Smart Radio - Building Music Radio On the Fly

Conor Hayes & Pádraig Cunningham
Computer Science Department
Trinity College Dublin
conor.hayes@cs.tcd.ie
www.cs.tcd.ie

Abstract. This paper describes the development of a networked music application at Trinity College Dublin. Smart Radio is a web based client-server application which uses streaming audio technology and collaborative recommendation techniques to allow users build, manage and share music programmes. While it is generally acknowledged that music distribution over the web will dramatically change how the music industry operates, there are few prototypes available to demonstrate how this could work in an managed way. The Smart Radio approach is to have people manage their music resources by putting together personalised music programmes. These programmes can then be swapped using techniques of collaborative recommendation to find similarities between users. The smart radio system currently runs within the Computer Science Intranet with permission from the Irish Music Rights Organisation (IMRO). It is a prototype system for an "always on" high bandwidth Internet connection such as ADSL.

1. Introduction

The availability of high quality digitally compressed music on the Internet has caused waves in the music distribution industry. The sale of music has been a highly centralized activity involving the record companies who record artists, distributors who ship music media items such as CDs, and retails outlet who make music artefacts available to the consumer. CD prices are generally fixed across geographic borders (government taxes and shipping costs differentiate one region from another). Up to recently the basic purchase unit of music has been a collection of tracks by a single artist or band. Though compilation albums of mixed artists are available, their number are generally low compared to single artist items. However, the music industry does not offer the facility to purchase in-store personally built compilations even though this activity is commonly carried out by consumers using blank tapes and, of late, recordable CDs. A common off-shoot of this activity is that home-made compilations are swapped or given as presents despite the fact that this contravenes the artist's copyright. With the advent of MP3 compression techniques and the ease with which people can now send data over the Internet, near CD quality digital tracks are suddenly easily available for millions of people to download and store locally. The piracy aspect of this activity is greater than 'home taping' in that an unregulated distribution channel (The Internet) had been put in place. The music industry has little agreement on issues of digital authentication, or on how to decentralise their sales operations to take into account the new paradigm of personalised delivery over a network. Furthermore, there is continuing debate on the payment system suited to music download.

The Smart Radio Project was initiated to address these issues. It does so by assimilating two paradigms:

- The regulated distribution of music on the web
- The service of personalised radio programming

In the section 2 of this paper we will describe our motivation for using streaming music technology as opposed to downloadable MP3s. Section 3 will address how we use collaborative recommendation techniques to provide in effect a personalised radio service to users. We argue that the most useful unit of recommendation is a music programme, a user built compilation. We show how user feedback occurs at two levels of granularity - at the track level and at programme level. Further to this we discuss how the Smart Radio recommendation system encourages community participation by allowing users know who their most consistent neighbours are. In this way, a recommendation system encourages rather than replaces social processes [1].

Sections 4 and 5 describe the system architecture and operation. Sections 6 and 7 discuss proposed payments systems, and the potential for a Smart Radio system in conjunction with higher bandwidth technologies such as Asymmetric Digital Subscriber Lines (ADSL).

2. Music Formats

Smart Radio uses streaming audio technology developed by Real Networks[1]. The architecture is not tied to this technology but at the time of development Real Networks provided the most popular streaming client player.

Streaming technology was chosen (as opposed to a complete download model) for a number of reasons. The service paradigm that we chose to examine was that of personalised radio. Since music radio is well understood concept, by tying together the concept of a music programme with a personalised recommendation system, we could provide an intuitive stream of music service. For example, if the basic unit of recommendation is an mp3 file, the time involved in downloading several of these to a hard drive and then setting up a play-list is quite considerable. With the streaming technology used by Smart Radio, a user can login, choose a recommended programme or a favourite programme, and have a music programme stream to the desktop almost immediately. As well, as the benefits of immediate delivery, we considered streaming technology to be a useful solution to the piracy

[1] http://www.realnetworks.com

issue. The streaming audio files are not saved to the user's hard drive, or posted on the internet for illicit distribution. Another benefit of a service where there are no downloadable files is that the user has a roaming profile and his music programmes and recommendations are available wherever he logs on. Though disk space has become cheaper, there is considerable management involved in holding a large digital library on a local machine. If each file has been paid for then back-ups need to be made in case of disk failure.

