Working Paper Series
ISSN 1177-777X

History Navigation in Location-Based Mobile Systems

Knut Müller & Annika Hinze

Working Paper: 07/2010
1st December 2010

Department of Computer Science
The University of Waikato
Private Bag 3105
Hamilton, New Zealand

History Navigation in Location-Based Mobile Systems

Knut Müller[1] and Annika Hinze[2]

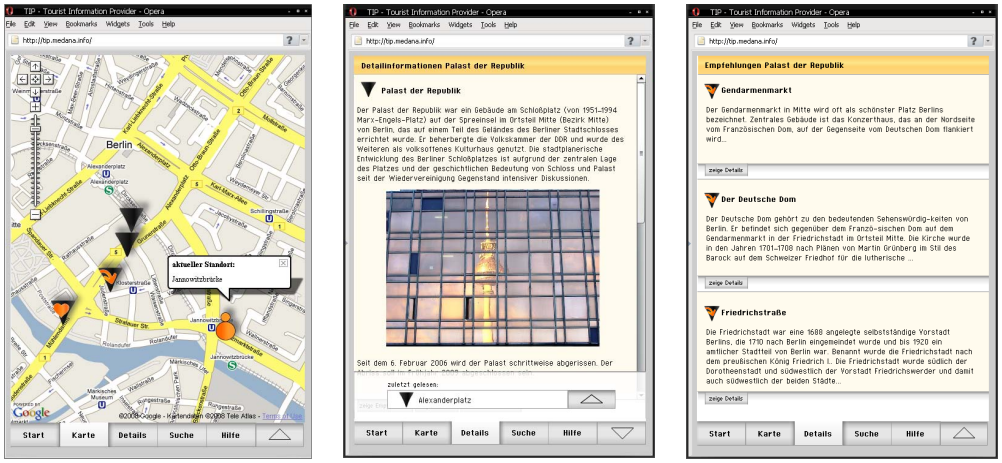
[1]Humboldt-Universität zu Berlin, Germany
knut.mueller@informatik.hu-berlin.de,
[2]University of Waikato, New Zealand
a.hinze@cs.waikato.ac.nz

1 Introduction

The aim of this paper is to provide an overview and comparison of concepts that have been proposed to guide users through interaction histories (e.g. for web browsers). The goal is to gain insights into history design that may be used for designing an interaction history for the location-based Tourist Information Provider (TIP) system [8]. The TIP system consists of several services that interact on a mobile device.

Why do we need an interaction history? A typical user of the TIP system may not remember all places they visited and pages they have seen. We like to give the users the opportunity to discover points-of-interest they have encountered before and whose information they may have forgotten: They could use the history function to virtually walk backwards on their route. The TIP system will thus support the users by reminding them of things they have seen and of information that has been presented to them before.

We selected three different approaches to analyze: A whiteboard system with multiple-levels of history [1], the well-known web browser back button [3], and the representation of history as trees within a web browser [11]. In the remainder of this section, the TIP system is introduced and we give a brief motivation for the project.



(a) map service (b) information service (c) recommendation service

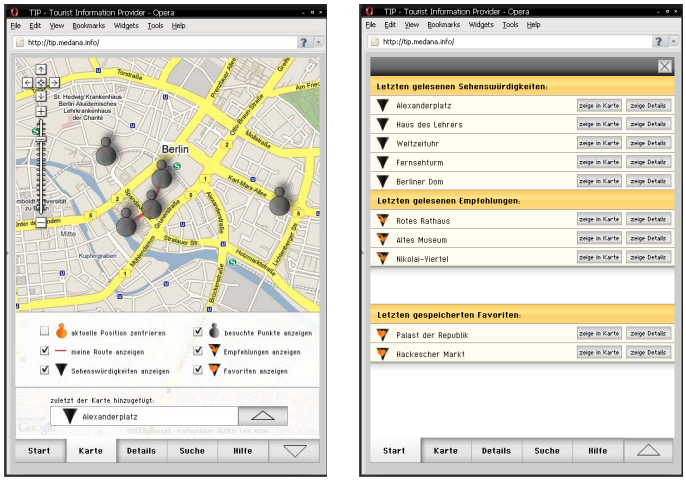
Figure 1: Services provided by the TIP system [13].

1.1 Tourist Information Provider (TIP)

The Tourist Information Provider (TIP) is a mobile software system for portable devices [4, 5, 6, 7]. The typical user is a tourist visiting a unknown city assisted by TIP.

