Workshop on Case Based Reasoning and Personalization

6th European Conference on Case Based Reasoning ECCBR 2002

September 4th, 2002
Aberdeen, Scotland

Position Papers

Mehmet H. Göker, Barry Smyth (ed.)

Introduction

With information becoming a commodity and access to data becoming ubiquitous, computer users are increasingly faced with the problem of locating information relevant and interesting to them rather than data in general. Customizing data access, retrieval, processing and display according to the users' preferences is the main task of current personalization systems. These systems obtain user preferences through interactions with users, summarize these preferences in a user model and use this model to adapt and generate customized information or behavior. They deliver the customized results in the manner that is most desirable for the current user, thereby increasing the quality of both the interaction and the generated result.

Case Based Reasoning (CBR) has been used in a variety of application areas to retrieve relevant information. While typical personalization approaches on the Internet focus on computing similarities between users of the sites, leaving the actual content out of scope, CBR typically focuses on the content, rather than on providing information based on user characteristics. Initial applications of CBR concentrate primarily on the application-specific customization of retrieval and adaptation, and not on retrieving and delivering user dependent, customized information. With the advent of personalization, a new direction of CBR research and systems has appeared. The developed personalized CBR systems adapt their knowledge containers according to the characteristics of individual users (or user groups) in addition to the application area.

The use of CBR methods for personalization and the development of hybrid systems have introduced a need to integrate case-based methods with alternative techniques such as collaborative filtering. The *Workshop on Case Based Reasoning and Personalization* during the *6th European Conference on Case-Based Reasoning ECCBR 2002* brought together researchers from both the Personalization and CBR communities to discuss and share their ideas and to demonstrate developed systems. The following are the position papers some of the participants submitted to the workshop for discussion.

Mehmet H. Göker
Menlo Park, August 9th, 2002

Program Committee:

Ralph Bergmann	University of Hildesheim	bergmann@dwm.uni-hildesheim.de
Pádraig Cunningham	Trinity College, Dublin	Padraig.Cunningham@cs.tcd.ie
Mehmet Göker (Chair)	Kaidara Software	mgoker@kaidara.com
David Leake	Indiana University	leake@cs.indiana.edu
Barry Smyth (Chair)	ChangingWorlds	Barry.Smyth@Changingworlds.com

Table of Contents

User Context and Personalisation

Ayse Göker[1], and Hans I Myrhaug[2]

[1] School of Computing, The Robert Gordon University,
Aberdeen AB25 1HG, Scotland
asga@scms.rgu.ac.uk
[2] SINTEF Telecom and Informatics,
N-7465 Trondheim, Norway
hansim@sintef.no

Abstract. The importance of user context as a means of delivering personalised and context-sensitive systems is discussed. Relevant aspects of personalisation and context technology are covered. The intention is to inspire those interested in Case-base reasoning and personalisation from background and experience in other disciplines such as information retrieval, adaptive user interfaces, user modelling and mobile computing. Descriptions of personalisation and context are followed by their use in information retrieval and their importance and use in ambient computing. Relevant literature that may be a motivating source for interested readers are provided. Various questions are also raised in initiating discussion on this topic.

1 Personalisation

Personalisation is about tailoring products and services to better fit the user. There are several ways of achieving this. The main ways are by focusing on the user needs, preferences, interests, expertise, workload, tasks etc. We advocate *user context* as a means of capturing all these.

Personalisation can be achieved by tailoring products and services either to large user groups, smaller interest groups, or the individual user. The degree of personalisation that your business chooses depends on the competitor's behaviour, the internal resources, the market, and the customer. Normally, the main reason for personalisation is that you believe you can establish a better relationship with a customer if you do so. This in turn can lead to increased competitiveness, which can result in increased or maintained income.

An example of personalisation can be in car purchasing: once you have selected the car model, you can tailor it with extra equipment, colour, dashboard interior, seat textile, the engine, and special wheels. The interesting aspect of this is that the customer is prepared to wait for a delay in delivery in order to receive a personalised product and the manufacturer sells the car/product before it has even been made – although the individual components may exist prior to assembly.

Given this type of customer behaviour, it seems likely that there will be a demand for these personalised services to be life-long or at least longer term. Banking and finance businesses have demonstrated that it is possible to establish long-term business-to-customer relationships. Thus, we conjecture that personalisation based upon the user context is one way of achieving this and a standardised way of understanding context (i.e. modelling/representing the context) is important in enabling this.

2 Context

A context can be defined as a description of aspects of a situation. In this way, context can seem similar to cases in case-base reasoning. A context as an internal representation in the computer should be a structure for information units and data. It is also natural to refer to contexts that are more or less similar to other contexts.

Context technology is a mechanism that can capture the concepts and relations between these concepts. However, we argue that there should be some common structure for user contexts, which is easy to reuse across domains. What makes domains differ is mainly that the relevance and importance of concepts within the context structure differ. Hence, it is possible to have redundant items in the context because their relevance can change over time.

Context information can be used to facilitate the communication in human-computer interaction. The use of context is becoming important in interactive computing. Recently, there has been much discussion about the meaning and definition of context and context-awareness. These are exemplified strongly in two recent workshops: DARPA [1] and UM2001[8] and some EU projects. However, this kind of information (context) is still not utilised much and the concept of context is not yet well understood or defined. Additionally, there exists no commonly accepted system that supports the acquisition, manipulation and exploitation of context including information units and data.

Items in a context may be exploited by adaptive information services including those for the Web search environment and those for users who are increasingly mobile. Three important aspects of context can for instance be where you are, whom you are with, and what resources are nearby you. This information is more likely to change often for the mobile user.

One challenge of mobile services is to make use of context information and exploit the change of context. We think that service vendors should have a common tool or method to explicitly model context with, because all service vendors will then be able to provide the users with context-sensitive, personalised services and products - independent of the runtime technology.

3 Context and system adaptation

The roots of personalisation of information systems can be traced back to the early adaptive user-interfaces, personal assistants/agents, and adaptive information retrieval. Relevant readings in these areas can be found in [2, 5, 6]

Most of the approaches started with users' needs, preferences and expertise. Some of these approaches also merged with work on user modelling [also see UM conferences]. User modelling is both the process of modelling the user as well as the outcome i.e. the user model. Other approaches involve detecting patterns in user behaviour when searching for information. A complementary approach can be found where the system designer decides that changes in the environment should lead to system adaptations. Few systems have been made which achieve this. Some of these systems have been referred to as context-aware applications and others as affective user interfaces. Context-aware applications have mostly focused on location-awareness and mobility since monitoring context is difficult with present technology.

Although there is a relationship between a user model and user context, the problem with adaptive systems based upon only user models is that changes within the environment or situation cannot be naturally modelled with user models. For example, it is not easy to say that a PDA, map, building, cockpit and so on are an integral part of the user model. Rather it can be easier to state the reverse: that the user is a part of the environment.

4 Context and information retrieval

When discussing the information retrieval process, often the focus is on the individual activities such as formulating queries, searching document collections and presenting returned documents. However, there are situations where we need to go beyond analysing these individual activities in isolation, and consider the groups of these activities. Spink et al [7] show that nearly 60% of users had conducted more than one information retrieval (IR) search for the same information problem. In their research, they refer to the process of repeatedly searching over time in relation to a specific but possibly evolving information problem as the successive search phenomenon.

Contextual information plays a more important role in the study of successive searches than that of isolated searches since the contexts behind a series of successive searches are probably closely related to each other, if not the same. However, finding contextual information is a difficult task even for successive searches, especially if the searches are launched on the Web. Previous studies have demonstrated that less information is available about the users and their information needs on the Web, not to mention the fact that Web searches are shorter and search statements contain less terms than their counter parts in traditional IR searches [4].

An individual information retrieval activity may be informative sometimes, but a collection of search activities provides much more information about the topic and the context if they are organised according to their time order and related search topic. It is likely that consecutive activities related to one topic can share the same context. It is, therefore, reasonable to say that the information about search topics is an important component of the context behind the users' searches or retrieval need.

