PATS: Realization and User Evaluation of an Automatic Playlist Generator

Steffen Pauws
Philips Research Eindhoven
Prof. Holstlaan 4 (WY21)
5656 AA Eindhoven, the Netherlands
+31 40 27 45415
steffen.pauws@philips.com

Berry Eggen
Philips Research Eindhoven, and Technische Universiteit Eindhoven / Faculty of Industrial Design
Eindhoven, the Netherlands
j.h.eggen@tue.nl

ABSTRACT

A means to ease selecting preferred music referred to as Personalized Automatic Track Selection (PATS) has been developed. PATS generates playlists that suit a particular context-of-use, that is, the real-world environment in which the music is heard. To create playlists, it uses a dynamic clustering method in which songs are grouped based on their attribute similarity. The similarity measure selectively weighs attribute-values, as not all attribute-values are equally important in a context-of-use. An inductive learning algorithm is used to reveal the most important attribute-values for a context-of-use from preference feedback of the user. In a controlled user experiment, the quality of PATS-compiled and randomly assembled playlists for jazz music was assessed in two contexts-of-use. The quality of the randomly assembled playlists was used as base-line. The two contexts-of-use were 'listening to soft music' and 'listening to lively music'. Playlist quality was measured by *precision* (songs that suit the context-of-use), *coverage* (songs that suit the context-of-use but that were not already contained in previous playlists) and a *rating score*. Results showed that PATS playlists contained increasingly more preferred music (increasingly higher *precision*), covered more preferred music in the collection (higher *coverage*), and were *rated* higher than randomly assembled playlists.

1. INTRODUCTION

So far, music player functionality that has been designed for accessing and exploiting large personal music collections aims at providing *fast* and *accurate* ways to retrieve relevant music. This type of access generally requires well-defined targets. Music listeners need to instantaneously associate artists and song titles (or even CD and track numbers) with music. This is not an easy task to do, since titles and artists are not necessarily learnt together with the music [8]. In our view, selecting music from a large personal music collection is better described as a search for poorly defined targets. These targets are poorly defined since it is reasonable to assume that music listeners have no *a-priori* master list of preferred songs for every listening intention, lack precise knowledge about the music, and cannot easily express their music preference *on-the-fly*. Rather, choice for music requires listening to brief musical passages to recognize the music before being able to express a preference for it.

If we take music programming on current music (jukebox) players as an example, it allows playing a personally created temporal sequence of songs in one go, once the playlist or program has been created. The creation of a playlist, however, can be a time-consuming choice task. It is hard to arrive at an optimal playlist as music has personal appeal to the listener and is judged on many subjective criteria. Also, optimality requires a complete and thorough examination of all available music in a collection, which is impractical to do so. Lastly, music programming consists of multiple serial music choices that influence each other; choice criteria pertain to individual songs as well as already selected choices. A means to ease and speed up this music selection process could be of much help to the music listener. PATS (Personalized Automatic Track Selection) is a feature for music players that automatically creates playlists for a particular listening occasion (or *context-of-use*) with minimal user intervention [7].

This paper presents the realization of PATS and the results of a controlled user experiment to assess its performance. PATS has been realized by a decentralized and dynamic cluster algorithm that continually groups songs using an attribute-value-based similarity measure. A song refers to a recorded performance of an artist as can be found as a track on a CD. The clustering on similarity adheres to the listener's wish of coherent music in a playlist. Since it is likely that this coherence is based on particular attribute values of the songs, some attribute values contribute more than others in the computation of the similarity by the use of weights. At the same time, the clustering allows groups of songs to dissolve to form new groups. This concept adheres to the listener's wish of varied music within a playlist and over time. Clusters are presented to the music listener as playlists from which the listener can remove songs that do not meet the expectations of what a playlist should contain. An inductive learning algorithm based on decision trees is then employed that tries to reveal the attribute values that might explain the removal of songs. Weights of attribute values are adjusted accordingly, and the clustering continues with these new weights aiming at providing better future playlists.

