

Paths and Contextually Specific Recommendations

Matthew Chalmers, University of Glasgow

www.dcs.gla.ac.uk/~matthew, matthew@dcs.gla.ac.uk

Abstract

We describe the Recer system and its underlying model of user activity, paths. A path is a timestamped log of activity in one of more tools, such as web browsers and text editors, gained by client-side tracking. By default, recent user activity is taken as representing the current context, and as an implicit query to the database of paths. The result is a set of recommendations specific to that context, and also reflecting a group's patterns of past activity with the logged tools. We give some examples of its use before going on to discuss how recommender systems' ability to handle heterogeneous data types allows extension of the model of user context to include locations and artefacts in both the physical world and virtual worlds. Our aim here is to have a much more realistic and broad notion of context and activity implemented in our systems, moving beyond an implicit assumption that all that is significant about a person's activity is on their desktop. We outline our current work extending Recer to track multiple devices and media so that recommendations refer to people, places and things—both physical and digital. Recer will become part of a larger system aiming to support awareness and collaboration amongst people exploring a collection of documents about Glasgow, the physical city and a VR model of the city centre.

Introduction

Collaborative filtering (Goldberg 1992) or recommender systems (Resnick 1997) depend on a model of each person using them. This is the representation of the current user's activity or information need, and the basis for personalising its provision of information. Most recommender systems collect information about each user over some time, to build up a profile of individual choices or interests. This representation is generally unordered temporally, although some approaches do focus on temporal patterns of user activity. One such approach relies on time-stamped logs of user activity—'paths' through information—and represents activity or need as the recent part of the user's path e.g. the set of URLs loaded into the user's web browser within the last few minutes (Chalmers 1998). As in other recommenders, this path forms a model of the user that avoids content analysis, and this lets us handle complex and heterogeneous data at the cost of collecting significant data on user activity. In a way unlike most recommenders but rather like information retrieval systems, path systems allow the user to form a contextually specific representation of their current information need (Chalmers 1999). One influence on us here was the Remembrance Agent (Rhodes 1996), which tracked activity in an editor to form implicit IR queries. Whether active or passive, the result is a query or context that the system responds to with recommendations.

In this way, path systems aim to do more than make personal or individual recommendations. They also aim to be contextually specific in their recommendations. An issue here is what one records in the path. This may be addressed by focusing on fine-grained actions such as activity within and between web pages (Villa 2001). Another approach is to focus on the media one tracks and records. As we describe below, we have enriched paths by including file activity in the *xemacs* editor so that, for example, recommendations are made on the basis of a combination of web pages and images browsed along with the code files recently read or edited. Recommendations can therefore be 'documents' of various types such as HTML pages, JPEG images, Java source code, Emacs Lisp, and so forth. Recommender systems' ability to handle heterogeneous data lets us bridge between tools and types usually treated in isolation. We can record and combine a range of tools and media, and use the temporal patterns of user activity that interweave them to make recommendations that span them.

We are now broadening our range of tools and media significantly further as part of our work within Equator, a large interdisciplinary research consortium in the UK funded by the EPSRC. Equator is an Interdisciplinary Research Centre, involving roughly 40 researchers and 50 postgraduate students working in eight UK universities over six years. It is centred on research in computer science and informatics, but strongly psychologists, sociologists, artists and designers are also Equator members. Its focus is on blending and bridging between digital and physical media,

on interweaving electronic devices and information with physical objects and spaces. We work at three levels, improving our understanding and theories of interaction, exploring deep adaptation and flexibility of infrastructure, and designing new devices and technologies. A young web site exists at www.equator.ac.uk, and information more local to Glasgow is available at www.dcs.gla.ac.uk/equator.

System development at Glasgow focuses on path systems that record user activity in terms of physical, informational and virtual context, tracking activity amongst a group of people using wearable computers, hypermedia and VR systems. We will then feed back recommendations of documents, artefacts and places, both physical and virtual. We can then combine physical location, gained via technologies such as GPS and ultrasound tracking, informational location in terms of recently used documents, and virtual location in terms of position and motion in maps and virtual environments. For example, someone using a mobile computer in a museum may browse web documents about exhibits nearby. These documents may link through to a 3D model of related cultural institutions, which the user might then use to find out about examples of interesting architectural design in the same city. That user would gain recommendations based on their location in the museum, the documents they have just read and the places they have examined in the virtual city. Theory development at Glasgow focuses on the interdependence of formal representations, such as paths and IR indices, on interaction and collaboration. We wish to find an accord between computer science's models of information, interpretation and interaction, and those of contemporary semiology and philosophy (Chalmers 2001).