The Web is a source of information and Web user searches can be analysed to detect patterns in search behaviour and information needs in order to effectively deal with their subsequent needs. Collaboration amongst users has been a prominent research topic since the start of the Internet. Given the rather limited amount of information available on individual Web users and the unreliability of their identification process, the Web environment makes collaborative approaches more appealing.

A personalisation approach that was originally developed within the context of a traditional bibliographic retrieval system [3] has been adapted and extended with a collaborative model for the Web retrieval environment. The transition of information search environments from a traditional library to Web then to a ubiquitous one presents new challenges.

5 Context is important for ambient computing

The use of user context in ambient computing is needed for several reasons: users are increasingly mobile and require ambient computing with context-aware applications; and they need personalised information services to help them in their tasks and needs. We argue that the challenge which ambient computing applications will face is complex and can not be solved easily with isolated approaches to wireless technology, miniaturised devices, context-aware applications, information retrieval, or user modelling.

Rather, an integrated approach is needed where system designers, programmers, content service providers, and most importantly the mobile users get the support and help they need in order to find ambient computing useful and user-friendly. To this end a user context, which builds bridges between user modelling, information retrieval, and context-aware application is presented.

6 Case: User Context in AmbieSense

As an example, the AmbieSense system implements a general context-aware technology that is proposed as a solution with a unifying framework for exploiting user contexts in ambient computing.

The standardisation of user context in AmbieSense is motivated by the generic user needs that occur when we combine the following facts: (1) users want useful services that are personal, context-sensitive, and life-long (2) computers are used as tools for knowledge and experience sharing, (3) users want to be mobile.

The belief is that personalised and adaptive services, which increasingly operate in a mobile society, need effective knowledge and experience sharing. This is only possible to achieve if one can link relevant information units (e.g. various kinds of files) into explicit and individual user contexts. The user and personal assistant, actuators, and sensors should be able to update the contextual information that together comprise a user context. Modelling context should therefore follow an approach that is model-based but extensible.

User Context in AmbieSense - A generic user context consists of five parts:

- Environment context
- Personal context
- Task context
- Social context
- Spatio-temporal context

(1) Environment context – this part of the user context captures the entities that surround the user. These entities can for instance be things, services, temperature, light, humidity, noise, and persons. Information (e.g. text, images, movies, sounds) which is accessed by the user in the current user context is all part of the environment context. The various networks that are in the surrounding can also be described in the user's environment context.

(2) Personal context – this part of the user context consists of two subparts: the physiological context and the mental context. The first part can contain information like pulse, blood pressure, weight, glucose level, retinal pattern, and hair colour. The latter part can contain information like mood, expertise, angriness, and stress etc. Some contextual information are quite static while others are rather dynamic in time.

(3) Task context – this context describes what the persons (actors) are doing in this user context. The task context can be described with explicit goals, tasks, actions, activities, or events. Notice that this also can include other persons' tasks (that are within the situation). For example, in a car with a driver and passengers, the situation can include the driver driving the car, passengers doing various things such as reading, watching the car TV, listening to music on the personal stereo. Thus, driver's task context can include information about the tasks his/her passengers are up to. For example, if one of the passengers is the driver

(4) Social context – describes the social aspects of the current user context. It can contain information about friends, neutrals, enemies, neighbours, co-workers, and relatives for instance. One important aspect in a social context is the role that the user plays in the context. A role can be described with a name, the user's status in this role, the tasks that the user can perform in this role, and the various sub-roles that the role can have. A role can in addition be played a social arena. A social arena has a name like "at work" and has a geographical area.

(5) Spatio-temporal context – this context type describes aspects of the user context relating to the time and spatial extent for the user context. It can contain attributes like: time, location, direction, speed, shape (of ob-

jects/buildings/terrain), track, place, clothes of the user and so on. i.e the spatial extension of the environment and the things in it.

6 Important questions to be addressed

Below are some questions to inspire further discussion.

When applying context in a variety of search environments, how best can the function of the search intermediary be met? For example, Web search engines do not have the help of human intermediaries, in contrast to the case in traditional retrieval environments. Unfortunately, from a retrieval perspective, the Web is a vast heterogeneous database covering a large variety of topics at different depths. A search intermediary was able to establish the context of a user's search for information, and hence advise and guide a user when searching. It has been argued forcefully that exploiting the user's context has the potential to improve Web retrieval systems as more information is available about a user and his/her information need.

What are the common aspects between context and Case-based reasoning? User contexts cannot naturally be described as problems and solutions because it is often impossible to know what the problem is now or in the future – as is exemplified in information retrieval. It can be that for future retrieval this is obvious once you start to share your user context with other users. Modelling user contexts may seem unnatural if the context consists of problems with solutions. However, relevance and importance seem natural.

What about sharing user contexts and privacy issues? Users may want explicitly share their contexts with others. Personalised systems may need to monitor the contexts and any changes in the context so as to improve system adaptiveness and context-sensitivity. There are important user privacy and ethical issues that need to be addressed.

These questions, other arising issues, and possible solutions can be further discussed in considering personalisation and the possibility of hybrid approaches for users.

Further Literature and References

1. DARPA Workshop on Meaning Context, (2001). via personal communication P. McDowell, Naval Postgraduate School.
2. Edmonds, E.A. Adaptive Man Computer Interfaces. In Coombs, M. J. And Alty, J.L (Eds). Computing skills and the user interface. *Computers and People*, (1981). 389-426, Academic Press.
3. Goker A. Context Learning in Okapi. *Journal of Documentation*, (1997) 53(1):80-83.
4. He D., Goker A., and Harper D. Combining evidence for automatic web session identification. Journal of *Information Processing and Management*. 38 (2002) 727-742.

5. Myrhaug H. and Thomasen (1997). A new taxonomy of adaptive-user interfaces. NIK Proceedings. Tapir Forlag.
6. Schneider, Hufschmidt, Kuhme, and Malinowski (Eds), Adaptive user interfaces, North-Holland. 1993.
7. Spink A., Wilson T., Ellis D., and Ford N. Modeling user's successive searches in digital environments. D-Lib Magazine, 1998.
8. User Modeling Conference, Sonthofen, Germany. Workshop on User Modelling for Context-Aware Applications, 2001. http://orgwis.gmd.de/gross/um2001ws/papers

Recommendation and Personalization From Item to Collection, from Client-Server to Peer-to-peer New Frontiers for Case-based Personalization

Paolo Avesani and Paolo Massa

ITC-IRST,
Via Sommarive 18 - Loc. Pantè, I-38050 Povo, Trento, Italy
{*avesani,massa*}*@itc.it*

Mainly the personalization or recommendation services are designed to support the choice of single good from a catalogue of alternatives (e.g. books, movies, music, ...). Our claim is that more often such a choice is a more complex task that ask for the user to aggregate a set of goods (compilations, travels, ...). The personalization process becomes in this context a mixed-initiative interaction between the user and the system that can not be reduced to a one shot step of information retrieval.

In this perspective it is straightforward that the case-based reasoning can help to design effective solution for case-based personalization systems. CoCoA[1], a Compilation Compiler Advisor, is a system that enables the design and the delivery of a full personalized audio compilation. Exploiting the notion of the query by example the user can obtain recommendation on similar compilations giving as input the current sketch of compilation.

In CoCoA a specific similarity metric has been designed to overcome the lack of a rich content description that usually applies in many domain. A collaborative filtering approach has been adopted to exploit the knowledge encoded in the set.

CoCoA Radio [2] is a different application where we assume that the knowledge of a collection, a playlist, is encoded also in the order relation. In this case a rankboost approach has been proposed to provide group recommendation.

While the evaluation of the specific techniques, collaborative filtering rather than vector similarity, could be assessed quite easily, the evaluation of their use for personalization systems remains an open issue. We have elaborated, in collaboration with the Trinity College Dublin, a new methodology [5] that promotes an on-line pairwise comparison between two competing solutions. One of the main motivation is related to the difficulty to collect meaningful dataset to accomplish off-line evaluation.