3. Collaborative Recommendation

Collaborative Recommendation techniques have proven a attractive methodologies for making recommendations where there are few or unhelpful features by which to categorize assets. Some of the earliest work in this area was done in the area of music recommendation [2]. The techniques have come into their own with the growth of e-commerce stores such as Amazon.com. The basic collaborative recommendation model supposes that a person's rating of a new resource can be predicted by looking at how other similar users have rated the same resource.

The system makes a correlation between a user and other users whose profile are similar according to a similarity metric. From this correlation it then calculates items that the group have in common, which are missing from the active user profile. Smart Radio uses a measure of co-linearity as the correlation between the active user and the user population. This measure, known as the Pearson R coefficient, has been reviewed and found to be a satisfactory indicator of user similarity [3].

The technique used in Smart radio is to choose the top *n* neighbours for each user and then use these neighbours as the source of recommendations. For items in the neighbourhood that the active user has not yet rated predictions are made using a weighted average of deviations from each neighbour's mean vote:

$$p_{a,i} = \bar{r}_a + \frac{\sum_{u=1}^{n}(r_{u,i} - \bar{r}_u) * w_{a,u}}{\sum_{u=1}^{n} w_{a,u}}$$

Equation 1: Predicting user a's rating for item i

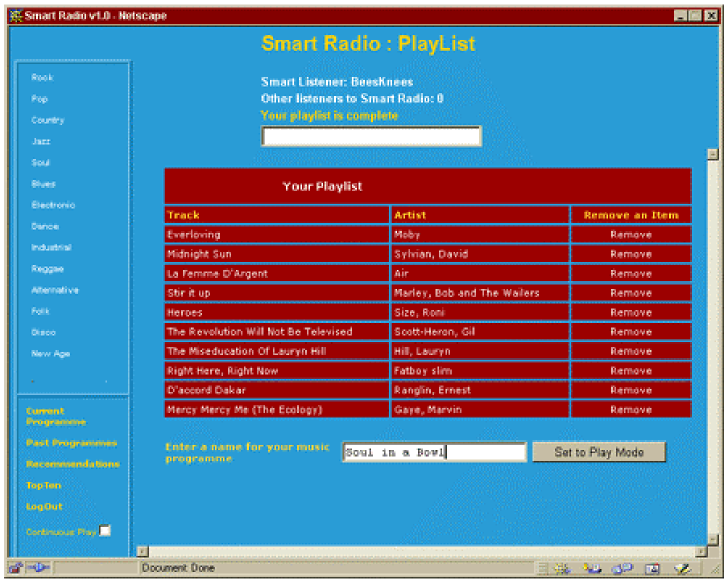
pa,i is the prediction for user a for item i. $r_{u,i}$ refers to user u's rating for item i. $\bar{r}_a$, $\bar{r}_u$ refers to the rating average calculated for user *a* and *u* respectively. *n* is the number of neighbours and $w_{a,u}$ is the weight between user *a* and neighbour u .

This weight is given by the pearson correlation coefficient between user *a* and user *u:*

$$w_{a,u} = \frac{\sum_{i=1}^{m} r_{a,i} * r_{u,i} - \frac{\sum_{i=1}^{m} r_{a,i} \sum_{i=1}^{m} r_{u,i}}{m}}{\sqrt{\left(\sum_{i=1}^{m} (r_{a,i}{}^{2} - \frac{(\sum_{i=1}^{m} r_{a,i})^{2}}{m})\right)\left(\sum_{i=1}^{m} (r_{u,i}{}^{2} - \frac{(\sum_{i=1}^{n} r_{u,i})^{2}}{m})\right)}}$$

Equation 2: Pearson correlation coefficient

In equation 2, *m* refers to the number of items the two users have in common. In order to ensure that correlations are not being calculated over a small number of common items a further weight is applied. With our current user population we found it was necessary to have rated 20 items in common before good recommendations were being made. Therefore, if a pair of users has less than 20 items in common the correlation obtained by the Pearson measure is devalued by m/20.



