

A Framework for Personalizing Action History Viewer

Masaki Ito
niya@ht.sfc.keio.ac.jp

Jin Nakazawa
jin@ht.sfc.keio.ac.jp

Hideyuki Tokuda
hxt@ht.sfc.keio.ac.jp

Graduate School of Media and Governance
Keio University
5322, Endo, Fujisawa, Kanagawa, Japan

ABSTRACT

This paper presents a programmable analysis and visualization framework for action histories, called *mPATH framework*. In ubiquitous computing environment, it is possible to infer human activities through various sensors and accumulate them. Visualization of such human activities is one of the key issues in terms of memory and sharing our experiences, since it acts as a memory assist when we recall, talk about, and report what we did in the past. However, current approaches for analysis and visualization are designed for a specific use, and therefore can not be applied to diverse use. Our approach provides users with programmability by a visual language environment for analyzing and visualizing the action histories. The framework includes icons representing data sources of action histories, analysis filters, and viewers. By composing them, users can create their own action history viewers. We also demonstrated several applications on the framework. The applications show the flexibility of creating action history viewers on the mPATH framework.

Keywords

Action history, visualization, visual language

INTRODUCTION

In the ubiquitous computing environment where computers and sensors are embedded in our surroundings, it will be possible to recognize our action and record it. From the accumulated action history, we will be able to get highly abstracted context information of the human activity. These information are used to develop context-aware applications, and also used to provide us with useful information. Well presented action histories help our life such as retrieving memory and sharing our experiences.

Several representation applications of the action history have also proposed[7][6]. These systems represent user's activity to provide functionalities of navigation and indexed action histories. However these representations are designed

for a specific use, and hence users can not customize them to acquire personalized view of their action histories. For example, though PEPYS[7] can organize human action histories based on the location- and time-axis, it can not handle additional information, such as images, related to an action history item. Activity Compass[6] provides a navigation functionality based on a location track analysis, which also lacks diversity of location history analysis. The action history viewers, therefore, should provide users with personalized views for enabling them to analyze, recall, talk about, and report their action histories.

In this paper, we propose a new framework for creating the personalized action history viewer. The framework provides users with programmability by a visual language environment for analyzing and visualizing the action histories. The framework includes icons representing data sources of action histories, analysis filters, and viewers. By composing them, users can create their own action history viewers.

This paper is organized as follows. In the next section, we introduce scenarios where various visualization technique for analyzing action history. The third section shows current techniques for analysis and visualization and clarify the requirements. In the 4th section, we introduce mPATH as a framework of personalizing action history viewer. Next three sections introduce the usage of the mPATH framework and show several applications. We evaluated the system in the following session and introduce related works. In the final session, we conclude this paper and suggest future work.

SCENARIOS

We introduce two scenarios where visualization method are shared and easily developed by users. These scenarios show usage of action history viewer with personalizing function. The feature helps our communication and the deep understanding of past activity in the scenarios.

Navigation of a Travel Memory Last week, Alice has traveled Kyoto, Japan. When she arrived at Kyoto station, she borrowed a PDA with GPS as a guide for sightseeing. While she was in Kyoto, the PDA gave her a guidance at a tourist attraction and assisted her in planning her travel. When she left Kyoto, she returned the PDA and received a small memory card in which her location track data, activity history, and

photos she took were recorded.

Today she is talking about her travel with her boyfriend, Bob. She inserted the memory card to her PC, and a map of Kyoto with her action history shown. She said this was her travel, but she found the map were not enough to represent her impression. She searched for a visualization method of travel experience and downloaded it.

The method analyzed her travel log and calculated weights of each tourist attraction by her walking speed and number of pictures. The method visualized the map of Kyoto with distortion of the weights, she could intuitively know her travel.

She thought shopping is also important to calculate the weights, she changed the parameters of analysis and generated a map which highly reflect her impression. The map realized smooth understanding of her travel for him.

Development of Analysis and Visualization Method Now, Bob wants to go to Kyoto. He asked her to go with him, but she refused because she has just been there. He then decided to show the attraction of Kyoto she did not know.

