Using User Models in Music Information Retrieval Systems

Wei Chai Barry Vercoe
MIT Media Lab
{chaiwei, bv}@media.mit.edu

ABSTRACT
To make multimedia data easily retrieved, we use metadata to describe the information, so that search engines or other information filter tools can effectively and efficiently locate and retrieve the multimedia content. Since many features of multimedia content are perceptual and user-dependent, user modeling is also necessary for multimedia information retrieval systems, e.g., music information retrieval systems. Furthermore, to make the user models sharable, we need standardized language to describe them. In this paper, an XML-like language is proposed to describe the user model for music information retrieval purposes. We also propose some paradigms to acquire, deploy and share the user information to improve current music information systems. A prototype system, MusicCat, is analyzed and implemented as a case.

KEYWORDS
User modeling, music information retrieval, XML, MPEG7

1. INTRODUCTION

The Internet has become one of the main sources of multimedia content for entertainment, education, business, or other purposes. Effective and efficient methods to retrieve data from Internet are desired, and music is one of the important media falling in this situation.

There are many websites for music sale, advertisement and sharing. However, the interfaces for users to find things are not convenient, since only category-based browsing and/or text-based searching are supported on these sites. There has been some research to improve the interface. For pull applications, content-based search techniques are studied and some prototype query-by-humming systems have been developed. Users can input the main melody of the music via web browser by uploading pre-recorded humming [3][4][5], or typing the string to represent the melody contour [6][7]. There are also some systems [8][9], in which users can search the sound libraries according to the acoustic features like sound effect, rhythm, etc. For push applications, collaborative filtering model is mostly used to recommend music to users. This is a technique for making personalized recommendations from any type of database to a user based on similarities between the interest profile of that user and those of other users [2]. Feature-based models [10] are studied as well to model the correlation between the user's preference and the features of music.

For content-based search or feature-based filtering systems, one important problem is to describe the music by its parameters or features, so that search engines or information filtering agents can use them to measure the similarity of the target (user's query or preference) and the candidates. Collaborative filtering model usually avoids this task by measuring only the similarity of users, not music, but the bootstrapping may sometimes be hard and take a long time.

MPEG7 [11] (formally called "Multimedia Content Description Interface") is an international standard, which describes the multimedia content data to allow universal indexing, retrieval, filtering, control, and other activities supported by rich metadata. It is obviously useful for multimedia information retrieval systems. However, the metadata about the multimedia content itself are still insufficient, because many features of multimedia content are quite perceptual and user-dependent. For example, emotional features are very important for multimedia retrieval, but it is hard to describe it by a universal model since different users may have different emotional responses to the same multimedia content. We therefore turn to user modeling techniques and representation to describe the properties of each user, so that the retrieval will be more accurate and efficient.

This paper is organized as follows. In section 2, we argue user modeling is necessary for music information retrieval systems. In section 3, we describe the main issues involved in user modeling. In section 4, we propose some paradigms, in which user modeling can be very helpful. In section 5, an XML-like user modeling language is proposed to make the user information more sharable and thus the systems more open. At the end of the paper are future work and conclusions.

2. BENEFITS OF USER MODELING FOR MUSIC INFORMATION RETRIEVAL

Using user models for music information systems has several benefits.

- A user model is necessary to specify some perceptual features.

Music features can be classified as quantitative and qualitative. Although the classification is not absolute, we use these in this paper for the sake of simplicity.

> - Quantitative features: background information (composer, artists, age, genre, etc), time signature, key signature, tonality, tempo, instruments, music structure, pitch, loudness,

melody, rhythm, all the other parameters that could be derived from the music, etc.

- Qualitative features: rating, functionality, emotional features, etc.

Quantitative features (note some of them are perceptual) are easy to describe because they depend almost entirely on the music itself and will not differ much from person to person. Thus, these features can be described as triples like <music, feature, value>.

Qualitative features (all of them are perceptual) are more difficult to describe because they depend on the users to high extent. If a user tells the search engine that he wants some "happy" music or "good" music, it is hard for the search engine to select accurately without any information about that user. Thus, these features should at least be described as 4-tuple like <music, user, feature, value>. Even more complicated, music may sound different to the same person in different contexts, for example, the user wants some "music suitable for lunchtime", in that case, these features should be described as 5-tuple like <music, user, context, feature, value>.

- A user model can be used to reduce the search space.