Smart Radio : PlayList

Smart Listener: BeesKnees
Other listeners to Smart Radio: 0
Your playlist is complete

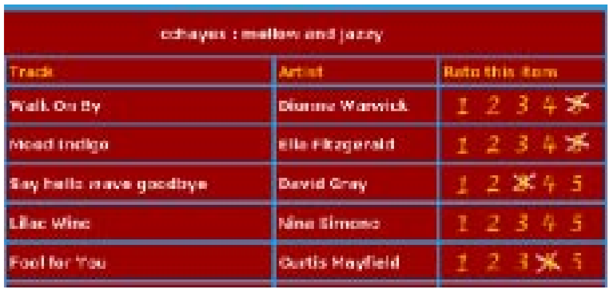
Your Playlist

Track	Artist	Remove an Item
Everloving	Moby	Remove
Midnight Sun	Sylvian, David	Remove
La Femme D'Argent	Air	Remove
Stir it up	Marley, Bob and The Wailers	Remove
Heroes	Size, Roni	Remove
The Revolution Will Not Be Televised	Scott-Heron, Gil	Remove
The Miseducation Of Lauryn Hill	Hill, Lauryn	Remove
Right Here, Right Now	Fatboy slim	Remove
D'accord Dakar	Ranglin, Ernest	Remove
Mercy Mercy Me (The Ecology)	Gaye, Marvin	Remove

Enter a name for your music programme: Soul in a Bowl

Figure 1: Naming a recently built play list

User ratings are gathered using explicit and implicit user feedback. Users can explicitly choose to rate individual track items or individual programmes on a scale of 1 – 5, where 5 is the top score. Implicit feedback is important for the less interactive or casual user [4]. The reasoning behind this is that the casual user wants to interact as little as possible with the system and will intervene only when the system gets things badly wrong. In terms of implicit feedback the system allocates a score of 4 to recommended programmes and constituent items that have been saved to a user's profile, or to programmes that have been built from scratch by the user. Programmes can be put together in a matter of seconds by simple clicking on the desired music item and adding it to the current play list (See figure 1). Music items are indexed currently by genre, which is not satisfactory since the genre feature is not finely grained enough to capture the nuances within music types. Further research is planned on how to allow user indexing of music assets. Once a programme has been built it can be played immediately and is automatically saved to the users profile for future retrieval. Programmes that are played more than three times are awarded the top score of 5, even though the average rating of constituent items may be lower. Our theory is that a well chosen collection of music has greater value than the sum of its constituent items. For one thing, there is some work involved in putting together a programme so there is some value in choosing something "off the shelf". For another, a collection of music may contain the difficult to quantify feature of "mood" which depends on the collected items being played together. This feature is apparent where users amend their ratings for individual items as they appear in different programmes. Figure 2 illustrates an excerpt for the programme *mellow and jazzy* in which the user *cchayes* has rated four out of the five shown items. If cchayes chooses *mellow and jazzy* again he will be shown his ratings for the individual items within the programme and he may recast his vote. This facility is important because music taste does shift, and user profiles will have to move to reflect this. It is entirely probable that a user will cease to become a recommender in one neighbourhood only to have moved to another.



cchayes : mellow and jazzy

Track	Artist	Rate this item
Walk On By	Dionne Warwick	1 2 3 4 [✓ 5]
Mood Indigo	Ella Fitzgerald	1 2 3 4 [✓ 5]
Say hello wave goodbye	David Gray	1 2 [✓ 3] 4 5
Lilac Wine	Nina Simone	1 2 3 4 5
Fool for You	Curtis Mayfield	1 2 3 [✓ 4] 5

Figure 2: a portion of play list entitled *mellow and jazzy*

4. Smart Radio Architecture

Smart Radio is a web based client server application. Users must login through their browser, after which they may choose to build new programmes from scratch, play past favourite programmes, or choose programmes recommended by their neighbours. Server side programming was done with Java servlets, Java server pages and Java beans running on the servlet engine in Sun Microsystem's Java Web Server. Servlets provided an elegant *state* management facility for Smart Radio. Each user logs in and is provided with a virtual session on the Smart Radio server. Since http is a stateless protocol, Java servlets allow unique session Ids be set as cookies in the users browser. These ids are returned to the server with every http request. The session id can be validated and the resources associated with each valid id retrieved from the servlet engine's hash table. In the case of Smart Radio, each online user has a neighbourhood object and a recommendation object associated with his active profile.