Data become to be a critical issue not only for the company or the reaserchers but also for the users. For the service providers data represent the competitive advantage that prevent new opponents to delivery an alternative solution. For the reasearches data are the a strong requirement to enable the validation of innovative working hypothesis. For the users the data, the personal data, are the precondition to obtain an effective personalized service. Up to now for example an amazon.com user can not take advantage of a personalized service from barnesandnoble.com because the personal profile belongs to the previous provider.

The current client-server approach to personalization prevents the users to exploit their personal profiles moving from one service to another and rewards the bigger providers getting more and more difficult for a new provider to collect enough information to deliver a personalized service. To overcome these drawbacks we are exploring a new scenario that could emerge moving from a client-server perspective to a peer-to-peer (P2P) approach.

First of all in a P2P view the personal profile information should be located by the user. It should lower the redundant elicitation of personal preferences and at the same time it will open a new challenge to the personalization services: the interoperability with non homogeneous representation of user profiles.

Although this basic assumption could seem too ambitious on the contrary it is becoming one of the precondition for the emergent scenario of ubiquitous computing. The hypothesis that the personal profile has to stay with the user is mandatory if we assume that we are approaching a situation where through our palm or cellular phone we will receive personalized information from an augumented environemnt.

A second concern is the privacy. Data related to a user are usually a very sensible source of information and their management should be under the control of the user. The privacy requirement should preserve the control of the user profiles without to prevent the information gain that could derive from sharing with other peers. A work has been already developed that combines privacy and collaborative filtering [4].

A further concern related to a P2P view on personalization are the reputation metrics. In this context it is not sustainable the assumption that every peer is trustworthy and it becomes mandatory to deal with the issue of malicious disrupters [3, 6]. Peers are invited to share not only data related to their own personal profiles but also their reputation models.

References

1. S. Aguzzoli, P. Avesani, and P. Massa. Collaborative case-based recommender systems. In *European Conference ob Case-Based Reasoning*, Aberdeen, Scotland, 2002. Springer Verlag.
2. P. Avesani, P. Massa, M. Nori, and A. Susi. Collaborative radio community. In *Proceedings of Adaptive Hypermedia*, Malaga, Spain, 2002. Springer Verlag.
3. M. Blaze, J. Feigenbaum, J. Ioannidis, and A. Keromytis. The role of trust management in distributed systems security. In J. Vitek and C. Jensen, editors, *Secure Internet Programming: Security Issues for Mobile and Distributed Objects*. New York, NY, 2000.
4. J. Canny. Collaborative filtering with privacy. In *IEEE Conference on Security and Privacy*, Oakland, CA, 2002.
5. C. Hayes, P. Massa, P. Avesani, and P. Cunningham. An on-line evaluation framework for recommender systems. In *Workshop on Recommendation and Personalization Systems*, Malaga, Spain, 2002. Springer Verlag.
6. S. Sen, A. Biswas, and S. Debnath. Believing others: Pros and cons. In *Proceedings of the 4th International Conference on MulitAgent Systems*, pages 279–285, 2000.

Information Ordering vs Information Filtering*

Keith Bradley[1], Barry Smyth[1]
[1]Smart Media Institute, Department of Computer Science,
University College Dublin, Dublin 4, Ireland
Keith.Bradley@ucd.ie, Barry Smyth@ucd.ie

1 Introduction

The CASPER project investigates the potential for personalization in the online recruitment domain. In particular, for reasons of privacy, security, and computational efficiency, it adopts a client-side personalization strategy that is designed to operate as a post-processing stage for a more traditional server-side information retrieval or recommender system. In addition, CASPER's personalization strategy is used as the basis for information ordering rather than information filtering, a decision which, we argue, is appropriate for information retrieval tasks, such as job searches, where recall takes precedence over precision as the primary success criteria.

2 Information Filtering *vs.* Information Ordering

We can consider two types of personalization strategy – *personalized information filtering (PIF)* and *personalized information ordering (PIO)*. The former focuses on the *selection* of information items that are relevant to a particular user. For example, PTV is a well-known personalized television listings system that constructs personalized TV guides for users by selecting TV programmes that are relevant to a given user based on his or her viewing habits [1]. Irrelevant items (programmes) are filtered-out prior to presentation to the user. PIF is perhaps the most common form of personalization, whether personalized TV guides or news articles, travel suggestions or restaurant recommendations. PTV focuses on presenting the user with a small set of highly relevant recommendations only, and thus benefits from a high precision potential. However, its filter-out strategy is also its Achilles heel since it runs the risk of incorrectly eliminating items that may well be relevant, and as such it can suffer from poor recall characteristics.

Personalized information ordering is a related, but alternative strategy. Instead of filtering out apparently irrelevant items, the strategy simply re-ranks these items according to their predicted relevancy for the user in question. Thus items that are likely to be preferred by the user appear first and those that are less relevant, or even

* The support of the Informatics Research Initiative of Enterprise Ireland is gratefully acknowledged.

irrelevant appear last. This strategy tends to enjoy much improved recall when compared to PIF, for obvious reasons, but at the expense of precision.

We argue that in the online recruitment domain it is more important to focus on recall than precision. Precision is important in information retrieval and recommendation tasks where the user is likely to be interested in find one particular information item, for example a restaurant to eat at or a movie to watch. In these tasks, the key success criteria is to make sure that the user is delivered at least one item that is relevant to her personal needs. In contrast, in the online recruitment domain, the user is looking for a new job and will typically be interested in viewing a range of jobs to apply for. In this way it makes good sense to present the user many job options, ranking them, rather than filtering them, according to their relevancy. There is little to be lost by recommending an irrelevant, lowly ranked job, but a lot to be gained by ensuring that all relevant jobs are presented.

3 CASPER

CASPER is a two-stage system. During stage one a user's query is used as the basis for a "traditional" server-side similarity-based search of a recruitment case-base – alternatively CASPER could use a standard data-base retrieval engine during this stage. The main objective of the stage-one retrieval is to identify those job cases that are broadly similar to a query, and therefore likely to be relevant to the particular user.

In CASPER, at the client-side, the user can browse through the search results, look for more detail on particular jobs, and ultimately apply for jobs online. This activity information is used as a means of judging recommended jobs as being relevant or irrelevant to the target user, and this serves as the basis for a graded user profile of positive and negative preferences. These profiles reside on the client-side and the second stage of processing uses the graded cases in a user profile to predict relevancy scores for future job recommendations using a nearest-neighbour case-based prediction technique. We have evaluated a range of different prediction strategies in order to find one that is sufficiently accurate (at both positive and negative predictions) and sufficiently robust under noisy conditions – the predicted user grades exhibit high degrees of noise given the difficulties associated with using behavioural indicators as the basis for relevance. To date a number of successes have been forthcoming and CASPER has proved to be an effective and efficient job recommendation system under a variety of experimental conditions.

4 References

1. Smyth B & Cotter P (2000) Sky's the Limit: A Personalised TV Listings Service for the Digital Age. Journal of Knowledge-Based Systems, 12(2-3), 53-59.
2 Bradley K., Smyth B. (2001) Improving Recommendation Diversity, Proceedings of the 12th Irish Conference on Artificial Intelligence and Cognitive Science, Maynooth, Ireland.

User Aspects of Situated CBR Systems

Jörg Cassens

Norwegian University of Science and Technology (NTNU),
7491 Trondheim, Norway,
jorg.cassens@idi.ntnu.no,
http://www.idi.ntnu.no/~cassens

Introduction

In the Division of Intelligent Systems in Trondheim, we focus on systems which do not only learn from experience, but also incorporate given general knowledge to solve the problems (see e.g. [1]). This is refered to as knowledge-intensive CBR.

The group is aiming towards building a framework for such CBR systems. This involves identifying usable knowledge and reasoning structures as well as questioning how to embed the system in user tasks.

This paper deals with embedding CBR systems into human decision processes. It is heading towards Human Computer Interaction in a wide sense, in particular the question of how to design systems which are not only tools in a conventional sense, but have intelligent abilities.

