DAILY CONTEXT-ADAPTIVE PRESENTATION DRIVEN BY PERSONAL DATA STORE

Hiromu Ogawa, Kinji Matsumura and Hiroshi Fujisawa

NHK (Japan Broadcasting Corporation), Japan

ABSTRACT

Smart environments are expected to enable broadcasters to develop "context-adaptive content presentation" technology, through which heterogeneous smart and Internet-of-Things (IoT) devices in the environment work together to autonomously select and present appropriate content according to a user's context. However, it is impractical for a single broadcaster to implement all the functions in heterogeneous user environments. Therefore, a loosely coupled mechanism that allows devices and software developed by different manufacturers and service providers to work flexibly in various combinations is required. Thus, we designed a system architecture that realizes content presentation based on the user's moment-to-moment situation by utilizing context recognition and a personal data store. We prototyped the system and confirmed its feasibility even if the context estimation and content presentation modules were constructed independently. The result demonstrates the scalability of our system architecture and the potential impact of adaptive media in the smart era.

INTRODUCTION

The Internet has facilitated people's ability to choose and access information fit to their preferences based on a wealth of available data. However, individuals have limited time to access information in their daily lives, meaning that access to information of relatively low-interest decreases as access to information of high-interest increases. Therefore, certain users may have less access to information that should be widely shared, such as socially relevant public and political topics, resulting in unequal levels of knowledge in relation to those with more access (1). This can lead to a split in public opinion. Broadcasters primarily transmit various information through broadcasting. However, owing to changes in people’s information contact styles, it has become difficult to disseminate such information sufficiently through broadcasting alone. Therefore, we are exploring ways to extend the broadcasting service platform and boost the convenience of the services, thereby increasing the opportunities to present information to users.

One possible approach is smart technologies for living environments using the Internet-of-Things (IoT) and smart devices. Efforts are underway across industries to create smart cities, homes, and other living environments (2). These smart environments are expected to enable users to receive services in a personalized manner in their current context with minimal efforts. If our living environment changes in such a way, broadcasters can make their services readily accessible to broader users than today. For example, suppose a user watches a video on TV in the living room but interrupts viewing and moves to another room. If the smart environment detects the user's movement and automatically continues the video on the device in a new location, the user is encouraged to continue watching it. In another example, a user performs daily activities (e.g., brushing their teeth) and feels bored. By detecting such user requests and presenting content (e.g., the latest news) through a nearby device (e.g., a smart mirror or communication robot), the system can motivate the user to watch the content while performing daily activities (3). It contributes to the broadcaster's mission of delivering information about the public interest to the audience.

The difficulty of constructing a complex system is a challenge for broadcasters in realizing context-adaptive content presentation. The system requires several complicated procedures: detection of events based on IoT sensor data, estimation of the current situation, selection of an appropriate presentation method, and presentation through the coordinated control of smart/IoT devices. Implementing all the functions in heterogeneous user environments is impractical for a single broadcaster. Therefore, a loosely coupled mechanism that allows devices and software developed by different manufacturers and service providers to work flexibly in various combinations is required.

In this study, we have designed a “context-adaptive content presentation” system architecture to realize content presentation according to the user's moment-to-moment situation by utilizing knowledge-driven context recognition and a personal data store (PDS). The architecture realizes context-adaptive presentation, even if the context estimation and content presentation modules are constructed independently.

The context-adaptive approach to content presentation proposed in this study was developed as one of the technologies comprising web-based broadcast media, which we are exploring as part of this research. Web-based Broadcast Media is a new media realized by combining broadcasting with the web. In designing the architecture of the context-adaptive presentation system, we considered consistency with a set of technologies that we are developing for web-based broadcasting media. This study also provides an overview of web-based broadcast media and their technology groups.

The remainder of this paper is organized as follows: Section 2 provides an overview of web-based broadcast media and their technologies. Section 3 describes the requirements for a context-adaptive presentation system. Section 4 describes the structure of our proposed context-adaptive presentation system. Section 5 describes the implementation of a typical use case of a context-adaptive presentation in a smart home. Additionally, the evaluation of the proposed system is described. Finally, Section 6 presents our conclusions and discusses future work.

WEB-BASED BROADCAST MEDIA

What is Web-Based Broadcast Media?

With the spread of the Internet, the variety and volume of information that people can access have become enormous, and the means of accessing this information have become more diverse and complex. Consequently, some users feel burdened to create opportunities to access information provided by broadcasters, which is thought to result in them losing access to such information.