TIP provides a number of services for tourists. Here we focus on a selection of three services: A map service, an information service and a recommendation service (see Figure 1). The map service displays the current position of the user and points-of-interest on a map. A user's movements are represented on the map and additionally a user can select points-of-interest on a map and receive additional information. The information service displays textual and visual information related to points-of-interest. The recommendation service detects semantic correlations between points-of-interest and recommends related points to the user. As a user walks through the city, they can navigate their way using the map to access information according to points-of-interest at their current location or by following recommendations.

Each interaction with the system may generate a history entry. These entries could be accessed by the user through a history or a back button such as the ones typically used in web browsers. However, what is meant by "going back" through history in a location-based system? Going back is an action used to reach a previous system state. The state may be a global or local one: Global states influence all services, local ones refer to the state of a single service.



(a) history on map (b) history in list

Figure 2: History entries in TIP [13].

1.2 Motivation

The storing of history entries in the TIP system is triggered by interactions with TIP services. Each entry consists of a timestamp, the service which triggered the recording and the geographical coordinates where the TIP user is located.

These history entries are represented in a two-dimensional time-location graph on the map (see Figure 2(a)). The concept of TIP as a location-based and service-oriented system prompts three groups of questions:

1. How many histories do we need? Should each service use its own history or should all services share a global history?

2. How to manage histories? Is it useful to cluster histories by locations or by services or both? Is a time-based order encoded into histories (as typically done in web browsers)?

3. Do users understand the history behaviour? Will users expect and understand the displayed behaviour? Is a user aware of different modes in a service (physically visited versus virtually visited points-of-interest)?[1]

This project aims to give answers to all of these questions.

[1]This phenomenon is called *mode confusion* of the user [6]. The term is inspired by aviation, where a pilot can be in mode confusion because of the amount of options, buttons and switches he has to control.

(a) bird's eye view (b) route on map

Figure 3: Berlin with points-of-interest from scenario.

1.3 Structure of the Paper

We design a scenario in the next section, so that we have a reference for all system examples. In the subsequent three sections, we introduce three approaches to present histories. For each system we will give a system overview, a description of its history representation and a comparison to the TIP system. The system overview contains the system's functionality and possible user interactions. The history representation describes what information the system remembers and how the user can access them. In the comparison sections, we compare each system to the TIP system to identify which aspects may be used in this context.

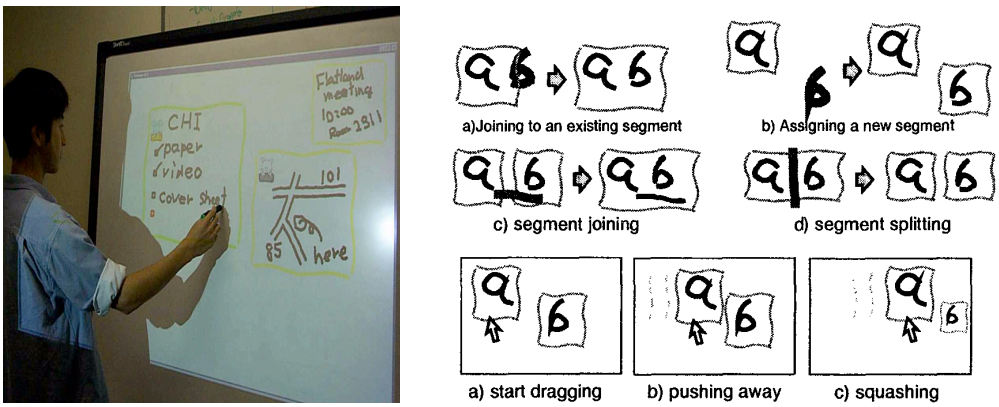
In Section 6, we provide a summary of insights and recommendations of how to design history concepts in TIP.

2 Scenario in TIP

To compare the TIP system with the three different history systems, we design a scenario of a tourist visit to Berlin. Our tourist Lisa visits four places in Berlin: The boulevard *Unter den Linden*, the *Brandenburg Gate*, the *Holocaust Memorial* and the *Reichstag building* (see Figure 3[2]).

Scenario Yesterday Lisa stopped her visit Unter den Linden and starts today on Ebertstraße. Walking south on Ebertstraße she reaches the Holocaust Memorial. After taking some photos, she turns towards the Brandenburg

[2]Bird's eye view and route with courtesy of Google Maps (`maps.google.com`).



