Grant Agreement No.	**ICT-2009-270082**
Project Acronym	**PATHS**
Project full title	**Personalised Access To Cultural Heritage Spaces**

D 4.3

Additional Applications

Authors:	George Chrysochoidis, i-sieve
Contributors:	Paul D. Clough and Mark Stevenson, USFD Eneko Agirre and Arantxa Otegi, UPV/EHU Kate Fernie, MDR

Project funded under FP7-ICT-2009-6 Challenge 4 – "Digital Libraries and Content"	
Status	Final
Distribution level	Public
Date of delivery	24/01/2014
Type	Report
Project website	http://www.paths-project.eu
Project Coordinator	Dr. Mark Stevenson University of Sheffield

Change Log

Version	Date	Amended by	Changes
0.1	12/12/2014	Paul Clough (USFD)	Outline + recommender system
0.2	13/12/2014	Eneko Agirre, Arantxa Otegi (UPV/EHU)	Additional recommender system
0.3	23/12/2014	George Chrysochoidis, i-sieve	Mobile application
0.4	16/01/2014	George Chrysochoidis, i-sieve	Additional text
0.5	20/01/2014	Kate Fernie, MDR	Edited text
1.0	24/01/2013	Mark Stevenson, USFD, Kate Fernie, MDR	Final

Contents

1. Executive Summary

The objective of PATHS has been to establish a flexible system architecture that allows the investigation of functionalities supporting exploration and discovery in cultural heritage collections. The initial deployment of PATHS prototypes focused on browser-based applications for access via desktop machines. In subsequent development, additional applications have been created to demonstrate the implementation of functionalities from the PATHS system on mobile devices. Additionally, research has been carried out on the implementation of recommendation services.

In recent years there has been an explosion in the use of mobile devices, such as smart phones and tablet computers for accessing all kinds of information. The 2011 "Culture on the go" report investigated Europeana's usage with reference to this explosion in mobile access. The report concluded "it might be expected that mobiles will extend the reach of websites and draw in a wider range or people (more digital natives perhaps)" and mobile devices will become a very significant means of access to Europeana; the impact on other cultural websites is likely to be similar.

This deliverable describes the PATHS mobile application. After taking into consideration the findings of the "Culture on the go" report a decision was taken to develop the application on an Apple iPad platform. An application was designed that included selected functionality from the desktop version, including path following and search functionalities, which are typically offered by cultural institutions in mobile applications. The PATHS mobile application will be made available through Apple's App store. A Facebook application to share Paths was envisaged in the work plan. The project has experimented with reusing the software components developed for the mobile application in Facebook to give users the ability to explore PATHS functionality and to publish items on their timeline.

In addition to the mobile application, this deliverable also describes the recommender system developed for the PATHS desktop application. Recommender systems are increasingly used to bring relevant content to users' attention and are part of a set of techniques for personalizing services to the needs of individual users. User interaction data can be used to implement "people who viewed this also viewed" functionality; however, the sparseness of data available from cultural heritage information systems means that different approaches need to be explored. In PATHS, content-based approaches using the semantic similarity between items were used to suggest content to users. In addition, the deliverable describes a series of experiments that were carried out to explore and evaluate alternative log-based and content-based recommendations. The experiments show that a combination of those alternatives leads to better results

2. Introduction

The objective of PATHS has been to establish a flexible system architecture that allows the investigation of functionalities supporting exploration and discovery in cultural heritage collections. The initial deployment of PATHS prototypes focused on browser-based applications for access via desktop machines. In subsequent development, additional applications have been created to demonstrate the implementation of functionalities from the PATHS system on mobile devices. Additionally, research has been carried out on the implementation of recommendation services.

2.1. Culture on the go

The “Culture on the go” report published on Sept. 2011 investigates Europeana’s usage in reference to the recent explosion in mobile access [Ciber, 2011]. The report concludes by stating that: “The real change for Europeana has not been in smart-phones but in tablets. The iPad has achieved a breakthrough making the tablet (big touch-screen, un-encumbered by wires or peripheral devices) a popular platform where previous attempts have failed. It redefines the consumer 'personal computer' experience; in fact it is an “interweb” access-device rather than a computational machine. It makes apparent the difference between telephone/internet access and PC as office machine (even if office at home). Tablet-oriented interfaces are influencing design of PC interfaces e.g. Gnome3, KDE4. The iPad has shown the way to go and is now being chased by rivals such as Android.”

2.2. Native and Web Mobile Applications – A Brief History

Within a week of the original iPhone launch in June 2007, the first iPhone DevCamp was held in San Francisco, CA. When the iPhone was originally released, there was no SDK available. Therefore, all the original iPhone applications were web-based.

When the iPhone first launched, the iPhone’s OS was less powerful than the newer phones on the market today, and being on the EDGE network, downloads were painfully slow. With these limitations, a main focus in developing applications was ensuring less than 10 KB downloads, less than 10 KB of images, and less than 10 KB of JavaScript. At the first iPhone DevCamp, participants developed their own documentation, helping each other gain the skills to develop web-based iPhone applications. Originally, there was no default on Orientation Change event. Instead, a timer was added to regularly check the phone’s orientation, and switched CSS classes with JavaScript based on the returned value.

During that first weekend after the iPhone’s launch, Joe Hewitt wrote iUI, the first JavaScript and CSS library for the iPhone and shared it with the developers present. Hewitt, Nicole Lazarro, and three others created Tilt, the first iPhone game that used iPhone's motion-sensing capacity. Dori Smith created iPhone Bingo, a purely JavaScript iPhone game. Richard Herrera, Ryan Christianson and Wai Seto created Pickleview, a Twitter/Major League Baseball AJAX mash-up that allows users to virtually watch any baseball game and tweet about it. For the first time, multiple background images were used, border images, CSS3 selectors, and opacity without having to worry about supporting a multitude of browsers, browser versions, and operating systems.

For the first nine months of the iPhone's life, there were only web applications and Apple-controlled native applications: there was no native iPhone app development in the wild. Because of bandwidth limitations and a dearth of Apple developer documentation, iPhone web applications didn't skyrocket. Because of the inability of the iPhone WebKit Safari browser to access native iPhone OS features, web application development for the iPhone did not take off. Application development for the iPhone finally skyrocketed with the release of the SDK.

The iPhone SDK was first released on March 6, 2008. The iPhone SDK allowed third-party (i.e., non-Apple) developers to make applications for the iPhone (and later the iPod touch and iPad), with availability in the App Store following in July of 2008. With the release of the SDK, and the opening of the App Store, not to mention the ability for developers to make money from selling their Apps in the App Store, the focus of iPhone development quickly and wholeheartedly switched to building native iPhone applications. The fact that the focus of iPhone application development has been mostly on the development of native iPhone applications since the release of the SDK makes sense to a great extent. In 2008, the limitations of web-application over native-application development discouraged focusing on web apps, as the following lists show:

Cons for web apps in 2008

- 10 MB file-size limit in iPhone Safari
- Lack of storage for data via web apps, and very limited cache
- Lack of support for most CSS3 and HTML5 features in not only Safari for the iPhone, but all browsers

Pros for native apps in 2008

- Ease of development using XCode
- Ability to sell applications in the App Store

In 2013, however, the tables have turned. The arguments for developing web apps versus native apps has caught up, if not surpassed, the arguments against, as the following lists show:

Pros for web apps in 2013

- Easier to build and iterate (developers can push multiple times a day, providing for quick iteration)

- Uses existing skills in HTML and CSS (building upon skills rather than requiring developers to master completely different ones)
- Same technology, same platform
- Potential to be cross-platform

Cons for native iPhone apps in 2013

- 3-week+ approval process for distributing in the App Store
- Risk of censorship of content and non-inclusion by application stores
- $99+ annual Apple Developer membership fee, plus 30% sales fee
- Long waits to push code changes to production, as well as for users to sync and update their application (with HTML5, your changes are live immediately)

2.2.1. PhoneGap - Bridging the Gap Between Native and Mobile Applications

PhoneGap is a mobile application development framework, based upon the open source Apache Cordova project that provides a solution for building cross-platform mobile apps with standards-based Web technologies like HTML, JavaScript, CSS.