We are attempting to develop a means for broadcasters to continue fulfilling their role as part of society's information infrastructure, even when the Internet plays a primary role in information access. We aim to realize a user-centric media experience in which appropriate content is provided on appropriate devices in a format that matches the user's individuality in any situation. User-centric media is expected to reduce the burden on users when they access information. Therefore, if broadcasters can provide users with user-centric media services, the perceived burden on users when accessing information will be reduced, preventing them from abandoning their access to that information due to the burden. Consequently, delivering information to a significant number of users is expected.

However, independently developing and providing information services for all situations in life has become extremely difficult for individual broadcasters.

To meet these challenges, we propose web-based broadcast media and are developing this framework as a new form of media that combines broadcasting and web technologies. This new approach was designed to deliver information owned by broadcasters to users across various environments. The web provides a unified application platform that is available in many different environments, and services designed to operate in various contexts are being developed based on device- and OS-agnostic web technologies. In addition, many applications developed on the web are highly reusable, and new technologies and services can be built by combining the technologies, services, and data provided by different developers. Therefore, by combining broadcasting and web technologies, broadcasters can make their content available across a wider environment than can be achieved using conventional broadcasting services alone. This could involve either incorporating web functionality into broadcasting services or creating web services that incorporate broadcasting technologies. We are currently researching and developing several technologies for web-based broadcast media. These technologies can be classified into three broad categories (Figure 1): viewing application technology, content and data processing technology, and cloud-native content distribution technology.

