Empowering End-users to Find Point-of-interests with a Public Display

Tetsuo Yamabe Yasuyuki Washio Sota Matsuzawa Tatsuo Nakajima
Dept. of Computer Science, Waseda University
{yamabe, y-washio, sotam, tatsuo}@dcl.info.waseda.ac.jp

ABSTRACT

It is expected that public displays compensate the shortcomings of mobile web search. Public displays have a big advantage in its large display size and touch-based interaction for multi-player's use, but sometimes end-users such as elderly people do not have enough skill to make sufficient queries that derive interested information from the Internet. In this paper, we point out an important aspect of public displays: an interaction medium in social communication. Since the public display users are expected to have similar interests about the local area, search queries and results could be reused in a future search. Our approach aims at assisting end-users' point-of-interests search by providing the local contents, which are composed of the knowledge mashed up from web services and local search history. As a proof of concept, we developed a map-based tourist navigation service and performed preliminary experiments. In the experiments, we elaborately analyzed subjects' interaction process and figured out design issues for further improvements, such as search history presentation.

Categories and Subject Descriptors

H.3.5 [**Information Storage and Retrieval**]: Online Information Services—*web-based services*; H.5.2 [**Information Interfaces and Presentation**]: User Interfaces—*interaction styles, user-centered design*

General Terms

Design, Experimentation, Human Factors

Keywords

Public display, ecosystem, programming by demonstration, route navigation, mash-up.

1. INTRODUCTION

Web services on the Internet provide a variety of useful information for tourists, such as train timetables, maps, and local restaurants recommendation. Thanks to the progress of mobile computing technologies, today we can access the Internet and retrieve necessary information on demand - it allows us to visit unfamiliar places without a careful preliminary survey. Sometimes mobile devices are not suitable for complex search, however, because the size of mobile display is too small to show entire search results. Moreover, frequent interaction with tiny keypads frustrates a user, since it enforces them to allocate much cognitive resources for a task [10]. Even though they are skillful to manipulate the device, it is often time-consuming to interact with large-scale contents (e.g., maps).

Using interactive public displays (e.g., public display with a touchable surface) is an alternative approach to the mobile search. Usually a public display is larger than a desktop computer's display, so a user can look and easily recognize large-scale information at a glance. Unlike conventional bulletin boards, contents on a public display can be dynamically changed according to a user's context [2]. Public displays can be also networked with the Internet and a personal mobile device so that personalized contents can be sent to/from a public display. As shown in [4, 7], potential new services on the interactive public displays have been proposed with exploring the feasibility of service deployment in the real world. Moreover, modern cities equip public displays at the important points of transportation service nowadays, and the cost of displays has been getting cheaper. We believe that it is feasible to apply public displays as a platform of web search, in a great range of application domains.

One important issue on the application design is the diversity of users. Unlike personal computers and mobile devices, a public display is supposed to be used not only by skillful users, but also the novices who do not have sufficient technical literacy. Making a search query to derive an exact answer from such a flat and boundless Internet is quite difficult for them, since the search syntax is too flexible and the vast amount of data could be hit as a search result. Search process requires both knowledge and skill. Moreover, this issue cannot be completely resolved by replacing a mobile device with the large workspace on a public display. One possible approach to support a end-user is pushing the contents that the user might be interested in, instead of increasing the flexibility of search interface.

Another issue is the locality of information. Users on the go often want to get the information that relates to their context (e.g., time and place). For example, a digital signage on the train platform indicates train arrival time and its destination. Such information is mostly important to the peo-

©ACM, 2010. This is the author's version of the work. It is posted here by permission of ACM for your personal use. Not for redistribution. The definitive version was published in ICPS2010, July 13-15, 2010, Berlin, Germany

ple who are waiting for the train, and temporary consumed before the train arrivals - they will forget the information after leaving the platform. Usually a public display is stably located at a certain place as well, and temporal and local contents should be prioritized than static contents. Stable information can be widely accessed from portable computers anytime. However, nomadic users tend to have the niche interests (e.g., near pubs opening now, good photo shooting spots for current lightning condition), which are not available on the Internet.

Our approach aims at solving these two design issues with knowledge mush up and reusable search query mechanism. We focused on public displays' role in social communication, and designed an ecosystem to realize the query reuse. Public displays are installed into popular public spaces and act as a medium of communications among general public. It means that the trend of search can be analyzed from previous queries, and then optimized information can be offered to newly joined users. For example, if an end-user can use the search query that is previously made by a skillful user, it will drastically shorten the time for information search. Moreover, the search history will be useful for a future use as well, since the people around the public display are expected to have similar interests. Our approach enables to collect skillful users' knowledge and interests from the queries, and cluster its results in order to use a public display as an access point to the regional information.

