Smart Radio - a proposal

Conor Hayes, Dr. Padraig Cunningham

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

Abstract. There is significant value in having predictions for an item before deciding whether to invest time or money in consuming that item. In a web based scenario where the items are multimedia items such as audio, recommendations can be made to users based on an understanding of their previous consumption or their indications of likes and dislikes. We examine two types of recommendation: content based and non-content or collaborative recommendation. We then apply out thinking to the area of new internet services such as online radio, and propose an architecture for an intelligent music radio system. We then suggest the efficacy of using conceptual clustering techniques in such a paradigm.

1. Introduction

The growth of intelligent systems on the web is driven by the need to assist users in finding or configuring information relevant to them. As greater amounts of services, products and information become available on the Internet intelligent *personalised* assistance will become a necessity if people are to efficiently locate and use web resources.

Predictive utility becomes an important issue in this context. In certain domains there is significant value in having predictions for an item before deciding whether to invest time or money in consuming that item (Konstan et al. 1997). In each domain predictive utility is not simply a measure of accuracy; it is a measure of how effectively predictions influence user consumption decisions. A domain with high predictive utility is one where users will adjust their decisions a great deal based on predictions.

In this paper we review two methodologies suited to maximising predictive utility, case based reasoning (CBR) and automated collaborative filtering (ACF). The domain to which we have chosen to apply these is that of online music recommendation. We develop this idea from a marketing slant and then put forward an application architecture with a functionality similar to that of radio. This radio application, however, is constantly monitoring its listeners' likes and dislikes and suggests new items using predictions made by a hybrid ACF/CBR system. Our research proposes to extend this recommendation system using conceptual clustering techniques.
In section 2 we compare the strengths and weaknesses of CBR and ACF and conclude that a hybrid approach overcomes the weaknesses of both methodologies. We introduce the personalised recommendation systems used by online CD vendors such as Amazon and CDNOW. In section 3, we

argue that the marketing aspect of Internet commerce has been neglected. The smart radio system we propose is both a marketing tool and a service. Its marketing facility is indirect in that listeners are provided with a choice of music to their taste, and have the possibility of purchasing. Otherwise, the system operates as a radio. Section 4 introduces the precedent for internet radio, and the growing trend for radio programmes to be downloaded on demand. Sections 5 and 6 describe the radio application we have in mind, the user requirements and the technology we plan to use in its implementation. Section 7 illustrates our proposed hybrid ACF/CBR distributed architecture and introduces the concept of a *play-list*, a mutable collection of tracks, which can evolve over time. we outline other research ideas associated with this application: the feasibility of distributed CBR/ACF retrieval, the need for a filtering system to keep track of items listened to over time, so as not to have the same items reoccurring at too short an interval, and the concept of the trusted *recommendee*. In section 8, we apply the ideas for a recommendation system outlined in section 2 specifically to the domain of internet radio, and we suggest a role for clustering techniques to build new play-lists on the fly. The possibility of using the system CD player to seed an ACF system is also presented.

2. CBR and ACF

Case Based Reasoning (CBR) is a *content based* methodology (Balabanovic, Shoham 1997) suited to providing expert advice in domains that traditionally required the judgement of an expert who could recall and readapt solutions to similar past problems.
Typically CBR proceeds by finding a similarity between a query case and a subset of cases in its case-base memory. Cases are described by feature value pairs and there are several methods available to organise case memory and calculate similarity between case items.

Automated Collaborative Filtering (ACF) is a non-content based recommendation methodology. Recommendations are made not on the basis of similarities between a user profile and feature descriptions of the domain items, but on the similarities between the user and other users.

Both methodologies have their benefits and drawbacks. The feature based retrieval of CBR means that it will consistently find similar items to that specified in a user profile. However, it will tend to over-specialise, not finding new items outside the user profile that might be of interest. It is also limited by the granularity of its case description, and cannot make subtle distinctions between items which are similar in feature terms but intuitively dissimilar in the eyes of the user. Building such refinement into a CBR system means further refining the granularity of its case description, leading to knowledge engineering overheads, longer retrieval times and larger storage requirements. It could be argued that no case description will ever be fine enough to satisfy the intuitive distinctions people make between apparently similar items.

We could position a purely Automated Collaborative Filtering at the other end of the content-based retrieval scale. No descriptive features at all are used to recommend particular items, instead users are described in terms of their likes and dislikes of individual items.