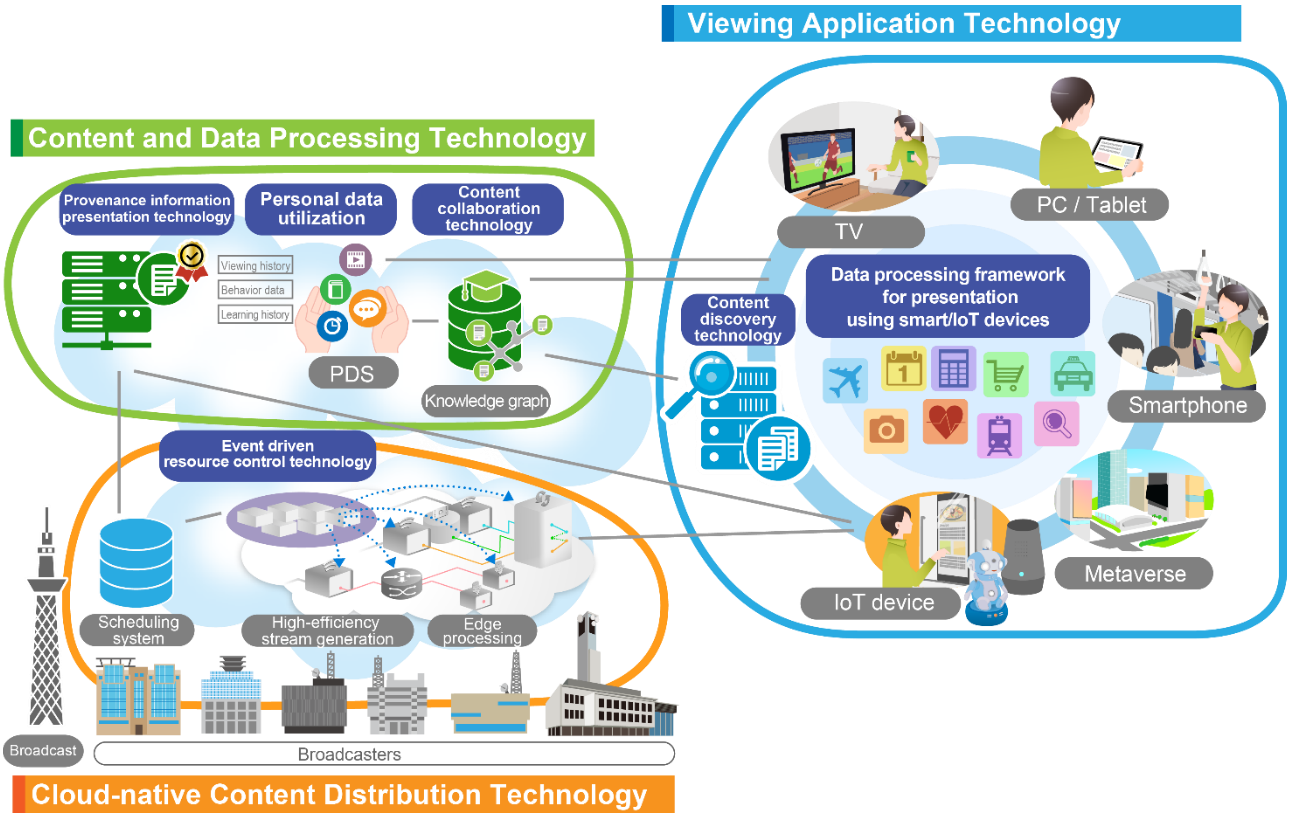

Figure 1–Technology configuration of Web-based media

As discussed in this paper, context-adaptive presentation is one of the service forms that web-based broadcast media aim to realize. In our research and development of web-based broadcast media, we studied the following technologies that are closely related to context-adaptive presentation.

Data Processing Framework for Presentation Using Smart/IoT Devices

The first is a data processing framework for presentations using smart/IoT devices (4). This framework helps select and present context-appropriate content from providers and smart/IoT devices with different output capabilities in the user's vicinity by treating the metadata of the elements involved in the media experience.

In this framework, the elements involved in a presentation are divided into three categories: people, content, and environment (including devices). Examples of data classified under "people" include the user's broadcast program viewing history, web video viewing history, and profile. Examples of data classified as "content" include content titles, formats (video, audio), and information from where the resource was obtained. Examples of data classified as "environment" include the names of available devices in the user's surroundings and their functions. These three types of data are recorded as machine-readable structured data using semantic web technology (5).

Developers can achieve autonomous content presentation using appropriate devices and suitable presentation methods by developing applications within this framework.

User-centered Personal Data Management and Utilization Model in Broadcasting Services

The second is a user-centered personal data management and utilization model for broadcasting services. We examined the broadcast service based on PDS (6). A PDS is a mechanism through which individuals proactively manage and control their own data.

A key advantage of a PDS is its high interoperability between services based on the user’s personal settings. Typically, data from services offered by different providers are collected and managed separately by each provider. This prevents the data from being used across different services according to user preferences. In contrast, in the user-centered management model considered here, data for all services are aggregated in each user’s PDS, and users can manage the data associated with their accounts. This approach enables the framework to utilize data from one service for the provision of other services based on user choices.

This mechanism enables a service to obtain the personal data of a targeted user, such as a user profile and viewing history, which that service would not know alone. Services can be improved suited to the user's needs and situation based on this information.

Remaining Problems

As described above, some technologies that contribute to the realization of context-adaptive presentations have already been developed in the process of web-based broadcast media research and development. However, though these technologies provide a mechanism for presentation based on context data when given, they do not provide a specific mechanism for obtaining context in a smart environment. Hence, we have to explore specific methods for acquiring user context in a smart environment. Therefore, we will begin to examine this method in the next section.

ISSUES AND REQUIREMENTS FOR DAILY CONTEXT-ADAPTIVE PRESENTATION

This section summarizes the issues that service providers, including broadcasters, must solve to provide context-based content presentations in a smart environment. Next, the functional requirements of the system are described. In this study, we focused on two issues.

Issue A: Difficulty in Context Estimation by Service Providers

The general process for executing context-dependent services in a smart environment is as follows: (a) Estimate the context based on sensing results from multiple IoT/smart sensors. (b) Determine the services (content to be presented and presentation methods) based on the estimated context. (c) Control of IoT/smart devices to execute an action. For installing IoT/smart sensors, it is more practical for a service provider to obtain information from a group of sensors installed independently of the service and acquire context rather than installing a group of sensors dedicated to the service in each user's environment. However, context estimation requires information on the installation status of sensors for each user, such as the sensor type and location, in addition to sensor sensing values, and it is necessary to train a context estimator for each user's environment. Service providers find it difficult to implement complex context estimators. Therefore, the following functional requirement is set (A): service providers should be able to obtain the estimated value of the context without estimating the context themselves.

Issue B: Coordination with Various Services

The UX of context-adaptive presentation can be improved by coordinating services such as delivering information that the user has not been exposed to in broadcasting or other services through context-adaptive presentation. To realize such coordination, it is necessary to simultaneously reference the user's context log and the usage logs of other services. Therefore, the following functional requirement was set (B): the system must be able to uniformly handle context and usage logs of other services.