Since he is an amateur programmer, he decided to develop a visualization method in which attractive places where she did not visit were emphasized. He at first searches for a web guide of Kyoto in which tourist attractions are ranked and contains many pictures. Then he developed an algorithm to find attractions she did not visit by comparing the web guide with her tour data. He designed to visualize a map of Kyoto with many photos of the attractions.

He uploaded his algorithm and asked her to download and apply it to her memory. She noticed unknown attractions in Kyoto and decided to visit again.

VISUALIZATION OF ACTION HISTORY

In this section, we define action history, and mention current techniques of visualization and analysis of action history. Then we clarify requirements of the visualization system.

In this paper, an action history is an aggregated form of information which contains location, date and description of action about certain person. Location track data obtained by GPS is one example of action history. Digital photo data is also an action history if it contains time stamp and location information as its meta information.

Current Visualization

Several visualization methods were developed to represent action history. A text-based representation like a diary is simple, and used in PEPYS[7]. To emphasize temporal aspect of action history, a table of action history ordered by date is used. To represent spatial aspect of action history, a map which is overlaid with location tracking data or other icons suggesting activities in certain places is used.

To represent highly abstracted context of action in addition to simple activity or movement, analysis method of action history is also important. There are several researches to analyze location track data captured by GPS and extract context information of an actor like daily patterns of movement and interests [1][6][8]. Some of these researches utilize additional geographic information to detect an activity in a certain place.

Current visualization systems and analysis methods, however, are designed for specific use of action history. To represent action history in various aspects as the scenario shows, flexible analysis and visualization features are required.

Requirements

For flexible programming of visualization mentioned in the scenario, following features are required to the system.

Flexibility of Data Input The system must treat various types of data available in the ubiquitous computing environment simultaneously. Action histories are characterized by their description of location, contents of what we did there, and the way of acquisition of the data.

The system also need to treat geographic information for analysis of action history. A digital map and the Yellow Pages with address are examples of geographic information.

Providing Programmability of Analysis and Visualization For all users of the system, it must be easy to create their original visualization method. For skillful users, the system must provide a flexible programmability. Even for unskilled users, the system must provide possibility of changing a visualization method.

Sharing existing methods, which are programmed by third parties, also increases programmability of the system. Using existing method reduces cost of creating new analysis and visualization method.

mPATH FRAMEWORK

We developed a programmable analyzing and visualizing framework for action history named mPATH. In this section we introduce the features and implementation of the mPATH.

Approach

The followings are the approaches taken by the mPATH framework to accomplish aforementioned flexibility.

Component Based Architecture For reducing the cost of developing a new analysis and visualization method, we divided the method into several components. We defined three types of component, data source, viewer and filter. Users can create visualization method by combining existing components.

Standardization of Internal Data We defined a unified type of data in the system. The description of location, date and other information are unified. When we input action history

or geographic information, they must be converted into single type of data.

The type of data is used to exchange action history between components. Since data type is unified, we can easily design a component for several action history. The feature also realize flexible combination of components.

Using Data Flow Style Visual Language To control combination of components, we provide a visual programming system of data flow style. Visual programming reduces difficulty of creation of original visualization method by combining components.

Implementation

We implemented a prototype of mPATH framework with Java. Our system is implemented as a GUI application and consists of 14,000 lines of Java language. Figure 1 shows a screen shot of the system.

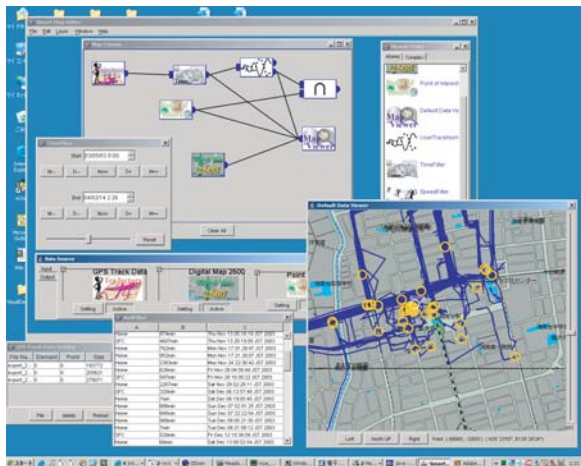


Figure 1: Screens hot of the system

In the current implementation, we mainly focus on realizing visual programming for data analysis and visualization. This version works as a platform for creating analysis method by data flow style programming. Exporting and importing of program function is still under construction. We are planning to use XML to exchange them.

