been used for various entities in computer and communication systems for a long time already.

When the entity is an ordinary computer, the logs are kept in the computer-log file to automatically record all changes of hardware and software made on the computer. When the entity is a web server, the server-log records all clients’ access history to the server, and enables its administrator to know where clients were from, when they accessed the server and which web pages were accessed so as to improve the quality of web services. It is reported that the size of the computer and communication log data may reach hundreds of gigabytes of data per day for a large organization, and log management (LM) has become an interesting but challenging research topic [6].

Since year 2000, a new area called lifelog has emerged, and it is a sequence of data records about personal life experiences of a special entity, i.e., an individual. The initial lifelog ideas may date back to Fuller’s Dymaxion Chronofile which documented his life as completely as possible [7], as well as Bush’s MEMEX which can keep written memos.
and their related materials so as to enhance human’s memory [8]. In recent years, some projects, e.g., LifeLog/ASSIST [9], MyLifeBits [10], LifePod [11], Ubiquitous Memories [12], SenceCam [13], as well as our LifeImages [14], are trying to realize the Chronofile’s and MEMEX’s ideas by automatically recording all the states and acts of an individual’s daily life with continuous capture using various sensors. These captured data records can be regarded as a personal database of life experiences, which can be retrieved whenever necessary and used as reference to improve the person’s life quality.

Different from the above mentioned computer-log and lifelog, which is targeted for a single entity such as a web server or an individual person, the spacelog on the other hand is generally for multiple entities, and their existence states and activity experiences are in a real space or physical environment such as a laboratory, a classroom, a library, a home, a clinic, a shop, a restaurant, a farmland, a construction field, etc. The entities in a space can be people, facilities, artifacts, and space-related states, e.g., temperature distribution, air quality, noise level, sound source, etc. This variety of possible entities in a space marks one essential difference of the spacelog from the computer-log and the lifelog.

Although the spacelog partially shares one common feature, i.e. records of a specific physical site, with surveillance monitoring cameras and sensor nets, they are greatly differentiated by the number of medium types used. The surveillance cameras are mainly to take and keep the visual media data, and a sensor net is often used to sense some specific type(s) of media. In contrast, a spacelog system may be involved in collecting many kinds of media data via different sensing devices distributed in a space, and all the logs will be kept integrally in some databases. In other words, spacelog can be seen as extensions of the surveillance cameras and sensor nets in terms of recorded medium types.

Therefore, the spacelog is a continuous collection of digital records for multiple entities in multiple media with different types of devices. It can be regarded conceptually as a special database of a physical space, and functions as a context history in a smart space system or as a “memory organ” of an intelligent spatial environment. Furthermore, the intelligent capability of smart spaces/environments can be greatly enhanced when incorporated with the spacelog, and the whole individual’s lifelog may be compiled by combining spacelogs from different environments where the individual has ever stayed.

The spacelog data is continuously collected using various devices such as cameras, microphones, sensors, RFID tags, etc., which are either fixed somewhere in a physical environment, e.g., a home, a laboratory, an office, a classroom and a library, or carried by users/robots that can move in the environment to get log information at different locations. The data acquired by these media devices are raw data, which are diverse, and are of the following characteristics.

- **Heterogeneous:** different spaces, devices, networks, media, entities, and a variety of applications as well as users, make spacelog data in heterogeneous forms. It is thus necessary to find the common features/rules, general methodologies, uniform data format, effective approaches and associated techniques to deal with such great heterogeneity in spacelog systems.
- **Massive:** fast and continuous accumulation of data from a real space along time creates massive spacelog data. Suppose the average data amount generated by all sensing devices in one second is \( R \, B/s \) (Bytes per second), then the data amount will be roughly
30RMB/year, that is, about 30TB in a year if \( R = 1MB/s \) just for a single environment. The accumulation of such massive data must be considered as one crucial characteristic in designing how to efficiently and effectively capture and store/retrieve data.