ARCHITECTURE OF DAILY CONTEXT-ADAPTIVE PRESENTATION SYSTEM

Based on the arguments in the previous section, we propose an architecture for a context-adaptive content presentation system, as illustrated in Figure 2.

Overview

The system consists of IoT/smart sensors and content presentation devices. Additionally, it uses several functions implemented in the cloud and PDS. The primary function of the system is to estimate the user's context based on sensor values and determine the content to be presented and its method by referring to the logs of the estimated context and those of the use of other services.

Context is divided into two categories, lower-order context, and higher-order context. Lower-order context is defined as a context that can be directly estimated from sensor values. For example, "walking" or "turning on the coffee machine" are included. Hereafter, lower-order contexts will be referred to as "events. On the other hand, higher-order context is defined as a context consisting of a sequence of multiple lower-order contexts. For example, "taking a break" is included. Hereafter, higher-order contexts will be referred to as "context." The values of events are defined in an event ontology. The relationship between events and contexts is also defined in a context ontology.

The process of the context-adaptive presentation can be broadly divided into three steps: (a) Sensors detect events from sensor values, (b) a context estimate section estimates context based on a series of events, and (c) a content presentation application decides a content presentation method based on context. Each functional section exchanges data via PDS. Event and context data are structured in a machine-readable manner based on event and data ontologies. Therefore, each functional part obtains a common understanding of the data to be exchanged.