As a proof of concept, we prototyped a tourist guide service for public displays. In the next section, we introduce the ecosystem with sample scenarios and explain how a public display's knowledge is enriched. In Section 3, the essential parts of the navigation service's system design are explained. We describe the detail of system implementation in Section 4, and show experimental study results in Section 5. Based on the experiments, lastly we conclude this paper with discussing the application design issues to improve our system in future work.

2. PUBLIC DISPLAYS IN AN ECOSYSTEM

In this paper, we chose tourist navigation as a service to demonstrate the ecosystem of public displays' contents. In this section, firstly we present a sample scenario, and after that we explain how the ecosystem works in detail.

2.1 Sample Scenario

2.1.1 Scene 1 (Finding a Restaurant)

User A was at station. He needed to attend a meeting at his office an hour later, and he wanted to have lunch shortly before that. He found a public display in the station, and started to look for good restaurants with the navigation service. Firstly he confirmed that the area nearby the station is shown as a map with pointing his current location in the center. On the right side of the map pane, genres of point-of-interests (POIs), such as restaurants, landmarks and shopping stores, were shown as a list. Then he touched the *"Restaurants"* button and shortly after listed items changed to show restaurant categories (e.g., cafe, pub, dining kitchen). He chose the *"Hamburger"* button, and also touched the *"Highly ranked but inexpensive"* button to screen out the results. He realized that the map indicated the restaurants far from the station, so the *"Within a 10 minutes' walk"* button was also selected and three restaurants remained in the map. He was still worried about their foods quality, so lastly touched the *"Reviews"* button to show other users' comments about the restaurants. As the navigation service gathered various types of information from the Internet and showed in a well understandable way, he could decide the restaurant for lunch without struggling with mobile web search.

These all buttons in this scene (e.g., *"Restaurants"*, *"Hamburger"*) are called *Filters* in our system, and they are used to screen out information on the map as the scenario mentioned. Filters have a hierarchy structure and available filters at a certain point change according to previously applied filters (e.g., the *"Hamburger"* button does not appear until the *"Restaurants"* filter is applied).

2.1.2 Scene 2 (Route Planning)

User B was on a trip and arrived at the station to see the sights of the town. She was completely a stranger there, but had not conducted enough preliminary survey since she was so busy before the trip. Shortly after, she found the public display and started to plan a tour route with the navigation service. Firstly she applied *"Sightseeing"* filter and continuously selected *"Museum"*, and also tried *"Sightseeing"* and *"Temple"* filter combinations to find POIs. Since she roughly knew the name of famous museum and historic temples, the filter based search helped to reach the exact location on the map. She wanted to check in the hotel after seeing the museum and temple, but no reservation was made at that moment. Then she started over map search with keeping the results of previous search (i.e., the museum and temple) as POI marks. The *"Accommodations"* filter was applied firstly, and *"Hotels"*, *"Hot spa"* filters followed after that. Then she chose one of the hotels appeared on the map and made a reservation on the phone.

Finally, three POIs were shown on the map: the museum, the temple and the hotel. Then she pushed *"Route"* button to find the routes to go around the town, which starts from the station. The routes allowed her to drop into both the museum and temple, and the hotel was destination. She transferred the route information into her mobile phone so that she can check the route while on the move.

2.1.3 Scene 3 (Reusing the Search History)

A few days after the user B's trip, user C happened to visit the town to attend a conference. Since by chance he could allocate time for sightseeing before the conference opening, he started to go around the town. Firstly he went to the station to gather information, and found the navigation service running on the public display. Then he started to use the service in order to grasp the town's overview.

When he applied *"Sightseeing"* filter, the navigation service showed the other users' search history as recommended routes. The user B's search results were also included in the history, and it met user C's interests, because he was roughly aware that the town was famous for ancient temples and its historic view. Since he had no idea about the exact name and place of famous temples, the history was helpful to initiate route planning. He checked the route's contents in detail and realized that there was a famous museum in the town. He was also interested in the museum, since lots of comments from previously visited users indicated the museum was must-to-see. He specified his hotel as the final destination, and made a route as user B did in the Scene 2. After he

left the town, he added his comments by accessing the web link that remains in his mobile phone. Since the navigation service allows remote access from conventional computers such as laptops and mobile phones, users can add information from their experiments anytime.

2.2 Knowledge Localization Process

As introduced in the sample scenario, we propose the ecosystem that works based on CGM (Consumer Generated Media) mechanism. CGM is an approach to compensate the gap between real consumers' needs and the preliminary provided contents that are made by service designers. Moreover, it also enriches the entire contents level, since users are motivated to compete with other users by creating attractive contents. This cycle evolves a web service quickly, but on the contrary creating media itself requires a certain time and effort to amateur creators.

In our system, users' POI search composes of the combination of filters, and search histories are stored in the service's database. This means that every user plays both roles of contents creator and consumer. Filter-based search allows users to unconsciously embody their vague knowledge about POIs in a reusable manner, without coercing elaborate data input and formatting process. On the other hand, a user can obtain other users' knowledge from the history shown on the map. Unlike conventional web search, this approach works effectively, since public display users at a certain place are expected to share similar interests. Figure 1 illustrates the interaction process in the ecosystem.