Similarities between the tastes of different users are then used to recommend or advise against items. This method of recommendation has been called *social information filtering* in that it models the word of mouth recommendations made amongst people, particularly where there is a shared interest. Clearly this methodology fills the gaps outlined by the content-based methods of CBR. New items of a type not yet rated by the user can confidently be recommended. Furthermore, it harnesses the ability of people to make subtle distinctions between seemingly similar items.

The drawbacks of ACF are that it requires a large user group to be effective in finding accurate similarities between users and that it is largely inaccurate until it has been adequately *bootstrapped* - until a user has rated enough items to be associated with a user group. Furthermore, an ACF system requires dynamic input from the world outside its domain of users. For ACF to be effective over time users cannot solely rely on ACF recommendations to find new items of interest. In order to keep the system from running dry they must find new items by external means and rate them. However, if the number of users is small relative to the volume of information in the system there is the danger of the coverage of ratings becoming very sparse, making it difficult to find recommendable items in the first place.
Another problem is that for users whose tastes are obscure there may not be other users who are particularly similar, and accurate recommendations may be impossible.
Because ACF is a non content based methodology the lack of access to the content of each item means that users will not be rated as belonging to a similar group unless they have rated the exact same item. This may be misleading in circumstances where certain items may be very similar.

Interestingly, while ACF plugs the gaps in a purely content based CBR approach, the latter provides some solutions to the drawbacks of ACF just outlined. It is not surprising therefore that there have been successful hybrid implementations of the two methodologies (Balabanovic & Shoham 1997, Smyth, Cotter & O'Hare 1998)

2.1 Internet Commerce and Smart Recommendation

One area of Internet commerce in which there has been significant growth is the area of online retailing of books and CDs. **Amazon**, the largest online book seller has now expanded its market to include sales of CD products. The companies retailing these items have begun to use personalised recommendation systems. Amazon, for instance, greets every user by her name and suggests a number of books that the user might like. These recommendations are based on likes or dislikes the user has expressed while setting up a profile for Amazon. However, unless the user actively seeks to change the profile or purchases a product, her recommendations remain the same.

CDNow is one of the largest online CD retailers. Like Amazon it provides a facility to search their data base according to *artist, title*, *song title*, *record label* and *soundtrack*. Their site includes a weak collaborative recommendation filtering system.
Like other online music sites CDNOW provides excerpts from album tracks in the form of streaming audio files.

Amazon and CDNow both have taken steps to assist online shoppers find a product personally suited to them. Indeed several reviews have concluded that for successful commerce on the web, retailers must steer away from the idea of an online catalogue and provide intelligent sales support as a good shop assistant might. [Hayes et al., Wilke et al.] Indeed, based on this traditional notion of commerce, Wilke et al. develop a sophisticated model of the different stages entailed in an electronic commerce transaction. This approach is an excellent analysis suited to the idea of a proactive shopper who specifically goes to a site to find or configure a product suited to her needs.

3. Marketing:

However, not all sales transactions are conducted in a proactive way. People are encouraged to buy from a variety of sources - TV, Radio, Magazines and papers, word of mouth. The e-commerce scenario described by (Wilke 1998) does not take into account how people come to arrive at an internet shop in the first place, or what gives them the idea to buy a certain product. In other words the pre-sales situation described by Wilke assumes that the customer has sought out the vendors e-boutique and that a sale will follow as a matter of consequence, once both vendor and purchaser are content. This is a reassuring position. Prior to this however, there must be some technique whereby the potential customer enters an online store. Why should the customer decide to buy from one vendor as opposed to another? In marketing terms what can one vendor offer that makes returning to shop an attractive prospect. This question becomes more pertinent when the goods being sold are not proprietary but standard goods such as books and CDs. Cutting Prices and offering special deals are methods designed to attract customers, but not necessarily their loyalty. Online retailers are currently attempting to woo customers by making claims about their size rather than service. The rivalry between Barnes and Noble (The world's largest booksellers online) and Amazon (The world's largest online bookseller) is an example of this.

It is our belief that an automated system of personalised assistance and service is the key to bringing about customer loyalty. However, only Amazon currently appears to have a cohesive policy towards customer assistance. We have noted the potential for internet brokerage firms (Hayes 1999), and in this paper we will present our ideas for a personalised recommendation system that can act as a brokerage service and a CD marketing device.