When AI systems are considered not as a replacement of, but a supplement to human work, the question of an adequate form of interaction arises. An AI system is to a certain degree trespassing the boundary of the computer system as a tool, and extending this to it acting as a partner in a work flow.[1]

Situated AI and Personalisation

A situated system interacts with the world. It observes the parts of the world visible to the system, and adapts its behavior according to a changing environment, and, to a certain degree, vice versa.

Previous work in our group (see e.g. [4]) has argued for the necessity to ascribe certain capabilities to intelligent systems if these systems shall interact with the user in a meaningful way.

We model the context in which the system is implemented with the help of the Actor Network Theory, ANT (see e.g. [5] and [6]). This approach tries to avoid the trap of either overstating the role of technological artifacts in a socio-technological system or underestimating their normative power by applying the same framework to both human actors and technological artifacts.

[1] This relates to different perspectives on Human Computer Interaction, like the tool perspective, the media perspective, or the dialogue perspective. For a detailed discussion of different perspectives, see e.g. [2] or [3].

When focusing on the interaction of a particular user with the system, we use the semiotics approach (see e.g. [7]) to understand the peculiarities of interaction with intelligent systems. On the background of semiotics, human communication is a sign process. In contrast, conventional computer systems are only processing signals, lacking the necessary interpreting capabilities humans have. Both processes have to be coupled.

We now have to bring together this two theoretical frameworks. We argue that in order to make intelligent systems work not merely as a tool or a media, but as actants to whose decision abilities a human user can subscribe, the system must appear to the human user as-if it was capable of a meaningful interaction, hence to participate in a sign process. This includes, but is not restricted to, the ability of the system to adapt itself to different users and their preferences and needs.

With the theoretical background outlined, we have the means of understanding the interaction process in general. In the next step, the aforementioned inter-individual variations have to be analyzed in greater detail. The users will trust the reasoning capabilities of the system only when it is able to show a meaningful behavior in eventually very varying usage situations. Imagine as an example that the same decision support system is used by people with different expertise in the problem domain, than the system has to adopt its explanation of found solutions in several different ways.

Ongoing Work

The semiotics perspective is helpful to understand medial aspects of Human Computer Interaction, e.g. how knowledge is communicated. It is, however, not as helpful to analyze their use as instruments for achieving a predefined (by the human) goal in the work process and especially to understand the transformation of the artifact itself during this process.

In my opinion, the Activity Theory (AT, see e.g. [8] and [9]) can be suitable to cover also this aspects in our framework. Its focus lies on individual and collective work practice. One of its strength is the ability to identify the role of material artifacts in the work process.

AT strictly distinguishes between human and technological aspects and it might therefore at first sight look as if the theory is incompatible with ANT. But this holds also for semiotics, and we already saw how semiotics can be fruitful when applied to intelligent systems in contrast to conventional information systems.

Another problem which needs to be addressed is to operationalize the theoretic framework. For now, we have hints that constraints on the design process exist, but we have no methodology for creating systems with the desired capabilities.

Using CBR for enabling a decision support system (which itself is CBR based) to observe its own user interaction coupled with using a priori knowledge about interaction processes in knowledge-intensive CBR systems is a starting point for the development of such a methodology.

References

1. Aamodt, A.: Knowledge Acquisition and Learning by Experience – The Role of Case-Specific Knowledge. In Tecuci, G., Kodratoff, Y., eds.: Machine Learning and Knowledge Acquisition – Integrated Approaches. Academic Press (1995) 197–245
2. Svanæs, D.: Understanding Interactivity. PhD thesis, NTNU, Trondheim (2000)
3. Donker, H.: Didaktisches Interaktions- und Informationsdesign. PhD thesis, Carl von Ossietzky University, Oldenburg (2002)
4. Pieters, W.: Free Will and Intelligent Machines. Project Report, NTNU Trondheim (2001)
5. Latour, B.: Technology is Society made Durable. In Law, J., ed.: A Sociology of Monsters. Routledge (1991) 103–131
6. Monteiro, E.: Actor-Network Theory. In Ciborra, C., ed.: From Control to Drift. Oxford University Press (2000) 71–83
7. Nake, F.: Human-Computer Interaction – Signs and Signals Interfacing. Languages of Design **2** (1994) 193–205
8. Bødker, S.: Activity theory as a challenge to systems design. In Nissen, H.E., Klein, H., Hirschheim, R., eds.: Information Systems Research: Contemporary Approaches and Emergent Traditions. North Holland (1991) 551–564
9. Wygotski, L.: Ausgewählte Schriften Bd. 1: Arbeiten zu theoretischen und methodologischen Problemen der Psychologie. Pahl-Rugenstein, Köln (1985)

Recommendation in Context

Pádraig Cunningham, Conor Hayes

Department of Computer Science
Trinity College Dublin
Padraig.Cunningham@tcd.ie, Conor.Hayes@cs.tcd.ie

Introduction

Our current interest in personalisation is in making recommendations that are relevant in context. Our work is focused on SmartRadio, an internet-based music radio system. Music items in SmartRadio are organised into playlists. Users can create these playlists by selecting from available items and these playlists can be recommended to other users using collaborative (ACF) and similarity based recommendation [1]. The user provides explicit ratings to the system through the interface shown in Figure 1.
From talking to users of SmartRadio (it is currently deployed on the TCD intranet), it seems clear to us that recommendations should be tuned to the current listening patterns of a user. A user may be interested in Country, Folk and Rock music as evidenced by high ratings in their profile; however, the system should be less inclined to recommend Rock music if they are currently only listening to Country and Folk.

Figure 1. A playlist in Smart Radio. The icons to the right indicate the user's ratings.

This need for recommendations to be relevant in context is by no means restricted to the SmartRadio scenario. If I place a book on Machine Learning in my shopping basket at Amazon it should focus recommendations on similar topics rather than on any different topics that are in my profile. In fact Amazon can do this very well. In the next section we give a brief account of how we produce recommendations that are relevant in context in SmartRadio. The paper concludes with a statement of the implications that an emphasis on context has for evaluation of recommender systems.

2-Stage Recommendation

According to the argument above, recommendations should be more strongly based on recent items in the users' profile. This is achieved in SmartRadio by using two views on the user's listening data. Firstly, the full user history is used to produce ACF recommendations. These are then refined based on similarity to a short term user profile, which we term the *listening context* (see Figure 2). This two-fold strategy is a novel type of MAC/FAC retrieval [3].

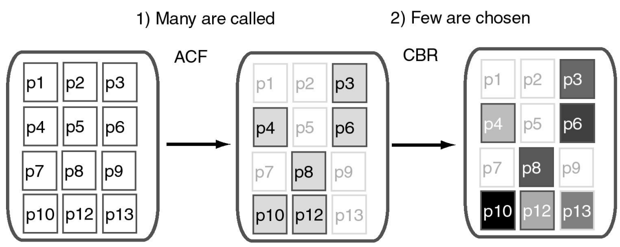


Figure 2. 2-stage retrieval: the darker shaded cases in the third stage indicate cases which best match the listener's current listening context.

Implications for Evaluation

Konstan and Riedl [3] suggest that the evaluation of recommender systems may be either, **Off-line** where the performance of a recommender mechanism is evaluated on existing datasets or **On-line** where performance is evaluated on users of a running recommender system. They argue that on-line evaluation is problematic because of the need to field a fully engineered system and build up a community of users. Consequently they favour off-line evaluation, not because it is better but because it is easier to do. However, it is difficult to see how off-line evaluation can measure how relevant a recommendation is to the current context. For this reason we have proposed an on-line framework for evaluating recommender systems that captures context [2].

References

1. Hayes, C., Cunningham, P., Clerkuin, P., Grimaldi, M., Programme driven music radio, *ECAI'02*, F. van Harmelen (ed.): IOS Press, Amsterdam, 2002.
2. Hayes, C., Massa, P., Avesani, P., Cunningham, P., An on-line evaluation framework for recommender systems. In *Workshop on Recommendation and Personalization Systems*, Malaga, Spain, 2002. Springer Verlag.
3. Gentner, D., and Forbus, K. D. 1991. MAC/FAC: A model of similarity based access and mapping. In Proceedings of the Thirteenth Annual Conference of the Cognitive Science Society. Erlbaum
4. Konstan, J.A., Riedl, J., Research resources for recommender systems. In *CHI' 99 Workshop Interacting with Recommender Systems*, 1999.