The next sections describe the older version of the recommender system and give some examples of its use. Then we go on to discuss our current work extending that system to work with a wider range of interactive media, and the way in which this ongoing implementation work is driven by broader notions of context, recommendation and activity than most collaborative filtering systems.

Paths and the *Recer* System

Extending the recommender system approach to take more account of the temporal order of activity is at the core of paths and the *Recer* system. Most of *Recer* is written in Java, although some parts involve Unix shell scripts and Lisp. It also involves a commercial relational database, Oracle, as the repository of paths.

A path is a time-ordered history of a person's use of information. Each person's use of symbols, such as URLs accessed in a web browser, is logged over time, and this path serves as the system's representation of the person's activity. All our logging is done on the client side, in order to bridge between different web sites, different desktop tools and different media. Unlike most recommender systems, the context of activity is an essential part of *Recer*'s approach. Each path entry, i.e. each recorded instance of a symbol's use, can be associated with a context in the form of the temporally neighbouring entries in that path. We then treat a person's most recent path entries as a request for recommendations i.e. as a possibly implicit statement of current context and of information need. As we describe below, this current context drives the recommendation process in that we use the pattern of past occurrences of recent symbols and the contexts of those past occurrences as the source of recommendations.

Recer is dependent on external systems to initialise and extend the set of symbols recorded in paths and used in recommendations. Paths would be useless without people's use of Web search engines and file browsers, and indeed editors, mail tools, magazines, journals and so forth. We see this not as a weakness, but as realistic and pragmatic acceptance of the interdependence and holism of tools, and their embedding in the wider world. We interleave information tools in everyday use, and the sets of objects they operate on overlap. One of the improvements to the system of (Chalmers 1998) described in this paper is that we now record activity in the *xemacs* text editor, in the same path as that of the web browser. Web activity can thus potentially trigger recommendations for local files, and vice versa, and we are looking towards further expansion of the range of tools and media *Recer* works with. The last section describes this in more detail.

We have experimented with two modes of activity logging. The first records only the URLs and filenames that identify the documents active in an editor and a browser i.e. no document content. In the *xemacs* editor, a timer triggers a check every five seconds for a change between editor buffers. When one is detected, and there is a filename associated with the new buffer, the filename is written out to a temporary log file along with a timestamp roughly representing the time of the buffer switch. A shell script, *emacsWatcher*, periodically pipes any new lines added to the log file into a Java program that enters a path entry for each via a JDBC connection. The name of the

database table to write to, e.g. *path_matthew*, is determined by the username property of the Java environment. In the *Muffin* web proxy, a Java program freely available from *muffin.doit.org*, each time a new page is accessed we insert a path entry for the URL in question.

In the second logging mode we create a line of text once every second, containing the nearest complete word or symbol to the *xemacs* insertion point, the current time, and the filename associated with the current editor buffer. This is only written to the log when the symbol differs from the previous recorded symbol. In the Web browser, the proxy records the URL and all unique terms in the page, including contained URLs, each time a new page is accessed. Again, all path entries for the same person are written to the same table. Words and URLs inside documents clearly may be significant, but we have suspended experimentation with this logging mode for the moment. The primary issue was that the computational load of logging interfered with interaction with the web browser and editor. Another issue was that of finding which words are worth logging and worth recommending. Using recent symbols helps to filter unhelpful recommendations, and we are interested to see if we can use this approach as a less blunt instrument than IR's traditional stop lists.

In a path table, each entry or row is a triple: (*timestamp, symbol, source*). In the case of a content symbol, the second element represents the symbol itself while the third element represents the document or file it was found in. Otherwise, the third is a duplicate of the second. For efficiency and to save space, in both logging modes we use the hash function of (Pearson 1990) to make 32 bit fingerprints for symbols and sources. A separate lookup table stores (*fingerprint, string*) pairs. Also, each path table has a primary key based on both the timestamp and the symbol, and an index based on this key.

We treat the user's most recent path entries as an implicit request for recommendations. Periodically, the recommender takes these entries as a set, and searches the database for past occurrences of each one. (At this stage it also collects the set of symbols used in the later 'exclusion step'.) This search can cover one's own path and/or a selection of others' i.e. the set of paths is a resource shared amongst those who contribute to paths. By offering selectivity of which paths (or rather, whose paths) to draw from, we allow people to use knowledge of their colleagues' skills and interests to steer the recommendation process. In another change to the system of (Chalmers 1998), we no longer do a brute force search across the entire path database, sliding windows from the start to the end of each path table. The key and index for each table lets us narrow our search to where at least one past occurrence of a recent symbol exists, which greatly aids speed and scalability.