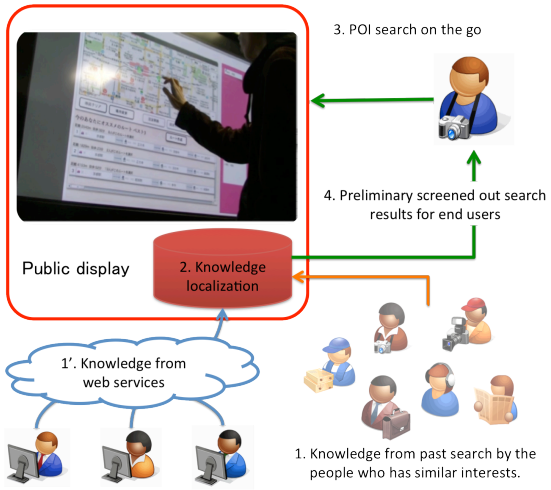

Figure 1: Public display as a knowledge collecting system in an ecosystem.

As the numbers in the figure indicate, the ecosystem roughly composes of four steps: 1. collect knowledge from search histories and web services, 2. rank locally interested knowledge from the trend of search on the public display, 3. interact with new users, 4. provide requested information with suggesting possible options from the localized knowledge.

The service accepts inputs not only from the remote users who have personal computers, but also from the users who directly interact with the service running on a public display. This mechanism allows the service to analyze the trend of interests at the area, with keeping the remote users as an information source. For example, conventional map services, such as Google Maps[1], do not take the locality of information into its system design. Thus every area is shown in a same way, even though each has an unique trend in regard to peoples' interests. Some cities might be famous for well-designed buildings, and other some might be nice place to go hiking. Peoples' interests will differ as place changes, so contents on the map should adapt to that's trend.

However, such contents localization process requires time and human resource cost. As aforementioned in the introduction, the trend often creates niche markets; and locally interested information tends to be temporary useful for smaller communities. For example, if the area has a great view from a mountain, photo-shooting spots will be a major interest among tourists. Moreover, additional information, such as current lightning condition and sample composition described by skillful photographers will be helpful to make a plan for those who are about to have a stroll. Thus our system aims at automating the process by reflecting the real users' interests, which are obtained from the public display installed at the area. Our approach also compensates the shortcomings of the locality in the web services' knowledge, since such information is created and consumed in the local interaction loop [1].

As aforementioned, a public display gathers information and screens out it in order to shorten an end-user's interaction process. Since a variety of people use a public display, applications running on the display should cover both skillful users and the novices who lack of technical literacy. Thus our approach supports two paths for information search: flexible manipulation for the former users and selection from limited options for the latter users. As seen in conventional map services, our system allows skillful users to manipulate the service; and its result (e.g., filter combination and searched contents) is saved for future reuse. Moreover, the record is ranked according to its usefulness (i.e., how often used for search), and thus the search history that matches the local interests will remain in the end. Technically influent users will use the remained search results as the other path. The highly ranked search results are helpful to reach the localized knowledge, since sometimes it is hard for the users to appropriately combine filters.

3. SYSTEM DESIGN

In this section, we break down the ecosystem into core features and explain technical points.

3.1 Filter-based Search

As explained in Section 2, our system applies filter-based search. Filters have a metaphor for extraction, and they will remind users to find interesting contents from a map. Since a public display can show a large map, the amount of contents could be enormous. Users need to screen out them step-by-step with specifying category, genre, and etc. Filters also have a metaphor for combination, thus we applied this filter-based search for the first prototype.

In contrast, most conventional map applications adopt two different approaches: text search and tree-based search. The text search requests a user to input keywords or an address to find an interesting place. This approach is highly flexible, but sometimes it is difficult for an end-user to devise the keywords. It could be also considered that users feel