(a) Implementation.

(b) Segmenting, Moving and Squashing.

Figure 4: Flatland, Computer Augmented Whiteboard System [1].

Gate and walks to have a rest at Pariser Platz in front of the Brandenburg Gate. Sitting there, she wants to know whether she has missed any points-of-interest. She uses her TIP system to virtually go back via the Brandenburg Gate to Ebertstraße. At the northern end of the street she arrives at the Reichstag building.

3 Flatland: Multi-Level Undo and Redo

3.1 System Overview

Edwards et al. developed a temporal model for multi-level undo and redo for an electronic whiteboard [1]. Their electronic whiteboard is a computer-augmented system, called Flatland [9, 10].

Flatland presents a user model in which the whiteboard is loosely subdivided into regions of activity called segments (see Figure 4(a)). The empty area of the whiteboard represents the root segment. When some strokes are drawn on the root segment, automatically a new segment is created and a frame is drawn around the strokes. Segments can be merged, split, dragged, pushed and squashed by gestures (see Figure 4(b)).

3.2 History Representation

Every segment has an infinite undo and redo function. The user can access any past state of each segment stored in the linear history. With the undo and redo function of the root segment (the underlaying whiteboard) all segments

at once can be changed to global states in the past.

Unfortunately, the paper gives no information of how the Flatland system handles branches in history: We assume that the Flatland history operates like the Timewarp history [2] – a project published by the same authors. In Timewarp uses two history views: a slider to select states within the current linear history line and a graph with all existing history states.

When splitting one segment into two segments, two new segments are created in the history structure. Then the corresponding content of the areas of the original segment is copied to these. The original segment is removed. Merging is similar: a new segment is created and the content of both segments is copied to the new segment. The original segments are removed.

The user does not have to delete segments when no space is left. By moving segments, other segments are pushed and if needed squashed, so that they take less space on the whiteboard.

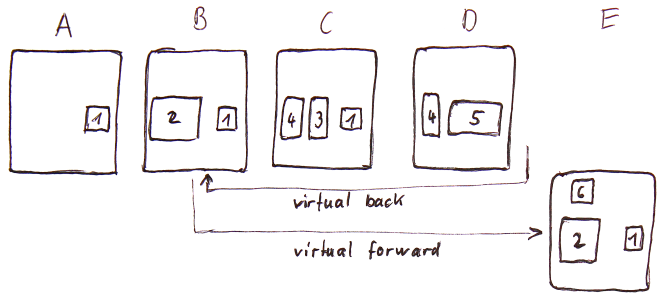
3.3 Comparison to TIP

Although the TIP system and the Flatland system differs in size, mobility, and maybe also in the number of users, we can show similarities. The Flatland segments behave similar to points-of-interest in TIP: The segments have locations on the whiteboard and the points-of-interest have locations on the map. While the points-of-interests behave like segments, the information presented by the TIP services can be considered as content of the Flatland segments.

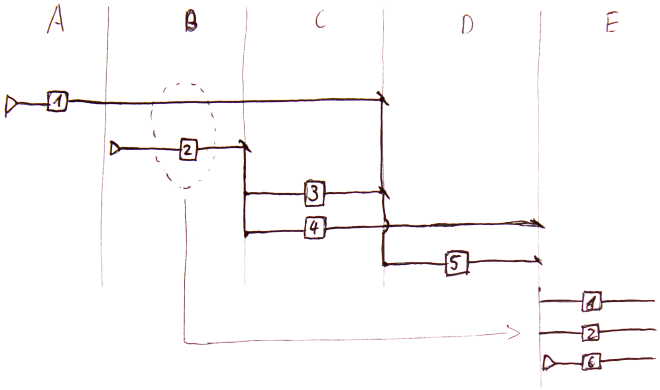
Moving backwards in history of a TIP services can be considered as moving backwards in the history of a single Flatland segment. Moving back in the history of the TIP map can be considered as a global moving backwards in Flatland history.

Therefore, we can reproduce the TIP scenario on the Flatland whiteboard (see Figure 5): At A and B the user creates new segments on the whiteboard. Segment 1 relates to the boulevard Unter den Linden. Segment 2 relates to Ebertstraße, the start of Lisa's trip next day. From Ebertstraße Lisa chooses first to walk to Holocaust Memorial (Segment 4) and then to Brandenburg Gate (Segment 3). This is shown in C as a split of Segment 2 into the segments 3 and 4. From Brandenburg Gate (Segment 3) she walks towards Unter den Linden (Segment 1) until she reaches Pariser Platz (Segment 5). This is displayed as a merge of segments 1 and 3 to Segment 5 in D.