The resulting applications are hybrid, meaning that they are neither truly native (because all layout rendering is done via web views instead of the platform's native UI framework) nor purely web-based (because they are not just web apps, but are packaged as apps for distribution and have access to native device APIs).

2.3. Facebook

In the early summer of 2007, when Facebook opened its doors to free registrations from the wild and untamed Internet, a lot of people started using it and became part of a large social graph, which is a model and codification of user relationships with other people. A little later Facebook released an API to let developers create and manage custom web applications. Suddenly Facebook became a massive base of users, connected with their personal circles of friends and colleagues, which could lead to applications that operate within this new flourishing ecosystem. More recently there has been a decline in the popularity of Facebook particularly amongst younger people.

2.4. Recommender systems

Increasingly recommender systems are being used to assist users with information discovery by bringing relevant content to users' attention. They are part of a wider set of techniques for providing personalization: the tailoring of systems or services to the specific needs of individual users or communities [Smeaton and Callan, 2005; Adomavicius and Tuzhilin, 2005]. Recommendation mechanisms provide advice on objects depending on the user context or profile. They can be broadly classified by the strategy they employ (content-based or collaborative filtering) and by the recipient of the recommendations (individual user or group recommendations). Recommender functionality (and personalization more generally) has been proven useful when providing information access to cultural heritage [Ardisson, et al, 2012].

The PATHS project is investigating ways of assisting users with exploring a large collection of cultural heritage material taken from Europeana[1], the European aggregator for museums, archives, libraries, and galleries [Agirre, et al, 2013a; Fernie, et al, 2012]. A prototype system has been developed that includes novel functionality for exploring the collection based on Google map-style interfaces, data-driven taxonomies and supporting the manual creation of guided tours or paths. Another aspect being explored is the use of recommendations to promote information discovery. To date we have been exploring non-personalized recommendations based on item-to-item co-occurrences. These provide recommendations of the kind “people who viewed this item also viewed this item.” Co-occurrence information (items that have been viewed consecutively in the same session) has been mined from a sample of Europeana logs to power the recommendations. Additionally, we provide links to “related items”, a form of content-based recommendation, based on identifying ‘similar’ items and classifying the type of relation.

In this report we describe our recommendation work to date, difficulties in implementing recommendations and our experiments to check alternative techniques to improve recommendations.

[1] Europeana website: http://www.europeana.eu/portal/

3. PATHS Mobile Application

After considering the “Culture on the go” report a choice was made concerning the development platform for the PATH’s mobile app. The iPad with its iOS operating system was clearly well ahead of all other rivals, especially non-iOS platforms, taking a total of 40% of all mobile users, as shown on the following figure taken directly from “Culture on the go” report.

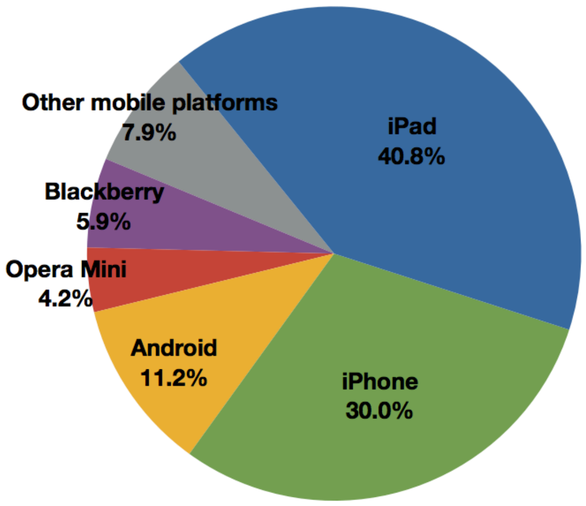


Fig 1. “Culture on the go” report: Europeana mobile page views by platform worldwide

The main purpose of the mobile app is to demonstrate the usefulness of components developed in the other work packages for users of mobile devices and social networking sites.

3.1. iPad - Device Specifications

At the time of development the iPad had reached its fourth model version. Device characteristics vary among the iPad models. The following table displays the various model specifications.

Model	**iPad (1st generation)**	**iPad 2**	**iPad (3rd generation)**	**iPad (4th generation)**	**iPad Mini**
Release date	April 3, 2010	March 11, 2011	March 16, 2012	November 2, 2012	November 2, 2012

Display	9.7 inches (250 mm) multi-touch display with LED backlighting and a fingerprint and scratch-resistant coating			7.9 inches (200 mm) multi-touch LED displa y
	1,024 × 768 pixels at 132 ppi		2,048 × 1,536 pixels at 264 ppi (Retina Display)	1,024 × 768 pixels at 163 ppi
Operating system	iOS 5.1.1	iOS 6.1.3		
Dimension s	9.56×7.47×0.528 in (243×190×13.4 m m)	9.5×7.31×0.34 6 in (240×186×8.8 mm)	9.5×7.31×0.37 in (240×186×9.4 mm)	7.87×5.3×0.28 in (200×130×7.1 m m)
Camera	N/A	720 p HD still and video camera 0.7 MP, 30fps and 5× digital zoom	1080p HD still and video camera 5 MP, 30fps and 5× digital zoom	

Table 1. iPad models technical specifications

Screen resolution and screen estate on the larger iPad models proves to be a key feature in a way that mobile users can now benefit from a wider range of functionality, similar to desktop users (Ciber, 2011). For this reason the PATHS mobile application was developed and tested mostly on the larger iPad models to take advantage of its screen size. Care has been taken to accommodate for the different screen resolutions, by creating all graphics in the various sizes needed for correct display.

3.2. PATHS Mobile Application Specification

The PATHS mobile iPad application will serve mainly for demonstrating the usefulness of components developed in the other work packages for users of mobile devices. Although it was initially planned to meet all “Required” or “Must” user requirements from the desktop web application, some parts, like path creation, were left out as they would be out of scope for our main demonstration purpose or too “heavy” to be useful in a mobile environment.

3.3. Technologies – Framework and Libraries

The following list includes the main technologies and frameworks used for implementing the PATHS mobile application.

Technologies:

- **HTML5**: It has been in the works for many years, since efforts began in 2004 on what was originally called Web Applications 1.0. While not finalized, some parts are fairly complete and already supported, often fully, by modern browsers. Modern, or A-grade, browsers include Safari, Chrome, Internet Explorer 10+, Firefox, and Opera. IE8 and older is not in this list. IE9 has some HTML5 support, but is a browser that is holding back the Web. So, while not all browsers provide support for HTML5, it is supported by all WebKit / Blink browsers, Opera Mobile,5 Firefox OS, and the new Windows phones. iPad's and generally iOS's web engine is based on WebKit.

- **JavaScript**: An interpreted computer programming language. As part of web browsers, implementations allow client-side scripts to interact with the user, control the browser, communicate asynchronously, and alter the document content that is displayed. Currently many JavaScript frameworks and libraries are available and can be used as higher level programming JavaScript interfaces, both for desktop and mobile use.

- **AJAX**: Asynchronous JavaScript and XML (AJAX) is a group of interrelated web development techniques used on the client-side to create asynchronous web applications. With Ajax, web applications can send data to, and retrieve data from, a server asynchronously (in the background) without interfering with the display and behavior of the existing page. Despite the name, the use of XML is not required (JSON is often used instead).

- **CSS**: The style sheet language used for describing the look and formatting of a document written in a markup language, used for styling web pages. The current version, CSS3, provides us with some new great features. CSS3 selectors, allow targeting just about every element on the page without adding a single class, including media queries to enable responsive web development.