- **Illegible**: The acquired data directly coming from devices and sensors is raw, original detailed records but with least semantic meanings. The main problems with raw data are that, (1) it’s huge with high data redundancy; (2) less readable for users; and (3) less relational among data. The important and challenging issue is how to process and interpret the raw spacelog data and make them meaningful.

Due to the above log data characteristics, it is not trivial to process, analyze, manage and utilize the raw data. A whole spacelog system can be developed based on a multi-layer general model, as shown in Fig. 1.

![Fig. 1. The general multi-layer model of a spacelog system.](image)

The lowest layer is the real space/environment where various entities including human users, appliances, facilities and other physical objects are placed or located. The second layer from the bottom covers all devices including various sensors, RFID tags/readers, robots, hard discs and their associated communication networks. With the great heterogeneity of the devices and networks, a middleware layer is necessary to provide unified programming abstractions and common services for spacelog application developers to interact with these heterogeneous devices across different networks. Above the middleware, is the spacelog database layer, which keeps all the log data. Here, a unified data format should be defined to cope with different types of media and original styles captured from various sensing devices. Before saving the log into the database, the sensed raw data is necessarily processed to reduce data redundancy, and indexed to properly organize the spacelog data. As we mentioned earlier, the data on the spacelog database is usually massive and illegible, and it is then necessary to have many effective tools to automatically retrieve, analyze, mine, and summarize the logs. All of these tools are placed at the spacelog utilities layer on top of the spacelog database. The application layer is about concrete spacelog applications which provide corresponding services directly to users. A spacelog system can be called LabLog if it mainly serves people in a laboratory, or called HomeLog when it works in a home to serve a family. The topmost layer, *i.e.*, the user interface, offers convenient interfaces between users and applications.
Furthermore, the whole system should be manageable and scalable to enable convenient installations and controls of a new device/network, a new utility, and a new application. It must also be very reliable and secure to protect the importance and privacy of some spacelog data.

Actually, many technical issues exist when designing and developing practical spacelog systems. For example, what log data should be captured using what devices? How and where to capture? What pre-processing on the raw log data is necessary to remove redundant or useless data? How to efficiently and synchronously keep the records from different devices with varied media formats into the available storage? What’s the unified format of log records for storage, representation, presentation, etc.? What are the general models and approaches for spacelog analysis, mining, indexing, summarization, retrieval, presentation, key information extraction, etc.? How to update and manage spacelog considering system scalability, reliability, and security as well as privacy protection? How to use log data conveniently and flexibly by other modules or applications? What are the suitable interfaces and interactions between human and spacelog systems?

Our current study on spacelog are focused on some core issues in the following aspects: (1) a general scalable framework/middleware to deal with heterogeneous devices, networks, data types and so on; (2) basic relationships and principles to acquire more information using less number of sensors as well as their optimal installation positions and combinations; (3) flexible spacelog acquisitions using robots and their cooperative work with other devices in smart spaces; (4) a spacelog database and related techniques to extract and index captured raw spacelog data with a unified spacelog description scheme for effective retrievals and summarizations of interested states and events occurring in an environment.

In the next section, we present a gateway-based smart space system that can support a variety of spacelog acquisitions, administrations and applications due to its great scalability. Two application prototypes called LabLog and HomeLog are given as spacelog case studies in sections 4 and 5, respectively.

3. THE GATEWAY-BASED SMART SPACE SYSTEM

As we previously explained, the spacelog can be regarded as the context history or a special database in a smart system. One of the fundamental problems in smart space systems is the great heterogeneity of devices with different functions, hardware, software, networks, and so on. A common middleware is generally regarded as one important solution to deal with the heterogeneity of devices [15]. A smart space middleware can be based on either the popular technologies such as Jini [16], OSGi [17] and SLP [18], or by some special ones, e.g. United Space [19], Comet-based framework [20], Mobile Code [21], and Dynamic Binding [22]. It is generally assumed that all the devices must strictly follow the specifications, interfaces and procedures defined by the middleware. This assumption works well for relatively large devices that have enough memory capacity and computational power to run the necessary software requested by such a middleware. It is, however, impractical for very small devices such as many sensors and actuators to run the requested software due to limited memory and computation capacities. Actually, many available sensing and actuation devices are developed by different manufacturers
using their own interfaces, programming languages, software environments, etc.