The system collects the context of each past occurrence of each recent symbol i.e. a window containing the path entries immediately following the past occurrence. It tallies the symbols collected from these windows and then, in a final exclusion step, discards any symbols that were recently used. Symbols used recently thus form a dynamic stop list, helping to filter out unnecessary and commonly used symbols. Symbols not filtered out are presented to the user as recommendations. The system then sleeps for a time before waking, collecting a new set of recent symbols and beginning again. To afford some privacy, path logging can be turned on and off at will.

The recommender's interface is split into two panels. The first displays recommended symbols, split into three subgroups: URLs, local filenames, and 'everything else'. (In the case of no content being logged, the third list is empty.) The second panel has buttons for selecting paths to draw from and for triggering an immediate recommendation, as well as four sliders which set periods of time: the recent period used to define the current context, the window after each past occurrence, the recent period used for excluding symbols from recommendations, and the sleep period between recommendations.

Clicking on a URL sends it to the web browser to be loaded (and logged). Clicking on a local filename triggers *xemacs* to load and log it. Note that a URL or filename can be recommended even if no-one has actually visited it. If a URL occurs frequently enough in the content of files or pages associated with recent work then it can gain enough tallies to be recommended. In the *Everything else* list, clicking pops up a window which shows a list of the URLs and filenames which were recorded as containing the selected symbol. A click in this pop-up list triggers display as before. We do not identify the paths (and people) that contributed most to recommendations. No-one has so far objected to the idea, but then no-one has experienced it. Less conservative trade-offs are possible, of course, balancing the utility of finding out about contributors and the potential invasiveness in making explicit exactly who used what, and when (Bellotti 1993). Indeed, we are negotiating with a governmental organisation to run a large-scale trial of Recer, and they require that identities are revealed in order to support the sharing of expertise and

experts across the organisation.

When past occurrences of recently used symbols are temporally near to each other, their contexts or windows overlap. Symbols within overlaps then get higher tallies. Highly ranked symbols tend to be, therefore, those symbols most consistently used just after the periods most similar to recent activity. We assume that consistency of use in similar contexts is a good indicator of appropriate (literally, utile) recommendations. Just as the stereotypical recommender system draws recommendations only from the most 'similar' people to oneself, path systems draw recommendations from the most similar contexts to one's own. Statistics for recommendation and visualisation thus form categorisations or abstractions over path entries that are not fixed *a priori*. Instead, recurrence statistics are made anew for each person at the time of, and based on the context of, each recommendation operation. They are also ephemeral, in that once a new path entry is made after a recommendation e.g. upon clicking on a recommended URL, the old recurrence statistics are no longer valid. Finally, we note that adaptation can occur even if one refrains from logged activity. One's recommendations may vary as other people's activities extend their paths.

Examples

Showing screendumps of Recer's output would take up rather too much page space, and they have been shown in other papers, and so in this section we offer descriptions of recommendations. In a first example, we discuss a set of recommendations made without content logging at a time when the author was resident in Zürich, Switzerland. This example is intended to demonstrate how recommendations, such as ski information given weather pages, might not be obviously useful until one considers the author's particular context and the history behind it. If the mountain weather was good, he might ski. Skiers living in Zürich often checked the weather before getting details of piste conditions and cable car times. In this case, the author got the Klosters information from his own path, and the Arosa information from a colleague's path. Never having been to Arosa, or to the Arosa web site, the recommendation was therefore both novel and relevant. The example also offers a contrast with content-based tools whose likely recommendation, we suggest, would simply have been for yet more weather pages.

As discussed in (Chalmers 1998), co-occurrence statistics were used in the layout algorithm of (Chalmers 1996), and the results browsed using the visualisation tool of (Brodbeck 1996). The intention is to show more of the context and structure that lies behind recommendation lists. Layouts involve a relatively small number of symbols, which tend to form 'strands.' Each strand is a sequence of symbols collected prior to symbol tallying i.e. one of the periods of past activity used in the recommendation process. Strands cross at multiply accessed symbols, and so symbols used at diverse periods of time, but adjacent to these recurrences, are brought near to each other. Such a visualisation is made in under 30 seconds on a Silicon Graphics O2.