- **SVG**: Scalable Vector Graphics (SVG) is an XML-based vector image format for two-dimensional graphics that has support for interactivity and animation. The SVG specification is an open standard developed by the World Wide Web Consortium (W3C) since 1999. SVG images and their behaviors are defined in XML text files. As XML files, SVG images can be created and edited with any text editor. WebKit-based browsers support SVG and can render the markup directly.

Frameworks and Libraries:

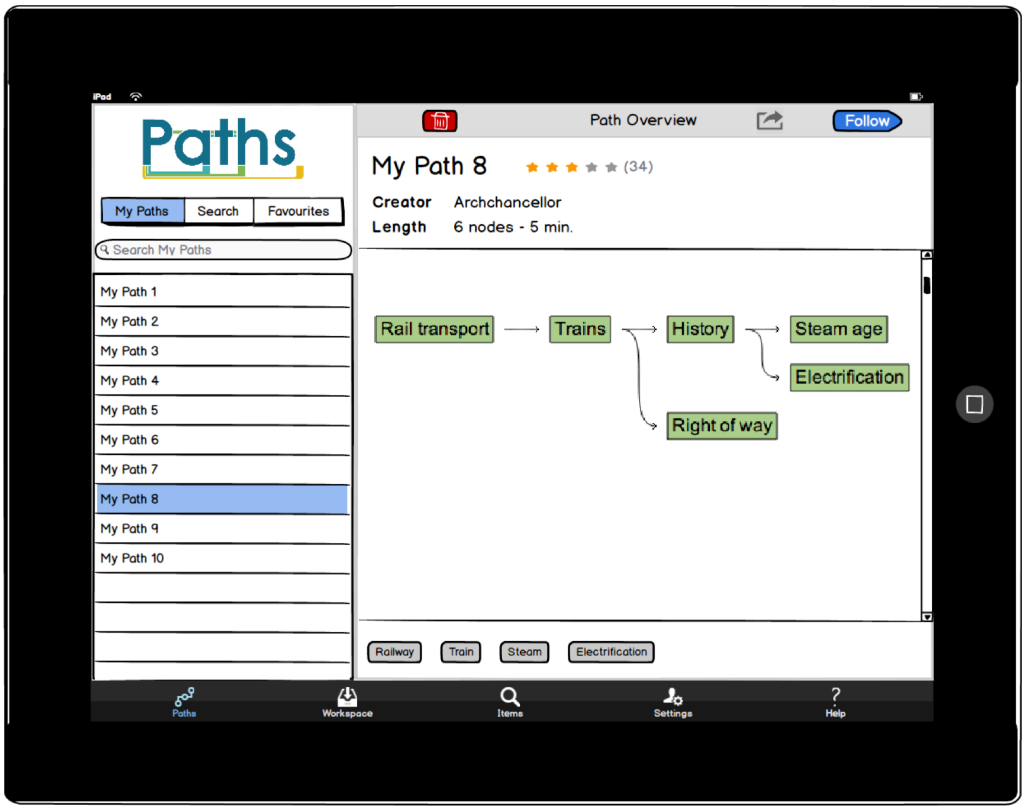
- **jQuery Mobile**: It is a touch-optimized HTML5-based web framework and user interface system designed to make responsive web sites and apps that are accessible on all smartphone, tablet and desktop devices. It is based on the popular jQuery JavaScript framework.

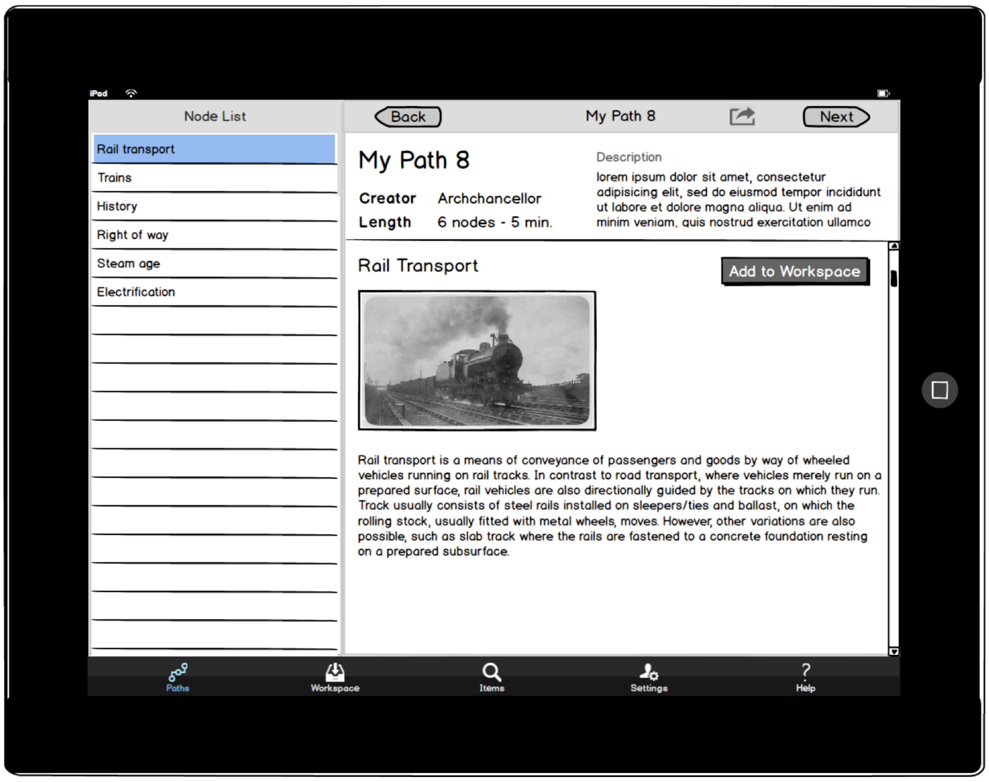
- **Raphaël**: It is a small JavaScript library that simplifies working with vector graphics on the web. Raphaël uses the SVG W3C Recommendation and VML as a base for creating graphics. This means every graphical object you create is also a DOM object, so you can attach JavaScript event handlers or modify them later.

- **FastClick**: It is a small JavaScript library used for eliminating the 300ms delay between a physical tap and the firing of a click event on mobile browsers. The aim is to make mobile web applications feel more responsive.

3.4. User interface - Functionality

An application mockup was created as a roadmap for the actual UI implementation. The following figures present some screens extracted from the mockup.





Rail transport
Trains
History
Right of way
Steam age
Electrification

My Path 8

Creator	Archchancellor
Length	6 nodes - 5 min.