To handle these small and maker-dependent devices, one approach to improve the middleware is to add special functional modules to the middleware for connectivity extension to these devices. As a result, the middleware will become very complex especially when different types of devices exist, and have to be reconfigured whenever new types of devices are introduced. That is, this approach will face the scalability problem if there are many devices that do not follow the standards originally specified by the middleware. In order to keep a common middleware with flexible scalability, we have adopted a two-level middleware architecture, which consists of a set of Smart Space Gateways (SSG) and a Smart Space Server (SSS) as shown in Fig. 2.

![Fig. 2. The gateway-based smart space system architecture.](image)

The SSGs are for the interactions with various sensing and actuation devices, and the SSS is for management of all SSGs in a smart space. That is, the device heterogeneity is dealt with by the SSGs, which are scalable and managed in a unified mechanism by the SSS. The context data captured by a sensor will be forwarded by its corresponding SSG to the SSS that passes the data to Spacelog Database (SDB). The SDB will make some necessary processing (e.g. compressing, formatting and indexing) of the sensed raw data and then save the data into some storage like a hard disc. Atop the SSS and SDB, the Smart Space Applications (SSA) can be built, and the Space Log Utility (SLU) may provide a set of tools and APIs to support development of SSAs. Users will exploit the SUI (Space User Interface) to interact with the smart space system.

SSG and SSS are the two core parts in our smart system middleware, and they are designed using the SOA (service oriented architecture) approach [23]. Fig. 3 shows the components and architecture of an SSG that consists of both the hardware and software. Although the SSG hardware may contain some embedded sensors and actuators, it provides wired ports of GPIO (General Purpose Input Output) as well as some wireless network connections to other sensors and actuators. That is, the devices connected to an SSG are extensible according to special requirements of the desired applications. The electricity provision to the SSG can be from either the conventional power line or some
kind of battery. In the latter case, power consumptions should be taken into considerations for the connected devices and all related software in the SSG.

The SSG software is made up of four services; *i.e.*, communication, discovery, sensing, and actuation. The communication service runs on the protocol stack of a wireless network such as BlueTooth, ZigBee or other WPAN (Wireless Personal Area Network). It is a basic service for communications between SSG and SSS. The discovery service is used by an SSG and an SSS to automatically find each other and periodically exchange management related messages. According to requests from an SSS, the sensing service in an SSG will control the start/stop working of sensors and send the sensed data to the SSS. Similarly, the actuation service will send control commands and values to actuators based on requests from the SSS. An SSG can communicate not only with the SSS but also with other SSGs and other wireless devices such as wireless sensors, RFID readers and robots (see [24]) so long as they are equipped with the same network and protocol stack. Fig. 4 shows the software components and architecture of the SSS that runs on a computer inside a smart space.

![Fig. 3. The architecture of smart space gateway (SSG).](image)

![Fig. 4. The architecture of smart space server (SSS).](image)
It acts as a centralized administrator of the whole system for a single smart space and provides basic functions necessary for management of SSGs, control of sensors and actuators, storage of sensed data, display of system working states, access of information, etc. These functions are carried out with the seven service modules, among which the four services of communication, discovery, sensing and actuation in SSS work as pairs with the corresponding four services in the SSG.

4. LABLOG ACQUISIONS AND RETRIEVALS

The “space” in the spacelog can be any real site, such as a home, a laboratory, a classroom, a kindergarten, an office, a shop and a clinic. It can be some subspace, e.g. a refrigerator and a toolbox in a space, or a hyperspace like a hospital and a campus that consists of multiple spaces. Accordingly, the general term spacelog can be substituted by more concrete ones such as lablog, homelog, refrigeratorlog and campuslog for specific places/sites. Because of its wide coverage, the spacelog can be used in many people’s daily environments for a variety of novel services. The possible applications may fall into the following categories: (a) Retrievals of necessary detailed information from the spacelog; (b) Summarization of main activities related to an interested entity in some period; (c) Detections of critical changes using spacelog for prompt reactions; (d) Mining of important events or key data; and (e) Awareness of the exact situations using both current contexts and past log information. This section describes the LabLog prototype to acquire the log data and retrieve some information which occurred in our laboratory, and the next section gives the HomeLog, as one automatic summarization of a kid’s activities in a home.