[1]http://code.google.com/intl/ja/apis/maps/

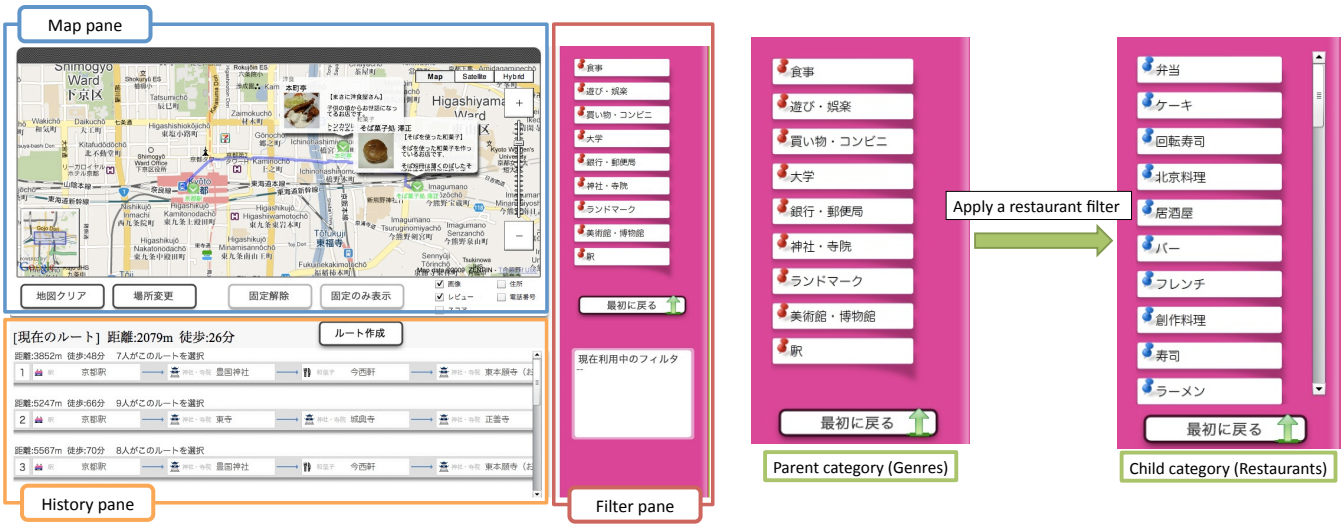

Figure 2: The tourist navigation service running on a public display. The user interface composes of three main panes: map pane, filter pane and history pane (left side). Item list in the filter pane changes according to a user's choice (right side).

annoying when the service returns unexpected results due to inappropriate keywords. Moreover, text input forces a user to type keys to organize a word, and as a result the user needs to keep their hands raised for contentious interaction. This is an important design issue for public display applications, if the service does not support additional devices (e.g., keyboard, mobile devices) for text input.

The tree-based search composes contents into clusters based on its attributes (e.g., category). Each cluster is linked with others, and as a result a tree-based structure is organized. A user can follow the link between the nodes (i.e., clusters) to find search results. As the filter-based search realizes a stepwise search, with this tree-based search a user can gradually get close to interesting information. However, the tree-based search requires lots of steps and it is difficult to predict results until reaching to the deepest nodes.

The filter-based search is less flexible than text input, but increasing the variety of filter can solve this problem. Moreover, there is the trade-off between the flexibility and the universality (i.e., allow end-users to use the service without sufficient knowledge). Compared with the tree-based search, the filter-based search shows its search results on a map immediately, and thus a user can confirm results at every operation.

3.2 Programming by Demonstrations

The filter-based search also enables to reuse the search process as a script. Once a skillful user creates a filter set, other users also can use the combination in a future search. As aforementioned, we assume that users who have similar interests will share the script; and the script will reflect the characteristics of the region where the public display is located. Moreover, since the filter set can be flexibly decomposed and reconstructed, a user can customize the script for own purpose.

This feature was designed with keeping the concept of *programming by demonstrations* in mind. It is difficult for end-users to make a complex script with a public display, even though each filter is easy to reuse. For example, either changing parameters of a filter or setting a conditional branch between combined filters requires certain programming skill and knowledge. Therefore, our approach aims at creating a script without forcing a user to write a program consciously. Our system only requires users to demonstrate search process by manipulating GUI widgets. The programming by demonstrations is realized by tracking the user's operation, and the demonstration is recorded and compiled into a reusable script [3].

3.3 Web Service Mash-up

Contents are another important aspect of our system. When we search information on the Internet, usually we use other web services in parallel, and compare and/or compensate the results to improve the quality of outcome. A search result in one service will be used as a keyword for other web services. Mash-up is more sophisticated style of such multi-service-based search: it combines multiple services into one service [9]. For example, a map service can show restaurant information with review results and photos, if it is mashed up with restaurant search service.

Since multiple people share a public display, the transaction time for each interaction (i.e., search) should be shorter than personal computers. Moreover, launching multiple services in one window could be confusing to end-users. On the other hand, contents should be enough attractive and consist of variety types of information. For example, in terms of format, text comments and images should be available to convey information in both verbal and nonverbal ways. Furthermore, from the aspect of variety, amateur creators' contents derived from CGM mechanism should be provided as well. Some web services only manage the contents prepared by companies and designers, but CGM is also important in order to explore niche markets and stimulate the ecosystem.

For the first prototype, we mashed up four different web services. In the next session, we explain the implementation of our system in detail.

4. IMPLEMENTATION

In this section, we explain the detail of system implementation from two main aspects: user interface and system architecture.

4.1 User Interface

As shown in Figure 2, our tourist navigation service's window consists of three main panes: map pane, filter pane, and history pane.

4.1.1 *Map Pane*

The map pane is the area for showing a map, and it also indicates POIs with corresponding information. At the default position, the public display is shown at the center of the map. Then a user can move shown area as they like by dragging the map. Moreover, the map is scalable, thus a user can zoom in/out it if necessary.