Comparing branching in history (moving back and choosing an alternative) is difficult, because it is not clearly described by Greenberg et al.. We assume that Flatland history behaves like Timewarp history. Thus Lisa's



(a) states on the whiteboard



(b) states in the history

Figure 5: Flatland example according to TIP scenario.

virtual going backwards from Pariser Platz (Segment 5) to Ebertstraße (Segment 2) can be considered as a global back to B. There she virtually walks on Ebertstraße (Segment 2) until she reaches the Reichstag building (Segment 6).

The Flatland system stores every interaction with the whiteboard in the history: Each drawing, moving, splitting or merging is recorded. In addition, the users rarely need to remove segments from the whiteboard, because the segments are squashed if necessary. The multiple layers of history give the user the opportunity to access past states of each segment independent from the other segments. Due to the fact that Flatland generate a global history out of all local histories, each past state of the Flatland whiteboard as a whole can be reaccessed. However, the behaviour while branching from past states remains unclear.

4 Back Button: Linear History without Loss

4.1 System Overview

Tauscher and Greenberg analysed sequences of page visits to produce a recurrence distribution [12]. They found that 58% of page visits are, in fact, revisits. The rest of visits belongs to new pages. With an 39% chance the next page is one of the six last pages.

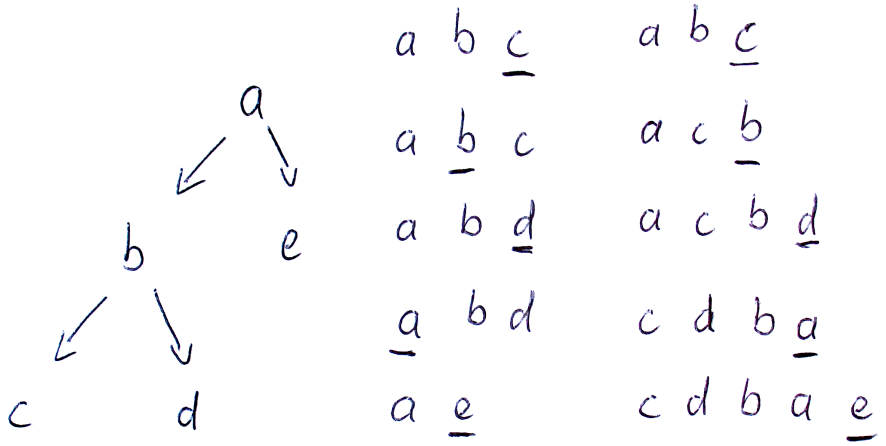
This motivates Greenberg et al. to propose an alternative behaviour for a web browser back button [3]: Unlike commonly used Back Button systems, their back button will not lose pages in history. Although the paper is from 1999 the back button is still present in all browsers.

Why is it used anyway? The back and forward buttons are simple to use, even without completely understanding the underlying model. The two buttons do not take much space on a screen.

4.2 History Representation

The common stack-based back button concept implicates loss of history states. Greenberg et al. developed four methods to avoid these losses: (1) pure recency, (2) recency with adding spokes only, (3) recency with adding hubs and spokes and, finally, (4) recency with adding temporal ordering enhancement [3]. All methods work on a linear history list.

- Pure recency simply adds each page visited to the top of the list. Unfortunately, this results in a loop between the last two pages.
- Recency with spokes move a pointer through the history list. Recency with adding spokes only adds every newly visited page to the end of the list, regardless of the position of the pointer, any older duplicates are removed. This results in a long history list and requires a lot of clicks to go back to a desired page in the history.
- Recency with adding hubs and spokes adds both the current page and the newly visited page to the end of the list. This shortens the history list, but the temporal order is destroyed.
- The last method, recency with temporal ordering enhancement, records back steps in a second list. When a new branch is opened by visiting a new page, the second list and the new page are added to the end of the history list.



(a) page graph (b) stack-based (c) recency with temporal ordering enhancement

Figure 6: Comparing stack-based history versus recency with temporal ordering enhancement history.

Figure 6 compares this last approach to the usual stack-based approach.