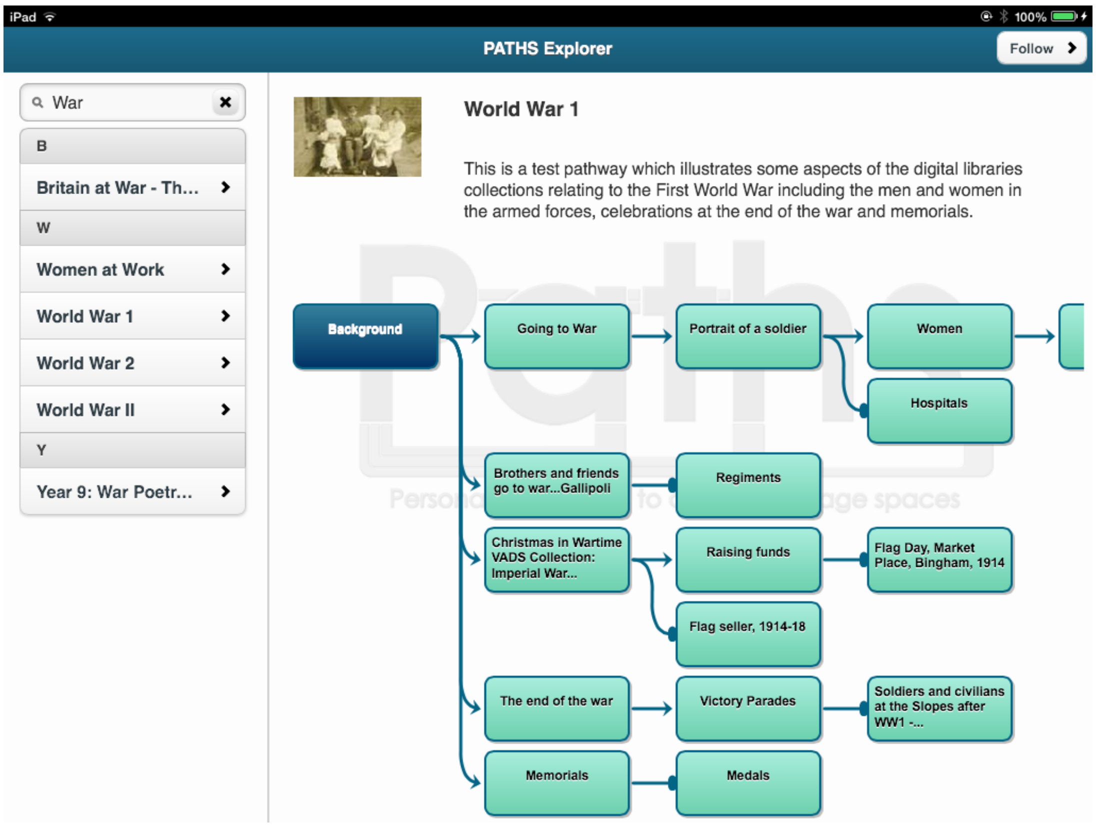
Description
lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco

Rail Transport

Rail transport is a means of conveyance of passengers and goods by way of wheeled vehicles running on rail tracks. In contrast to road transport, where vehicles merely run on a prepared surface, rail vehicles are also directionally guided by the tracks on which they run. Track usually consists of steel rails installed on sleepers/ties and ballast, on which the rolling stock, usually fitted with metal wheels, moves. However, other variations are also possible, such as slab track where the rails are fastened to a concrete foundation resting on a prepared subsurface.

Fig 2. PATHS mobile application mockup screens

The actual implementation of the application uses two main screens: one for searching paths and another one for following paths, as depicted in figures 3 and 4 below.

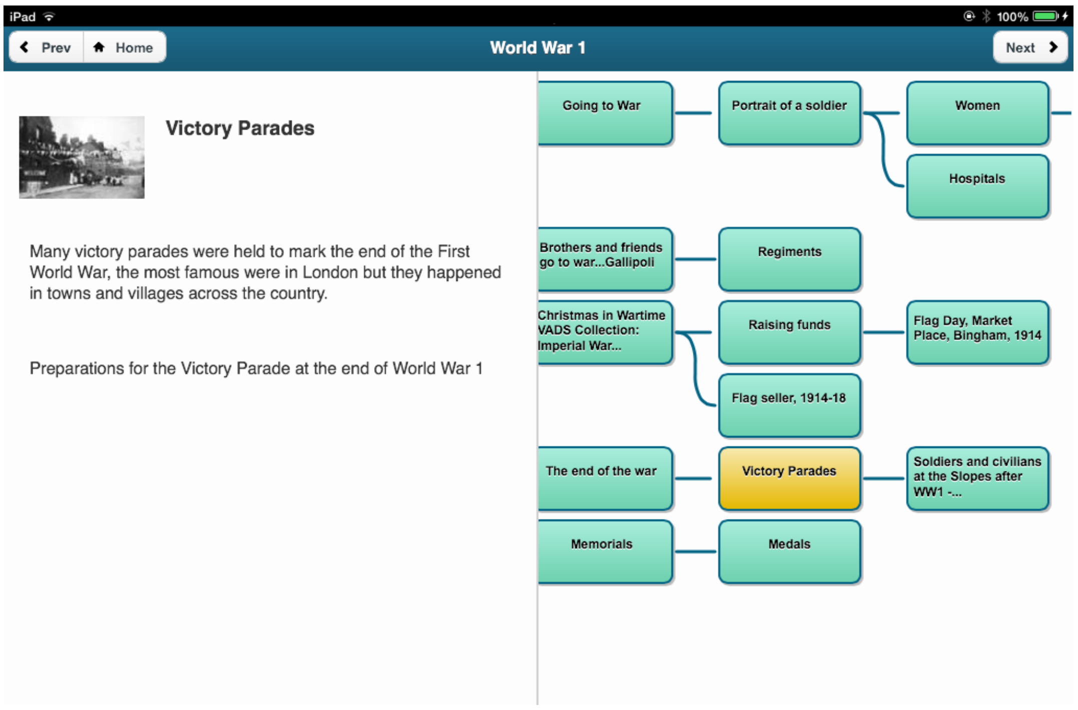


World War 1

This is a test pathway which illustrates some aspects of the digital libraries collections relating to the First World War including the men and women in the armed forces, celebrations at the end of the war and memorials.

Fig 3. Initial screen: Searching for paths

The searching paths screen, which is also the initial screen, includes a search box for and a filtered view list of the results. It also includes an image overview of the path, using nodes and connections between them. Each node is clickable. When the user clicks on a node a popup dialog appears where the node information is displayed. In this modal window the user has the option to click on a follow button, in which case he is transferred to the follow screen. The UI elements described above are shown in figures 5 and 6.

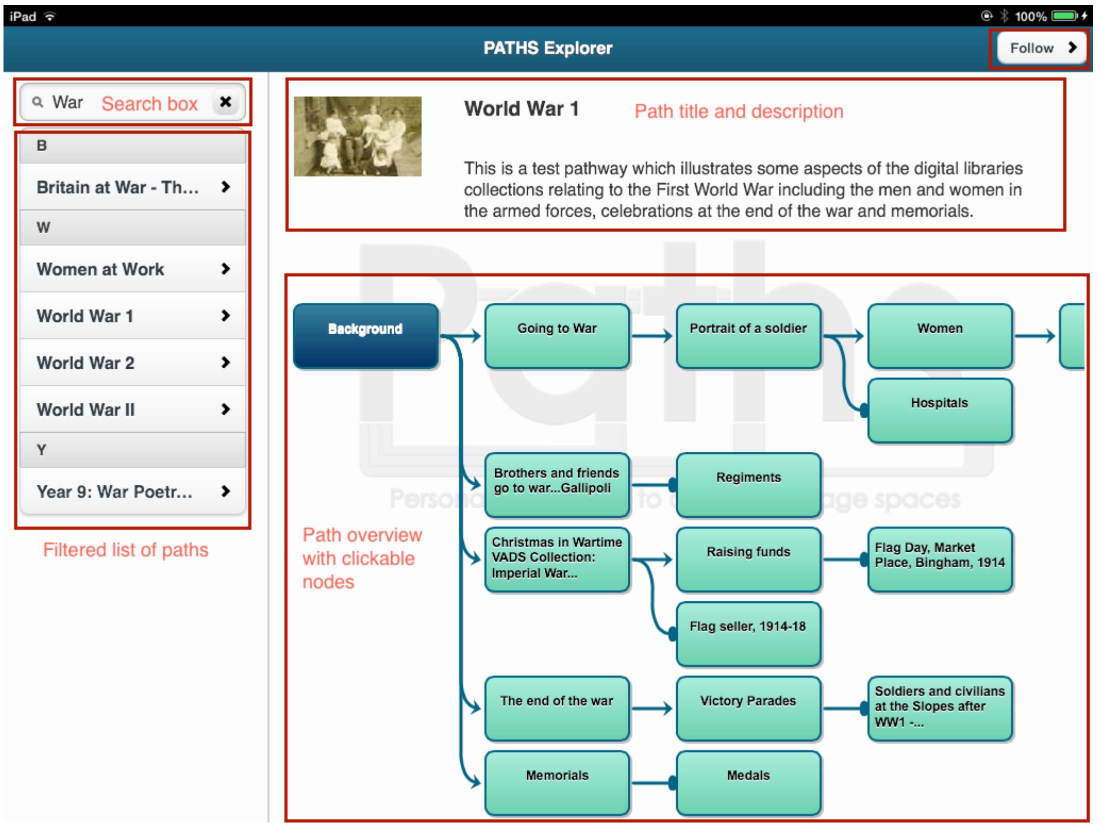

iPad
100%
Prev
Home
World War 1
Next
Going to War
Portrait of a soldier
Women
Hospitals
Brothers and friends go to war...Gallipoli
Regiments
Christmas in Wartime VADS Collection: Imperial War...
Raising funds
Flag Day, Market Place, Bingham, 1914
Flag seller, 1914-18
The end of the war
Victory Parades
Soldiers and civilians at the Slopes after WW1 -...
Memorials
Medals

Victory Parades

Many victory parades were held to mark the end of the First World War, the most famous were in London but they happened in towns and villages across the country.