When a user applies the filter that specifies POIs' category (e.g., restaurant), corresponding points are marked as *markers*. Then, additional filters are applied to the markers on the map. Figure 3 shows an example of the transition of map pane. If a user selects a review filter, review comments and its rank are retrieved from a web service. Such information is shown in the markers' bubble so that the user can confirm the POI's detail at a glance. If the user applies a route filter, a route from the display's location to the point is shown as a line. This feature is also implemented with using another web service's functionality.

Figure 3: An example of the information that appears on the map pane.

The map pane also has menu buttons on the lower side of the pane. These are "map clear", "location change", "unmark bookmarks", and "show bookmarks". With the map clear button, all markers on the map are cleared, so a user can refresh the pane and start from an initial state. The location change button allows a user to change the focused area. The unmark bookmarks and the show bookmarks buttons are used for bookmarking feature. In the search process, a user can partly keep markers even though the map clear button clears other search results. This feature is mainly used for route creation, since the user often needs to connect different types of POIs as mentioned in the sample scenario. In this case, the results of previously conducted search need to remain on the map, even though a new search started.

4.1.2 *Filter Pane*

The filter pane shows available filters as a list. The filters have a parent-child relationship as Figure 2 shows. In the example, filters in the parent category specify POIs' genres (e.g., *Restaurants*, *Universities*). If the *Restaurants* filter is applied, then the available filter list changes to specify restaurants' types (e.g., Japanese, Chinese). To apply the filters, the user has to drag-and-drop a filter from the list onto the map pane. Applied filters are shown in the window on the lower part of the filter pane. If already applied filters need to be canceled, each filter has a "close mark button" so that the user can remove it by clicking the button.

4.1.3 *History Pane*

The history pane shows the filter combinations that are previously created by other users. In Figure 2, the histories of created routes are shown for an experimental study. The filter combination is ranked higher as it is reused more, so that highly ranked combinations represent the locally interested points. Moreover, highly ranked combinations are shown as recommendations from the system. Thus it helps for end-users to choose an appropriate filter, since they cannot create a search query by themselves. The user can directly apply the filter combination by selecting options from this history pane.

4.2 System Architecture

The tourist navigation service is a client-server system. The client application runs on a networked public display and it handles a user interaction part. The user interface is implemented with Flash (Action Script 3.0). The client application communicates with a server application running on a remote computer. The server offers REST interface to receive the client's HTTP request and also to transfer stored contents as XML format. The server application is implemented with Ruby on Rails (2.3.3) and it stores application data into a relational database (SQLite 3) with O/R mapping. The stored data consists of data that is currently shown on the display (e.g., shop data, review data) and search history data. Figure 4 shows the detailed system architecture of the client and server applications.

When a user interacts with the service via a public display, the client application creates a HTTP request according to the selected filter. Each filter has an identity number corresponding to filter category (e.g., id_10: *Restaurants*, id_11: *Hamburger*). The HTTP request contains the ID number, and in some cases additional data is also included so that a query to web services can be created in a supported format. For instance, both the *Restaurants* and the *Hamburger* filters are handled as different HTTP requests with unique ID numbers. However, in the server side, they are aggregated into one query, since the web service that provides restaurant information requires a query in a specific format. After

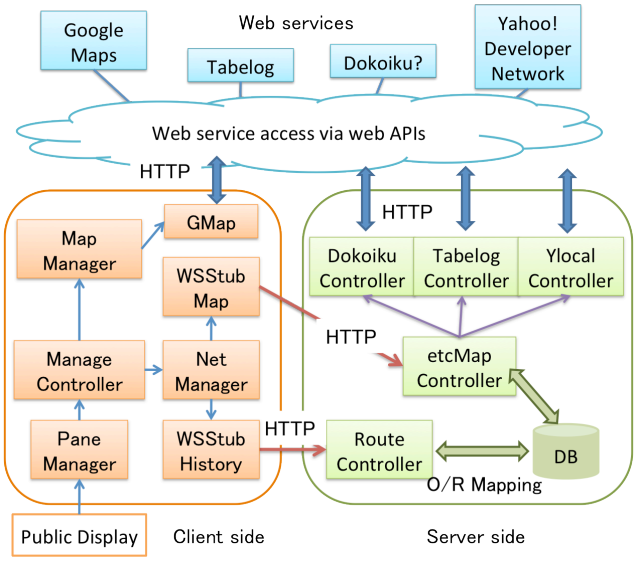

Figure 4: System architecture of the tourist navigation service.

the web service responded to the query, the server application again interprets the data and sends it to the client application in XML format. At the same time, the server application stores the ID number that corresponds to the applied filter into the search history database. Following sections explain the system components that are illustrated in Figure 4.

