Second Screen Displaying Life Logs and TV Contents Information

Manabu Motegi[1], Kensaku Komatsu[2], Yoichi Takashima[1], Toru Kobayashi[1]
NTT Cyber Solutions Laboratories, NTT Corporation[1]
NTT Communications[2]

1 Introduction

Recently, HTML5 has been attract rising attention. It will allow us to write applications that mirror native applications by using only Web standard skills such as HTML, JavaScript, and CSS. Moreover, Linked Data is an approach that discloses structured data on the Web. In the LOD(Linking Open Data) project, the data set published under an open license is converted into RDF(Resource Description Framework) and put on the Web. Furthermore, tablet PCs are rapidly being adopted and are reforming our image of mobile devices. Moreover, the life log service on the Cloud that accumulates various information has become widely used. However, a remaining problem is that there is no one method that permits easy access to the relevant information from the vast amount of information that is accumulating. Our solution is to extract personalized information without intervention for display on tablet PCs etc. As a result, information whose presentation is hard to achieve by intentional retrieval can be displayed.

2 Concept

Fig.1 shows our proposed concept.

The existing approaches to information retrieval general require the user to open a retrieval page on a PC, and to input the key word needed to direct retrieval. However, both actions are troublesome and represent a considerable barrier to users who have low IT skills. User who have high IT skills also wants to reach information with fewer actions. We examined the techniques by which retrieval could be made easier. Our focus was display of the user's life log information on his/her tablet PC.

In daily life, watching television programs is a key method of acquiring information. If there is a television at home, the preference of the user is reflected in the programs watched. We have developed a system that presents information related to watched television programs. As a result, information is extracted from the large life log and presented on networked appliances such as the digital photo frame, and the tablet PC. The user can access appropriate life log information in an appropriate place because information related to television content is pushed to these terminals.

The concept chart of Fig.1 shows three use cases.

In the first example, a family is watching a personal health program on TV. Life log information on each member's health is presented on their own tablet PC. For instance, when the program discusses the problems caused by a lack of physical activity, recent pedometer charts are presented on the father's tablet PC.

In the second example, a family is watching a travel information program on TV. If this family visited the sightseeing spot being introduced by the program, the family pictures are displayed in the digital photo frame. Moreover, photographs that all family members took

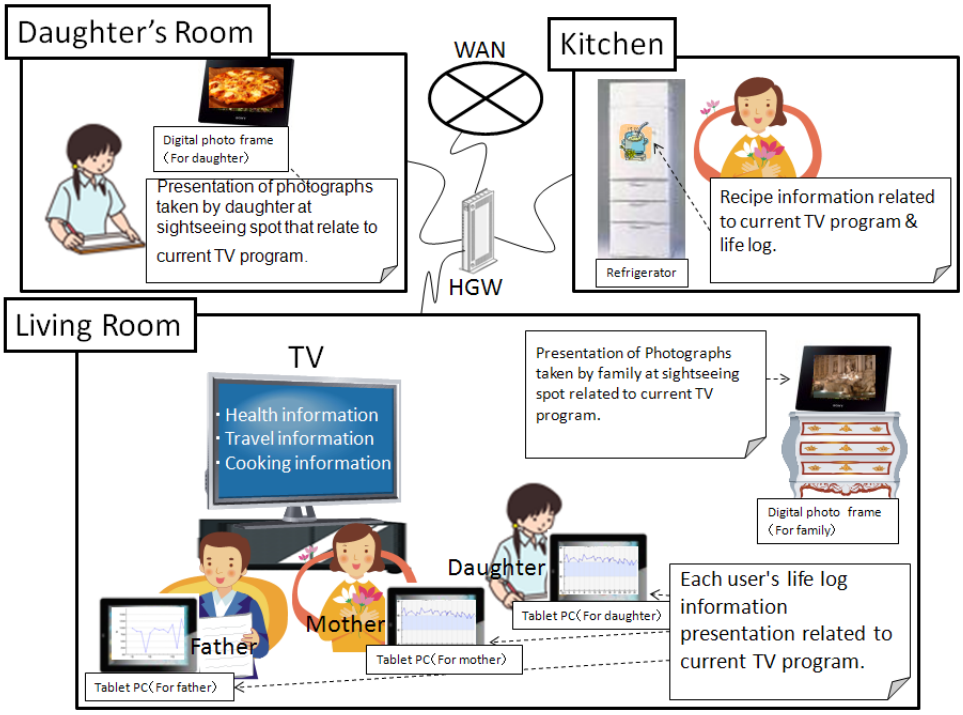

Figure 1: Concept.

are displayed in the photo frame in the living room. The photo frame in the daughter's room displays only the travel photographs that the daughter took.