A Collaborative Learning & Content-based approach for adapting to users needs (Position Paper)

Andreas Jedlitschka, Klaus-Dieter Althoff, Markus Nick

Fraunhofer Institute Experimental Software Engineering (IESE),
Sauerwiesen 6, D-67551 Kaiserslautern
{jedl| althoff | nick}@iese.fraunhofer.de

In most knowledge intensive areas users are overflowed with information. The aim is to assist users in getting the right information at the right time (e.g., for a the task at hand), at the right place (laptop, handheld), and in a presentation that is, on the one hand, adapted to the user's information requirements (skills and needs) and, on the other hand, to the resource limitations of the device (small interface). In this position paper we will focus on the right information and on the user adaptation step. Therefore we briefly introduce a generic context model followed by the explanation how case-based reasoning (CBR) can support this.

For the first step we will reduce the number of problems with modeling unknown users by focusing on intranet solutions. Here we can assume a well defined environment for the IT users (e.g., roles like department head, administrator). Thus, for most of the users, tasks and workflows can be fixed within processes.

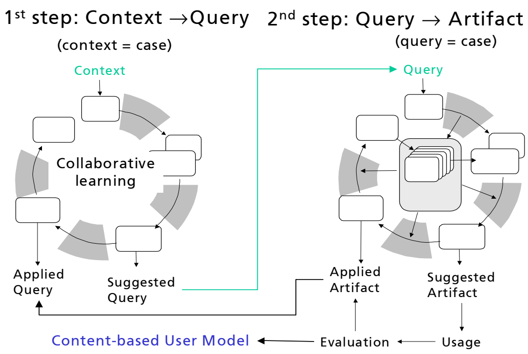
Focusing on admins, which are experts for their topic, it is a fact that in the fast evolving IT sector, it is difficult to keep pace. Our system aims at supporting the administrator through experience-based pre-evaluation of threat potentials and proposing escalation hierarchies ("at the right time") and through an experience-adapted presentation. Therefore, information about the person, his role (admin), his tasks (check logfiles), and the entire environment are necessary. Such information is stored in the context model (object-oriented model).

For a person we have to store typical attributes like the name but also working hours and, especially for task force members, mobile phone numbers, which are used by the system to automatically send a short message in case of events. The level of expertise and experience are stored in an association to task. The initial value for the level of expertise is given by the person itself. The level will be increased by successfully performing a task but can be decreased by the system in case the person has not performed a task for a longer period of time or by the person itself.

Role is an abstract description of the job a person is responsible for (more than one person can have the same role). Additionally, the role contains an abstract role model (aggregated user model), which is used for collaborative learning. So we assume that for persons performing the same role similar information is useful. Also, they will query the experience base in a similar manner. Our approach uses this assumption to push the information to the users if a specific task occurs and other similar users have contributed information or asked questions which led to useful information. With the association to the task, access rights and responsibilities are described.

We distinguish between regular tasks and dynamically arising tasks. A typical task for admins is the analysis of logfiles. This task can be regular, on the one hand, because it is defined in the security management process, and dynamic, on the other hand, because of suspicious actions discovered by the system. For each task documents in the form of detailed descriptions and short checklists are available. Because of the maintenance cycle, documents are relatively static. Thus, annotations are used for dynamic user input of, e.g., tips and tricks, lessons learned, and guidelines. The document describes the workflow of the task. The system leads the user through the task and points to open issues (e.g., occurring after interrupts). The user can decide whether he wants to use this support. His behavior is stored in his user model, so the system can refer to the former wishes of the user (content-based). Each description contains text fragments and a corresponding graphic, which supports navigation and overview. Each text fragment (e.g., single work instruction like change disk) is stored in a database. Together with the graphic, it is loaded dynamically. So we have the possibility to build user-adaptive tasks descriptions. For the resulting task, a checklist can be generated. It is necessary to have an attribute in the characterization part of the fragments that describes whether an instruction is optional. For expert users, a minimal but complete task is granted. The usage of task fragments and annotations is the basis for continuously improving the processes, because the experiences of the users influence the system and other users.

With CBR so called problem-solution pairs can be stored and cases are used for case-based retrieval. With this terminology, the context is treated as a "problem". When a new context is obtained, for example, because the user steps forward in his task, this context is handled as a new case, which has no solution yet. Similarities are used to retrieve the most appropriate matching case, which then includes a solution representing queries successfully used in a former context. In the sense of user modeling, this is a collaborative learning approach, because similar contexts are used for estimating the actual context. The query gained in the first CBR cycle is the input for the second CBR cycle, resulting in a similar case with a query-artifact pair. This artifact is presented to the user. Users can also pose a new query, which is then stored in the query case base.



The evaluation of delivered information by the user is necessary for future evolution of the whole system. If the user agrees with the information, the value of the query that led to the content and the content itself are increased (the same will happen to a successfully reused context-query pair). In case of rejection, the value of the query for the next retrievals is decreased. With his collaboration, the user also contributes to his own model. The evaluation data deliver data about the actual interests and refine the context. So we also have the content-based approach to user modeling. The result of the two-step CBR and the user evaluation are used for better forecast of users' needs with regard to information.

Personalized CBR: Challenges and Illustrations*

David Leake

Computer Science Department, Indiana University, Lindley Hall 215
150 S. Woodlawn Avenue, Bloomington, IN 47405, U.S.A.
leake@cs.indiana.edu

Personalization research has made significant progress on the task of recommending appropriate items from a set of alternatives. This has resulted in valuable technologies for satisfying user needs, and some of these technologies apply case-based reasoning with considerable success. By helping highlight the need to reflect user preferences and customize information presentation, personalization research raises an important question for CBR research in general: When and how should be CBR knowledge containers and processing steps be refined to reflect personal information needs? The resulting refinements need not be restricted to systems that provide recommendations; in fact, personalization may be appropriate for the results of any problem-solving or interpretive CBR systems whose users have varying needs.

Developing systems reflecting individual needs requires replacing the *task-centric* view common to many CBR systems—that there is a single solution for each problem—with a *user-centric* view that supports multiple solutions, based on the user as well as the problem situation.[1] A case-based explanation system, for example, might need to generate different explanations depending on the particular user: A device failure might be explained differently for a mechanic trying to fix the problem ("part X has failed"), a designer trying to determine how to change the design to prevent similar future failures ("part X is subjected to more vibration than anticipated"), or someone interested in business strategy ("manufacturer Y uses cheap parts") [2].

To provide the solutions that a user needs, CBR systems must address the challenge of determining what those needs are. Doing so requires augmenting traditional *situation assessment*, aimed at the domain problem, with *needs assessment* methods that may fall anywhere in the spectrum from fully automatic—requiring no additional user input—to requiring extensive user input or interactions. Once needs have been identified, another challenge is how to combine personalization with the CBR process. Just as a CBR system's knowledge can be placed in multiple knowledge containers, personalization-relevant knowledge may be distributed among different knowledge containers, and personalization processes can take place at different steps in the CBR cycle.

The CBR group at Indiana University has pursued a number of approaches to providing personalized information, addressing different knowledge containers and processing steps:

– **Personalizing indexing with automatically-generated context descriptions:** The Calvin project [6] develops methods for supporting task-driven research. Calvin automatically stores cases recording which information resources researchers consult

* This research is supported in part by NASA under award No NCC 2-1216.