The biggest marketing tools for music products (the CDs, the fashion regalia and ticket sales) are radio and television. Because popular music is such an aspect of fashion, word of mouth and peer influence is also a major factor in music sales. Whereas, television and music magazines actively market the music industry (television asks you to buy the image behind the music as well as the song itself). Radio, incidental listening (hearing music at your friends house or at a club)and word-of mouth are a much more passive marketing methods.

The marketing aspect of this application follows along this line. In one sense it provides a service much like music radio. In another sense, it is a marketing tool providing the listener with a free sample of what is available to buy.

4. Radio

The success of cheap streaming technology has meant that Radio is no longer confined to its geographical position. The MIT List of Radio Stations on the Internet (http://wmbr.mit.edu/stations/list.html) contains links to over 6000 stations, many of whom broadcast their signal over the internet using real audio software http://wmbr.mit.edu/stations/list.html You can for instance listen to Jazz broadcast from San Diego on KIFM (http://www.kifm.com/) or Classical music broadcast from Chicago on WFMT (http://www.wfmt.com/). These stations are enabled by Real Audio technology produced by Progressive Networks.

People listen to the radio specifically to hear music, assuming of course they have chosen a music station. However, listening to music is very often a background, passive affair. Unlike TV which demands your attention and your immobility, listening to music allows you to do other things. For instance, in the office environment people listen to music while working on their PCs.
The BBC and RTE offer certain programmes in Real Audio Format, which can be listened to at any time. It seems to be only a matter of time before people can choose to listen to their radio programmes at any time of the day.

5. The application

This application extends the idea of listening to music tracks one at a time on a CD retail site to that of a personalised play-list or juke box. This takes the ideas already implemented on on-line CD retail sites where selections from recommended CDs are made available, and crosses it with the ideas behind personalised TV and Radio stations.

The site would provide a personalised play-list with one song playing after another. The play-list recommendation is made using content-based methods such as CBR to find a play-list that is acceptable. Collaborative Filtering techniques then make recommendations based on the play-lists other listeners have chosen. The listener can choose any of the songs on their play-list and the applet will take them to a secure transaction site where they can purchase the full CD.

This method of marketing is indirect. The site provides a service where listeners configure their play-lists and have music to their desk top all day long, much like a radio station. Recommendations are made based on the listening pattern of other listeners.

Taking the idea of personalised radio further, the user will be able to choose from a list of regional news services that can bring them news on the hour or several times a day.

5.1 Technology

The implementation we have described would use a Java Client to play Real Audio Files which have become the standard for streaming media on the Internet. Real Audio Files can be played from Java (http://www.real.com/devzone/tools/demos/java/rajava.html)

The Server side CBR and Collaborative Filtering system are also implemented with Java servlets.

This research wishes to extend the ACF method of finding similar user groups, by looking at new ways of clustering users according to other concepts.

6. What are the user requirements ?

The user of this system is *casual* (Loeb 1992). She is much less inclined to engage in lengthy interactions with the system in order to articulate current information needs and provide explicit feedback.

The user should be able to quickly locate and configure a play-list of music. The music chosen should be appealing to the listener and new music recommended should appeal to her based on the profiles of listeners like her. The play-list is not completely configurable – the user is allowed swap out a number of tracks.
New tracks will have to be introduced into the system on a non random basis, and offered as part of suitable play-lists (A Folk fan might be very disconcerted by the new *Prodigy* single).

The client interface should provide several options and facilities:

- Continuous auto-play: once the user is happy with her profile, the system lines up one play-list after another. The ACF/Clustering engine finds a new play-list before the old one has finished.
- Profile Reconfiguration: the user may, at any time, intervene to reject aspects of the submitted play-list.
- Play-list adjustment: the user can modify a play-list using derived attributes such as *more alternative, more traditional, jazzier etc*
- Call a play-list by mood*: Sad, Romantic*
- Find out more about the artist.
- Buy the CD online.
- Feedback: the listener may provide feedback on an individual play-list and/or its component tracks.

Two types of feedback are available: explicit or implicit

Explicit:

- During configuration – refusing a particular artist or track.
- During playback – indication of like/dislike.

Implicit:

- not rejecting or indicating dislike of new material offered or allowing new material to be played a second time.
- Playing CD tracks over and over again. This feature is part of the extended application detailed in section 8.4.