2. PATS: EASY WAY TO SELECT MUSIC

Some widely used terms such as context-of-use and music preference need further clarification. Also, we tell what we mean with minimal user intervention and explain the requirements for PATS.

2.1 Context-of-use

We define *context-of-use* as the real-world environment in which the music is heard, being it a party, romantic evening or the traveling by car or train. The use of this concept is thought to be a powerful starting point for creating a playlist or as an organizing principle for a music collection.

In every-day language, the terms *music preference* and *musical taste* are intuitively meaningful and apparently self-evident. They are interchangeably used to refer to the same concept. We make a distinction between the two, following the definitions as given by Abeles [1].

Musical taste is defined as a person's slowly evolving long-term commitment to a particular music idiom. Its development is assumed to depend on the cultural environment, the major consensus [3], peer approval, musical training [4], age as an

indirect factor [5][11] and other personal characteristics. Personal music acquisition behavior over time is likely to represent the development of a person's musical taste.

On the other hand, music preference is defined as a person's temporary liking of particular music content in a particular context-of-use. It is instantaneous in nature and subordinate to the musical taste of a person. Music is deemed to be preferred if its musical features suit particular activities, moods or listening purposes. Therefore, the context-of-use is supposed to produce constraints and opportunities for what music is preferred. It sets what kind of music should be selected and what kind of music should be rejected. North and Hargreaves [10] showed that music preference is associated with the listening environment and that people prefer to use different descriptors for music to be listened to in different environments. For instance, music for a dance party sets up desirable and undesirable criteria on tempo, rhythmic structure, musical instrumentation and performers, which are likely to be different for a romantic evening, for dull or repetitive activities or for car traveling.

However, an indefinite number of contexts-of-use may exist; they all produce different criteria for preferred music. In addition, the particular experience to listen to given music does not need to be the same in similar contexts-of-use or a given context-of-use is unlikely to be best provided with exactly the same music, over and over again. In other words, music preference changes over time.

2.2 Interactive control of PATS

When using PATS, the link between a context-of-use and a playlist is established by choosing a single preferred song that is used to set up a complete playlist. Thus, music listeners only have to select a song that they currently want to listen to or that they prefer in the given context-of-use. This selection requires minimal cognitive effort as it may be the result of *habitual behavior* or *affect referral*. People may choose a song that is chosen always in a similar context-of-use, that was selected last time in a similar context-of-use, or that was given much thought lately.

After selecting a song, PATS generates and presents a playlist, which includes the selected song and songs that are similar to the selected one. While listening, a music listener indicates what songs in the playlist do not fit the intended context-of-use. As only a decision of rejection is needed for a small number of songs, this task makes only a small demand on memory processes. This user feedback is used by PATS to learn about music preferences of the listener and to adapt its compilation strategy for future playlists. If the system adapts well to a listener's music preferences, user feedback is no longer required. Moreover, PATS does not require any other user control actions.

2.3 Requirements

Ideally, PATS should make music choices that would have been made by the music listener in case no PATS was available. Therefore, it uses attribute information of music on which human choice is largely based, and generates playlists that are both coherent and varied.

Jazz was chosen as a music domain in this long-term research project, as jazz contains a variety of well-defined styles or time periods serving a diverse listening audience and its appreciation is largely insensitive to temporarily prevailing music cultures and movements.

2.3.1 Attribute representation (meta-data) of music

Music listeners use many different musical attributes for their music choice. Talking about and judging popular and jazz music in terms of musicians, instruments, and music styles is common. It is therefore reasonable to represent songs as a collection of attribute-value pairs (meta-data). We have created and collected an attribute representation for jazz music of 18 attributes, in total. Their values were primarily extracted from CD booklets, discographies, books on jazz music education and training, and systematic listening. A listing of all attributes and an instance is given in Table 1.