[1] It is possible, of course, to revise the problem description to include user information; this transforms the system's task to explicitly include generating personalized output.

during their decision-making and uses these cases to proactively suggest information resources to consult in similar future task contexts. The system automatically characterizes patterns of user document accesses over time, which it uses to generate user and context profiles. There are in turn used to guide retrievals [1]. Calvin relies entirely on unobtrusive monitoring; no additional information is requested from the user. DRAMA [5], which explores the use of concept map cases for design, allows users to specify additional information to help focus retrieval. The PRISM project [4] applies context descriptions to select relevant sources, and—in concert with Calvin's context descriptions—could select user-appropriate case-bases for future processing. In all these approaches, personalization targets indexing and retrieval.
- **Personalizing case representations with concept maps:** The DRAMA and CMap Suggester [3] projects support using concept maps as cases, to enable flexible case representations serving idiosyncratic user needs. The Suggester helps retrieve cases that not only have similar contents, but similar representations, to increase understandability and facilitate re-application. In this approach, personalized retrieval and personalized cases are used in concert; the personalization is integrated with the CBR process, with interactive user adaptation
- **Personalizing case evaluation to guide user-centric adaptation:** ACCEPTER's case-evaluation process [2] adds usability evaluation to the standard case evaluation process, to focus additional adaptation on satisfying the user's information needs. Thus ACCEPTER can be seen as augmenting the task-centric CBR cycle with another phase, which may be considered *usability assessment*, analogous to situation assessment but revising solutions to individual needs.

These systems provide a sample of the diversity of approaches that can be applied to integrating personalization into the CBR knowledge containers and processing steps. A challenge for personalized CBR is to bring diverse methods together, applying them strategically in a unified framework that brings them to bear where they will most benefit the user.

References

1. T. Bauer and D. Leake. Wordsieve: A method for real-time context extraction. In *Modeling and Using Context: Proceedings of the Third International and Interdisciplinary Conference, Context 2001*, Berlin, 2001. Springer-Verlag.
2. D. Leake. Goal-based explanation evaluation. *Cognitive Science*, 15(4):509–545, 1991.
3. D. Leake, A. Maguitman, and A. Cañas. Assessing conceptual similarity to support concept mapping. In *Proceedings of the Fifteenth International Florida Artificial Intelligence Research Society Conference*, pages 168–172, Menlo Park, 2002. AAAI Press.
4. D. Leake and R. Scherle. Towards context-based search engine selection. In *Proceedings of the 2001 International Conference on Intelligent User Interfaces*, pages 109–112, 2001.
5. D. Leake and D. Wilson. A case-based framework for interactive capture and reuse of design knowledge. *Applied Intelligence*, 14:77–94, 2000.
6. David Leake, Travis Bauer, Anna Maguitman, and David Wilson. Capture, storage and reuse of lessons about information resources: Supporting task-based information search. In *Proceedings of the AAAI-2000 Workshop on Intelligent Lessons Learned Systems*, Menlo Park, CA, 2000. AAAI Press.

Preference-Based Feedback for Recommendation[*]

Lorraine McGinty, Barry Smyth
Smart Media Institute, Department of Computer Science,
University College Dublin, Dublin 4, Ireland
Lorraine.McGinty@ucd.ie, Barry Smyth@ucd.ie

1 Introduction

Many recommender systems operate as interactive systems that seek feedback from the end-user as part of the recommendation process in order to revise the user's query and guide the recommendation process (eg, [1]). Four different feedback strategies have been incorporated into recommender systems to a lesser or greater extent. *Value elicitation* and *tweaking* are feature-level techniques in the sense that the user is asked to provide information about the features of a recommended case; in the former they must provide a specific value for a specific feature (eg. show me *PCs* whose *type* is *laptop*) while the latter requests that they simply express a directional preference for a particular feature value (eg. show me *PCs* that are cheaper that the current recommendation). In contrast, *ratings-based* and *preference-based* feedback methods operate at the case-level. In the former the user is asked to rate or grade the recommended cases according to their suitability, while in the latter the user is simply asked to select one of the current set of recommendations that is closest to their requirements (or indeed farthest from their requirements).

Value elicitation, tweaking, and ratings-based feedback have all been used in a variety of recommender systems. However, preference-based feedback is a less popular choice. It seems that the reason for this is the assumption that this simple form of feedback carries very little information to guide the recommendation process. Neverthess this form of feedback does have a couple of major advantages – it requires very little effort for a user to indicate a simple preference, and also, users are often able to indicate a preference even when they have very little understanding of domain features. In our work we believe that these advantages suggest that preference-based feedback deserves further attention. In fact we suggest that even simple preferences can prove to be a valuable guide for a recommender system.

2 Comparison-Based Recommendation

The comparison-based recommendation approach is an iterative recommendation technique. During each recommendation cycle the user is presented with a set of *k*

[*] The support of the Informatics Research Initiative of Enterprise Ireland is gratefully acknowledged.

cases and asked to indicate a preference for one of these. This preference feedback is used to update the current query, which is then used as the basis for the next cycle. The process finishes when the user indicates that an appropriate case has been presented; see [2] for further details.

This comparison-based recommendation framework allows us to evaluate different strategies for converting simple user preferences into valuable guidance for the recommender. In particular, we have explored a number of different strategies for updating the user query during each cycle. For example the simplest strategy is the *more like this (MLT)* technique, which transfers the entire preference case to form the new query. The problem with this approach is that the presence of a feature in the preference case does not necessarily mean that the user has a preference for more cases with this feature. For example, in a PC recommender, a user may have indicated a preference for a $1000 PC with a 14" monitor because of its price, even though they are ultimately looking for a larger screen size. In other words the MLT strategy will tend to overfit the query to the current preference and ultimately mislead the recommender. An alternative strategy is to be more selective in the way that features are transferred to the query. For example, the *partial MLT (pMLT)* update strategy only transfers those features of the preference case that are not present in any rejected (non-preference) cases. In this way spurious preferences can be eliminated from the query. A number of other update strategies have also been explored, including ones that attempt to weight the query features and others that capture negative preferences from the rejected cases.

Initial results indicate that preference-based feedback can be used to efficiently guide the recommendation process. They also confirm that the simple MLT strategy, which is commonly used by a number of search engines, can lead the recommender astray following false preferences. In contrast, the partial MLT and weighted MLT strategies result in more efficient recommendation sessions with an average reduction in the number of cycles needed to locate a target case of up to 30%.

Currently we are exploring different ways of extending preference-based feedback in comparison-based recommendation. For example, manipulating the diversity of retrieval results and accumulating preference information over a number of cycles shows some promise for further improvements.

3 References

1. Burke, R., Hammond, K. and Young, B.C. (1997) The FindMe Approach to Assisted Browsing. IEEE Expert, 12(4), 32-40.
2. McGinty, L, and Smyth, B. (2002) Comparison-Based Recommendation. Proceedings of the 6th European Conference on Case-Based Reasoning. Aberdeen, Scotland.

Dimensions of Personalization and their Effect on the Knowledge Containers in a CBR System

Mehmet H. Göker

Kaidara Software
330 Distel Circle, Suite 150, Los Altos, CA 94022
mgoker@kaidara.com

1. Dimensions of Personalization

The ever-growing amount of accessible data makes intelligent information search and selection without computational aid a hopeless venture. Personalization becomes an ever more desirable feature for IT systems. A computer system should ultimately be sophisticated enough to take individual variations in preferences, goals, and backgrounds into account and generate, select, and present personalized information. The goal of personalization is to make the interaction with a system subjectively more effective and efficient. Personalized systems obtain user preferences trough interactions with users, summarize these preferences in a user model and utilize this model to adapt themselves to generate customized information or behavior.

Personal preferences can have an effect on the *data processing level*, the *information filtering level*, and *the interaction and information presentation level* of a system. On the data processing level, the algorithms used on a data set to generate new information can be varied in accordance with the user's preferences. During information filtering, the results of the data processing algorithms can be screened based on the preferences of the user and subjectively irrelevant choices can be eliminated. The presentation of the information as well as the interaction with the user is also subject to personal preferences and needs [1].

Typically, a personalized CBR system will adapt itself by modifying the way it selects suitable items from the set of previously generated information in the case base, i.e. on the information filtering level.

In order to retrieve personalized solutions with a CBR system, it is necessary to acquire and model the preferences of the users along several dimensions. A user may have preferences with respect to:

- specific *items* (information entities),
- the relative importance of an *attribute* used in describing these items,
- *values* for an attribute of the items,
- the *combination* of certain attribute-value pairs, and
- the *diversity* of the suggested items and values.