Preparations for the Victory Parade at the end of World War 1

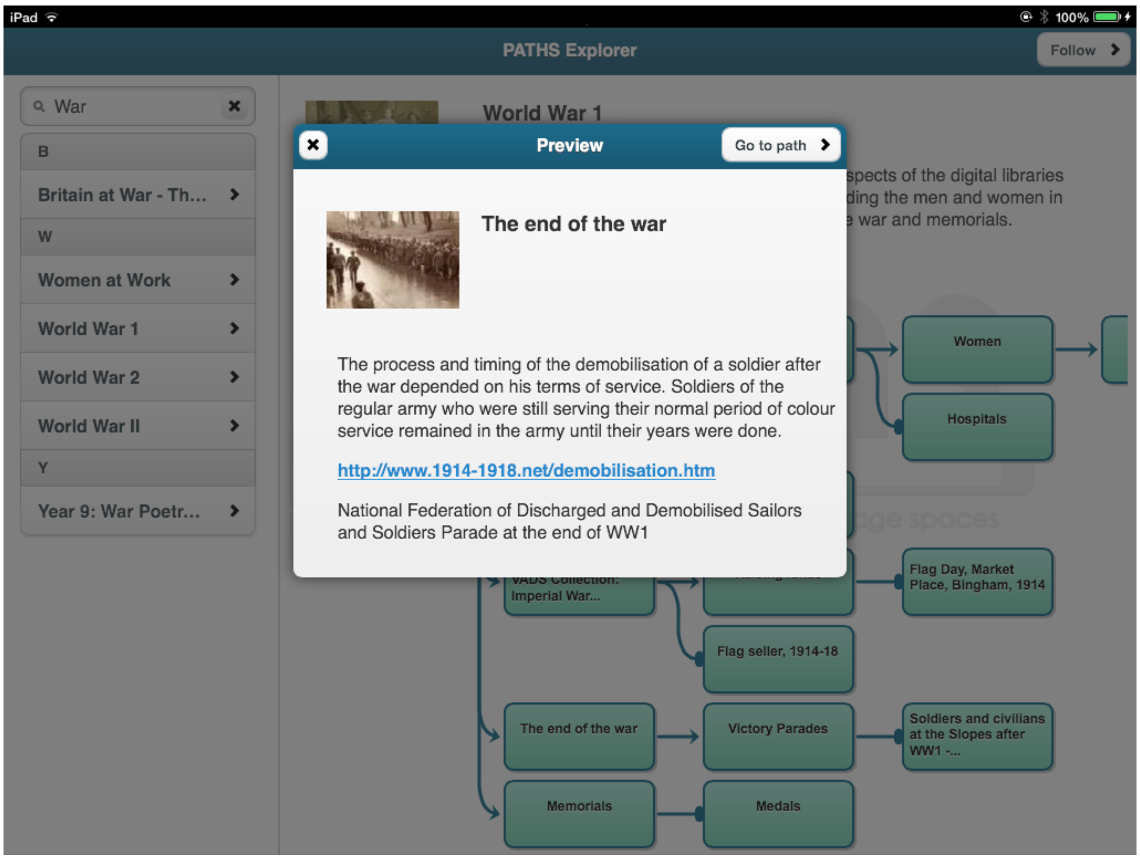
Fig 4. Following a path



World War 1

This is a test pathway which illustrates some aspects of the digital libraries collections relating to the First World War including the men and women in the armed forces, celebrations at the end of the war and memorials.

Fig 5. Search screen: Basic UI components



The end of the war

The process and timing of the demobilisation of a soldier after the war depended on his terms of service. Soldiers of the regular army who were still serving their normal period of colour service remained in the army until their years were done.

http://www.1914-1918.net/demobilisation.htm

National Federation of Discharged and Demobilised Sailors and Soldiers Parade at the end of WW1

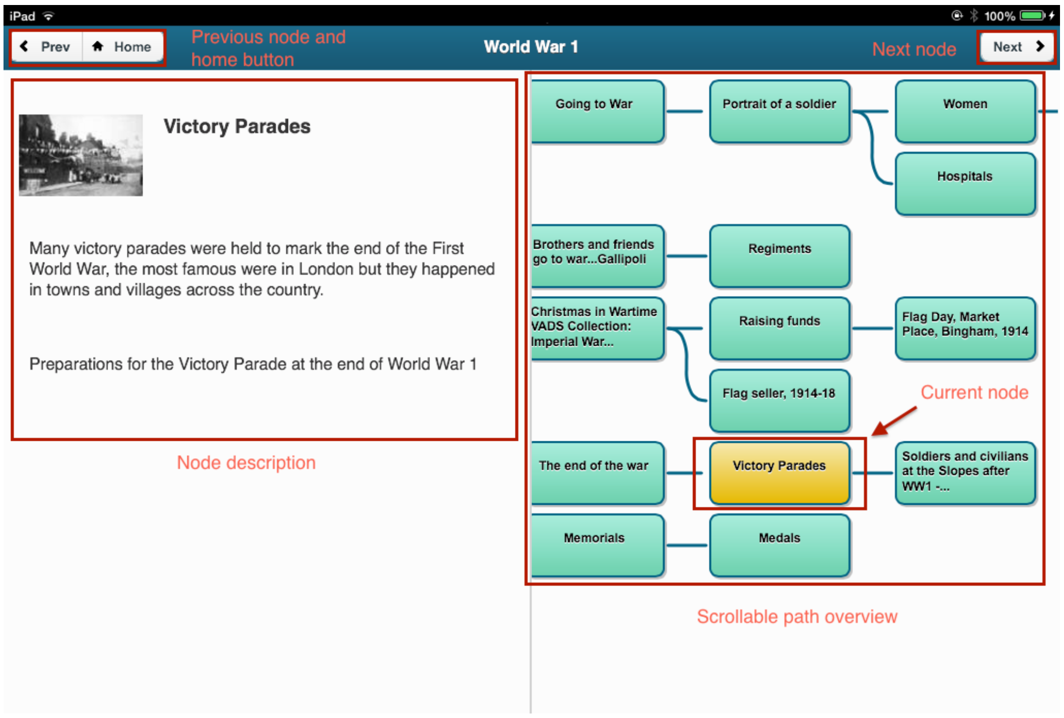
Fig 6. Node information in modal window

The follow-a-path screen includes a left pane where node information is displayed and a right pane with a scrollable overview image of path, which is also clickable. The current node is highlighted, as shown in figure 8. The user can follow a path sequentially or click on nodes to be able to hop to a more distant node inside the path.

When the next step leads to more than one node a modal window appears with a full list of available next nodes. The user is able to choose by clicking on the node's button and proceed with following the path. This function is shown in figure 8.

3.5. Deployment - The App Store

Additionally to developing the mobile web app, we used some services installed in an Apache server from where all requests to the PATHS services where made. This way we had no issues with cross-domain AJAX requests from within the mobile application. After some testing, bug fixing and some final changes, the application was wrapped inside the PhoneGap framework's code to be deployed within the App Store.