Table 1. Attribute representation for jazz music.

Title	*Title of the song*	'All blues'
Main artist	*Leading performer/band*	Miles Davis
Album	*Title of album*	'Kind of blue'
Year	*Year of release*	1959
Style	*Jazz style or era*	postbop
Tempo	*Global tempo in bpm*	144
Musicians	*List of musicians*	Miles Davis, John Coltrane, Cannonball Adderley, Bill Evans, Paul Chambers, Jimmy Cobb
Instruments	*List of instruments*	trumpet, tenor saxophone, alto saxophone, piano, double bass, drums
Ensemble strength	*No. musicians*	6
Soloists	*Soloing musicians*	Miles Davis, John Coltrane, Cannonball Adderley, Bill Evans
Composer	*Composer of the song*	Miles Davis
Producer	*Producer of the song*	Teo Macero, Ray Moore
Standard/Classic	*Standard or classic jazz song?*	Yes
Place	*Recording place*	New York
Live	*In front of a live audience?*	No
Label	*Record company*	CBS
Rhythm	*Rhythmic foundation*	6/8
Progression	*Melodic/harmonic development*	modal

Results of a focus group study showed that the set of attributes and their values is sufficient to express reported preferences for jazz music. In this study, participants were instructed to assort a set of 22 jazz songs into a preferred and rejected category and verbalize their decisions. Many of the criteria elicited could be expressed as a logical combination of attribute-value pairs.

2.3.2 Wish for coherence

Coherence of a playlist refers to the degree of homogeneity of the music in a playlist and the extent to which individual songs are related to each other. It does not solely depend on some similarity between any two songs, but also depends on all other songs in a playlist and the conceptual description a music listener can give to the songs involved.

Coherence may be based on a similarity between songs such as the sharing of relevant attribute values. When choosing music, music listeners tend to focus on relevant attribute values for reducing the available choice set of songs and for making different songs comparable. This includes eliminating songs with less relevant attributes values and retaining only the ones with the more relevant attributes values. Choice on the basis of elimination is a

common strategy in every-day choice tasks[13]. For instance, a music choice strategy is to first reduce the choice set by eliminating those songs that do not belong to a particular music style or in which a particular musician did not participate, before continuing further search.

2.3.3 Wish for variation

Variation refers to the degree of diversity of songs in an individual playlist and in successive playlists. It contradicts the requirement for coherence. Variation is a psychological requirement for continual music enjoyment by introducing new musical content and making the outcome unpredictable. It produces surprise effects at the music listener such as the re-discovery of 'forgotten' music.

As music preference changes over time, the most elementary requirement is that not exactly the same music should be repeatedly presented for a given context-of-use. Also, music within a playlist should be varied as the experience of each additional song in a playlist may decrease if it contains features that are already covered by other songs in the list.

2.4 Realization

PATS makes use of a two-step strategy in interaction with the user. First, songs are clustered based on a similarity measure that selectively weighs attribute values of the songs. Clusters are presented as playlists to be judged by the user on suitability for a desired context-of-use. Second, an inductive learning algorithm is used to uncover the criteria on attribute values that pertain to this judgment. The weights of the attribute values involved are adjusted accordingly for adapting the clustering process.

2.4.1 Similarity measure

If it is known that a set of songs is preferred (or fit a given context-of-use), then it is likely that preference can be generalized to other songs based solely on the fact that they are similar. Although a similarity measure may not provide all explanatory evidence for stating preference, it is an essential component for providing some choice structure amongst songs. The used similarity between songs is based on a weighted sum of their attribute similarities.