The gateway-based LabLog prototype is conceptually shown in Fig. 5. Some sensors/actuators may already be embedded in an SSG, and external sensors (e.g. Phidgets [25]) or actuators can be connected to an SSG via either wired I/O ports or wireless communications. SSGs can be distributed somewhere inside the space such as on the floor, wall, table and door, and they can also be carried by moveable robots and human users. To be flexible in system deployment, all SSGs are equipped with some wireless
network (e.g. BlueTooth or ZigBee) for their inter-communications between the SSGs and the related SSS that runs on a computer.

So far, the LabLog prototype has been implemented using Java programming language. Its SSS can run on an ordinary PC installed with a Java VM using JSE 1.60_07 on Windows OS environment. And its SSG runs on SunSPOT (Sun Small Programmable Object Technology), a small device developed by Sun Microsystems [26].

The SunSPOT has been embedded with an acceleration sensor, a light sensor, a temperature sensor and several three colored LED actuators, and also provided a set of GPIO ports for connections to other external sensors and actuators as shown in the left of Fig. 6. As shown in the upper right of Fig. 6, the SunSPOT offers a compact Java virtual machine called Squawk [27] on which a MIDlet can run directly. The MIDlet is a compact Java application used for Java-enabled cell phones and other small information end devices. Actually, our SSG has been developed as one MIDlet. The bottom-right of Fig. 6 shows the SunSPOT network stack that adopts low power IEEE802.15.4 for its physical and MAC layers, on which a lowpan (low power PAN) layer has been implemented. The SunSPOT provides two basic data transmission schemes: Radiogram and Radiostream, which function similarly as the Internet UDP and TCP protocols.

The storage service is to provide necessary APIs for an application SSA to set up a database, manage sensed data storage and retrieval of demanded data records for various purposes. Different from conventional data, the sensed data or spacelog is closely associated with the location and corresponding time in sensing the data. Therefore, the smart space database is a typical spatial-temporal one in which the associated spatial and temporal attributes should be added to every sensed data record. The spatial attribute can be the position information of an SSG that may be fixed or moveable. The temporal attribute is usually the sensing instance based on some synchronous timing scheme between multiple SSGs and their SSS. The popular RDBMS MySQL 5.0 has been adopted as one type of database in LabLog and the following database operation APIs are provided: setupDatabase, createTable (table_name, attributes), deleteTable (table_name), writeData (table_name, record), readData (table_name, record), and executeSQL (SQL_code). When an SSS runs, a GUI will appear on the host computer as shown in Fig. 7. The GUI provides the necessary system information to the developer or administrator including a
list of SSGs discovered in the smart space and all services working on the SSS. The embedded application (SSAe) is also manipulated via this GUI.

To test the whole system of Lablog prototype, an experiment environment has been set up with a set of SSGs distributed in our laboratory for making a spacelog application to record the sensed data in the laboratory as shown in Fig. 8.

As shown in Fig. 8, SSGs are placed on the door (three of them), on the robot and table, and others on the partition boards in the room. The experiment was done in 24 hours (15:00, Jan. 7, 2009 – 15:00, Jan. 8, 2009). The SunSPOT’s battery can last up to about 50 hours when it is idle; but during our experiment, it only lasted for about 21-23 hours. This is because of the large power consumed when running the SSG software and conducting wireless communications.

Fig. 9 shows the sensed raw data of acceleration, temperature and light from the sensors embedded into SSGs that were put on the door and the robot during the experimental period. It can be seen from the figure that the temperature varies in different locations of the room during the day, even though the air conditioner in the room was work
ing during the whole period. The sensed light data from the robot indicated whether the room was in use or not, while the sensed acceleration data on the door showed how often people have entered/left the room in a day. We can combine the sensed light and the acceleration data from the door to know that if someone entered the room for a short time.