Item preferences manifest themselves in the user having a bias for or against a certain item, independent of its characteristics (*item preferences*). The preferences regarding an attribute represent the relative importance a user places on the attribute while selecting an item (i.e. how important is cuisine vs. price: *attribute preferences*). Preferred values show the user's bias towards certain types of items (e.g. Italian restaurants vs. French restaurants: *value preferences*) whereas preferences for certain property combinations represent certain constraints with respect to the combined occurrence of characteristics in an item (accepts Mexican restaurants only if they are cheap: *combination preferences*). While the item preferences are related to single items, the attribute, value, and combination preferences are applicable to the retrieval process in general and the *diversification preferences* model the suitability of an item or value at a given time [2].

2. Effects of User Preferences on the Knowledge Containers

Personalization along the dimensions mentioned above will modify the behavior of a structural CBR system by adapting the knowledge containers [3]. The following table lists some possible interactions for both positive and negative feedback regarding each dimension. The table is not meant to be exhaustive and can be expanded.

Kn. Cont: ***Preference***	***Vocabulary***	***Similarity Metric***	***Adaptation Knowledge***	***Case Base***
Attribute	Potential removal if not of interest.	Update of weighting factor(s).	Modify effect on adaptation in multi-attribute adaptation rules.	Indexing strategy can be adapted.
Value	Potential removal if not of interest.	Update of similarity metric.	Default value can be set. Adaptation rules can be updated.	Default value can be set.
Item	Values that are unique to this item can be removed.	Exceptions in similarity metric.	Default to equivalent but preferable item.	Potential removal from case base.
Combi-nation	Attributes may need to be combined.	Multi-attribute similarity metric.	Rules for cross-attribute Query completion and case adaptation can be learned.	Indexing strategy can be revised.
Diversifi-cation	Time-dependent query and case represen-tation.	Time-dependent similarity metric.	Time dependent adaptation rules can be learned.	Clusters of cases can be build to suggest ‘equivalent alternatives’.

Table 1: Effects of Personalization on the Knowledge Containers

Since the concept of a container includes the possibility of moving knowledge from one container to the other without changing the total amount of information (at a given time), it is obvious that these effects will influence each other and can potentially be implemented in one container rather than the other.

3. Future Work: Personalization as Maintenance of a CBR System

If we define the goal of CBR system maintenance to be the preservation, restoration, or enhancement of system performance in a given context, we can claim that personalization can be viewed as CBR system maintenance in the context of a specific user and at a given time. Personalization will ensure that the performance of the system does not degrade and potentially is enhanced. The comparison of CBR systems that were initially identical but have been personalized for different users could also provide means to automate some of the rather difficult decisions required in CBR system maintenance.

4. References

[1] Göker M., Smyth B. *'Delivering Personalized Information: What You Get Is What You Want'* , in Bergmann R, Wilke W. (eds.), 'Special Issue on Artificial Intelligence and E-Commerce', German Artificial Intelligence Journal (Künstliche Intelligenz) 1/01, pp 17-22, January 2001.

[2] Göker M., Thompson C. (2000), 'Personalized, Conversational Case-Based Recommendation', pp. 99-111, in E.Blanzieri and L. Portinale (eds.), "Advances in Case-Based Reasoning, Proceedings, 5th European Workshop on Case-Based Reasoning, Trento Italy, 6-9 September 2000", LNAI 1898, Berlin, Springer Verlag, 2000

[3] Richter M. (1995), "Introduction", in Lenz M., Bartsch-Spörl B., Burkhardt H. D., Wess S. (Eds.), "Case-Based Reasoning Technology, From Foundations to Applications", Lecture Notes in Artificial Intelligence Vol. 1400, pp.1-15, Springer-Verlag, Berlin, Heidelberg 1998. Also: Richter M., "The Knowledge Contained in Similarity Measures", Invited talk at ICCBR95, http://wwwagr.informatik.uni-kl.de/~lsa/CBR/Richtericcbr95remarks.html

Personalizing Navigation Structures in a Mobile Portal*

Barry Smyth
ChangingWorlds Ltd., South County Business Park,
Leopardstown, Dublin 18, Ireland.
Barry Smyth@changingworlds.com

1 Introduction

The mobile Internet (epitomized by WAP) has so far failed to meet user expectations. Many factors have been responsible – unreliable early handsets, limited content, slow connections, and poor portal usability. Today, the first 3 of these issues have been largely solved through device and infrastructure improvements but usability remains a problem, limiting the ability of users to locate, and benefit from, wireless content.

The usability problem is that users spend too much time navigating to content through portal menus, a problem that is exacerbated by the fact that users are usually charged for this navigation time. A recent usability study indicates that while the average user expects to be able to access content within 30 seconds, the reality is closer to 150 seconds [1]. The result: today WAP offers users poor value-for-money.

2 Click-Distance & Personalized Navigation

Portal navigation effort, with a mobile handset, can be usefully modeled as *click-distance* – the number of menu *selections* and *scrolls* needed to locate a content item – and our studies indicate that many portals suffer from average click-distances (home page to content items) in excess of 15 to 20. One way to improve the usability of a portal is to optimize the click-distance to content sites. However, this is not possible using conventional portal design techniques because, inevitably, as more content is added to a portal, more menus and navigation structures must be added in order to help the user access this content. This is especially true in the mobile domain because there are severe limitations on the number of options that can be presented on a single menu page – for practical reasons a typical menu must contain less than 10 options.

However, using personalization techniques it is possible to reduce the click-distance of a portal by selectively reordering and promoting menu options in line with a given user's short and long-term preferences. The ClixSmart Navigator system by ChangingWorlds achieves this by using a multi-strategy personalization approach that combines probabilistic and collaborative techniques. A standard deployment architecture is outline in Fig. 1. When a user U selects a menu option O (1), the request is intercepted by the navigator server (2). The server updates (3) and accesses

* The hard work, and dedication of the ClixSmart Navigator team is gratefully acknowledged.

(4) U's profile and request the menu page M that corresponds to O (5) from the portal content database. The navigator server then adapts M for U by promoting and reordering its links as appropriate. This newly adapted menu M' is returned through the gateway to user U.

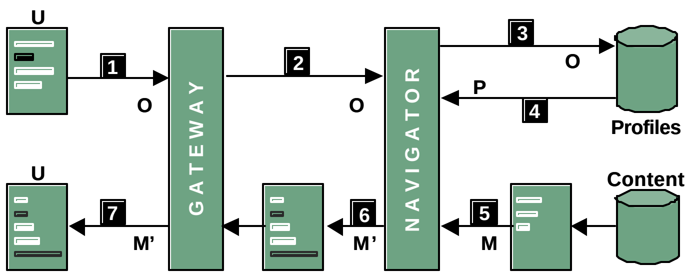


Fig. 1. The ClixSmart Navigator standard deployment architecture and information flow.

3 Discussion

Traditionally, the lion's share of attention, when it comes to personalization, has been directed at content personalization, with methods such as content-based and collaborative filtering proving to be effective at filtering information items according to a user's implicit and explicit preferences. In this position paper, we are emphasizing the important of personalized navigation, which is not concerned with identifying individual items of information (TV programmes, job adverts, news articles) that are relevant to the user, but rather seeks to identify navigation paths through a portal that lead to relevant content services. In this sense, we argue that personalized navigation represents the initial phase of portal personalization – helping users to discover and locate relevant content services – with content personalization playing a critical role once the user has located a particular service of interest.

The ClixSmart Navigator solution has shown click-distance reductions of over 50% with mobile usage increases in excess of 40% for a number of European mobile operators. In particular, the success of its personalized navigation strategy as a means of improving the mobile user experience is highlighted by the fact that these deployments have shown that for every second of navigation time that is saved, users are willing to engage in an additional 3 seconds of content time. In short personalized navigation is now recognized as a key technology for mobile portals going forward.

4 References

1. Ramsey, M. and Nielsen, J. (2000) WAP Usability Report. Neilsen-Norman Group
2. Smyth B & Cotter P (2002) Solving the Navigation Problem for Wireless Portals. Proceedings of the 15th European Conference on Artificial Intelligence. Lyon, France. IOS Press, pp. 608-612.