Let $O = \{o_1, o_2, \ldots, o_N\}$ denote the music collection containing N songs. Each song $o_i \in O$ is represented by an arbitrary ordered set of K valued attributes $A_k = V_{ik},\ k = 1, \ldots, K$ where A_k refers to the name of the attribute. A song is then represented by a vector $o_i = (V_{i1}, V_{i2}, \ldots, V_{iK})$. In our case, the domain of an attribute can be nominal, binary, categorical, numerical or set-oriented. For notational convenience, the value of $V_{ik} = (v_{ik1}, v_{ik2}, \ldots, v_{ikL_{ik}})$ is itself a vector of length L_{ik}. For most attributes, $L_{ik} = 1$, except for set-oriented attributes since they represent the list of participating musicians or the instrumentation as found on a musical recording. Likewise, non-negative weight vectors $W_{ik} = (w_{ik1}, w_{ik2}, \ldots, w_{ikL_{ik}})$ are associated with each attribute A_k and each song o_i. These weights measure the relevance of an attribute value in the computation of the similarity between songs.

For nominal, binary or categorical attributes such as titles, person names and music genres, the attribute similarity $s(v_{ikl}, v_{jkl})$ is either 1 if the attribute values are identical, or 0 if the values are different. More precisely,

$$s(v_{ikl}, v_{jkl}) = \begin{cases} 1 & , v_{ikl} = v_{jkl} \\ 0 & , v_{ikl} \neq v_{jkl} \end{cases}$$

For numeric attributes such as the global tempo in beats per minute or year of release, the attribute similarity $s(v_{ikl}, v_{jkl})$ is one minus the ratio between the absolute value and the total span of the numerical attribute domain. More precisely,

$$s(v_{ikl}, v_{jkl}) = 1 - \frac{|v_{ikl} - v_{jkl}|}{R_k}$$

The similarity measure $S(o_i, o_j)$ between song o_i and o_j is then the normalized weighted sum of all involved attribute similarities. Its value ranges between 0 and 1. More precisely,

$$S(o_i, o_j) = \sum_{k=1}^{K} \sum_{l=1}^{L_{ik}} w_{ikl} \cdot s(v_{ikl}, v_{jkl}), \quad \text{with} \sum_{k=1}^{K} \sum_{l=1}^{L_{ik}} w_{ikl} = 1,$$

where K is the number of attributes, L_{ik} is the number of values for attribute A_k, and $s(v_{ikl}, v_{jkl})$ denotes the attribute similarity of attribute A_k between song o_i and o_j.

Note that the similarity between any song and itself is identical for all songs, and is the maximum possible (i.e., $S(o_i, o_j) \leq S(o_i, o_i) = S(o_j, o_j) = 1$). This is evident since it is unlikely that a song would be mistaken for another.

Also, note that the similarity measure is asymmetric (i.e., $S(o_i, o_j) \neq S(o_j, o_i)$) because each song has its own set of weights. Asymmetry in similarity refers to the observation that a song o_i is more similar to a song o_j in one context, while it is the other way around in another context. It can be produced by the order in which songs are compared and what song acts as a reference point. The choice of a reference point makes attribute-values that are not part of the other song of less concern to the similarity computation. Music that is more familiar to the listener may act as such a reference point. Then, for instance, music from relatively unknown artists may be judged quite similar to music of well-known artists, whereas the converse judgment may be not true.

2.4.2 Cluster method

The similarity measure governs the grouping of songs in a cluster method. Cluster methods are traditionally based on optimizing a unitary performance index such as maximizing the mean within-cluster similarity. We have however the two-edged objective to group songs adhering both to the wish for coherence and to the wish for variation. The wish for coherence can be seen as maximizing within-cluster similarity, whereas the wish for variation should rather decrease this within-cluster similarity. To meet these contrasting requirements, a decentralized clustering approach is used in which the clustering is established at the locality of each individual song with little external main control of the global clustering process.