Experiment

Since June 2003, Ito, the first author of this paper, has been carrying a “Garmin eTrex Legend”[4], a handy GPS receiver. Table 1 shows the amount of captured data. While the experiment, he took pictures as action histories using a digital camera. The track data of the handy GPS and taken pictures are used as a data of development and evaluation of this system.

COMPONENTS FOR mPATH FRAMEWORK

We developed several components for mPATH framework. In this section, we classify them into data source, filter and viewer, and introduce components we developed.

Table 1: Captured Data of GPS and Digital Camera

| Date | size(byte) | track | point | picture |
|-----------|------------|-------|--------|---------|
| Jun. 2003 | 321,310 | 286 | 5,893 | 515 |
| Jul. 2003 | 294,518 | 297 | 5,336 | 131 |
| Aug. 2003 | 473,739 | 412 | 8,676 | 218 |
| Sep. 2003 | 365,855 | 298 | 6,730 | 37 |
| Oct. 2003 | 307,187 | 287 | 5,608 | 342 |
| Nov. 2003 | 193,772 | 153 | 3,573 | 54 |
| Total | 1,956,381 | 1,733 | 35,816 | 1,297 |

Data Source

To input various action history, we developed several components as data sources. These components access action history and transform the data into standard data format.

GPS Location Track Data Source GPS Location Track Data Source access files of location track information or a GPS device through an RS-232C interface. Acquired data are transformed into a group of points where actor passed by.

Photo Data Source Photo data source deals with image files taken by a digital camera. By reading time stamp and location in EXIF[5] information of the files, the components generate an action history of “taking pictures” in the standard data format.

The component can accept location track data. By matching timestamp of photos with location track data, it estimates location of pictures.

Map Data Source Map data source outputs map data in a vector format such as points of station, lines of road and polygons of buildings. Users can detect detail of action history by comparing it with the map data. Since the data was designed for GIS, we can get generic map by rendering them and use the map as a background of visualization.

Filters

Components of filter has one or more inputs and outputs. They input data of standard format and output the result of processing in the same format.

Time Filter Time filter extracts action histories during a specified term.

Speed Filter Speed filter filters data by its speed. It is useful especially for infer transportation from location track data.

Formalize Filter Formalize filter classifies a group of point data into movements and stops, and outputs as track and point data by two separated sockets. Users can change threshold of time to detect actor’s stop.

Matching Filter Matching filter has two input sockets: one for the map data and the other for a geographic coordinate.

The matching filter calculates the name of the specified coordinate from the specified map.

Count Filter Inside count filter, geographical region are divided into a grid. The filter counts input data in every grid. Users can know how many times he or she visited a certain point, i.e. a weight of the point, in an action history.

Viewer

We developed three viewers to visualize action history.

Normal Map Viewer To visualize spatial aspect of action history, we developed normal map viewer. In this viewer, every input are ordered by the geographic coordinates, so that generic map-like visualization is realized. This viewer accepts multiple data and overlays them.

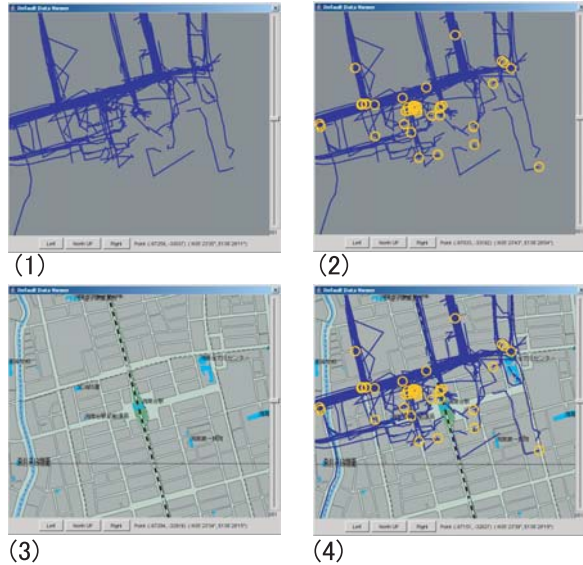


Figure 2: Normal Map Viewer

Table Viewer We used table type viewer shown in Figure 3 to visualize data output by the matching filter. In the left column, the name of the place he visited are listed, in center, time of stay are shown. In the right column, the date the user begin to stay are shown.