The Recommendation system:

- ACF/clustering techniques to recommend new play-list
- CBR to find specific or once off items
- Filter that keeps track of user frequency and the frequency of individual tracks – buffering of incoming items (It may not be a good idea to have the same tracks turning up every time a user logs in)
- Should be able to make connections between frequently rejected tracks in one genre or for one artist

7. Architecture

The architecture we proposed for this has its roots in the work our research group has undertaken on distributed CBR (Hayes et al. 1998). We were looking at suitable applications for the distributed Java engine which we had developed (Doyle 1999). The basis of this idea is that the retrieval process is distributed between the client and the server. At a certain point cases are passed to the client where retrieval continues locally. The point at which this transfer occurs is dependent upon the current network latency, the size of the case base and the number of features yet to be asked. We considered this type of application to be suited to this procedure because of the specific adjustments to their play-list a user may make. These adjustments which may entail rejecting individual tracks or artists in a play-list is analogous to the negotiation stage in an e commerce transaction described by Wilke. Furthermore since the case solutions are in fact a list of URLs, streaming can begin immediately with those tracks that are acceptable, while the user fine-tunes her profile.

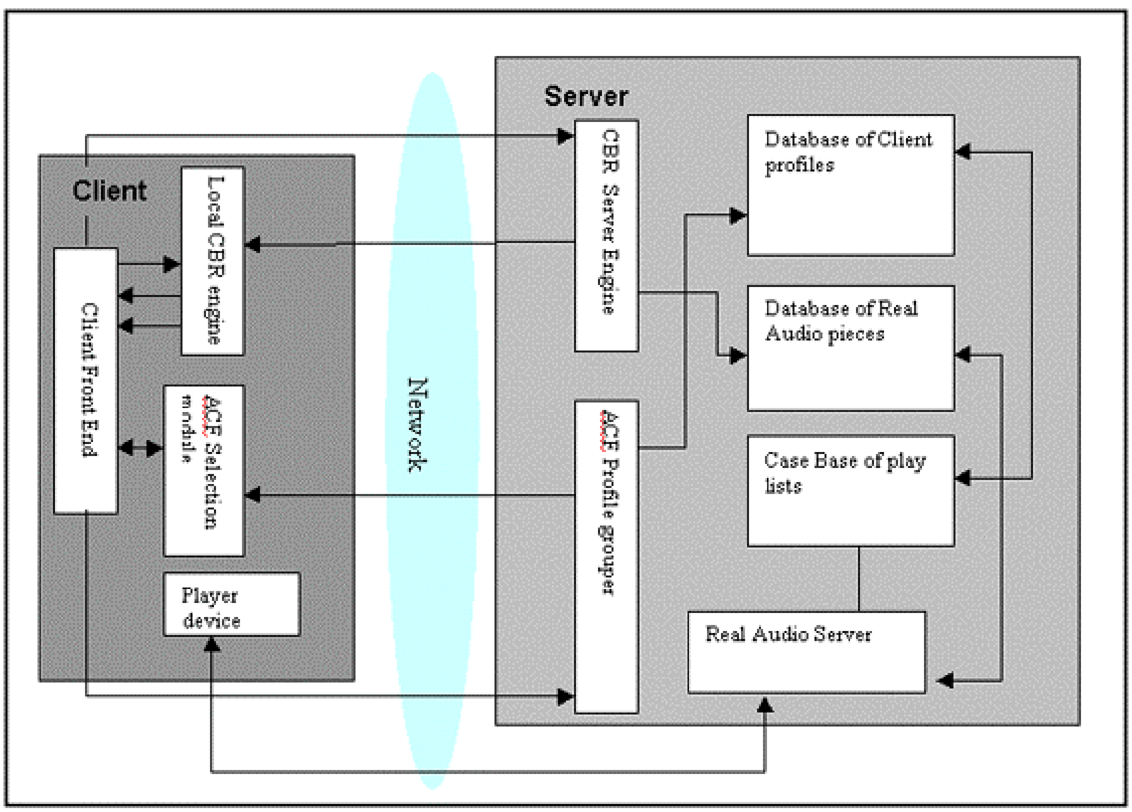

figure 1. CBR radio architecture proposal, version 2.0

Initially, our architecture design had distributed CBR components as described and a back end ACF recommendation system. Users would answer a series of refining questions after which a number of play-lists would be passed to the client and the refinement process proceeds locally. The agreed play-list is passed to the player device, and to the collaborative filtering device. The latter finds a neighbourhood of users similar to the current listener and configures a new play-list which is stored as part of the listener’s profile.