In a second example, recommendations were made via the second logging mode while working with *xemacs* on the recommender itself. URL recommendations were for JDK and Swing user interface toolkit documentation read during previous editing. Files recommended for reuse were other components of the recommender system, and were of various types: Java files, a shell script, a file of SQL statements and an Emacs lisp file. The *Everything else* recommendations were for the class *System*, often used in *println* statements to output (copious) debugging information, and variable names used elsewhere in the *recer* class such as *doc* and *recentSymbols*. If several programmers' paths were combined in this way, we believe the current system would recommend to each user, in his or her current work, software components that were used by others in similar work but that the user did not know about. Our system would then help support software reuse within work groups.

Current Work: Equator and the City Project

A large proportion of our current work in Glasgow is focused on a new project involving many media, technologies and partners. We are now making an extension to Recer similar to the *xemacs* extension described earlier, but somewhat larger in scale. As described in the sections above, we previously tracked and fed back to commonplace desktop tools, based on the view that richer models of user context and activity can be gained by taking account of the interdependence of such tools. Now we are looking towards still broader models of context and activity, and a larger and more varied set of tools.

The 'City' project focuses on information organised as tours, histories and narratives, and 'embedded' into the

physical objects, buildings and streets of the city. Just as paths wove together web pages and local files before, we extend the range of types represented in paths to both physical and digital objects. Initially we will focus on information about Charles Rennie Mackintosh, a Glasgow architect and designer who died in 1928 after a career spanning architecture, design and art. The project was in fact called 'Mackintosh' but, since we aim to extend our scope beyond C.R. Mackintosh and the museums and collections associated with him, which seemed rather narrow. Researchers from five universities (Glasgow, Southampton, Bristol, Nottingham, and University College London) are at the moment setting up infrastructure and making a number of existing systems interact—including Recer.

The City project explores new devices and information systems that support mobility, adaptation to context, and access to a mixture of information types. It is a vehicle for an approach to information and computation that takes fuller account of subjectivity and context than most. The project involves images, textual descriptions, and references to locations in the city, to artefacts in museums and exhibitions, and to electronic resources such as web sites. Beyond the storage of individual items, paths through them will initially be made 'by hand', via a simple editor, forming more explicitly designed tours or narratives. We do not intend to make these paths ourselves, but will invite a number of parties to contribute their own interpretation of the 'raw' data and of existing paths. We plan for these editors or authors to include curators of various Mackintosh resources, people who work with cultural information and institutions in Glasgow, members of design groups such as commercial design firms in Glasgow and local art schools, and residents of the city from various communities and backgrounds.

We are aiming for a running first prototype in August 2001. Initially we will use the static body of 'official' or 'designed' paths mentioned in the previous paragraph, and the collection of images, text and locations brought together by them. Thereafter we will allow user activity to form a growing or evolving system in two ways. Firstly, new associations within the collection will be made by the people involved in our demonstrations and trials. Their paths will afford new recommendations beyond those based on the 'official' authors. Secondly, new items will be added to the collection by our users. This will allow recommendations to extend beyond the 'official' or 'high culture' domain represented by many of the initial authors. If people visiting a given Glasgow museum consistently tend to drop into a particular café afterwards, or to visit a particular web site with cinema schedules, then we wish to represent that, rather than ignoring or suppressing their use of the city and its associated information.

We propose to support information access via audio, wearable computers, tablets and VRs. In the first case, we would generate a stream of verbal and non-verbal audio, supplied to a phone or to a computer. In the second case, we refer to a wearable computer with relatively small graphical displays, audio interfaces and a variety of context sensors integrated into a 'CyberJacket', as developed by U. Bristol. These wearables are of particular applicability to places such as the Mackintosh Room, where we aim to make available fine-grained tracking of location and to adapt system responses accordingly. By 'fine-grained' we mean to within around 15cm, as we are installing an ultrasound-based tracking system. A tablet refers to a laptop-like computer with wireless network communications, and it may have greater computation and display facilities than a wearable. In VRs, we refer to more high-end graphics systems that convey sophisticated 3D graphical display of virtual environments, including CAVEs (such as that of UCL and another local Equator affiliate, the Glasgow Science Centre).

An important aspect of this variety of media and devices is that we expect people's activities in each one to be perceptible in the others. This would certainly mean that use of each one would be represented in an information store used by all. We will record locations a person with a wearable walks through, the documents accessed via hypermedia systems, and the virtual locations a VR user moves through. Note that the use of these media may be interwoven in activity, for example in the way that documents may be shown on a wearable device or in a VR. They are also interwoven in that they refer to common place and artefacts, e.g. a given building may be referred to via its physical location, the web page about the museum there, or its VR model.