| Place | Time | Start |
|-------|-----------------|------------------------------|
| Home | 13 hours 37 min | Wed Jan 14 10:30:20 JST 2004 |
| SFC | 10 hours 40 min | Thu Jan 15 11:18:50 JST 2004 |
| Alome | 55min | Fri Jan 16 22:36:34 JST 2004 |
| Home | 16 hours 43 min | Sat Jan 17 21:52:29 JST 2004 |
| Alome | 1 hours 7 min | Sun Jan 18 21:28:26 JST 2004 |
| Home | 11 hours 13 min | Sun Jan 18 22:42:20 JST 2004 |
| Home | 55min | Mon Jan 19 20:00:01 JST 2004 |
| Home | 13 hours 9 min | Mon Jan 19 22:07:14 JST 2004 |
| Home | 17 hours 19 min | Tue Jan 20 20:20:03 JST 2004 |
| Home | 16 hours 18 min | Wed Jan 21 23:23:16 JST 2004 |
| SFC | 5 hours 25 min | Thu Jan 22 16:13:20 JST 2004 |
| Home | 15 hours 30 min | Fri Jan 23 01:37:36 JST 2004 |
| Home | 10 hours 19 min | Sat Jan 24 03:52:16 JST 2004 |
| Alome | 55min | Sat Jan 24 22:14:51 JST 2004 |
| Home | 22 hours 2 min | Sat Jan 24 23:26:52 JST 2004 |

Figure 3: Table Viewer

Weight Map Viewer To represent highly abstracted context of action, we developed weight map viewer. The viewer accepts weight of regions in addition to normal action history or geographic information. This viewer visualize the weight as a color or scale of each region as shown in Figure 4.

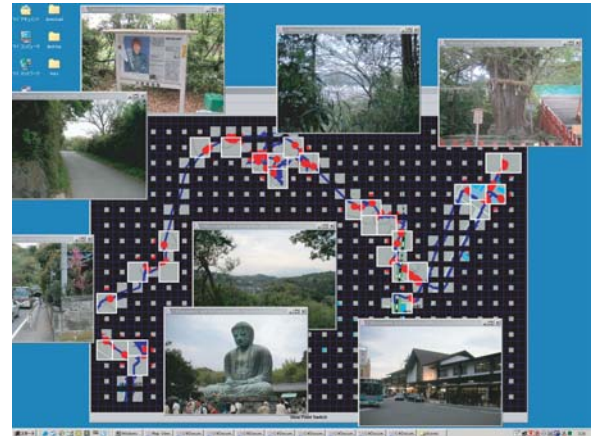


Figure 4: Weight Map

VISUAL PROGRAMMING

In this section we introduce a visual programming manner of mPATH system by creating map like viewer of action history. The main window of the mPATH system consists of mainly two windows, one is a palette and the other is a canvas. Figure 5 shows the detail.

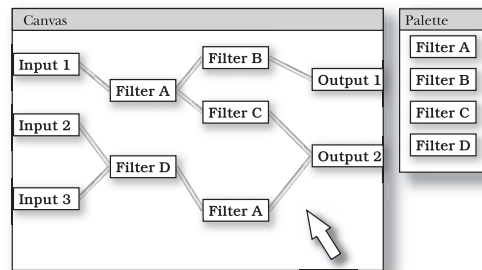


Figure 5: Visual Programming Window

In the palette, components of data sources, viewers and filters are registered as icons. By drag and drop of the icon, a component is copied and registered to the canvas.

In the canvas, every data is shown as lines from left to right. Each icon of component has sockets of data, Right socket means an output of the data, and left socket means an input. Icons only with right socket are data sources, and icons without right socket are viewers. Icons with both left and right socket are components of analysis method and work as filter of data. By clicking two icons on canvas, two components can be connected and disconnected.