5. HOMELOG ACQUISITIONS AND ACTIVITY SUMMARIZATIONS

HomeLog, as one kind of spacelog, is all about records related to a home. Our HomeLog prototype is designed to extract meaningful data from a large amount of acquired homelog data. Our focus is to know what a family member did in a certain period of time, especially for letting parents know their kid’s activities after school. Its objective is to automatically give a summary on the kid’s sequence of activities such as studying, doing homework, watching TV, using computer, playing game, reading a comic book, cleaning, doing some dangerous thing, etc. Fig. 10 gives an example layout of a home and the placement of sensing devices. Different from the previous LabLog which is for a single large room, the HomeLog system covers multiple relatively small rooms with different functions, e.g. living, dinning, studying, sleeping, etc.
In order to know what a kid is doing, various devices such as cameras, sensors, RFID tags and readers, robots and so on shall be installed at appropriate places in a home. Besides the above devices, there are other devices/machines used in HomeLog, such as a homelog database (HDB) to keep the acquired data, a smart home server (SHS) to manage devices and logs, and a PC/PDA or a cell phone for a parent to get information about his/her kid. Fig. 11 gives an outline of the HomeLog prototype workflow from data acquisition to kid’s activity summarization. The homelog service-ware includes pre-processing services, post-processing services, and access services.

The pre-processing is to turn incoming raw data into information for storing in a homelog database, further processing is the post-processing, or the direct use of a homelog application. For example, the collected camera sensor data can be filtered into different types of attribute-oriented information and stored in a homelog database for later use in applications.

The post-processing is to search, analyze, and summarize information from the homelog data into meaningful contexts for supporting homelog applications. This research takes an agent based post-processing approach in which multiple agents play their own roles and provide corresponding services in the process of turning information into meaningful contexts in applications.

The image data from the cameras and text data from the sensing devices with time stamps are stored in the homelog database under the hierarchical structure of the kid log’s id, each room’s id, and each device’s id. The search agent, analysis agent and summarization agent resided in the homelog service-ware are designated to provide related services.

The search agent takes a query request such as

| A query: kid, daily, life pattern, at home |
|---|---|---|---|
| (WHO) (WHEN) (WHAT) (WHERE) |

as its input. This request is explained as making a retrieval of the kid’s daily life pattern at home. The search agent queries for the set of data from the homelog database according to the request. The output is a set of data, which includes W4 (WHO, WHEN, WHAT, WHERE) associated but may be independent data.
The analysis agent takes the retrieved kid’s log data set as its input, analyzes the data set, and finds the associations and the correlations among the data set with reference to a particular event or activity along the time sequence axis with the temporal synchronization and the physical location constraint. As a result, it outputs a topological graph, which contains data nodes and links. In particular, each link is an m-relations tubular link, which explicitly expresses the relations of two nodes of data, exclusion, intercross, intersection, inclusion, equality, synchronization, or time sequence.

For example, the acquired data (5 data sets) from various sensors in the study room are queried, extracted, and expressed in Fig. 12. Each data set is a data node in the topological graph. To these, we can conclude that (1) the kid was studying (English) in the study room from the intersection of the data nodes 1, 2, and 3; (2) the kid switched from studying to reading a comic book for a break from the intercross of data nodes 3 and 4; (3) the kid left the study room from the data node 1.

The summarization agent takes the created topological graph as its input, makes a set of data with their relations meaningful to a specific objective by heuristic approach, and outputs a summarization as a service provided to the parents, as shown in the example given in Fig. 13. The main points of the heuristic based summarization are to model regular daily activities, create the corresponding heuristic graph examples, loosely match one of heuristic graph to a part of the created graph, and output the matched activities.