In the third example, the mother is watching a cooking program on TV. When the mother enters the kitchen, recommended recipes will be displayed on the monitor attached to the refrigerator. The system considers the family's previous menus and the TV cooking program and then generates recommended recipes.

As mentioned above, the action of watching the television program triggers retrieval of associated life log information and its display on the appropriate terminal or terminals.

3 System Framework

Fig.2 shows the system framework examined herein.

Contents producers like TV stations and cable-TV providers transmit time-tagged meta data, such as WebVTT(Web Video Text Tracks), of the programs to the service provider. Moreover, the service provider collects a variety of life log information for user profiling. The information that is appropriate for an individual user is acquired by using the profile and the meta data of TV program information. General information is acquired from the LOD site, and life log information is acquired from the life log information storage service. We consider the case that the user has up-loaded a lot of information to the life log information storage service. The service provider personalizes information for the user, and pushes this information to the user's terminal using Websocket. The information is displayed on the user's terminal via CSS.

A prototype that realizes this framework is described below.

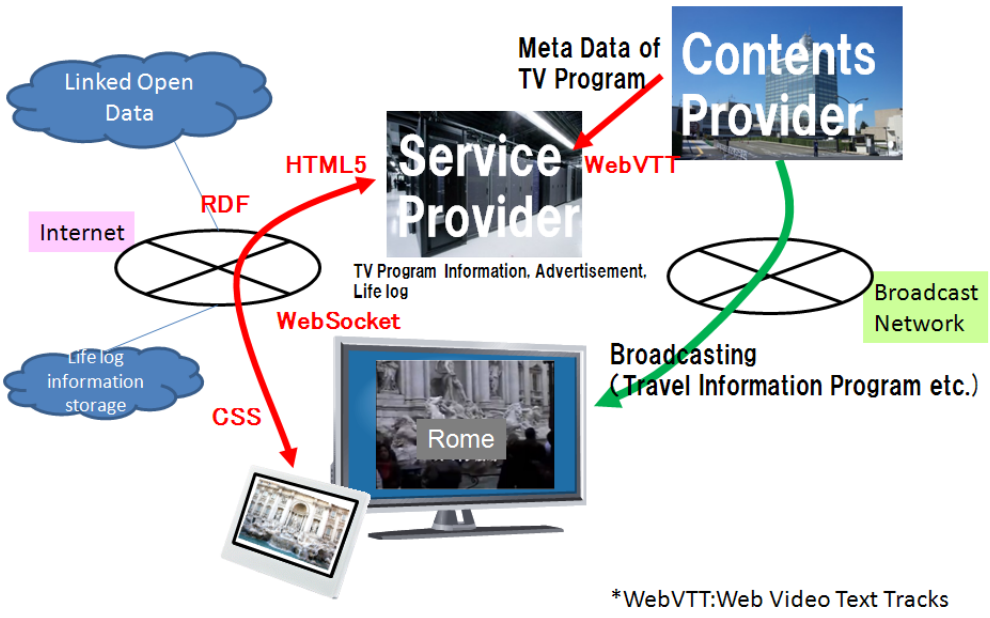

Figure 2: System Framework.

4 Prototype

Fig.3 shows the schematic of prototype system.

It was constructed to verify the above-mentioned functions. The second example described in Section 2 was the target of the prototype. The following assumptions were made.

- The photographs in the lifelog information storage service (taken by the user) are tagged with location information (latitude and longitude) and time taken.
- The contents provider is broadcasting a travel program on TV.
- The contents provider provides meta data about the travel program on TV to the service provider. The meta data links sightseeing-spot names to times of broadcast.

As shown in Fig.3, Apache is used as HttpServer, and node.js is used as the WebSocket server to push information to the tablet PC. Moreover, digital video contents are used instead of the television contents, and WebVTT is used as track data. As a browser that can display video contents, a Chrome build that can interpret WebVTT is used. As the terminal, an iPad was used, and Safari was used as a WebSocket browser. DBpedia on the Cloud was used to provide general knowledge, and Evernote was used as the life log information storage service. Comparing Fig.3 to Fig.2, HttpServer corresponds to the contents provider, and WebSocket Server corresponds to the service provider.