In this approach, songs are placed in a two-dimensional Euclidean space of a finite size. The number of dimensions is arbitrary. Songs move around in discrete time steps at an initially randomly chosen velocity. For that, a song has been augmented with position and velocity coordinates. Basically, at each time step, a randomly chosen song 'senses' whether of not any other song is in its nearest vicinity. Vicinity is defined as the area that is contained in a given circle centered at a song's current position in Euclidean distance sense. Vicinity checking has been realized by a constant time algorithm based on a spatial elimination technique known as the *sector method*. If the current song finds another song in its nearest vicinity, the similarity between the current song and the other is computed. This similarity value is used as a probability

measure to determine whether or not the current song groups with the other. Grouping can be seen as a one-way 'following' relation: each song groups only with one other song though multiple songs can group with the same song. It means that the current song adjusts its velocity to the velocity of the other song such that they stay close to each other in the two-dimensional space. It also implies that the grouping of the current song with another can have as side-effects that (1) a previous grouping in which the current song was involved will be broken and (2) the songs that 'follow' the current song are also indirectly involved.

From a global perspective, clusters are formed by the grouping mechanism and dissolved by the breaking up of groups (see Figure 1). Since the similarity measure selectively weighs different attribute values of the songs, clusters of songs arise that have several distinct attribute values in common. This is deemed to adhere to the wish for coherence. Since the content of a cluster varies continually in time, this is deemed to adhere to the wish for variation.

Eventually, when the user selects a preferred song, the cluster in which this song is contained is presented as a playlist. Special measures in the clustering process are taken to preclude clusters from becoming too big.

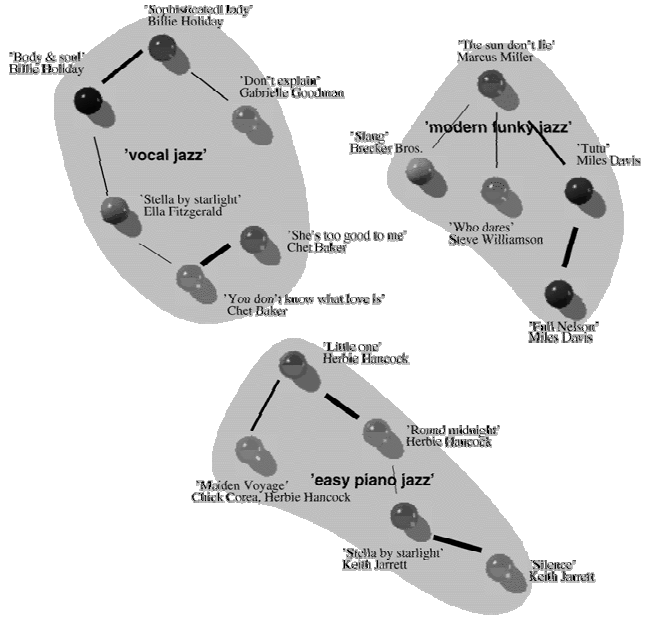

Figure 1. An ideal cluster result of songs that may represent a playlist suiting a particular context-of-use for listening to 'vocal jazz', 'modern funky jazz' or 'easy piano jazz' (cluster labels are added manually). Songs are represented by differently colored (or shaded) marbles. Similar songs have similar colors (shades). The lines connecting these marbles represent the grouping of songs in a cluster. The line width denotes the similarity between two songs.

2.4.3 Inductive learning

User feedback consists of the explicit indication of songs in a playlist that do not fit the intended context-of-use. In this way, it is known what songs in the playlist are preferred and what songs are rejected. An inductive learning algorithm based on the construction of a decision tree is used to uncover the attribute values that assort songs into the categories *preferred* and *rejected*.

A decision tree is incrementally constructed by a greedy, non-backtracking search algorithm in which the search is directed by an attribute selection heuristic. This heuristic is based on local information about how well an attribute partitions the set of songs (i.e., the current playlist) into the two categories under its values. Only attributes that are not already present in the path from the root to the current point of investigation are considered. The incremental nature of the process is characterized by replacing a leaf of the tree under construction by a new sub-tree of depth one. This sub-tree consists of a node, which carries an attribute that provides the best possible categorization, and branches that represent the partitions along the values of the attribute. This process is continued until partitions contain only songs of one category or no more songs are left. If no more attributes are left while the current leaf still contains preferred and rejected songs, the decision tree is *indecisive* for the songs involved. The constructed tree then contains interior nodes and branches specifying attributes and their values along which the songs in the playlist were originally partitioned into the categories *preferred* and *rejected* (see Figure 2).