4.2.1 *Client Side*

ManageCtrl class handles the events that are generated through the interaction with users. As aforementioned, a user interacts with GUI components on the panes, and PaneManager handles input events from GUI and update panes accordingly. Event type differs according to the user's input, and ManageCtrl class dispatches the event to other classes. When new information is acquired from other classes, ManageCtrl class sends a request to PaneManager, in order to update the panes.

MapManager handles all requests to the web service that provides map information. Google Maps is used in our system, thus we implemented GMap class as a proxy of Google Maps API. Map related functions, such as accessing map information, control the map (i.e., move and scale), and drawing routes, are invoked by MapManager through GMap class. NetManager handles communication between the server side applications. Two sub classes are implemented to offer HTTP access: WSStubMap class retrieves POIs' information for map drawing and WSStubHistory class retrieves search histories from the database.

4.2.2 *Server Side*

EtcMapController class returns POIs' information in response to the client application. The first prototype mainly focuses restaurants' guide, and three web services are used to fetch the relevant information: Tabelog[2], Dokoiku?[3], and Yahoo! Developer Network[4]. Tabelog provides restaurants' information, such as phone number and street address. Review comments, photos of the restaurant and ranking are available as well. Dokoiku? provides POIs in other categories, such as entertainment, shopping, convenience stores, universities, museums, and etc. By specifying a keyword from the basic information, a user can access category tags and reputation score. We also used Yahoo! Developer Network to show railway stations' information that is accessed through the *Stations* filter. The server side application has the proxy classes corresponding to each service, so three web service controllers were implemented in total. Fetched information from the web services is stored into the database as a cache, in order to decrease access overhead.

[2]http://tabelog.com/
[3]http://www.doko.jp/
[4]http://developer.yahoo.co.jp/

5. EVALUATION

With the tourist navigation service, we performed experiments to evaluate the concept of *public display in an ecosystem*. This experiment also aims at identifying design issues in application development for a public display. In this section, we explain the method and results of the experimental study.

5.1 Method

Below list describes the experiment's procedure.

1. Brief explanation about the tourist navigation service was given to subjects. Then, the subjects were allowed to try all features of the service freely for 10 minutes, in order to get used to its user interface.

2. After the practice step, two tasks were given to the subjects. Both tasks requested them to create a sightseeing route under given scenarios. In Task A, the subject is assumed to be at Sapporo St. in Japan, and they are about to start sightseeing. They have 3 to 4 free hours to go around the city and it is around noon at the moment, but they have not had lunch yet. Under this condition, the subjects can freely create sightseeing routes, but they need to visit at least one special landmark (a famous clock tower in the city) on the way.

3. Secondly, Task B was given to the subjects. In Task B, the subjects are assumed to be at Kyoto St. in Japan, and they are about to start a historic temple tour. It is about 14:00 p.m., and they already had lunch at that point. They also have 3 to 4 free hours for sightseeing, and no special landmarks were specified in the task.

4. After the experiments, the subjects were asked to fill in questionnaires with a web browser.

6 university male students (23 - 25 years old) joined the study, and all subjects were not familiar with the places used in the scenarios. In order to minimize an order effect, the subjects were divided into two groups. One group performed from Task A to Task B order, and the other group did vice versa. Experiments' process was monitored and recorded with a video camera in order to analyze how the subjects interact with the service on a public display. A Smart Board was used as a public display [8].

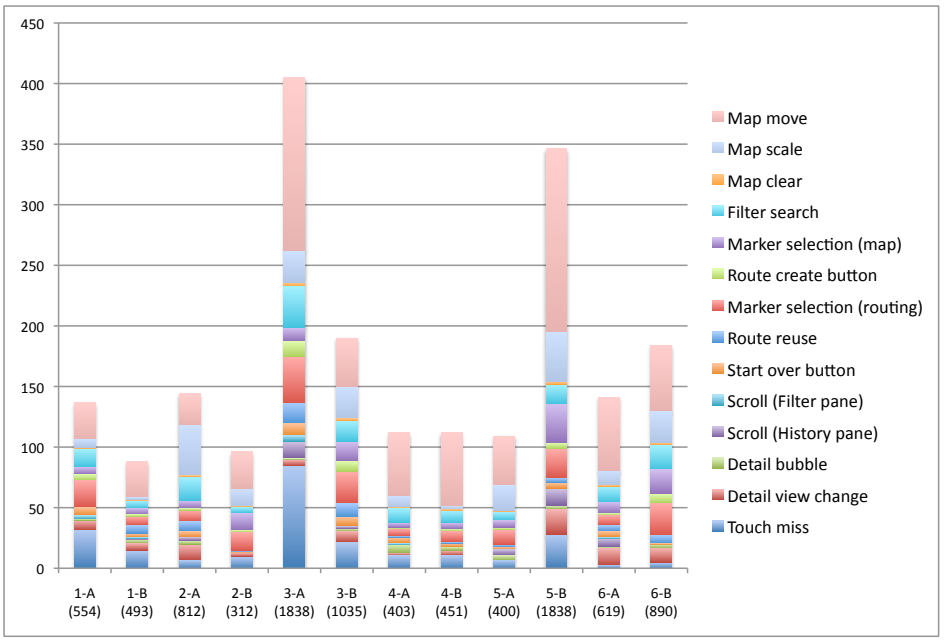

Figure 5: Classification of the touch interaction to the tourist navigation service. The X-axis shows subjects and tasks with indicating the time to complete the task (sec) in *Subject's ID-Task ID (Time)* format. The Y-axis shows the number of times the subjects touched the display in the experiments.