This is a dynamic domain which will require insight form an information filtering viewpoint as well as that from a collaborative recommendation viewpoint. Our research into collaborative recommendation techniques leads us to believe that the dynamic feedback an online community of listeners would stretch our understanding of profiling, and expose the limitations of current collaborative techniques. It is for this reason that we are researching incremental clustering methods.

We have chosen the idea of clustering tracks into play-lists of approximately 10 tracks. At this stage we are working under the metaphor of personalised radio, in which we want to help the user find "programme" of music. This is informed by the practicality of not having the user exactly chose the tracks he wants to hear. This is a recommendation system after all. Further to this, play-lists are not instances of the typical genre types assigned to music. We see play-lists evolving to reflect user genres.

Play-lists are fluid collections of tracks that evolve over time, reflecting the listening tastes of user groups. They are indexed in hierarchical fashion with classes containing both classes and sub-classes. This will allow play-lists to evolve over time and allow play-lists be returned for partial query matches. Our research foresees play-lists being constructed as part of an incremental clustering process similar to that described in (Lebowitz 1987).

CBR Retrieval takes place initially. Once a profile has been established, new play-lists are recommended using collaborative filtering/clustering techniques where neighbourhoods of similar play-lists are inferred.

The architecture in figure 1 s a second draft which has distributed CBR and ACF components operating in parallel. Distributed ACF system occurs when the server side ACF profile uses a matching algorithm to identify a number of profiles matching the current user. These profiles are passed to the client-side ACF selection module, where the most suitable play-list is configured with the help of the user. The user can reject selections made locally and the client can recommend other tracks by loosening the constraints on the selection algorithm.

7.1 Distributed Retrieval

The distributed retrieval ideas presented here have some similarity to Shimazu’s (1998) textual CBR system. As we have observed (Hayes 1998), this system, in which URLs are passed to the client after a MAC/FAC (Gentner 1991) retrieval process, could be adapted for retrieval of richer multimedia content. In our implementation several cases (consisting of differing play-lists) are brought to the client end and configured locally. Once the listener is happy with the play-list, the audio files indexed in the play-list case are requested from the client

end and gradually streamed. That the ’solution’ in this case involves downloading a lot of data is unimportant since the streaming can be used immediately as it arrives on the client desktop.
Our client side configuration is very similar to the final part of the negotiation process described by (Wilke 1998) inb the purchase of configurable products. Once the product has been chosen, there remains a final stage of configuring it specifically to the user’s requirements using a fixed set of components.

Similarly, once our recommendation algorithm determines the most suitable play-lists, it passes these to the client where recommendations are then made to the user. By sending the profiles to the client side the client application can generate a series of discriminating questions that help select the most suitable play-list for the user. In another case the user should be able top reject the selections locally and the client can recommend other programmes by loosening the constraints on the selection algorithm
To reiterate, carrying out a configuration stage client side makes sense for the following reasons.

1. Configuration might otherwise entail several connections to the server which could prove frustrating.
2. Standby track URLs in the neighbourhood of the user’s tastes are available locally should the retrieval server prove too busy at any time.
3. The server knows which tracks addresses are held locally and need not re-send them.

7.2 Filtering

User feedback be can sent to the filtering agent implicitly or explicitly, either by choosing not to hear a song from a play-list (or by replaying a favourite tune), by indicating a dislike of an artist or track, or by rejecting a complete play-list.

This raises the issue of dynamic profiling. The user profiles (in terms of what tracks they have chosen to hear) may change from hour to hour. The question arises as to whether recommendations should be based on user profiles recorded as a snap-shot in time or whether each user’s listening history should be generalised and used as a basis for recommendation. We would argue the latter. An interactive radio system will have to have a sense of how frequently the user uses the system and the frequency at which tracks have already been played. This would create an extra filter or buffer element in the user profile as described by Loeb (1992).

The system should take the decision to play the same tracks intermittently, as would a radio station. However, a filtering system would have to be put in place that would not allow the same material be played too regularly. In this case the filtering criteria described in Lyric Time would apply (Loeb 1992), where frequency of played items depends on the frequency of items played in the past. This contrasts the PTV system of (Smyth et al. 1998) which makes recommendations based on profiles of other users, but does not have a understanding of the watching history of the viewer.