First, Chrome browser on the client accesses the URL that displays Video contents. Chrome reads the video contents and the meta data described by the WebVTT form (In ① of Fig.3). ”The display beginning time, the finish time, and sightseeing spot name ” are described in the metadata. Chrome extracts the sightseeing spot name from this meta data, and sends it to the WebSocket server (In ② of Fig.3). Here, the problem is determining the relation between the sightseeing spot name included in the meta data, and the photographs in the life log. This problem is solved by recourse to DBpedia. The WebSocket server transmits the SPARQL sentence to acquire the outline of the sightseeing spot and the location information from the sightseeing spot name. The results that are received (In ③, ④ of Fig.3) are used to retrieve the life log photographs taken at or around the location (In

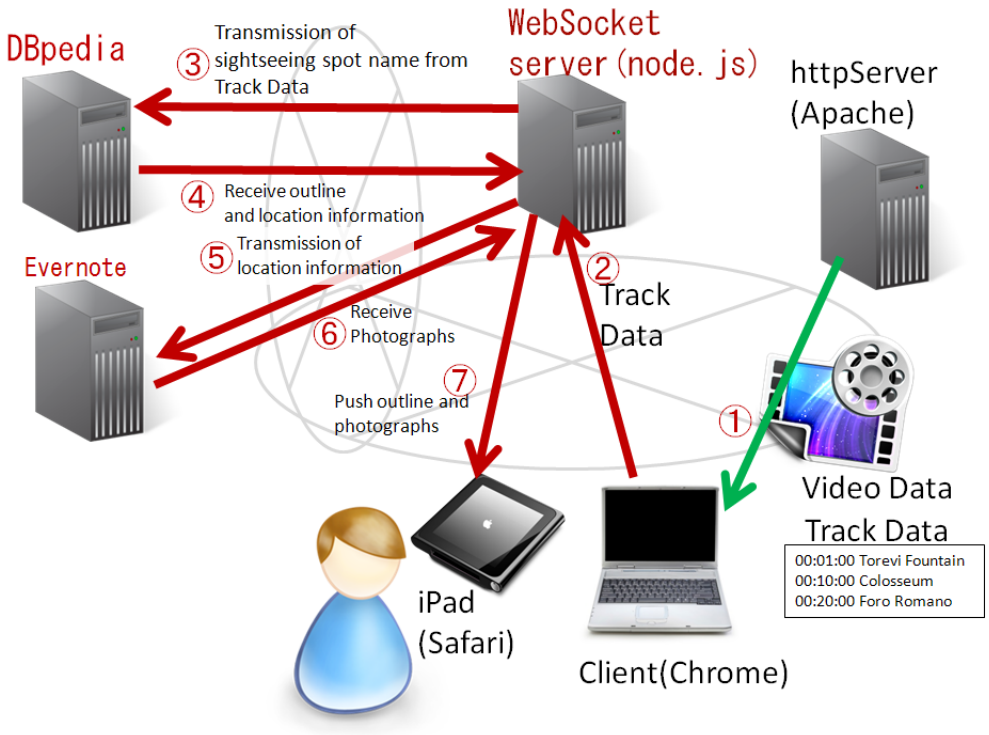

Figure 3: Schematic of Prototype System.

⑤, ⑥ of Fig.3) . The acquired information is pushed to Safari on iPad via Websocket (In ⑦ of Fig.3).

Since the prototype relates the television program contents to life log information and presents these related information, it realizes non-intentional information retrieval. This prototype used DBPedia to link television contents to the user's photographs. However, other approaches to realizing this linkage will be examined.

5 Conclusion

We examined techniques for presenting relevant information without intentional retrieval from life log data. Concretely, we paid attention to the television program being watched as the easiest way of acquiring the user's current interests. We showed the concept of presenting, on personal tablet PCs, the life log information related to the current TV program. A prototype was constructed by using standardized techniques: HTML5, LOD, and the life log storage service. This study used WebVTT as the program meta data description method. However, to promote the information presentation based on the watching program on TV, we believe that a standardized program meta data description method that can implement cooperation such as LOD and the life log storage service is necessary.