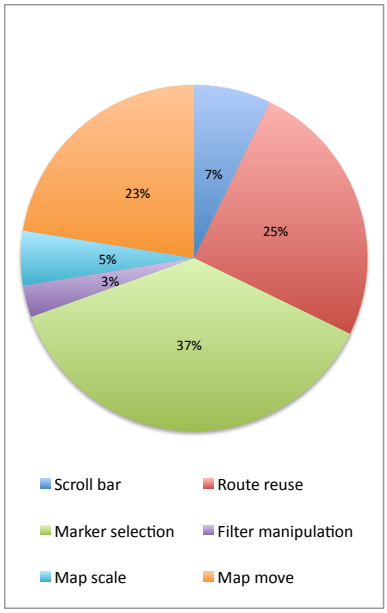

Figure 6: A breakdown of touch misses. The circle graph shows a ratio of touch miss patterns recorded in the experiments.

5.2 Results

Figure 5 shows the number of times each subject touched the display, and the ratio of its purpose. For example, even though the touch times and total time spent in the experiment differ according to individual subjects, map operations such as *"Map move"* (about 35% of total touch times in average) and *"Map scale"* (about 11%) keep the larger part of interactions. This is due to the balance of three panes. Since the history pane keeps a certain portion of area, the map pane is compressed and it requires the user to manipulate the map frequently. The resolution of used display is also problematic, since presentable information was limited. Fine-grained display could resolve this problem.

More importantly, the ratio of touch misses was also significant (about 11%). We analyzed the video and classified main results as shown in Figure 6. The ratio of *"Marker selection"* was relevantly larger than other manipulations. This is due to the size of interactive area, and it also becomes difficult to touch a marker precisely as the shown markers get denser and closer. This implies the filter-based approach does not always screen out information effectively, especially in the case that too much information appears in a small area. In addition to that, as the mash-up enriches contents, the volume of information that relates to one marker also increases. This is another issue of mass contents' appearance design for touch-based interactions.

Lastly, we evaluated the feasibility of the ecosystem from the experiment results. Figure 7 shows how many times markers in the search history were reused. In the experiments, randomly created routes were shown in the history pane and the subjects could make routes by referring the scripts. Each route has a couple of markers, so the subjects can select one of it and show corresponding markers on the map. Then with the bookmarking feature, they can create own route without removing the markers in the new search.

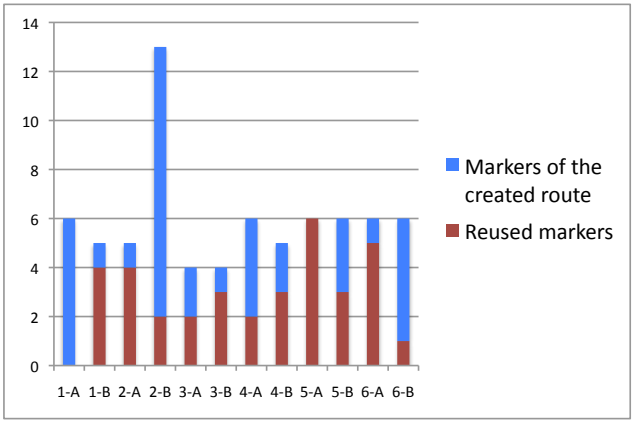

Figure 7: Number of times markers in the search history were reused in the experiments. The X-axis shows subjects and tasks in *Subject's ID-Task ID* format. The Y-axis shows the number of markers used in the experiments.

As the figure shows, about 50% of markers (35 out of 72) were reused in the experiments. Thus it could be said that the preliminary shown markers affected the subjects' choice in such a decision-free condition.

6. DISCUSSION AND FUTURE WORK

In this paper, we presented the concept of *"public display in an ecosystem"*, which adopts the display to an access point to the local knowledge. By utilizing the filter-based search, users can leave their search queries and results for future reuse, and it helps end-users who do not have sufficient technical literacy to perform information search. As a proof of concept, we developed a tourist navigation service for an interactive public display. Then we evaluated the feasibility of the concept, from both user interface and ecosystem design point of views. In the following sections, we discuss design issues found from the experiments and identify future directions to improve the system.