Finally, an ACF recommendation system depend upon the integrity of its users. ACF based systems are open to abuse since users may repeatedly bias their preferences in order to bring a product to the attention of other users. One such case is the un-moderated balloting mechanism available to voters in the Random house readers poll of the 100 greatest novels of the century. Due to multiple voting, The zealous advocates of Ayn Rand and L. Ron Hubbard have voted their authors into seven of the top ten positions[1].
A possible solution to this problem is the concept of the **trusted *recommendee* -** someone whose own profile consistently provides a basis for good recommendations to other users with similar profiles. Over time each user profile has associated with it a number of trusted profiles, that is, profiles from whom recommendations have been generated and have received a positive response.

Until we have decided on the key relationships between the clustered play-lists, user profiles and individual tracks we can not be more definite about the system architecture.

8. Recommendation Methodology research ideas

Section 2 of the paper introduced issues involved in using content and non-content based recommendation systems. We presented these ideas in terms of a pure CBR retrieval system and a pure ACF recommendation system. The strengths and weaknesses of the respective systems were highlighted, and it was evident that each system could potentially augment the weakness of the other.
In this section we present the evolution of our ideas on a recommendation system that uses CBR, ACF and conceptual clustering. To do this we present the system in terms of this dialectic of content and non-content based retrieval. Furthermore, we examine the elements involved in the recommendation system.

8.1 Non-content based recommendation system

In this implementation a purely ACF recommendation system recommends a series of individual tracks which are then played as streamed audio tracks.
Using typical ACF similarity measures, neighbourhoods of users are inferred from which new tracks are then recommended to its users. However, the nature of the radio domain we have specified raises some problems.
Recommending single tracks means that the listener has to regularly configure the system. We have specified that the system should be able to produce programme length "themes" of music. For this reason, we want to place some constraints on the configuration process. It does not makes sense to allow the listener choose music on a track by track basis. Ideally, the music played should be a mixture of known material and new material in a vein acceptable to the listener. Several tracks should be presented as part of the ACF process, some of which can be rejected and replaced by others.

[1] http://www.randomhouse.com/modernlibrary/100best/novels.html

If the similarity threshold metric is too high, user neighbourhoods are small, and the system may not be able to produce enough new recommendations at the one time. On the other hand if neighbourhood granularity is coarse, poor recommendations may be made.

In order to fuel ACF recommendation feedback on items external to those contained in the system has to be sought. Our original plan was to shoe-horn suitable new tracks. Certainly new tracks will periodically have to be introduced into the track database. It would be preferable if these were introduced on a non random basis, into user groups with which they have a high prospect of success. Section 8.3 will introduce clustering techniques that may prove suitable for this purpose. Section 8.4 will describe a modification to the system that overcomes the problems of finding external recommendations, and provides accurate feedback of the listening patterns of current users.

The User profiles in the ACF scenario will contain a list of likes and dislikes, as well as the frequency metrics described in section 7.2.

Track 406	
Artist	Beth Orton
Title	Don’t need a Reason
CD	Trailer Park
Genre	Folk, Alternative
Year	1997
label	Polygram

Play-list 104	Play-list 23	Play-list 45
Track 8	Track 42	Track 46
Track 406	Track 406	Track 56
Track 96	Track 916	Track 116
Track 408	Track 8	Track 8
Track 456	Track 56	Track 406
Track 365	Track 375	Track 35
Track 42	Track 12	Track 42

figure 2. Example of Cases used in the Radio CBR system

8.2 Content based recommendation system

In this scenario each track is described by a number of features which are freely available (figure 2). However, there are a limited number of features which we can retrieve on, some of which are less important than others, or may simply be dependent on the user. An obvious feature to use is the pre-assigned genre type. However, *genre* is too rigid a classification, it imposes a classification upon tracks that in an ACF system would be decided by neighbourhood consensus. It may be useful as a constraint - folk fans shouldn't get tracks marked as heavy rock. The *Artist* feature is again too rigid a classification. Listeners might

like one track by an artist, but hate others. It would be much more useful to infer similarity between artists. This would require knowledge engineering, specific user input or collaborative filtering techniques.
For instance, similarity tables could be drawn up to represent levels of closeness between genres or between artists. The metrics in these table would be subjective, unless a statistical analysis of likes and dislikes could be undertaken over a user population to find approximate similarity weights. Once again, this work could be done using clustering or collaborative filtering techniques. Such techniques might also produce multi-feature similarity descriptions.

With the content based model there is really not enough predictive features to perform accurate retrieval, and bad recommendations are likely unless retrieval can be correlated with user feedback.