Figure 6-(1) is a simple visualization of location track data

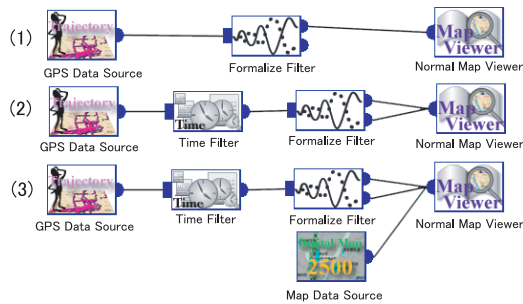


Figure 6: Example of Visual Programming

of GPS. In this program, GPS data are input to normalize filter. Only track data are input to a normal map viewer and shown in the geographic coordinates. The result of visualization is figure 2-(1). When we connect both track and point socket to the viewer, we can see stop point data with track data as shown in figure 2-(2). The operation on the canvas is immediately reflected on the viewer, therefore an interactive operation is accomplished.

By insertion of time filter between GPS data source and formalize filter as shown in figure 6-(2), we can control the term of location track data.

To visualize the detail of location track data, we use a map as a background. Figure 2-(3) is a result of connection between map data source and normal map viewer directory. By inputting the output of map data source to the viewer of track data as figure 6-(3), we can see track data with normal data as figure 2-(4).

APPLICATIONS

In addition to normal map viewer application mentioned as an example, we developed two applications on mPATH system. First one is listing points of interest system by analyzing stay stop in a certain place. Second one is a visualization system of travel activity focusing on traveler's interests on the places.

Listing Points of Interest System

We developed extraction system of user's interest point on mPATH system. An interest point is a place where user did something or stayed for a long time. We also detect name of the point using matching filter. In this system, we visualized interest points in a table. Figure 3 shows the result of the visualization.

Visualization with Weight

We developed an weigh method to classify regions by user's activity. In our algorithm, we at first divide the region into small grids. In each grid, we count the number of times we visited, the number of times we took pictures, and the number of times we shopped. Then we add each value and obtain the weight of each cell as a number.

We implemented this algorithm on the mPATH system, and

developed a visualization system reflecting the weight of the region. Figure 4 shows an example. In the example, we weighed each cell mainly by the number of pictures actor's took, and the weight of each cell is visualized as a scale. We realized showing photo by clicking on the map, this system can be used as a photo viewer.

EVALUATION

In this section we evaluate the performance of mPATH system. It shows if the response of the system is enough to develop interactive development of visualization method. It can also be used to find functions which should be improved.

For measuring the performance, we used the environment shown in table 2 and used data shown in table 3.

Table 2: Evaluation Environment

| | |
|--------|------------------|
| CPU | Pentium4 2.53GHz |
| memory | 1024MB |
| OS | Linux 2.4.22 |
| JDK | J2SDK 1.4.2_03 |

Table 3: Data

| Type | GPS location tracking | Map Data |
|-------------|-----------------------|----------------|
| description | Jun. 2003 – Nov. 2003 | Fujisawa city |
| size | 1,956,381 byte | 9,261,000 byte |

Overhead of component architecture

We measured the performance of the component architecture, therefore the architecture seemed to be an overhead compared to a hard-coding implementation. We measured simple visualization method with several time filters, in which we changed the number of the time filter.

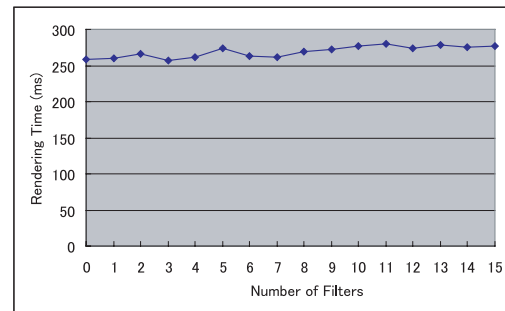


Figure 7: Measurement of the Overhead

Figure 7 shows the result of the measurement. While we increased time filter from zero to 15, the growth of the time is small. This result shows that the overhead of connecting analysis components is small.