Shared references and the shared database representation afford synchronous communication. Users, possibly using different media, can be made aware of each other and can communicate with each other directly. An example would be that of a person walking along the physical Buchanan Street, with a wearable that is tracked. That person's activity would be represented inside a virtual model of the city. Someone in London, exploring that virtual city, would have his or her activity tracked too. The first person's wearable could show if the Londoner was approaching for a chat, and similarly the Londoner could see and make contact with the Glaswegian.

Of course, the recorded activity in these various media will also support asynchronous communication. We will

graphically render relevant paths in maps and VRs, and also offer recommendations. Tracking activity in the virtual city, the physical city and the information about the city means that the patterns of stored symbols, the analysis of recent activity and the consequent recommendations will span all such media. Recer is being extended to recommend objects and locations, both physical and digital, with persistent identifiers akin to URLs. As someone changes location, their wearable can log the new location and obtain recommendations for people, places and things to go on to. As someone moves around a virtual world, their location can similarly be logged and woven into the system of paths and associations between objects of heterogeneous types. Our model of each person's activity is primarily constructed from identifiers of people, places and things in all the media we can track. Recent activity defines the context used in recommendations, and so our model of context is equally broad. Recommendations can be made of any or all such types because, like most recommender systems, our core algorithms do not depend on types but instead rely on activity patterns.

Another aspect of paths arises here. When people know they are being logged, they will shape their actions to suit how other people will perceive them. They will change their behaviour, because they know other people may perceive that behaviour later on. This is not so unusual, especially in the case of guides and curators who often lay out a path through a building or museum collection in order to provide visitors with an interesting or useful experience. This is an important aspect of paths: people can *design* them. People can either shape their actions to be a path for others to use, or they may use a tool to edit and design a path more explicitly.

This interweaving of physical and digital media, using activity to associate and design text, building, video, museum display, web page, room location, audio track and so on has been called *monomedia*. This stands in deliberate contrast to the rather wan term 'multimedia' that too often involves synchronised use of just two media, audio and video. (Thanks to Willem Velthoven and *monomedia.org* for this.) Activity stems from previous understanding, but also feeds back into understanding by creating or reinforcing associations between individual objects, individual spaces and individual people. A person's movement through data, through the city and through society adds to his or her understanding of information, places and people. One aim of this project is to support this interpretive process, improving information and information systems by representing and adapting with real use, making manifest more of the information and understanding that turn city spaces into 'places', and offering people useful and interesting ways to interact with each other.

References

- V. Bellotti & A. Sellen, Design for Privacy in Ubiquitous Computing Environments. In: G. de Michelis, C. Simone and K. Schmidt (Eds.) *Proc. Third European Conf. on Computer Supported Cooperative Work, (ECSCW'93)*, pp. 77-92. Kluwer, 1993.
- D. Brodbeck, M. Chalmers, A. Lunzer. & P. Cotture, Domesticating Bead: Adapting an Information Visualization System to a Financial Institution, *Proc. IEEE Information Visualization 97*, Phoenix, Oct. 1997, 73-80.
- M. Chalmers, A Linear Iteration Time Layout Algorithm for Visualising High-Dimensional Data. *Proc. IEEE Visualization 96*, Oct.-Nov. 1996, 127-132.
- M. Chalmers, K. Rodden & D. Brodbeck, The Order of Things: Activity-Centred Information Access. *Proc. WWW7*, Brisbane, April 1998, pp. 359-367.
- M. Chalmers, Comparing Information Access Approaches. *J. ASIS 50th Anniversary Issue*, 50(12):1108-1118, September 1999.
- M. Chalmers, Awareness, Representation and Interpretation, to appear in *J. Computer Supported Cooperative Work*, 2001.
- D. Goldberg et al., Using Collaborative Filtering to Weave an Information Tapestry, *Comm. ACM 35(12)*, December 1992, 61-70.
- P. Pearson, Fast Hashing of Variable-Length Text Strings, *Comm. ACM*, 33(6):677-680, June 1990.
- P. Resnick & H. Varian (eds.) Special Issue of *Comm. ACM* on Recommender Systems, 40(3), March 1997.
- B. Rhodes & T. Starner, A Continuously Running Automated Information Retrieval System, *Proc. Conf. Practical Application of Intelligent Agents and Multiagent Technology*, 1996.
- R. Villa, A Framework for Implicit Tracking Data, to appear in these proceedings, June 2001.