8.3 Non-content based recommendations using play-lists

Section 8.1 described the problems using individual tracks as the basis of recommendations. Since the object we are attempting to provide is a programme of music composed of individual music items, track by track recommendation entails too great a configuration overhead. It also may prove an unsuitable method for establishing a memory system, from which fast recommendations can be made.
In the following scenario as before, recommendations are made from neighbourhoods of similar users. However users are clustered according to their likes and dislikes of individual groups of tracks called play-lists. A play-list is not a unit item like track in the previous sections. It is composed of a number of tracks, each having a list of descriptive features. Play-lists need to be mutable entities whose make-up can be reconfigured through clustering techniques or manually by the user.
In the case of manual reconfiguration, the new play-list my be viewed as a specialisation of the original play-list, and the ACF user feedback algorithm will record a tapered positive response to both the newly configured play-list and the parent. Should this sub-class become the preferred configuration for several users, it becomes a parent class of its own.

The play-lists themselves are built manually by the developer at first, in order to bootstrap the system. However, we see new play-lists being arrived at by a process of incremental conceptual formation where tracks are clustered in a hierarchical fashion. From these hierarchical concepts we see new play-lists evolving.

8.4 Using the system CD player to seed ACF profiles
In section 2.2 and 8.1 we have mentioned the necessity for users of an ACF recommendation service to input new recommendations based on items encountered external to the system. These new items fuel a dynamic recommendation system, which would otherwise reach a state of stagnation. One issue for any ACF based system is that users may prefer to be passive consumers of recommendations rather than contributors. We have devised a novel way to elicit feedback on listening preferences using the system CD player. This model makes the reasonable presumption that a potential user of an online music system will also listen to music on their CD player. It also makes use of the fact that each CD title can be uniquely identified with a 8-digit hexadecimal number,

computed using data from the CD's Table-of-Contents (TOC) in MSF (Minute Second Frame) form.
Using the online facility of the CD database, we can build a listening profile of a users, that will provide regular feedback for ACF recommendations.