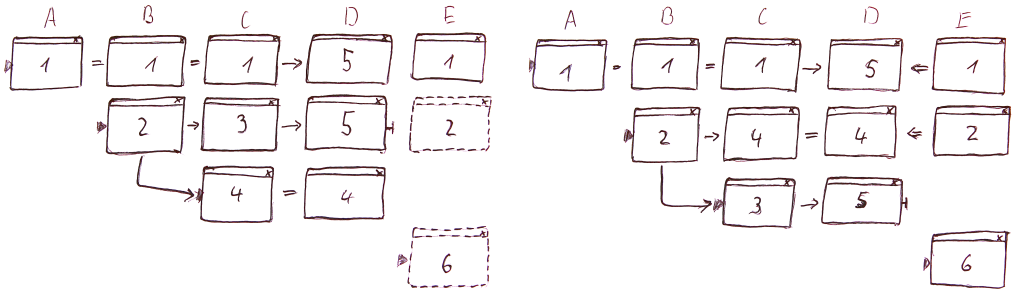
All these methods prevent from losing history states. However, the complexity increases, the history list become larger, caused by extra navigation through intervening children to navigate back up the tree. And the ease of use of the back function decreases because of the complexity, users may find the reordering of pages on the history list unpredictable. Greenberg et al. suspects this to be a minor problem, because most users will use the Back and Forward buttons in a mechanical fashion, repeatedly clicking the buttons until they recognise the desired page.

However, when Greenberg et al. wrote the paper, browsers did not have tabs implemented. Using tabs while browsing may be a practical work around to prevent loss of history states in stack-based histories.

4.3 Comparison to TIP

Obviously Greenberg et al.'s improvement is practicable to every software using back and forward buttons like web browsers. Our information and recommendations service are typical examples of this category. They mainly provide textual and pictured information.

The text-based TIP services can be easily adapted to the back button improvement. However the map service behaves different. Thus we decide to adapt the map view from our TIP scenario to the back button improvement.



(a) unreachable state B (b) reachable state B

Figure 7: Browser window history with reachable and unreachable state B.

Figure 7 shows two versions of this adoption.

Greenberg et al. considered a single window, single tab web browser, but the TIP system is a multi-service system. So we have to use multiple instances of web browser to rebuild our scenario, where each page represents a point-of-interest.

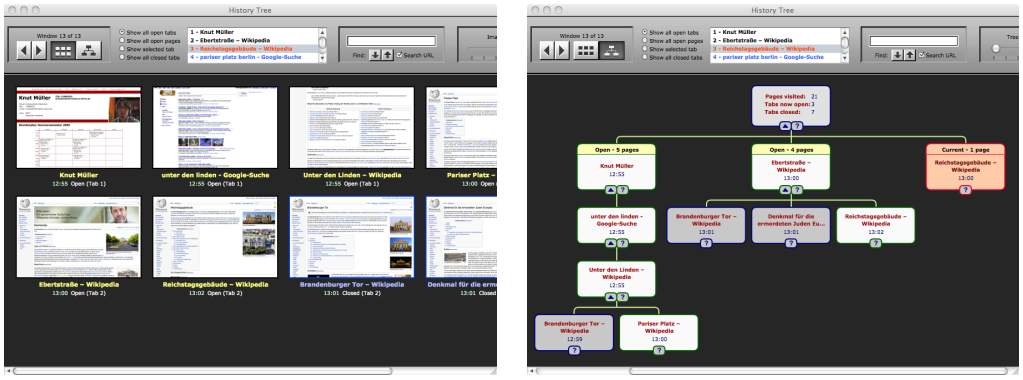
For example: Lisa visits the boulevard Unter den Linden in Page 1 (see Figure 7). Next day, Lisa starts her trip at Ebertstraße (Page 2 in a new window). There she chooses first to walk to Holocaust Memorial (Page 4) and then to Brandenburg Gate (Page 3). One of these pages needs to be opened in a new window (compare Figure 7(a) with 7(b)).

From Brandenburg Gate (Page 3) she walks to Pariser Platz (Page 5). At Pariser Platz she can see the boulevard Unter den Linden and remembers yesterday's trip. She closes this Page 5 and follows from Unter den Linden (Page 1) to Pariser Platz (Page 5).

Moving back to B requires backwards steps in two windows. This is not always possible: If we have closed the "wrong" window (see Figure 7(a)) we can not reach B again.

When Lisa reaches B, she recognises a recommendation and opens the Reichstag building (Page 6) in a new window.