One problem of current music information retrieval systems is performance. Usually the computational complexity in these systems is huge, while real-time response is required in most cases. If the system knows the user well in advance, it can search only the user's interest space instead of the whole music repository. This will improve both accuracy and efficiency.

- User modeling can make push service more accurate and easy.

User modeling has been used in push services. Collaborative filtering is already well known for being able to cluster users with similar interests according to their visit history, and to recommend music to one user if other users with similar interests like it. For push service, user modeling is especially important because the system can know each user's preference and then recommend accordingly. Although collaborative filtering can do this to some extent, it needs large user volume, heavy usage of the system and long-time bootstrapping. Since there is no general procedure in which feature-based models will be necessary for new music pieces, collaborative filtering can't make timely recommendation until some users encounter them via some other ways.

- User interface issues.

Different users may prefer different user interfaces in music information retrieval systems. For input, some users may sing well and prefer query-by-humming interface, some users may play instruments well and/or have strong music background and prefer some more professional interface. Some users may be very familiar with the background information of the music they are interested in, so text-based search is sufficient. Some users may be "active" listeners - they always know exactly what music they want to listen, while others may be "passive" listeners - they prefer others, e.g. agents, selecting music for them. For output, music visualization is a useful complement for music information retrieval systems, but different users also may have different preferences on visualization of one piece.

3. MAIN ISSUES

There are several important issues in user modeling for music information retrieval purposes or even more general multimedia retrieval.

- How to model the user?

Developing the user model has been studied in HCI field for years [1]. There can be two ways: (1) Explicit - using survey, dialog or any other methods to obtain the user knowledge directly (User-programmed). (2) Implicit - observing the user's behavior (Machine-learning) or inferring from domain knowledge or other user information (Knowledge-engineered).

Since a user's interest may change over time, user modeling should be a dynamic procedure, which can timely reflect this change.

- What information is needed to describe a user for music IR purpose?

  - Indirect information. User's demographic properties such as age, sex, citizenship, education, music experience, etc. can be used in collaborative filtering model to infer the user's interest space approximately.

  - Direct information. User's interests and definition of qualitative features, either of which could be specified by examples or by other features of music. User's appreciation habit should be described as well. For example, a user needs what kind of music in a particular context.

- How to represent, use and share the user model?

User model information can be represented in text format, so that a search engine or information filtering agent can use them to refine the result easily. The problem is that collecting user's accurate information is a lengthy procedure, and users can not tolerate a long-time training whenever they switch to a new system. A much better way is where all the music information retrieval systems can share the information in some way. Similar to MPEG7 concepts, we can use a standard language, in this paper for example an XML-like

language, to describe the information so that it can be easily understood and used by the retrieval systems. The paradigms for using this information and the language are discussed in the next two sections.

4. PARADIGMS

There can be several ways to use the user model information in music information retrieval systems. Generally the user modeling task can be done on either the server side or the client side.

- Server-side user modeling

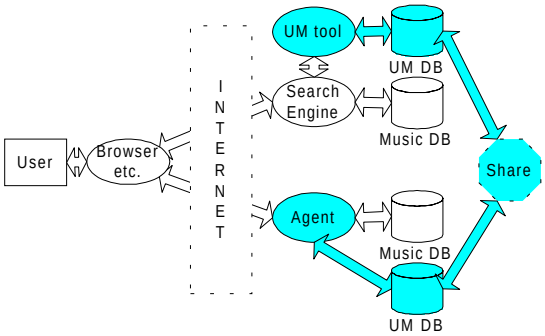

Fig.4-1 Server-side user modeling paradigm.

- Client-side user modeling

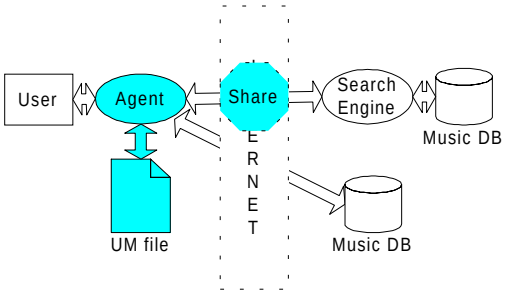

Fig.4-2 Client-side user modeling paradigm.

The agents in the above diagrams integrate the user modeling ability. Location of user modeling differs in:

- Easy/hard to obtain the user information timely.
  Client-side agent can obtain user information more easily and timely by observing user's behavior, learning from user's interaction with machine. Server-side UM tool or agent can only obtain this information from the data, which have been sent to server side.