Of course, the homelog provides various summarization services to the kid’s parents. The parents can always confirm or verify the kid’s activities by requesting video camera replay. From the summarizations, the system may give a warning or advice to the parents, for instance, the kid is playing video game at midnight, and the kid has not studied at all today. Especially, it is helpful to report any potential problems of the kid like: bad habit – reading comic book in the toilet; ill behavior – drinking alcohol; dangerous action – visiting prohibited web site; and potential addiction – playing game all the time. These are important means to prevent the kid from falling into dangerous situations and correcting the kid’s unwanted behavior at an early stage.
6. CONCLUDING REMARKS

Spacelog is a novel concept which we proposed recently [4] and it can be regarded as a context history database or memory organ of a computerized intelligent real spatial site or environment such as a laboratory and a home. To deal with the great heterogeneity in devices, networks and log data, we proposed a gateway-based system architecture which is very flexible in supporting a variety of spacelog acquisitions, administrations and applications. The core of the architecture is a two-level middleware, which consists of a set of smart space gateways (SSG) and a smart space server (SSS). That is, the device and network heterogeneities are dealt with by the SSGs, which are scalable and managed in a unified mechanism by the SSS.

Although the experiment results in the two prototypes LabLog and HomeLog have partially shown the feasibility and advantages of the gateway-based platform, many problems are left. Our next work is to implement SSG on other devices in addition to SunSPOT, find effective mechanisms to reduce SSG energy consumption, add the security management for accesses and utilizations of spacelog. Only preliminary research has been done in LabLog and HomeLog for the spacelog retrievals and activity summarizations, which are actually very hard and needs further studies in the future. It is also necessary to apply spacelog to a variety of real environments from small to large ones for different kinds of applications.

Spacelog is still new and many research issues remain in the system-level study which covers modeling, techniques, systems and applications. The spacelog looks sharing some common issues in other areas such as database, multimedia, sensor networks, pattern/activity recognition, AI, agent, data mining, etc. However, the question is that many issues in spacelog cannot be all solved by simply applying available techniques in these areas. Therefore, new approaches and techniques must be put forward to deal with the new features and solve the new problems that exist specifically in the spacelog, as well as its related systems and applications.

REFERENCES

24. T. Kawashima, J. Ma, B. O. Apduhan, R. Huang, and C. Rong, “A system prototype with multiple robots for finding u-objects in a smart space,” in *Proceedings of the


Jianhua Ma received his B.S. and M.S. degrees from the National University of Defense Technology (NUDT), China, in 1982 and 1985, respectively, and the Ph.D. degree in Information Engineering from Xidian University in 1990. He is now a Professor at the Faculty of Computer and Information Sciences in Hosei University since 2000. Previously, he had 16 years of teaching/research experiences at NUDT, Xidian University, and at the University of Aizu (Japan). His research interest includes networks, multimedia, agents, ubiquitous computing and intelligence.

Runhe Huang received her B.Sc. from National University of Defence Technology (NUDT), China, in 1982, and her Ph.D. in Computer Science and Mathematics from University of the West of England, UK, in 1993. She worked at NUDT in 1982-1988. She also had work at the University of Aizu for 7 years. She is now a Professor at Hosei University, Japan. Her research fields include CSCW, artificial intelligent applications, multi-agents systems, ubiquitous computing, and mobile computing.

Tomomi Kawashima received his B.S. and M.S. degrees in Computer and Information Sciences from Hosei University, Japan, in 2004 and 2009, respectively. He had 3 years experience as a software engineer at Silex Technology, Inc., and is now the software development manager at Digital Amuse Co., Ltd. In his M.S. studies, he developed a P2P service middleware called TOMSCOP over JXTA. His research interest includes network protocols, P2P communications, robot-based ubiquitous services, and 3D display with naked eye.

Bernady O. Apduhan received his B.S.E.C.E. degree from MSU-Iligan Institute of Technology (MSU-IIT), Philippines. He studied M.S.E.E. at the University of the Philippines-Diliman. He received his M.S. and Ph.D. degrees in Computer Science from Kyushu Institute of Technology (KIT), Japan. He held positions at MSU-IIT and KIT, and is currently a Professor at the Faculty on Information Science, Kyushu Sangyo University, Japan. His current research interests include grid/cloud computing, ubiquitous computing, mobile computing, and future information systems/applications.