The Back Button system stores each page opening in its history. Closings are not remembered: When the browser window is closed the history is lost. With the back button improvement each past state within a linear history can be reaccessed, past states in different branches too. However the history belongs only to one window without tabs. Thus there is no global history. In contrast to the stack-based behaviour, the history list becomes larger, so that navigating backwards takes more steps and the behaviour is more complex, so that it might not be understood.



(a) Grid of thumbnails view. (b) History tree view.

Figure 8: The History Tree window.

5 History Tree: Divergent History

5.1 System Overview

History Tree is an add-on to the firefox web browser written by Solomon [11]. It provides an extra window to the browser, extending its history views by a grid of thumbnails view and a history tree view (see Figure 8).

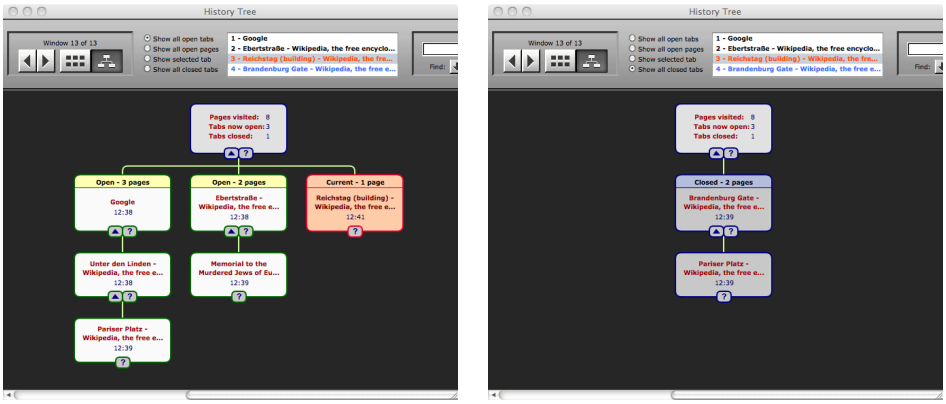
The add-ons menu bar provides controls to switch between these views and to select one of the open browser windows. Additionally, the user can filter the pages of a window by selecting a tab from a list of tabs or by states of tabs and pages. The main area shows the history tree.

5.2 History Representation

The History Tree add-on introduces three states of tabs and pages and relates them to colours: current page or tab – orange, closed pages and tabs – blue and not reachable by back button – grey.

The root of the history tree represents the current browser window. All its children are pages visited in this browser session (see Figure 8(b)). The first generation children of the root also represent the tabs of the current browser window. Branches in the history tree are results from hub and spoke navigation. Closed tabs can be displayed by selecting the option “Show all closed tabs” in the menu bar.

However, the History Tree add-on has no global history, just as the Back Button system: A newly opened window creates a new history tree and will not integrate into the existing tree.



(a) all open tabs view (b) all closed tabs view

Figure 9: The History Tree window with open and closed tabs view.

5.3 Comparison to TIP

To adapt the TIP scenario to the History Tree system we can use the same adaptation to the Back Button system (compare Figure 7(b) to Figure 9).

However, when a new window is opened in the back button adaptation we now open a new tab, so that we can use the advantages of the History Tree system.

Yesterday, Lisa visited the boulevard Unter den Linden in Page 1 (see Figure 7(b) and 9). Next day, Lisa starts her trip at Ebertstraße (by opening Page 2 in a new tab). There she chooses first to walk to Holocaust Memorial (Page 4 in same tab) and then to Brandenburg Gate (Page 3 in a new tab).

From Brandenburg Gate (Page 3) she walks to Pariser Platz (Page 5). At Pariser Platz she can see the boulevard Unter den Linden and remembers yesterday's trip. She closes this Page 5 and follows from Unter den Linden (Page 1) to Pariser Platz (Page 5).

Moving back to B requires backwards steps in two tabs. There her virtual walk brings her to Reichstag building (Page 6 in a new tab).

The History Tree add-on stores in its history all page and tab openings and tab closings within one browser window. Thus each page of a single browser window can be revisited. However there is no global history across multiple browser windows and the history is lost by closing its browser window. Although the system saves states from multiple tabs, no global back step across multiple tabs is supported. To reset the browser to a specific tab configuration, it might be necessary to move backwards in more than one tab. Additionally the History Tree add-on needs an extra window, which consumes much space on a screen.

	open	close	move	branch	split	merge	focus
Flatland	yes	yes	yes	yes	yes	yes	no
History Tree	yes	only tabs	yes	no	no	no	no
Back Button	yes	no	no	no	no	no	no

Table 1: Actions captured by the analysed systems.