This is a dynamic domain where users may seek new recommendations almost immediately once the current programme has played through. The first version of Smart Radio calculated new recommendations as they were requested by online users. A user-correlation matrix was held in memory and updated as each user required fresh recommendations. However, this method was not considered computationally or architecturally scalable. The current implementation separates the web server processes from the recommendation engine, allowing both to be scaled as necessary. A scheduler recalculates the neighbourhood for each *online* user on the half hour, and updates the database with the most current recommendations. The web server then queries the updated database for new recommendations. Currently the web server, the scheduler and the recommendation engine reside on a single machine. However, since the system is implemented in Java these modules can be easily moved to remote machines and would communicate using Remote Method Invocation.

The current database being used by Smart Radio is Microsoft Access which has proven adequate to the traffic generated within the Intranet. We will have to examine a scalable database solution if this work is to move to the Internet. Database access is handled by java database manager running on the server. The manager creates a pool of connections from which it allocates connections in a round robin fashion as requests are received from components of the system. If there are no connections available, the request is put on a queue, and assigned the first released connection.

A Real Audio Server runs in parallel with the Java Web Server. This server receives requests from the clients and streams audio pieces to the specified address. The Smart Radio Web pages are able to play the incoming stream by means of the Real Audio plug-in embedded in a HTML frame which remains constantly in place irrespective of what other pages the user requests from the Smart Radio server. In fact it is this plug-in that requests the file containing the addresses of constituent pieces of the programme ordered by the user.

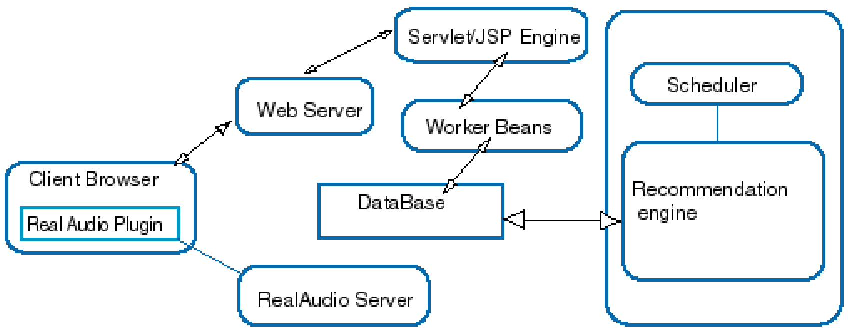


Figure 3. The architecture of the Smart Radio system

## 5.	Receiving a programme of music

There are two stages to receiving a programme of music. The user can choose to either configure a new programme, choose a recommended programme or a past favourite. When the user indicates that the programme is ready to play (he may want to edit it before setting it to play) the browser passes the session id to the plug-in, which requests the programme associated with this session id. The server has already configured a long XML string containing the URLs of the new programmes and stored it a hash table with the session Id as key. The XML file is written according to the format of the synchronized multimedia integration language, SMIL. The server receives the request and returns the SMIL string to the plug-in which begins buffering the first music item. It is important to note that programme files are created dynamically, and do not reside on the server. The database contains the URL for each music item and SMIL servlet builds the programme on the fly. Once the SMIL file is received, a hidden java applet monitors the activity of the plug-in and using browser scripting allows the user to see the status of the music stream (i.e. whether it is buffering, which track is playing and any connection errors).

6. Payment systems and future Viability

In Ireland, terrestrial radio is license-free for the user, though some of the television license funds may supplement radio programming. In order to comply with copyright law, radio stations have to pay a royalty fee determined by the Irish Music Rights Organisation (IMRO).