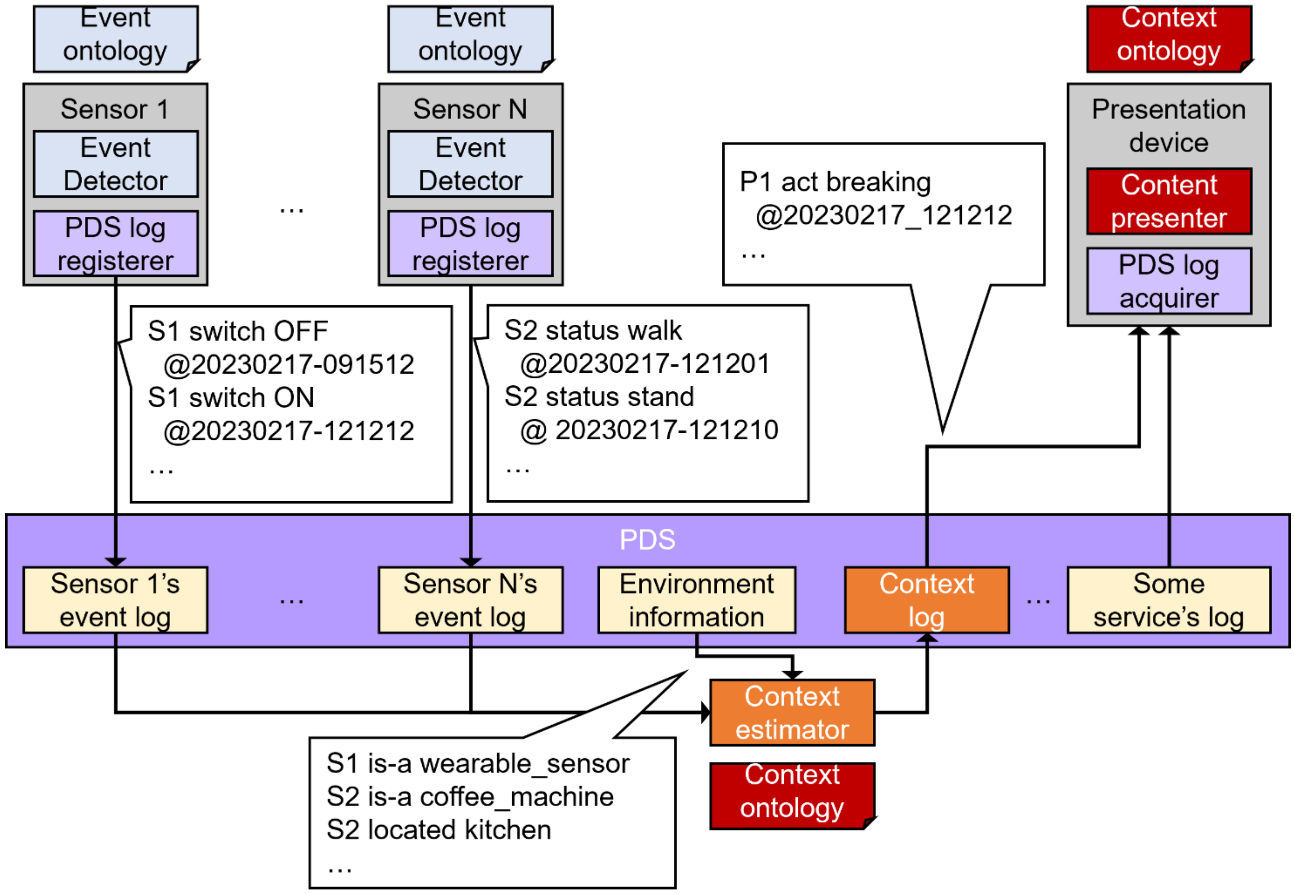

Figure 2–Proposed system architecture

System Configuration

The names and roles of the functional parts of the system are as follows.

- Sensors: Sense the environment, including user behavior.
- Event detector: This detects the corresponding lower-order contexts based on sensor values. These units detected events based on the vocabulary included in the event ontology.
- PDS log register: Structures the events detected by the event detector according to the event ontology and registers them in the PDS (The callout shows a rough image of the data to be registered in the PDS by the PDS log registrar. In reality, the data exchanged here are written in semantic web notations such as JSON-LD; however, the details are omitted in the figure.)
- PDS: Maintains a log of events detected by the sensors. It also stores the context logs estimated by the context estimator and the usage logs of other services. It is assumed that each user has a PDS, and the logs of the various sensors and services used by the user are aggregated.
- Context estimator: Aggregates logs of events detected by each sensor and environment information. Environment information is supplemental information about the installation status of the sensor, including the type of sensor and its location. Then, it estimates the currently executed higher-order context by referring to a series of events and the context ontology. The detected contexts are structured and registered in the PDS (the callout shows an approximate image of the context data).

- Presentation devices: Devices that present content, have a web browser, and can run web applications.
- PDS log acquirer: Monitors context logs for updates and acquires context and usage logs for other services specified in advance from the PDS when there is an update. It is assumed that the context and other service usage logs are structured.
- Content presenter: Determines the content to be presented and the presentation method based on the context and other service usage logs obtained by the PDS log acquirer. At this time, the context ontology is referenced to interpret the context logs. This decision process can be based on the data processing framework described in an earlier section (the web-based broadcast media section).

Addressing Functional Requirements

Response to requirement (A)
As mentioned earlier, we classified context into two stages: event (or lower-order context) and context (or higher-order context) (7,8), and described the relationship between these two stages as an external ontology. The event detector detects events based on the sensor values, whereas the context estimator estimates the context of the events. The content presenter decides how to present content based on the context and usage logs from other services. The input/output of each function module is set based on the event/context ontology. By loosely coupling each functional section in this manner, the content presenter can present the content according to the estimated context without information on the detailed processing of the context estimator. We assume that the content presenter is developed by the service provider, whereas the context estimator is developed by the IoT cloud service provider or housing business.

Response to requirement (B)
As mentioned previously, the PDS is configured to consolidate the context and usage logs of other services as structured data in a centralized manner. The PDS log acquisition part monitors updates to the context data in the PDS, and when there is an update, it acquires the context and usage logs of services that the user has specified or authorized in advance, and decides how to present the information. The structure of context logs follows a higher-order context ontology. In this manner, it is possible to make content presentation decisions considering the user's context and the usage logs of various services.

PROTOTYPING AND OPERATION VERIFICATION

To assess the validity of the design, a prototype of some of the functions of the proposed system architecture was built. Specifically, we assumed the typical service scenario described in an earlier section (INTRODUCTION).

Verification Scenario

Figure 3 illustrates the verification scenario. Suppose that Users A, B, and a friend of User A, are in their respective homes. User A has a TV in the living room, an IoT coffee machine, and a smart display in the kitchen. These TVs can register logs of the viewing context and content, and the IoT coffee machine can register switch operation logs for each PDS. A smart display can run a web application in a browser. User A wore an IoT sensor that could identify lower-order contexts, including walking. User B also had a TV in their living room, and User B permitted User A to view their TV viewing logs. The specific scenario flow is as follows. (S1) Initially, User A watches Program 1 in the living room. (S2) At this point, User A stops watching Program 1 and moves to the kitchen. (S3) Next, User A turns on the IoT coffee machine in the kitchen. (S4) Then, the presentation application running on the smart display in the kitchen automatically continues Program 1. (S5) Next, User B starts watching Program 2 on the TV at User B's home. (S6) Then, on the smart display in User A's home, the fact that User B had started watching Program 2 and a message encouraging the user to watch the program were displayed.