- Easy/hard to share the user model.
  To make the user model sharable and reusable, the user model can be represented in a standard format, so that it can be interchanged and understood. In server-side modeling systems, the task is between server and server; thus it will be huge, and for each user's visit the server will spend time finding that user's model first. In client-side modeling systems, whenever a client wants to retrieve the content from a server, it can send its user model first to the server then the server will parse the standard-formatted user model, do the retrieval based on that specific user model.

- Hard/easy to use collaborative filtering model.
  It is hard to use collaborative filtering model in pure client-side user modeling systems. A centered user model repository is very helpful for cross-user analysis and general statistic purpose.

- Far from/close to the music data.
  Client-side user model is far from the music repository, while server-side user model is close to it. Making the user model and music content closer may be helpful for indexing the music database more efficiently based on the user model.

- More/less privacy or safety.
  Client-side user modeling can have more privacy than server-side one. The client-side agent can even tailor the user model according to the reputation of the server system. On the other hand, for client-side modeling, there might be potential trust problems running others' code on the user's machine.

- Higher load on servers/more scalable.
  Server-side user modeling requires much more computation task and storage on the server machine, which might not be efficient.

In practice, hybrid systems of the above two are possible.

Example: MusicCat

Most current music recommendation systems deal with recommending new music to the user based on user's visiting history, user's demographic properties and the textual features of music. One thing is neglected: the user may also need some agent to automatically select music for him from his music collection according to the user's habit and current context.

MusicCat is such an agent, able to help the user define contexts and corresponding features of music that he wants to hear in those contexts correspondingly. Besides, the user can also define qualitative features of music based on quantitative features. For examples, the user just needs to tell the agent what kind of music he prefers to hear at what kind of context, like "I need fast and exciting music when I'm happy", "I need soft music to wake me up every morning at 8:00", "I need slow classical music, when I'm thinking", "I need rhythmic music when I'm walking", etc. Or, the user can define qualitative features, like "Romantic music for me means slow music with titles or lyric including word love", "My favorite music includes …" etc. Then, when the moment comes - the user tells the agent or a

pre-defined time approaches, or the user explicitly specifies some qualitative features, the agent can automatically, randomly and repeatedly choose music from the user's collection according to the pre-defined constraints without interruption until the user wants it to stop.

So far, MusicCat uses a profile-based user interface. Some learning techniques can be integrated into this system in the future. About one hundred midi songs are included in the database.The architecture of the system is shown in Fig.4-3. It adopts a client-side user modeling paradigm.

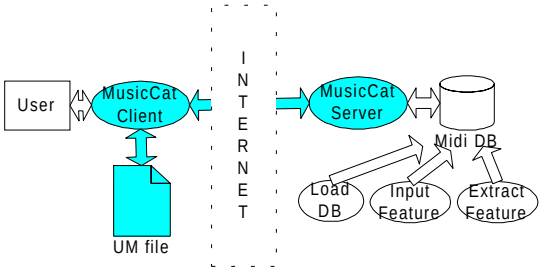

Fig.4-3 MusicCat system architecture.

In the system, we categorize the quantitative features of music into textual features including title, composer, genre, country, etc.; notation features including key signature, time signature, tonality, BPM, etc.; perceptual features including slow or fast, short or long, quiet or loud, non-rhythmic or rhythmic, and soft or exciting. User can define music corresponding to a particular context or qualitative feature based on these quantitative features (Fig.4-4).

In our system, we manually input the textual information. Notation features are automatically extracted from midi files. Perceptual features are computed from the parameters including average duration per note, average tempo, tempo deviation, average pitch change per note, etc. The weights of each parameter contributing to each perceptual feature are set manually. To make them more accurate, some psychoacoustic experiments should be necessary [10].

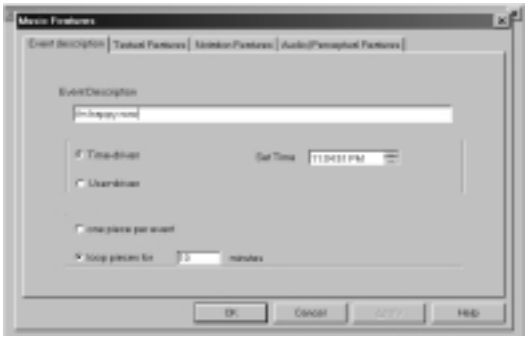

Fig. 4-4 MusicCat user profile dialog

The current language used in MusicCat to describe the user model, which will be sent from client to server whenever the user wants to hear music from MusicCat, is simply text CGI format. It works well in the current system. But in the future, a standard and more powerful description language should be necessary.