6.1 Improvements in User Interface Design

As explained in Section 5, there are two important design issues to improve the usability: the balance of the three panes and the capable volume of information for touch-based interaction. Furthermore, we realized that conventional GUI design does not well fit to public displays, since the focal area is limited in the interaction process. Even though the public display can be larger and larger, a user needs to stand at the position where they can reach the display. Therefore, the display gets closer, and as a result only limited area in front of their head becomes recognizable. Actually in our preliminary experiments, the history pane could not draw a user's attention enough, because it is shown on the lower part and out of their sight. Even though the histories are useful, it does not work if the user cannot notice it. One possible approach is to directly overlay recommendations onto the map pane so that the user can recognize it without switching focus to other parts. However, the amount of information and presentation timing should be well considered, otherwise it will also drastically increase cognitive workload.

Another issue is the consistency of metaphors. In the experiments, some subjects commented that the filter metaphor is a bit confusing since some other functions require pushing a button instead of drag-and-drop. Moreover, a user needs to keep raising their arm while the drag-and-drop gesture. It makes the user tired and degrades the usability. In addition, the animation effect of marker presentations did not remind the filters. This linkage between the metaphor and presentation is important for a user to understand how the script is composed, and manually improve the scripts that is automatically generated by their demonstrations.

6.2 Decision-making Support from Human Factor Aspects

In this paper, we mainly have discussed from a system perspective: applying programming by demonstrations to run the ecosystem and assisting end-users' information search. In addition to that, however, we should consider human factor aspects in the system design. For instance, public displays should be enough attractive for users to try the service, otherwise it disappears into background and ecosystem does not work [5]. Moreover, incentives should be given to users in order to keep the quality of contents and to motivate skillful users to create useful scripts. For example, reputation mechanism with economic incentives (e.g., virtual currency) is well used in human-power-based search. A questioner gives points to the user who gave a good quality answer, and it motivates users to actively contribute to the system. Moreover, other users can check the reputation so that they can decide whether the proposed answer is trustworthily or not.

It is also considerable that end-users still cannot find the scripts that exactly fit to their interests. Moreover, other users' history can be boring, even though the trends mined from history data is an effective approach to affect users' behavior. Serendipity is one important factor to engage end-users with the system [6]. Since our target users share niche interests related to location, there could be a lot of chances to direct serendipities around the public display. For example, Twitter[5] is the social communication media that enables users to distribute their short tweets over the Internet. There are also several mash-up services that show the message on a geographic map with reflecting the location where a user tweets. This geographically bound information could bring the serendipities, since users happen to be aware of new interests even if their prior interests were different. Particularly, it is important to design the system to increase the chance to find new interests, since it provides a good user experience and will keep the user to be involved in the ecosystem.

7. REFERENCES

[1] A. Angus, D. Papadogkonas, G. Papamarkos, G. Roussos, G. Lane, K. Martin, N. West, S. Thelwall, Z. Sujon, and R. Silverstone. Urban social tapestries. *IEEE Pervasive Computing*, 7(4):44–51, 2008.

[2] S. Carter, E. Churchill, L. Denoue, J. Helfman, and L. Nelson. Digital graffiti: public annotation of multimedia content. In *Proc. of CHI '04*, pages 1207–1210, 2004.

[3] J. Lin, J. Wong, J. Nichols, A. Cypher, and T. A. Lau. End-user programming of mashups with vegemite. In *Proc. of IUI '09*, pages 97–106, 2009.

[4] J. Müller, M. Jentsch, C. Kray, and A. Krüger. Exploring factors that influence the combined use of mobile devices and public displays for pedestrian navigation. In *Proc. of NordiCHI '08*, pages 308–317, 2008.

[5] J. Müller, D. Wilmsmann, J. Exeler, M. Buzeck, A. Schmidt, T. Jay, and A. Krüger. Display blindness: The effect of expectations on attention towards digital signage. In *Proc. of Pervasive '09*, pages 1–8, 2009.

[6] M. W. Newman, J. Z. Sedivy, C. M. Neuwirth, W. K. Edwards, J. I. Hong, S. Izadi, K. Marcelo, and T. F. Smith. Designing for serendipity: supporting end-user configuration of ubiquitous computing environments. In *Proc. of DIS '02*, pages 147–156, 2002.

[7] P. Peltonen, A. Salovaara, G. Jacucci, T. Ilmonen, C. Ardito, P. Saarikko, and V. Batra. Extending large-scale event participation with user-created mobile media on a public display. In *Proc. of MUM '07*, pages 131–138, 2007.

[8] S. Technologies. Smart board. http://smarttech.com/.

[9] G. Wang, S. Yang, and Y. Han. Mashroom: end-user mashup programming using nested tables. In *Proc. of WWW '09*, pages 861–870, 2009.

[10] T. Yamabe, K. Takahashi, and T. Nakajima. Towards mobility oriented interaction design: experiments in pedestrian navigation on mobile devices. In *Proc. of Mobiquitous '08*, pages 1–10, 2008.

[5]http://twitter.com/