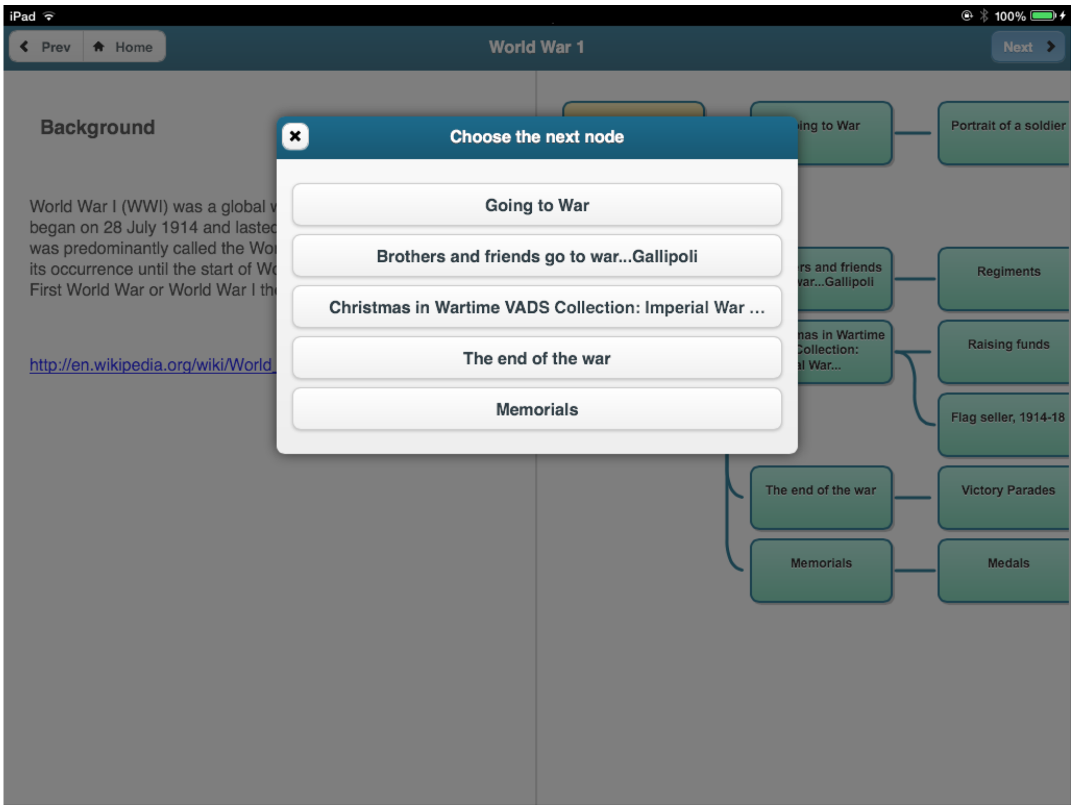


Victory Parades

Many victory parades were held to mark the end of the First World War, the most famous were in London but they happened in towns and villages across the country.

Preparations for the Victory Parade at the end of World War 1

Fig 7. Following a path: Basic UI components



World War I (WWI) was a global v
began on 28 July 1914 and lastec
was predominantly called the Wo
its occurrence until the start of W
First World War or World War I th

http://en.wikipedia.org/wiki/World_

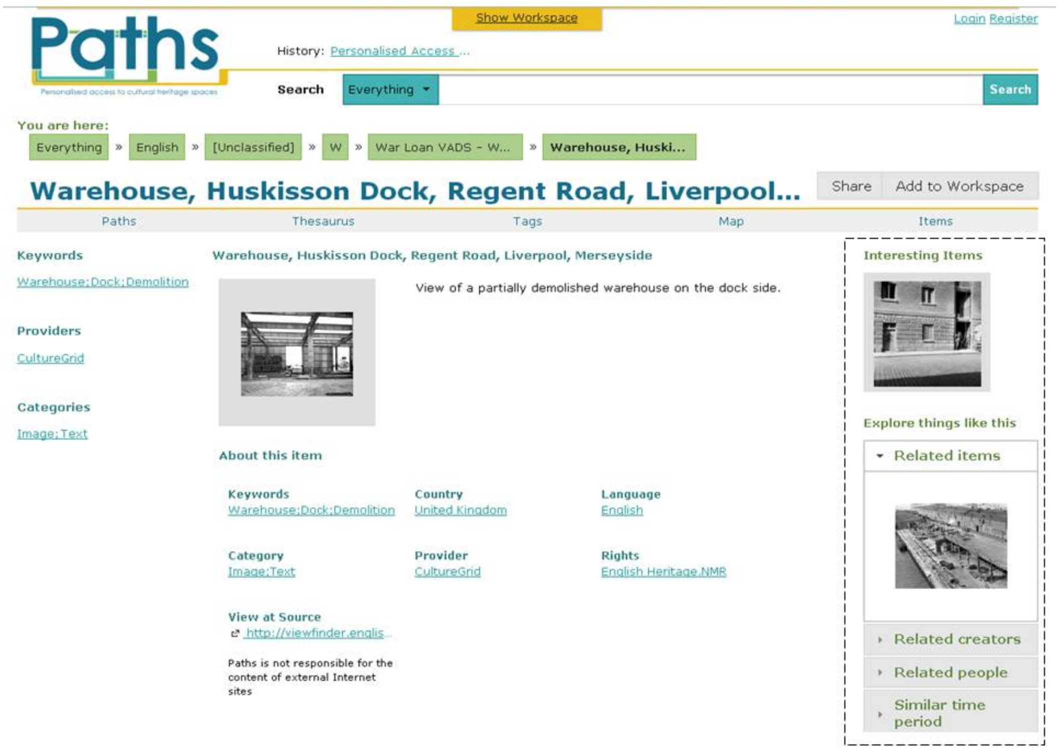
Fig 8. Choosing the next node

3.6. Facebook application

The PATHS project has investigated the development of a Facebook application. The aim has been to use the same technologies as those used in the mobile app, taking full advantage of reusable parts of code. The intention has been to demonstrate an application that would give Facebook users the ability to explore some of the PATHS functionality and publish related information in their timelines.

4. PATHS Recommender System

The current PATHS system interface is shown in Figure 9. At this point the user is viewing an item from the collection indexed in PATHS. This collection consists of approximately 540,000 items from items in Europeana that have English metadata. An additional 1.2 million Spanish items are in the process of being loaded. The prototype system as it stands supports "drill-down" exploration of the collection through the provision of a thesaurus, a tag-cloud, topic map, and faceted search. However, to support the exploration of individual items at a similar level, only "paths", manually curated narratives through parts of the collection, are available. To further improve the horizontal exploration facilities, particularly in areas of the collection not covered by "paths", we investigated non-personalized recommendations. This functionality is shown in Figure 9 in the dashed box (labeled as "interesting items"). This functionality was included in the implementation of Prototype 2 via the Recommendation Service.



Warehouse, Huskisson Dock, Regent Road, Liverpool...

Keywords
Warehouse;Dock;Demolition

Providers
CultureGrid

Categories
Image;Text

Warehouse, Huskisson Dock, Regent Road, Liverpool, Merseyside

View of a partially demolished warehouse on the dock side.

About this item

Keywords
Warehouse;Dock;Demolition

Country
United Kingdom

Language
English

Category
Image;Text

Provider
CultureGrid

Rights
English Heritage.NMR

View at Source
http://viewfinder.englis...

Paths is not responsible for the content of external Internet sites

Fig 9. An example screenshot of the PATHS prototype[2] when viewing an item

In the version of the recommendations implemented in the PATHS prototype we decided to focus on exploring non-personalized recommendations (i.e. the same recommendations are given to all users) rather than personalized recommendations. This was due to developing a reliable recommendation service in the time available for

[2] http://www.paths-project.eu/eng/Prototype

the launch of the second prototype. However, personalized approaches based on user models have also been explored as part of our research carried out in the PATHS project.