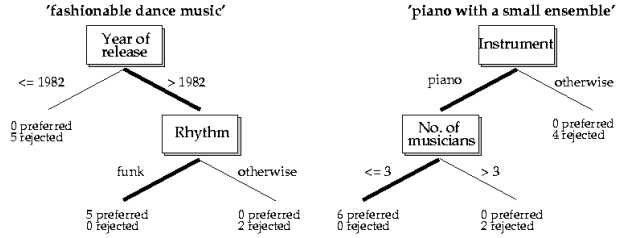

Figure 2. Decision trees to uncover the attribute values that assort songs into the categories *preferred* and *rejected* for 'fashionable dance music' and 'piano with a small ensemble'.

Given a decision tree, the categorization of a song starts at the root of a tree. Attribute values at the branches of the tree are compared to the value of the corresponding attribute of the song. A branch is then taken that is appropriate to the outcome of the comparison. This comparison and branching process continues recursively until a leaf is encountered at which time the *predicted* category of the song is known.

Decision tree construction algorithms differ in the type of heuristic function for attribute selection and the branching factor on each interior node. We have experimented with four different algorithms: ID3 [9], ID3-IV [9], ID3-BIN that is a variant of ID3 with a binary branching factor and INFERULE [12].

Basically, the ID3 family of algorithms uses a heuristic that is based on minimizing the entropy of the set of songs by selecting the attribute that makes the categories least randomly distributed over the disjoint partitions of the set along its values. In other words, it selects the attribute that has the highest *information gain (ratio)* heuristic when used to partition a set of songs. On the other hand, the INFERULE algorithm uses a *relative goodness* heuristic that selects an attribute *value* such that the category distribution in the resulting partitions differs considerably from the original set. This heuristic is especially useful if the available attributes are not sufficient to discern category membership for a given song [12]. This is also typical for our categorization problem for it is very unlikely that the set of music attributes used will cover the whole repertoire of music preferences. Since this heuristic considers attribute values instead of attributes, the result is a binary decision tree.

All algorithms were augmented with strategies to deal with attributes that are not nominal such as numeric attributes and set-oriented attributes, strategies to deal with missing attribute values, cases of equal evaluation of attributes (value) under the attribute selection heuristic and cases of indecisive leaves.

The four algorithms were assessed on their *categorization accuracy* and the *compactness* of the resulting decision tree using data sets of 300 jazz songs pre-categorized by four participants

and using training sets of different size to construct the tree. Categorization accuracy was defined as the percentage of songs in the complete data set that were correctly categorized as being *preferred* or *rejected*. Compactness was defined as the proportion of leaves that would be obtained by the least compact decision tree that is possible. The least compact tree is a tree of depth one that captures each song in a separate leaf. Compact trees have been theoretically proven to yield high categorization accuracy on 'unseen' data in a probabilistic and worst-case sense [2]. This suggests that it is wise to favor trees with fewer leaves, because these trees are supposed to be better categorizers solely on the fact that they have fewer leaves.

In short, the results showed that both ID3-BIN and INFERULE produced the most accurate decision trees for categorizing the data sets as being preferred or rejected under various training set sizes. In addition, INFERULE produced the most compact trees. ID3 produced the least accurate decision tree as it did not even exceed the categorization accuracy of a simple categorizer that randomly stated a given song as being preferred or rejected.

Obviously, the INFERULE algorithm was the best choice among the four alternatives to be incorporated in the PATS system. The input to INFERULE is the playlist in which songs are indicated as preferred or rejected by the user. The output is a decision tree that separates preferred and rejected songs on the basis of their attribute values. Weights of all songs in the collection are now adjusted in two stages, before the clustering is re-started.