State of Interest: Learning Personalized Utility Requirements

Armin Stahl

University of Kaiserslautern, Computer Science Department
Artificial Intelligence - Knowledge-Based Systems Group
67653 Kaiserslautern, Germany
stahl@informatik.uni-kl.de

Abstract. Defining similarity measures is a crucial task when developing CBR applications. Particularly, when employing utility-based similarity measures rather than pure distance-based measures one is confronted with a difficult knowledge engineering task. Especially if different users of a CBR system have different demands on the case retrieval this task becomes really complex and sophisticated. In such a scenario, identical queries may require different retrieval results depending on the context of the particular user because the utility of the cases for the current problem situation of the user may vary significantly.

Consider a product recommendation system in e-Commerce. Here the users are customers with individual preferences with respect to the offered products. For example, some customers focus more on the price of a product while others are mainly interested in the technical properties. These preferences, for example, can be represented in form of attributes weights, i.e. they can be encoded into the similarity measure used to retrieve suitable products. However, this approach may significantly increase the knowledge engineering effort when developing a recommendation system based on CBR. Instead of defining one domain specific similarity measure, one has to define several measures that consider the specific preferences of individual customers or customer classes, respectively. But even if one is willed to put up with this additional effort, it is still an open question how to acquire the required knowledge.

In our point of view, here a learning approach may help to facilitate both issues. Firstly, it may reduce the effort to define several similarity measures. Secondly, it is probably the only feasible way to obtain the required knowledge. To apply such an approach, one has to define one or several initial similarity measures that approximate the user specific utility measures as well as possible. During the use of the system one has to acquire feedback about the quality of the retrieval results from the users to learn more specific measures for each user or user class.

Intelligent Travel Recommendation

Francesco Ricci and Adriano Venturini

eCommerce and Tourism Research Laboratory
ITC-irst
via Sommarive 18
38050 Povo, Italy
{ricci,venturi}@itc.it

1 Recommender Systems

Major tourist web sites, such as Expedia or Travelocity, offers a selection of "predefined" tourist destinations, extracted from electronic catalogs of products and allow the user to select tourism products like flights, accommodations, cars and cruises. The man/machine interaction pattern, which is implemented by these web sites, is very simple and largely adopted by eCommerce applications. The user enters some constraints, expressed over features describing the products; the system compares the product features with the expressed constraints; and shows the matching products; the user examines the selected items, and, either selects and eventually buys one of them or changes the constraints and starts a new search. In tourism applications, this approach has some limitations. Tourist products have a complex structure, whose definitions have not been standardized [9]. The tourist decision process is much more complex [7, 5]. The main process variables cannot be easily translated into product features.

To overcome the above mentioned limitations, tourist web sites are incorporating recommender systems, i.e. applications that provide advice to users about products they might be interested in [2]. The two most successful recommender systems, triplehop.com [4] and vacationcoach.com can be classified as mainly content based. The user expresses needs, benefits and constraints using his language (attributes) and the system matches this description with items of the catalog that have been described with the same attributes. Content based approaches can be filter based retrieval, similarity based retrieval, and more recently, ordered based retrieval [1]. One of the problems of the filter based retrieval is that if no item matches the filter, the result set is empty. For this reason, triplehop and vacationcoach use the similarity based retrieval, which is well known in the CBR community. On the other hand, filtered based retrieval, as we have argued above, is very popular, easy to understand and useful. Therefore, there have been proposed some approaches were similarity based retrieval is integrated with filtered base [3] or were similarity and filter based approaches are converted in a new one, ordered based, where partial orders are built using a range of preferences/constraints input, including filters and similarity.

2 Intelligent Travel Recommendation

We have developed a novel approach to retrieval in recommender systems which combines logical filters and similarity-based ranking. We first support the user in building the logical filter by an interactive query management module (IQM) [6]. This enables to select a convenient number of product items which are then ranked according to the similarity with the items contained in a set of pertinent (similar) recommendation sessions. Therefore, ranking is provided by two similarity measures: item similarity and session (case) similarity.

ITR (Intelligent Travel Recommender) is built upon the above schema, and enables the user to bundle his own personalised travel by iteratively selecting *travel items* (e.g., a hotel or a visit to a museum or a climbing school). In this way, the user builds his *travel bag*, that is, a coherent (from the user point of view) bundling of products. The system exploits a case base of travel bags built by a community of users as well as catalogues provided by a Destination Management Organization (APT Trentino). The user's can pose constraints over the products in the catalog, and the system, in a conversational way, helps the user by suggesting changes to the query and ranks travel items, exploiting the case base. For a full description of the system please refer to [8]. Here we briefly describe the two major components: interactive query management and product item ranking.

2.1 Interactive Query Management

ITR tries first to cope with user needs satisfying the logical conditions expressed in the user's query and, if this is not possible, it suggests query changes that will produce acceptable results. This process is conversational: first, the user poses some initial constraints, then the system examines the produced result set and the available features. If too many items match the expressed conditions, the system suggest additional features that can be added to the query to tighten the result set. Here an hybrid unsupervised feature selection algorithm is used. This algorithm combines general expert knowledge about features' relevance with statistics over the data (entropy and fetures' mutual information). Vice versa, if no result is found, the system suggests the user some changes to the query that could produce some results. This stage is implemented by searching for those constraints that are responsible for the failure and building a limited set of alternative queries.

2.2 Product Item Ranking

After that the user, helped by IQM, has selected a reasonable amount of items, the system ranks them by exploiting the case base of previous recommendation sessions.

Firstly, the system uses similarity to retrieve the most similar old cases (travel bags), built by other users and stored in the case base. Cases are a complex hierarchical structures composed of: general wishes, travel products (items) selected by the user, and rates the user may have expressed on the travel items. The general wishes (e.g. travel party, the budget, the type of desired activities) are constraints and preferences that describe the desired travel in a more abstract language with respect the features used to describe the single products.

Secondly, the travel items contained in the past cases are then used to rank the items selected by the user. The basic idea is that items selected by the logical filter that are more similar to those contained in previous similar session must be ranked highly. Thus, similarity is used twice. First, to select the past cases (reference set). Then, to rank the result set of the user's query according to the similarity to the items contained in the reference set.

References

[1] D. Bridge. Product recommendation systems: A new direction. In *Procs. of the Workshop Programme at the Fourth International Conference on Case-Based Reasoning*, pages 79–86, 2001.

[2] R. Burke. Knowledge-based recommender systems. In J. E. Daily, A. Kent, and H. Lancour, editors, *Encyclopedia of Library and Information Science*, volume 69. Marcel Dekker, 2000.

[3] R. Burke, K. Hammond, and E. Cooper. Knowledge-based navigation of complex information spaces. In *Proceedings of the 13th National Conference on Artificial Intelligence*, pages 462–468. American Association for Artificial Intelligence, 1996.

[4] J. Delgado and R. Davidson. Knowledge bases and user profiling in travel and hospitality recommender systems. In *Proceedings of the ENTER 2002 Conference*, pages 1–16, Innsbruck, Austria, January 22-25 2002. Springer Verlag.

[5] D. R. Fesenmaier and J. Jeng. Assessing structure in the pleasure trip planning process. *Tourism Analysis*, 5:13–17, 2000.

[6] T. Gaasterland, P. Godfrey, and J. Minker. An overview of cooperative answering. *Journal of Intelligent Information Systems*, 1(2):123–157, 1992.

[7] Y.-H. Hwang, U. Gretzel, and D. R. Fesenmaier. Behavioral foundations for human-centric travel decision-aid systems. In *Proceedings of the ENTER 2002 Conference*, Innsbruck, Austria, January 22-25 2002. Springer Verlag.

[8] F. Ricci, B. Arslan, N. Mirzadeh, and A. Venturini. Itr: a case-based travel advisory system. In S. Craw, editor, *6th European Conference on Case Based Reasoning, ECCBR 2002*, Aberdeen, Scotland, 4 - 7 September 2002. Springer Verlag.

[9] H. Werthner and S. Klein. *Information Technology and Tourism - A Challenging Relationship*. Springer, 1999.