5. USER MODELING LANGUAGE

As mentioned above, user information is very valuable and needs to be shared in the future to make universal information retrieval possible. Here we propose an XML-like language, UMIRL (User Modeling for Information Retrieval Language), in which different systems may describe the user in this standard format to make the user model sharable and reusable. Although it is designed for music information retrieval purpose in this paper, it can be extended for general multimedia information retrieval as well.

The following are the design goals for UMIRL:

- UMIRL shall be compact.
- UMIRL shall be easy to type and read.
- UMIRL syntax shall be simple for the simple and common cases.
- UMIRL shall be expressed in strings that can easily be embedded in programs, scripts, and XML or HTML attributes.
- UMIRL shall be easily parsed.
- UMIRL shall be consistent with MPEG7.

An example of UMIRL is:

```
<user>
  <generalbackground>
    <name>John White </name>
    <education>MS</education>
    <citizenship>US</citizenship>
    <birthdate>9/7/1974</birthdate>
    <sex>male</sex>
    <occupation>student</occupation>
  </generalbackground>
  <musicbackground>
    <education>none<education>
    <instrument>piano<instrument>
    <instrument>vocal<instrument>
  </musicbackground>
  <generalpreferences>
    <color>blue</color>
    <animal>dog</animal>
  </generalpreferences>
  <musicpreferences>
    <genre>classical</genre>
    <genre>blues</genre>
    <genre>rock/pop</genre>
    <composer>Wolfgang Amadeus Mozart
</composer>
    <artist>Beatles</artist>
    <sample>
      <title>Yesterday</title>
      <artist>Beatles</artist>
    </sample>
  </musicpreferences>
  <habit>
```

```
    <context>I'm happy
      <tempo>very fast</tempo>
      <genre>pop</genre>
    </context>
    <pfeature>romantic
      <tempo>very slow</tempo>
      <softness>very soft<softness>
      <title>*love*</title>
    </pfeature>
    <context>bedtime
      <pfeature>romantic</pfeature>
    </context>
  </habit>
</user>
```

6. CONCLUSIONS AND FUTURE WORK

To make music information retrieval systems more efficient, both user modeling techniques and descriptions are important. Those are prerequisites for open and efficient personalized service. We propose the user modeling language because there hasn't been much attention on the ways to share the valuable user data. We believe the main issues discussed in this paper will be the most significant but also the hardiest points in this research area.

For MusicCat system, it would be better if we integrate learning techniques to do user modeling and support interface customization. Some wireless communication technology can be used to make the system portable. We can also use sensor-driven interface that can automatically detect the user context and then play corresponding music.

REFERENCES

[1] User Modeling in Expert Man-Machine Interfaces: A Case Study in Intelligent Information Retrieval, Giorgio Brajnik, Giovanni Guida, and Carlo Tasso, Systems, Man and Cybernetics, IEEE Transactions on Volume: 20 1, Jan.-Feb. 1990, Page(s): 166 –185.

[2] Social Information Filtering: Algorithms for Automating "Word of Mouth", Upendra Shardanand and Pattie Maes, Conference proceedings on Human factors in computing systems, 1995, Pages 210-217.

[3] Toward the digital music library: tune retrieval from acoustic input, Rodger J. McNab, Lloyd A. Smith, Ian H. Witten, Clare L. Henderson and Sally Jo Cunningham, ACM DL'96.

[4] MELDEX, University of Waikato's New Zealand, http://www.nzdl.org/.

[5] Query by Humming, musical information retrieval in an audio database, Asif Ghias, Jonathan Logan, David Chamberlin and Brian C. Smith, Proceedings of the third ACM international conference on Multimedia, 1995, Pages 231 – 236.

[6] THEMEFINDER, Stanford University, http://www.ccarh.org/themefinder/.

[7] SEARCH BY HUMMING, University of Southampton, http://audio.ecs.soton.ac.uk/sbh/.

[8] Content-based classification, search, and retrieval of audio Wold, E.; Blum, T.; Keislar, D.; Wheaten, J. IEEE Multimedia Volume: 3 3 , Fall 1996 , Page(s): 27 –36.

[9] MUSCLE FISH, http://www.musclefish.com.

[10] Music-Listening Systems, Eric Scheirer, PhD dissertation, 2000.

[11] http://www.darmstadt.gmd.de/mobile/MPEG7/.