The Smart Radio system may well be suited to a brokerage or middle-ware role between content vendors and listeners.

The first payment system to consider is a subscription or license model, similar to the subscriptions charged by cable companies. The subscription charge is for a service – access to streaming music as opposed to a series of audio downloads. Subscribers pay a flat fee per quarter. The brokerage firm then pays the content providers for their services. This would either be a flat fee or in proportion to the content they provide. Revenue would accrue to the Smart Broker from a proportion of the subscription fee and the direct marketing service being offered to music vendors, such as CD Now, for instance.

The second scenario involves charging per unit content received by the client. The most useful method here would be a micro-payment[2].

The client's relationship is with the Smart Radio broker. He may pay in advance into a personalised account. Micro-charges are extracted from his account per programme. An SSL connection will have to be set up to credit his account initially, but also *every time* a play-list is sent to his desk top since the client will have to be verified before charges are deducted from his account. This is difficult to scale since setting up an SSL connection has a computational over head. Certainly, a play-list is the smallest unit for which a user should be charged in a real-time environment

Of the two schemes, a periodic subscription charge may be the most suitable format since users may be hostile to the idea of paying for a single listen to an item rather than having it to download.

7. Internet Viability

Streaming Audio technology operates using TCP to establish a connection and the unreliable datagram service UDP to stream packets to the client. The unreliability of the UDP format means that packets may not arrive in time for playback due to network congestion. The problem is exacerbated by the transmission rates required to stream near CD-quality audio (even while highly compressed), approximately 128kbs. The Smart Radio system running on the computer science departmental intranet uses audio files encoded at bit rates between 44kbs and 128 kbs and occasionally encounters problems when traffic is high. On PC speakers the drop in quality caused by using a 44Kbs codec is not particularly noticeable.

[2] http://www.ipin.com

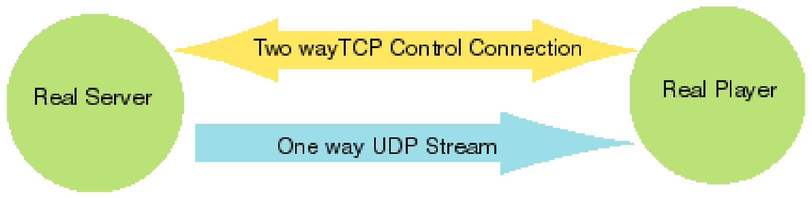


Figure 3. Streaming audio connections

However, even at this codec the transmission rate required would require most of the bandwidth available on good local telephone loop using a 56kbs analog modem. The system could run as an AM quality service on the local loop, but really would not be a viable service until internet dialup charges are free. It is for this reason we propose Smart Radio as the type of service suited to Asymmetric Digital Subscriber Lines (ADSL), a technology which is likely to change domestic use of the Internet dramatically. ADSL provides high bandwidth access to the Internet (up to 8Mbs) over existing local loop wiring. Since it uses frequency bands outside of that used for telephony (<4khz), ADSL does not have a dialup mechanism. It provides, in effect, an "always on ", dedicated line to the Internet for anyone with a telephone connection. It should be clear that these sort of conditions would prove ideal for an application such as Smart Radio.

References

[1] Hill, W., L., Rosenstein, M., and Furmas, G. (1995) Recommending and Evaluating Choices in a Virtual Community of Use. In proceedings of the conference on human Factors in Computing Systems (CHI95), 194-201, Denver, CO, ACM Press

[2] Shardanand, U., Maes, P. (1995) Social Information Filtering: Algorithms for Automating "Word of Mouth", In Proceedings of the Conference on Human Factors in Computing Systems (CHI95), 210-217, Denver, CO, ACM Press

[3] Billsus, D., Pazzani, M.J. (1998) Learning Collaborative Information Filters, in *Proceedings of the International Conference on Machine Learning. (ICML 98)* Morgan Kaufmann Publishers. Madison, Wisc.

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

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

[6] 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

[7] 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

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

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

[10] 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.

[11] 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)