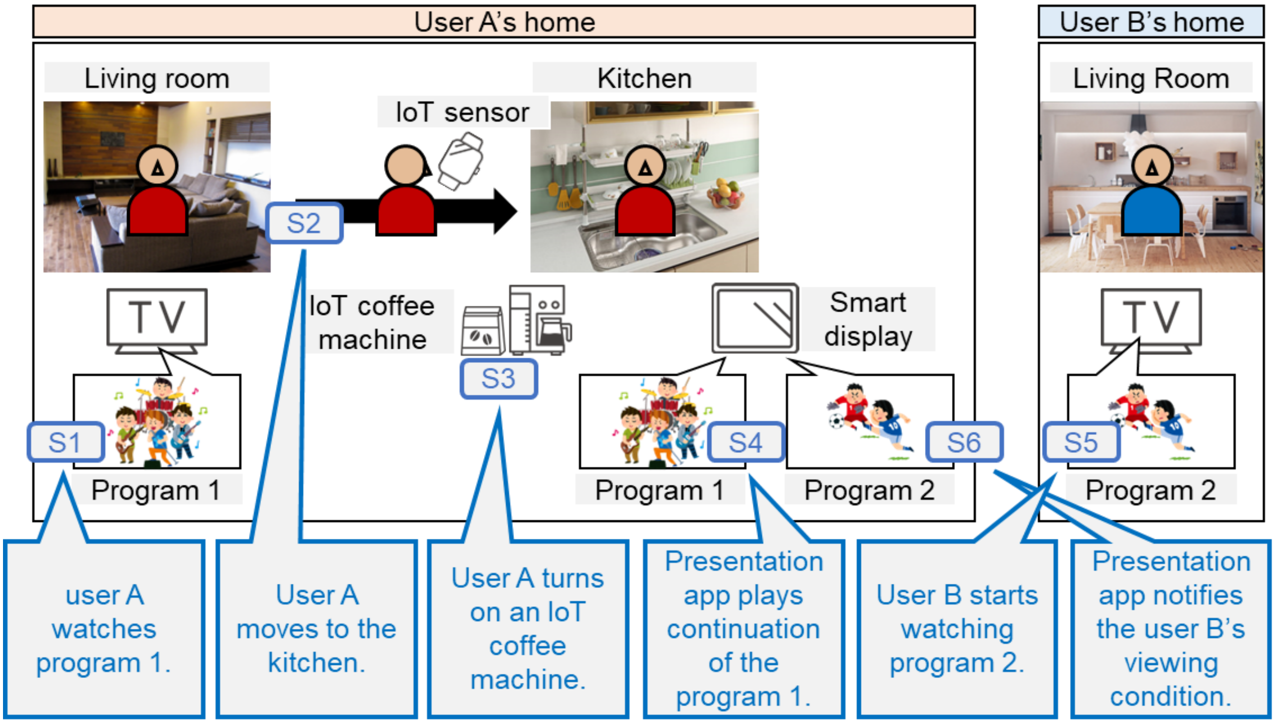

Figure 3–Verification scenario

Prototype Configuration

We implemented a process of exchanging data between each functional part using a PDS in the proposed system architecture. The content presenter was implemented as a web application and ran on a PC browser. The PDS server was implemented on the cloud based on the Solid (9) specification, and the PDS log registerer/acquirer was implemented using libraries developed by Inrupt (10).

Data Processing Flow

Figure 4 illustrates the prototype system configuration and the data processing flow of the system for the verification scenario described in the previous section. (S1) User A finishes watching TV in the living room and is about to move into the kitchen. At this time, it is assumed that the TV has written its viewing history to the viewing log store of User A's pod. It is also assumed that the event "TV turned on" is obtained by the context estimator via PDS. (S2) From this state, User A started moving into the kitchen. (S2-P1) Then, the IoT sensor detects the event that the user "moved" and writes this event to the event log file in the PDS. (S2-P2) Next, the PDS notifies the context estimator that the data have been written to the event log. (S2-P3) Next, the context estimator obtains the event log file from the PDS, and finds out that the event "move" has occurred. (S3) Next, User A arrives in the kitchen and turns on a coffee machine. (S3-P1) Then, the IoT coffee machine writes the event "coffee machine turned on" to the PDS, (S3-P2) the PDS notifies the context estimator, and (S3-P3) the context estimator acquires the event log file. (S3-P4) Next, the context estimator estimates that User A is about to take a "break" based on the sequence of events