6 Suggested Improvements for TIP

In the previous three sections we analysed three different history systems. In this section, we use insights from analysing the systems to suggest adaptions to the TIP system. We present our suggestions according to selected aspects of a history system.

Layers Layers are sub-histories or local histories for parts of the overall system. Several layers may be combined to an overall global history.

The Flatland system uses multiple local histories, one for each segment, and combines them to a global history for the whole whiteboard. The History Tree system and the Back Button system use one history for each browser window. These local histories are completely independent; a global history does not exist. Within the History Tree system, the history of each browser window is stored as a tree and displayed in two separate trees: One tree contains all open tabs and another tree contains all closed tabs.

The TIP system consists of several services effecting each other, each with a local history. However the TIP system should interact as one single application, thus we need a global history. We suggest a multiple layer history system which construct a global history out of the local histories.

History Content We identify the following actions that may be captured: open, close, move, branch, split, merge and change of focus. None of the the analysed systems capture all these actions; changes of focus are not captured by any system (see Table 1).

The Flatland system captures create, delete, move, split and merge actions of segments. Provided that the Flatland history behaves like the Timewarp history, the branch action is also stored Due to the fact that all segments are displayed at once, changes of focus are changes of the user's view direction, which are not detectable by Flatland. Creation and deletion of segments can be considered as open and close actions. Moving is a change of location of a segment on the whiteboard.

The History Tree system captures open, close and branch actions of pages

and close action of tabs. When a page is opened the History Tree system stores the page as a leaf of the previous page. Tabs are considered as pages without a previous page. They are children of the root which represents the browser window.

The Back Button system only stores page openings. Earlier representatives of the page are deleted from the list.

For the TIP system we suggest to capture open, close, move, branch and change in focus actions. Open and close actions especially belongs to text-based services like the information and recommendation service. The move action can be sub-divided into a physical movement and a virtual movement of the user. While virtual movement is generated when the users navigate to points-of-interest different to their current physical location. The branch action opens a new branch in the history tree of the TIP system. Branching is caused by browsing in one of the text-based services or by virtual back and forward movements.

We suggest to not only capture actions by order, which is sufficient for the three analysed systems, but also by time. The time reference is needed to display history entries by imprecise time specification.

Ordering of Entries History entries need to be sorted according to a specific order to display them. All analysed systems store or order their history events by time of triggered actions.

The Flatland system additionally stores changes of locations. But it only uses these changes to reset a past configuration of segments on the whiteboard, not for ordering.

Within the TIP system, it may be important to also remember which service caused a history entry at which location. Thus the history can be organised and browsed by time, location and service.

Complexity The challenge is to construct a comprehensive history presentation.

The Back Button system uses a straightforward interface; just two buttons, back and forward. Unfortunately unexpected pages in the history may confuse the user: It is possible that the target of the back button is not the parent of the current page like in stack-based back button implementation (compare Figure 6).

This particular risk of confusion is faced by displaying the history as a tree, like in History Tree system. However the history tree can grow in size so that it will not fit on the screen. Especially when it is displayed on a small screen like installed in mobile devices. The user may not get an overall

image of the history content, which might be confusing.

Due to the large screen size, the Flatland system can display its overall system state at once. However displaying branches in history will result in history trees with all their disadvantages.

On the map of the TIP system, all visited points-of-interest are displayed. The physically visited are connected by streets and paths. These streets and paths correspond to edges of a graph and the points-of-interest correspond to nodes of a graph. With the start point of a trip as root, this graph is a tree. Thus we suggest to use the map of the TIP system as basis for a history tree representation of the TIP history. To compensate the heavy screen consumption of a tree, we suggest to underlay the route with content of the TIP services, which was active at a particular position on the route. If the position is not unique the content is displayed ordered by time.

Overall we suggest to implement the TIP system with a multiple layer history consisting of several local histories, one for each service, and a global history, which connects all actions.

As actions we suggest to capture open, close, move, branch and change in focus actions.

The history entries of the TIP system should be browse-able (imprecise) time, location and service, so each interaction with the TIP system needs to store, which service caused the entry, at which location, at which time.

To keep the complexity handy we suggest an interface, which display the route of a trip underlayed with the content of the active service. Then the TIP history could be traversed first by location and second by time.

7 Conclusion and Future Work

In Section 1, we identified three groups of challenging questions, which we now attempt to answer.