In the first stage, the decision tree is used to categorize the complete music collection into the predicted categories *preferred*, *rejected* and *indecisive*. The latter category is required since there can be indecisive leaves in the tree. In the second stage, weights of attribute values are multiplied by a factor in the case of *preferred* songs and divided by this factor in the case of *rejected* songs. The factor is the multiplication of an arbitrary constant with $1/2^{l-1}$, where l denotes the level in the tree at which the attribute value occurs. The root of the tree is at level 1. It is assumed that attribute values occurring higher in the tree are more relevant than attribute values at lower regions of the tree. The weights of *indecisive* songs are left unchanged.

3. USER EVALUATION

A controlled user experiment examined the quality of PATS-compiled playlists and randomly assembled playlists. Participants judged the quality of both type of playlists in two different contexts-of-use over four experimental sessions. Playlist quality was measured by *precision*, *coverage* and a *rating score*. A post-experiment interview was used to yield supplementary findings on perceived usefulness of automatic music compilation.

3.1 Hypotheses

The quality of PATS-generated playlists should be higher than randomly assembled playlists irrespective of a given context-of-use. It is hypothesized that

1. Playlists compiled by PATS contain more preferred songs than randomly assembled playlists, irrespective of a given context-of-use.
2. Similarly, PATS playlists are rated higher than randomly assembled playlists, irrespective of a given context-of-use.

PATS playlists should adapt to a music preference in a given context-of-use. It is hypothesized that

3. Successive playlists compiled by PATS contain an increasing number of preferred songs.
4. Similarly, successive PATS playlists are successively rated higher.

Finally, PATS playlists should cover more relevant music over time of use than randomly assembled playlists. It is hypothesized that

5. Successive playlists compiled by PATS contain more distinct and preferred songs than randomly assembled playlists.

3.2 Measures

Three measures for playlist quality were defined: *precision*, *coverage*, and a *rating score*.

Precision was defined as the proportion of songs in a playlist that suits the given context-of-use. Ideally, the precision curve should approach 1, meaning adequate adaptation to a given context-of-use.

Coverage was defined as the cumulative number of songs that suits the given context-of-use and that was not already present in previous playlists. Over successive playlists, the *coverage* measure is a non-decreasing curve. Ideally, this curve should approach the total number of songs in all successive playlists, meaning nearly complete coverage of preferred material given the number of playlists.

The rationale of *precision* and *coverage* is that it is very likely that music listeners wish a single playlist to adequately reflect their music preference as well as that successive playlists cover as much different music reflecting their preference as possible.

A *rating score* was defined as the participant's rating of a playlist. This score was defined on a scale ranging from 0 to 10 similar to the traditional ordinal report-mark on Dutch elementary school (0 = extremely bad, 1 = very bad, 2 = bad, 3 = very insufficient, 4 = insufficient, 5 = almost sufficient, 6 = sufficient, 7 = fair, 8 = good, 9 = very good, 10 = excellent).

The post-experiment interview posed a single question concerning perceived usefulness of an automatic playlist generator (translated from Dutch): Do you find a feature that automatically compiles music for you a useful feature?

3.3 Method

3.3.1 Instruction

Participants were not informed about the actual purpose of the experiment being a comparison between two different playlist generation methods. Instead, they were told that the research was aimed at eliciting on what criteria people appraise music. They were informed about the global experimental procedures and the test material, and prepared for the relatively high demands for participation in the experiment since they had to return on four separate days, preferably within one week.

The two contexts-of-use in the experiment were described to the participants as 'a lively and loud atmosphere such as dance music for a party' and 'a soft atmosphere such as background music at a dinner'.