"TV turned on," "moved," and "coffee machine turned on." (S3-P5) Next, the context estimator writes the estimation results to the context log file. (S3-P6) Next, the PDS notifies the smart display about the log file update. (S3-P7) Next, the smart display obtains the log file, and finds out that the current context is "taking a break." Subsequently, the presentation engine determines the appropriate content presentation method for the user during breaks. The selection of the presentation method depends on the implementation of the presentation engine. In this implementation, the user can choose to continue with the content that the user has watched. The presentation engine then retrieved the viewing log from the PDS, checked the time when the video was interrupted, and played the video continuously. When (S5) User B starts watching TV at user's home, a similar process occurs. The detail is omitted.

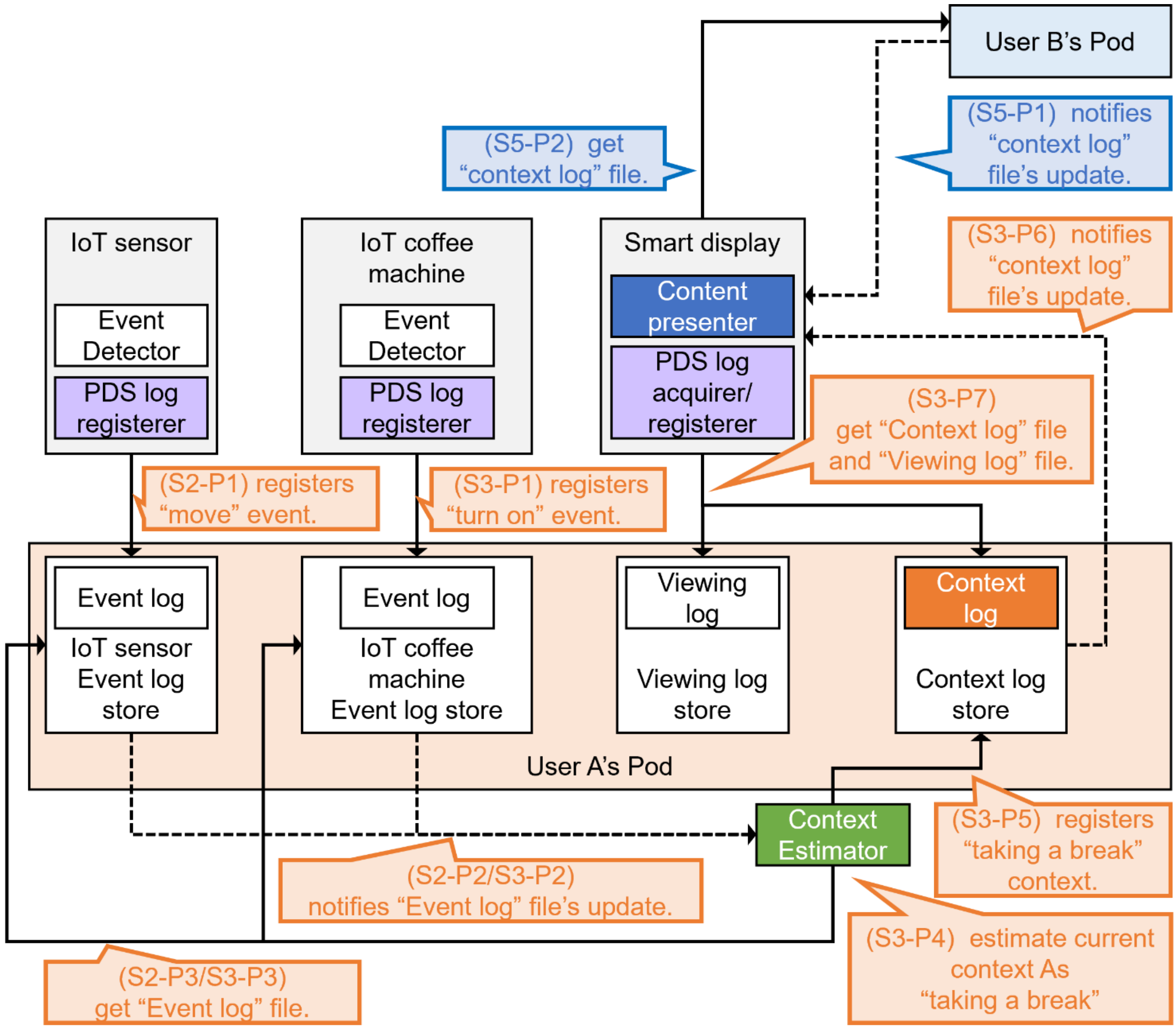

Figure 4–Prototype system configuration

Verification and Discussion

Based on the data processing flow described in the previous section, we checked whether program 1 is played on the smart display after the IoT coffee machine detects the event and after context estimation by the context estimator. As a result, we confirmed the validity of the data processing flow of the proposed system, as program 1 was played back from the time when the playback was interrupted by the TV.

In this paper, we have developed prototypes of low-order and high-order context ontologies based on existing ontologies such as (11). On the other hand, in order to improve the accuracy of the selection of content presentation methods according to context, it is necessary to clarify the relationship between context and desirable content presentation methods and design ontologies based on this relationship. Detailed design is a future issue.

CONCLUSION

This study provides an overview of web-based broadcast media was described first. Web-based Broadcast Media aims to realize a "user-centric media experience" in which individually optimized content is delivered to users in any environment. We then explained that broadcasting services could be expanded by combining web technology, which functions as an integrated application platform, with broadcasting technology.