Implementing "people who viewed this also viewed this"

Functionality of the kind "people who viewed this also viewed this" is typically implemented using collaborative filtering techniques that make use of data derived from user interactions to suggest items of potential interest. However at the time the recommendation functionality was being developed, the PATHS system was not used heavily enough to provide sufficient user-system interaction data for analysis. Instead we experimented with extracting interaction data from the transaction logs of Europeana. The rationale for this is that because we wanted to explore item-to-item recommendations in PATHS, then we could make use of data about items in the PATHS collection that had also been accessed through Europeana, a much more widely used system than PATHS. We also combined this with content-based approaches to identifying 'similar' items and classifying the type of relation.

We implemented a mechanism to automatically download transaction logs for the main Europeana portal on a daily basis. Currently we use a 6-months sample of logs (1 Jan to 30 June 2012), but have collected almost 2 years of data. We applied standard pre-processing, including the removal of lines not relating to user actions (e.g. cascading style sheets and images), removal of non-human actions (e.g. robots), session segmentation (based on a 30 min timeout between actions) and classification of requests (e.g. viewing an item). A 30 min timeout period of inactivity was selected based on previous research [Tanasa and Trousse, 2004; Catledge and Pitkow, 1995], but we recognize that a fixed timeout period does have limitations for reliably detecting sessions and warrants further investigation [Jones and Klinkner, 2008].

In total, the processed data consisted of 14,164,379 requests (3,245,766 sessions), with 53.7% of requests for item views. We filtered out those sessions without any request for items that map to the PATHS dataset. This resulted in 102,525 sessions (3.2% of the initial log) with 208,584 item requests. For each session we extracted sequences of 2 viewed items (ignoring all other request types). For example, for the action sequence $item_1 \rightarrow item_2 \rightarrow search_1 \rightarrow item_3$ we would extract the sequences $item_1 \rightarrow item_2$ and $item_2 \rightarrow item_3$. We ignored pairs containing repeated items (i.e. $item_1 = item_2$). This resulted in 55,521 different pairs of items and an average of 1.82 recommendations per item.

Implementing "Related Items"

For the "related items" functionality, the similarity between each pair of items is computed using a state of the art approach based on Latent Dirichlet Allocation over the text, allowing users to quickly find related items when browsing. An evaluation dataset was crowd-sourced to enable us to assess this approach [Aletras, et al, 2013]. In addition, a typed similarity approach is implemented to determine the 'type' of the relation, such as similar author, location, date, event, people involved or subject. With this extra functionality, users know *why* the system is making the suggestion, an aspect considered as important to recommender systems [Sinha and Swearington, 2002]. The approach is a combination of simple similarity heuristics, based on the appropriate metadata fields, and a lineal regression [Agirre et al, 2013b]. The latter method improved the results considerably, obtaining second position among several contenders at an open evaluation exercise[3].

4.1. Discussion

Like most cultural heritage systems the amount of interaction data generated by users of the PATHS system is insufficient for implementing "people who viewed this also viewed this" functionality, due to data sparseness. Therefore, we exploit usage information from a more widely used system (Europeana), but restrict the data to only those items we index. However, even using data from a more widely used system we can only make recommendations for 10.3% of items in the PATHS dataset.

A further issue is that only 2,407 pairs of items (4.3%) are viewed more than once which may be the threshold at which recommendations are acceptable. Therefore, we are also working on extracting more pairs between items based on transitivity (e.g. for the sequence $item_1 \rightarrow item_2 \rightarrow item_3$ we could also assume a relation exists between $item_1 \rightarrow item_3$) and duality (e.g. for the pair $item_1 \rightarrow item_2$ we could also extract $item_2 \rightarrow item_1$). Another approach to deal with data sparseness could be to map each item to a semantic category and then make recommendations at higher levels than item, i.e. suggest pairs of items for the same subject that are viewed consecutively.

One approach we adopted in the current prototype is utilization of additional content-based recommendations. These "related items" help to alleviate the problems of insufficient usage data. Combinations of approaches are commonly used to overcome the limitations in using collaborative filtering and content-based approaches independently [Adomavicius and Tuzhilin, 2005]. Further work being planned includes evaluating recommendations in a controlled lab-based setting and field trials. Also, we are developing personalized recommendations based on a session-based user model (i.e. the user profile is built up during a session from items viewed) and using PageRank to identify items of interest as shown in the next Section.

4.2. Additional Recommender System Experiments

Even only non-personalized and log-based recommendations are provided by the prototype, some additional experiments have been carried out using other kind of information (apart from logs) and to provide personalized recommendations, as well. In

[3] Semeval 2013: http://ixa2.si.ehu.es/sts/

this section the implementation of these additional recommender system experiments are described.

Details about the evaluation results can be found in Deliverable D5.3, which will be made available at: http://www.paths-project.eu/eng/Resources.

Ideally, recommendations should be provided on different scenarios of the prototype. The recommender system should take into account if the user is on a general landing page or they are viewing an item. Similarly, the recommendations should vary depending on whether the user is logged into the system. If so, the user profile information should be taken into account when generating recommendations. Thus, every recommender system that we have implemented is able to work in these three different scenarios:

- No profile, item page: when the user is viewing an item, but there is no profile information because they have not logged into their account.
- Profile, landing page: when the user is at the general landing page and, as they have logged into their account, their profile information is available.
- Profile, item page: when the user is viewing an item and, as they have logged into their account, their profile information is available.

In each of these scenarios the input information given to the recommender system is different. Thus, in each case the originated recommendations are influenced by the pieces of information available. In the case of an item page, it is important to take into account which is the current item the user is viewing. There is no such information in the case of a landing page. In case profile information is available, the recommender system should consider this information as some extra information about user preferences. In our case, a profile is defined as a set of three items (i.e. a set of user's three favourite items, or a set of most visited items for the user, or the last three items that the user has visited). In short, these are the inputs for the recommender system in each scenario:

- No profile, item page: the current item.
- Profile, landing page: a set of three items.
- Profile, item page: a set of three items and the current item.

Note that following this, “no profile, landing page” scenario does not have any input information. That is the reason why we have not implemented the recommender system for this case.

Algorithms

A number of recommender system algorithms have been implemented based on different information and using several tools and resources.

Baseline-random
This system chooses randomly an item from the whole PATHS collection. It is the same for all scenarios.

Simple-keywords
The system uses Solr[4] search platform over the index created with the items from the PATHS collection. Keywords from *dc:title* and *dc:creator* fields of the items are used as a query, and the retrieved items are returned as recommendations. It depends on the scenario which items are used for querying:

- No profile, item page: Keywords of the current item are used.
- Profile, landing page: Keywords of the profile items are used.
- Profile, item page: The two above retrieved ranked lists are combined, summing the scores of the retrieved items. To give less importance to the profile items than to the current item, the scores of the profile items' rank are multiplied by 0.5.

Simple-similarity
The system is based on the similarity links. The similarity between items is computed using a state of the art approach based on Latent Dirichlet Allocation (Aletras et al, 2013) after analysing the metadata information of the items. As a result, each item in the PATHS collection is linked to its top 25 relevant items together with confidence values (more details about similarity links are given in Deliverable D2.1). These links with their scores are used by this recommender system as follows:

- No profile, item page: Return as recommendations the similar items of the current item.
- Profile, landing page: First, get the links of similar items for each one of the profile items. Then, combine the lists, summing the scores in case an item is repeated in more than one resulting list. Return as recommendations the highest scoring items.
- Profile, item page: Same as above, but combining the links of profile items and the links of the current item. Before summing, the scores of the similar items for the profile items are multiplied by 0.5.