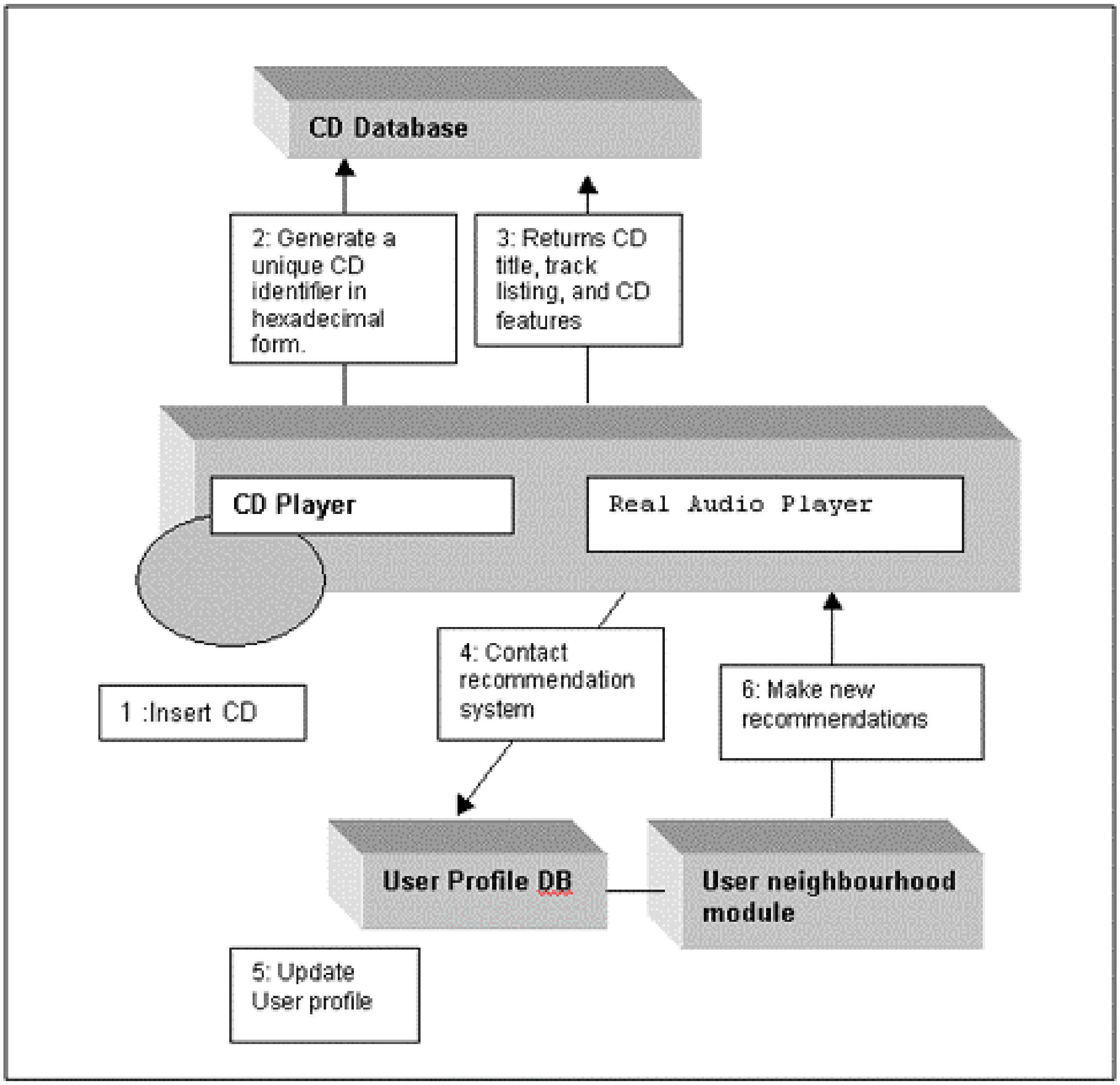

figure 3. Simple model for generating recommendations based on information extracted from user CD listening habits

9. Conclusions and further research issues

We have presented the strengths and weaknesses of content based and collaborative recommendation systems. Such recommendation systems apply the concept of predictive utility: a measure of value in having predictions for an item before deciding whether to invest time in consuming that item. We have presented an architecture for a smart radio system which uses a hybrid of CBR and ACF to recommend a selection of music called a play-list. Play-lists are mutable entities, whose composition we see being determined by techniques of conceptual clustering. The paper sets the scene for a sizeable period of research into smart recommendation, dynamic profiling, information filtering and the feasibility of using the unsupervised learning techniques of conceptual clustering.

10. References

Balabanovic, Marko; Shoham Yoav; (1997) Fab: Content-Based, Collaborative Recommendation. Communications of the ACM, March 1997, Volume 40, Number 3

Gentner, D., and Forbus, K. D. 1991. MAC/FAC: A model of similaritybased access and mapping. In Proceedings of the Thirteenth Annual Conference of the Cognitive Science Society. Northvale, NJ: Erlbaum

Hayes, C. Cunningham, P. (1999) Shaping a CBR view with XML - to be included in Case-Based Reasoning Research and Applications. Third International Conference on Case-Based Reasoning, ICCBR-99, Monastery Seeon, Munich, Germany, July 1999, Proceedings

Hayes, C. Doyle, M. Cunningham, M. (1998) CBR Net :- Smart Technology over a Network. TCD Technical report TCD-CS-1998-07

Hickie, J. Cunningham, P. Neural Networks for Collaborative Filtering: A Case Study in Web Advertising.

Konstan J.A., Miller B.N, Maltz D., Herlocker J.L., Gordon L.R., Reidl J.; (1997) Grouplens: Applying Collaborative Filtering to Usenet News. Communications of the ACM, March 1997-Volume 40, Number 3

Loeb, Shoshana (1992) Architecting Personalized Delivery of Multimedia Information. Communications of the ACM/December 1992/Vol.35. No12

Shardanand, U. Maes, P. Social Information Filtering: Algorithms for Automating “Word of Mouth”

Shimazu, H. (1998) A Textual Case-Based Reasoning System Using XML on the World-Wide Web. Advances in Case-Based Reasoning, Proceedings from the 4th European Workshop on Case-Based Reasoning, Dublin, Ireland, September 1998. Springer Verlag Lecture Notes in Artificial Intelligence 1488.

Smyth, B., Cotter, P., and O'Hare, G. M. P. (1998) Let's Get Personal: Personalised TV Listings on the Web. 9th Irish Conference on Artificial Intelligence and Cognitive Science. Dublin, Ireland.

Wilke, W. Lenz, M. Wess, S. (1998) Intelligent Sales Support with CBR in Case-Based Reasoning Technology. Lenz et al. Springer-Verlag Lecture Notes in Artificial Intelligence 1400(1998)