Evaluation of the Performance of the Visualization on the System

We measured the performance of a visualization system constructed on the mPATH system. We used simple visualization application in which location track data and map data are rendered on a same window, and measured the time required for rendering. Figure 8 shows the result. It also shows the result of rendering single location or map.

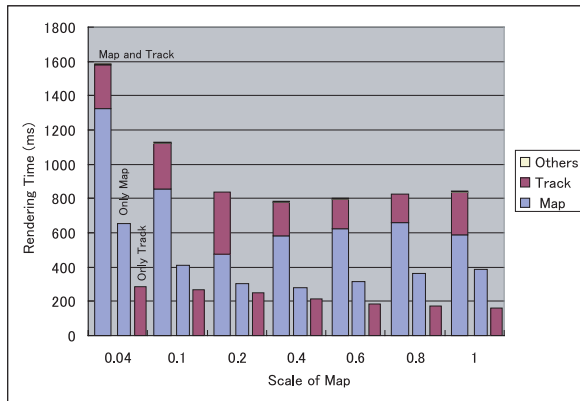


Figure 8: Application Performance

Lowest case of rendering time is about 800ms, and in the case of small scale takes more than 1500ms. In the overlaid case, the rendering time of the map are two times of single rendering of map, therefore it can be reduce.

RELATED WORKS

We introduce several researches and products as related work of mPATH framework. These systems does not treat action history, but provide flexibility of data analysis and representation by visual language manner.

DFQL

To improve query language of databases, various graphical query languages are proposed instead of text based language like SQL. DFQL[3] is one of the graphical query language for the use in scientific database. DFQL uses data flow language and enable analysis and visualization in addition to data retrieval which general graphical query languages focus on.

Max/MSP

Max/MSP[2] is a visual programming system for midi and audio signal. The system enables interactive processing of music stream with a graphical data flow language and is used to create electronic musical instruments or effectors with original sound algorism, and interactive media systems.

The visual language is used by many creator of electronic media and fruits of the programming are exchanged widely on the Internet. Since modules of the visual language can be developed with C language, many original modules are also distributed.

CONCLUSION

In this paper, we developed a framework to create original action history viewer by visual programming function of analysis and representation of action history. The system realized flexible visualization of action history. We developed several applications on the framework.

Implementation of Sharing Mechanism We are now implementing a mechanism to share the programs. We designed XML based description for sharing program. We will implement functions to import and export the XML. We also plan to develop sharing mechanism by providing servers on the Internet to upload and download the XML file.

Enable Visual Programming of Parameters of Modules We will extend the visual language and enable visualization of parameters of each modules. We will enable parameters of models to be treated as a modules in the system. This feature will realize coordination of each module and enable more complex analysis and visualization.

REFERENCES

1. D. Ashbrook and T. Starner. Learning Significant Locations and Predicting User Movement with GPS. In *Sixth International Symposium on Wearable Computers (ISWC 2002)*, pages 101–108, October 2002.
2. Cycling'74. Max/MSP. <http://www.cycling74.com/products/maxmsp.html>.
3. S. Dogru, V. Rajan, K. Rieck, J. R. Slagle, B. S. Tjan, and Y. Wang. A Graphical Data Flow Language for Retrieval, Analysis, and Visualization of a Scientific Database. *Journal of Visual Languages & Computing*, 7(3):247–265, 1996.
4. Garmin Ltd. Garmin eTrex Legend, 2001. <http://www.garmin.com/products/etrexLegend/>.
5. JEIDA. *Digital Still Camera Image File Format Standard (Exchangeable image file format for Digital Still Cameras: Exif) Version 2.1*. 1998.
6. D. J. Patterson, L. Liao, D. Fox, and H. Kautz. Inferring High-Level Behavior from Low-Level Sensors. In *Proceedings of The Fifth International Conference on Ubiquitous Computing (UBICOMP)*, pages 73–89, 2003.
7. M. G. L. William M. Newman, Margery A. Eldridge. PEPYS: Generating Autobiographies by Automatic Tracking. In *Proceedings of ECSCW '91*, pages 175–188, September 1991.
8. J. Wolf, R. Guensler, and W. Bachman. Elimination of the travel diary: An experiment to derive trip purpose from GPS travel data. *Notes from Transportation Research Board, 80th annual meeting*, January 2001.