Simple-categories
The system is based on the Wikipedia categories of the Wikipedia links found in the input items (this method is explained in more detail in [Fernando et al., 2012]) and it is implemented for different scenarios as follows:

- No profile, item page: Finds the set of all categories *C* that the current item belongs to. For each item in the whole collection, a score is assigned based on

[4] http://lucene.apache.org/solr/
4

how many of the categories of *C* the item belongs to. The highest scoring items are then returned as the recommended items.

- Profile, landing page: The same as above, but the set of all categories *C* belongs to the profile items, instead of to the current item.
- Profile, item page: Not implemented.

PPR-similarity

Next recommender systems use Personalized PageRank (PPR) algorithm. Given a graph, this algorithm ranks nodes according to their relative structural importance. When initializing, stronger probabilities could be assigned to certain nodes of the graph, in order to give more importance to those nodes. Given an item or some items, we used this graph-based algorithm to get some related items for recommendations. A publicly available implementation of the graph-based algorithm is used[5].

This recommender system exploits a graph created based on the similarity links (see “Simple-similarity” system’s description for more details about similarity links). Each node in the graph corresponds to an item of the collection and an undirected edge *e=(u,v)* connects two items if the item *v* is in the list of similarity links of the item *u*. After applying PPR over this graph, we obtain a ranked list of items. The top ranked items are returned as recommendations. The PPR algorithm is initialized differently in each scenario:

- No profile, item page: We set the element corresponding to the current item to 1 (and the other elements to 0) in the initial state of the random walk distribution vector *v*.
- Profile, landing page: We set the elements corresponding to the profile items to 1 (and the other elements to 0) in the initial state of the random walk distribution vector *v*.
- Profile, item page: We set the element corresponding to the current item to 1, the elements corresponding to profile items to 0.5, and the other elements to 0 in the initial state of the random walk distribution vector *v*.

PPR-logs

This system is also based on a graph and uses the PPR algorithm with the same initialization as PPR-intra for each scenario. But, instead of using intra-collection links to create the graph, we used logs extracted from the Europeana portal, as explained before in Section 2.1. The graph nodes are the items in the collection and there is an edge between two items if the pair exists in a log sequence.

PPR-categories

This system makes use of PPR algorithm as well (with the same initialization), but over a graph based on a Wikipedia taxonomy [Fernando et al., 2012]. Wikipedia categories and collection items are set as nodes in the graph. There is an edge between two category

[5] http://ixa2.si.ehu.es/ukb/

nodes if the categories are linked in the taxonomy, and there is an edge between an item node and a category node, if that item belongs to that category.

PPR-wiki

This system applies PPR algorithm over Wikipedia graph. In order to obtain the graph structure of Wikipedia, we simply treat the articles as nodes and the links between articles as the edges. The collection items are also added to the graph as nodes, and based on the background links to Wikipedia; items are linked to the articles on the graph as well. The initialization for each scenario is equally done as in the other PPR based systems.

The PATHS items have been automatically enriched with background links into relevant Wikipedia articles. To find appropriate Wikipedia links within the experimental subset metadata, the Wikipedia Miner API [Milne and Witten 2008] is used over title, description and subject fields. For each item, all candidate topic links identified by Wikipedia Miner are included. More details about Wikipedia links are given in Deliverable D2.1.

PPR-combination

A graph is created joining all the above graphs (similarity, logs, categories and wiki), and then PPR algorithm is applied over it, in the same way as other PPR based systems.

4.3. Evaluation of Recommender System experiments

The evaluation of the different variants described above showed that the best algorithm is different depending on the scenario: PPR based on logs is the best for cases where the landing page is proposed on past items from the user, but simple-keywords is the best for recommendations for the current item page. In addition, we tried a linear combination of several techniques. The best results in all cases were obtained when combining simple-keywords with PPR-logs and simple-similarity, beating the results of single algorithms for all scenarios.

These experiments show that the quality and coverage of the recommender system deployed in the final system could be improved easily using the combination of PPR on logs, simple keywords and similarity.

5. Conclusions

This deliverable describes additional applications and features of the PATHS system.

A mobile application has been designed and implemented to demonstrate that the PATHS functionality can be ported to a non-desktop device. Recommendation functionality has been developed to personalize the PATHS system.

The mobile application demonstrates the potential of deploying components of the PATHS system onto devices that are transforming public access to cultural heritage information. The “Culture on the go” report confirmed the impact that iPad’s are having on access to information and their potential for increasing use and bringing younger audiences to cultural heritage library systems. The PATHS mobile application shows the potential of offering novel forms of navigation through collections using path-ways that can be followed in the real world as well as online.

For the recommendation functionality, we described two areas of investigation: a set of “lightweight” non-personalized recommendation features that were implemented in the second prototype; followed by the design of novel algorithms for personalized recommendations. The results indicate that a combination of systems using the intra-collection similarity, keyword based search and query log information would improve the quality of recommendations. There is much scope for further research in the development of recommendation services for the PATHS project, including the recommendation of sequences (or paths) of items from a collection that match a given user model.

6. References

Adomavicius, G., and Tuzhilin, A. (2005) *Toward the Next Generation of Recommender Systems: A Survey of the State-of-the-Art and Possible Extensions*, IEEE Transactions on Knowledge and Data Engineering, vol. 17(6), pp. 734-749.

Agirre, E. et al. (2013a) *PATHS: A System for Accessing Cultural Heritage Collections*, In Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (ACL’13), Sofia, Bulgaria, August 4-9 2013, pp. 151-156.

Agirre, E. et al. (2013b) *UBC UOS-TYPED: Regression for typed-similarity*. In Proceedings of the Second Joint Conference on Lexical and Computational Semantics (*SEM 2013), Volume 1: Proceedings of the main conference and the shared task: Semantic Textual Similarity. Atlanta Georgia, June 13-14 2013, pp. 132-137.

Aletras, N., Stevenson, M., and Clough, P. (2013) *Computing Similarity between Items in a Digital Library of Cultural Heritage*, Journal on Computing and Cultural Heritage, vol. 5(4), Article 16.

Ardissono, L, Kuflik, T., and Petrelli, D. (2012) *Personalization in cultural heritage: the road travelled and the one ahead, User Modeling and User-Adapted Interaction*, vol. 22 (1-2), pp. 73-99.

Catledge, L., and Pitkow, J. (1995) *Characterizing browsing strategies in the world-wide web*, In Proceedings of the Third International World-Wide Web Conference on Technology, tools and applications, Vol. 27.

CIBER Research, (2011) *Culture on the go*, Europeana http://pro.europeana.eu/documents/858566/858665/Culture+on+the+Go

Fernando, S., Hall, M., Agirre, E., Soroa, A., Clough P. and Stevenson, M., (2012) *Comparing taxonomies for organising collections of documents*. In Proceedings of The 24th International Conference on Computational Linguistics (COLING 2012).

Fernie, K. et al. (2012) *PATHS: Personalising access to cultural heritage spaces*, in Proceedings of 18th International Conference on Virtual Systems and Multimedia (VSMM 2012), pp.469-474.

Jones, R., and Klinkner, K. (2008) *Beyond the session timeout: automatic hierarchical segmentation of search topics in query logs*, In Proceedings of the 17th ACM conference on Information and knowledge management (CIKM '08). ACM, New York, NY, USA, pp. 699-708.

Milne, D. and Witten, I. H. (2008) *Learning to link with Wikipedia*. In Proceeding of CIKM ’08, pages 509–518, New York, NY, USA.

Sinha, R. and Swearingen, K. (2002) *The Role of Transparency in Recommender Systems*, In Proceedings of the Conference of Human Factors in Computing Systems, 20-25 April, 2002, Minneapolis, MN, ACM, New York, NY, pp. 830-831.

Smeaton, A., and Callan, J. (2005) *Personalization and recommender systems in digital libraries*, International Journal on Digital Libraries, vol. 5(4), pp. 299-308.

Tanasa, D., and Trousse, B. (2004) *Advanced data preprocessing for Intersites Web usage mining*, Intelligent Systems, IEEE, 19(2), pp.59-65.