How many histories do we need? Should each service use its own history or should all services share a global history? In general we need one system-wide history. We would like to present to the user at most two histories at a time: the local history of the current service and the system-wide global history.

In storage we need access to one global history and to as many local histories as we have services. Whether the global history is stored and subdivided into several local histories or several local histories are stored and combined to one global history, does not matter.

How to manage histories? Is it useful to cluster histories by locations or by services or both? Is a time-based order encoded into histories (as typically done in web browsers)? The TIP system should capture each user interaction and store time, physical location, virtual location and service id. The physical location describes the location of the user in the real world. The virtual location describes where the user browses the TIP system on the virtual map when interaction occurred. The service id identifies the service, which triggered the history entry.

We expect that locations especially in combination with content from a service are remembered better by humans than time references. We may also store interactions with the history itself, even these interactions can help the user to remember certain situations. The TIP history could be implemented as a TIP service.

Do users understand the history behaviour? Will users expect and understand the displayed behaviour? Is a user aware of different modes in a service (physically visited versus virtually visited points-of-interest)? We will have to evaluate whether the users will understand the behaviour of the new TIP system. A user study could give us insights into whether locations combined with content of services are remembered better than time references and if the user is aware of the difference between virtual and physical location within the TIP system.

Future Work This paper provides an overview and comparison of concepts that have been proposed to guide users through interaction histories. We gained insights into history design, which we now can use for designing and implementing interaction histories for a new location-based TIP system.

We plan to evaluate these new TIP systems towards the previous three groups of questions.

References

[1] W. K. Edwards, T. Igarashi, A. LaMarca, and E. D. Mynatt. A temporal model for multi-level undo and redo. In *UIST '00: Proceedings of the 13th annual ACM symposium on User interface software and technology*, pages 31–40. ACM, 2000.

[2] W. K. Edwards and E. D. Mynatt. Timewarp: techniques for autonomous collaboration. In *Proceedings of the SIGCHI conference on Human factors in computing systems*, CHI '97, pages 218–225. ACM, 1997.

[3] S. Greenberg and A. Cockburn. Getting back to back: Alternate behaviors for a web browser's back button. In *Proceedings of the 5th Annual Human Factors and the Web Conference*, June 3 1999.

[4] A. Hinze and G. Buchanan. The challenge of creating cooperating mobile services: experiences and lessons learned. In *ACSC '06: Proceedings of the 29th Australasian Computer Science Conference*, pages 207–215. Australian Computer Society, 2006.

[5] A. Hinze and S. Junmanee. Providing recommendations in a mobile tourist information system. In *Information Systems Technology and its Applications, 4th International Conference ISTA'2005, 23-25 May, 2005, Palmerston North, New Zealand*, volume 63 of *LNI*, pages 86–100, 2005.

[6] A. Hinze, P. Malik, and R. Malik. Interaction design for a mobile context-aware system using discrete event modelling. In *Proceedings of the 29th Australasian Computer Science Conference - Volume 48*, ACSC '06, pages 257–266. Australian Computer Society, 2006.

[7] A. Hinze and A. Voisard. Location- and time-based information delivery in tourism. In *Proceedings 8 th International Symposium in Spatial and Temporal Databases (SSTD)*, pages 489–507. Springer, 2003.

[8] A. Hinze, A. Voisard, and G. Buchanan. Tip: Personalizing information delivery in a tourist information system. *Journal on Information Technology and Tourism*, 11(4), 2009.

[9] T. Igarashi, W. K. Edwards, and A. LaMarca. An architecture for pen-based interaction on electronic whiteboards. In *Proceedings of the Working Conference on Advanced Visual Interfaces*, pages 68–75. ACM, 2000.

[10] E. D. Mynatt, T. Igarashi, W. K. Edwards, and A. LaMarca. Flatland: new dimensions in office whiteboards. In *CHI '99: Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 346–353. ACM, 1999.

[11] N. Solomon. Firefox add-on: History tree. `http://normansolomon.org.uk/histTreeHelp/tutorial.html`, July 2009.

[12] L. Tauscher and S. Greenberg. How people revisit web pages: Empirical findings and implications for the design of history systems. *International Journal of Human Computer Studies, Special issue on World Wide Web Usability*, 47(1):97–138, 1997.

[13] C. Thunack. Konzeption einer Benutzerschnittstelle für ein kontextsensitives Informationssystem. Diplomarbeit, Fachhochschule für Technik und Wirtschaft Berlin, 2008.