Next, we propose a system architecture that enables broadcasters to achieve context-adaptive content presentation in a smart environment, which is required for web-based broadcasting media. The system utilizes knowledge of lower-level/higher-level contexts to enable broadcasters to realize context-adaptive presentations without performing complex context estimation. Subsequently, we prototyped a typical content presentation scenario and verified the proposed architecture. Thus, we demonstrated the possibility that service providers, including broadcasters, could realize context-adaptive services without installing a complex context estimation system in the user's environment by themselves.

We aim to design detailed low- and high-order context ontologies in the future. We investigated the relationship between context and appropriate content (3). We further investigate this relationship and reflect on it in the context of ontological design.

REFERENCES

1. P. Markus, 2007.Post-Broadcast Democracy: How Media Choice Increases Inequality in Political Involvement and Polarizes Elections. Cambridge University Press.
2. Cabinet Office, Government of Japan. Society 5.0. https://www8.cao.go.jp/cstp/english/society5_0/index.html
3. H. Ogawa and K. Matsumura. 2022. Home Routines Suitable for Incidental Exposure to News: Pilot Field Experiment. Ubicompiswc22adjunct: Proceedings of the 2022 ACM International Joint Conference on Pervasive and Ubiquitous Computing. September, pp. 94–96.
4. H. Endo et al., 2021. IOT-BASED MEDIA FRAMEWORK FOR PUBLIC SERVICE MEDIA: EXPANSION OF CURRENT DIGITAL BROADCASTING SYSTEMS. IBC technical paper. December. pp. 1 to 12.
5. W3C. Semantic Web. https://www.w3.org/standards/semanticweb/
6. Y. Yamakami, M. Ueno, and K. Matsumura. 2022. User-Centered Broadcasting Service Utilizing Personal Data Store. ACM International Conference on Interactive Media Experiences, Aveiro JB Portugal: ACM June, pp. 337 to 342.
7. L. Chen, C. D. Nugent, and H. Wang. 2012. A Knowledge-Driven Approach to Activity Recognition in Smart Homes, IEEE Transactions on Knowledge and Data Engineering. June, pp. 961–974.
8. D. N. Mekuria, P. Sernani, N. Falcionelli, and A. F. Dragoni. 2021. Smart home reasoning systems: a systematic literature review. Journal of Ambient Intelligence and Humanized Computing. April, pp. 4485 to 4502.

9. Solid. https://solidproject.org/TR/
10. Inrupt. Inrupt JavaScript Client Libraries. https://docs.inrupt.com/developer-tools/javascript/client-libraries/
11. S. Egami et al, S. Nishimura, and K. Fukuda, 2021. A Framework for Constructing and Augmenting Knowledge Graphs using Virtual Space: Towards Analysis of Daily Activities. IEEE 33rd International Conference on Tools with Artificial Intelligence (ICTAI), November, pp. 1226 to 1230.