At the first day, they were asked to imagine and describe personal instantiations of the two contexts-of-use, that is, the general circumstances in which the music would be heard. Three small tasks were intended to elicit some desirable properties of music suited in one of the two contexts-of-use. In the first task, participants completed a form in which they were asked to describe what music would be appropriate in the given context-of-use. In the second task, they were asked to compile a playlist by paper and pencil; they could select music from a list. Concluding, participants had to select a song from a list that they would definitely want to listen to in the given context-of-use. The list was alphabetically ordered by musicians and contained all songs in the collection. They had to do these tasks twice for each context-of-use separately. So, the results of these tasks were

personal instantiations of the two different contexts-of-use, an elicitation of the music that would fit the contexts-of-use and a 'highly preferred' song for each context-of-use.

For all four days, they were instructed to restrict their music listening behavior to the instantiation of each context-of-use. Also, the same 'highly preferred' song was used to set up a playlist for a given context-of-use.

3.3.2 Interactive system

An interactive computer application was implemented to listen and judge a playlist by using a standard mouse and a graphical user interface. Title, and names of composers and artists of a song were shown. Songs in a playlist were not displayed list-wise, but were presented one-by-one. Controls for common music play features and for going through a playlist were provided. Also, buttons for indicating preference in terms of 'good' and 'bad' per song in the playlist were provided.

Participants were instructed how to operate the interactive system. Information about interactive procedures to follow during an experimental session was readily available to the participants during the whole experiment.

3.3.3 Design

A factorial within-subject design with three independent variables was applied. The first independent variable *playlist generator* referred to the method used for music compilation, that is, PATS or random. The second independent variable *context-of-use* referred to the two pre-defined contexts-of-use, that is, soft music and lively music. The order in which the levels of *context-of-use* and *playlist generator* were applied was counterbalanced. The third independent variable *session* referred to the four experimental sessions in which playlists were listened to in a given context-of- use. These sessions were intended to measure adaptive properties and long-term use of the compilation strategies in terms of changes in *playlist quality* as a function of time.

3.3.4 Test material and equipment

A music database comprising 300 one-minute excerpts of jazz songs (MPEG-1 Part 2 Layer II 128 Kbps stereo) from 100 commercial CD albums served as test material. The music collection covered 12 popular jazz styles. These styles cover a considerable part of the whole jazz period. Each style contained 25 songs. Pilot experiments showed that the shortness and sound quality of the excerpts did not negatively influence judgment. The test equipment consisted of a SUN Sparc-5 workstation, APC/CS4231 codec audio chip, and two Fostex 6301 B personal monitors (combined amplifier and loudspeaker system).

Participants were seated behind a desk in front of a 17-inch monitor (Philips Brilliance 17A) in a sound-proof experimental room. They could adjust the audio volume to a preferred level. Both the mouse pad and the monitor were positioned at a comfortable working level.

3.3.5 Task

The task was to listen to a set of 11 songs (one-minute excerpts) that made up a playlist, while imagining a fixed and pre-defined context-of-use. Due to the size of a playlist, judgments of the songs were collected by presenting them in series. The songs were shown one at the time. Participants only had to decide which song did not fit the desired context-of-use, if at all. In the process of listening, participants were allowed to compare songs freely in any combination and cancel any judgement already expressed. There were no time restrictions.

3.3.6 Procedure

Participants took part in eight experimental sessions on four separate days, preferably within one week. The first session started with instructions and a questionnaire to record personal data and attributes. Use of the interactive system was explained and demonstrated. At each session, participants were alternately presented a PATS and a randomly assembled playlist with a pause in between. In four consecutive sessions, participants were instructed to perform music listening tasks by considering a fixed and pre-defined context-of-use. At the start of every four sessions, participants completed a form in which they described their context-of-use and what music would be appropriate in that context-of-use. In addition, they were asked to select a song from the music collection that they definitely would listen to in the given context-of-use. Both this song and the context-of-use had to be recalled each time a new experimental session started. A PATS and a randomly assembled playlist was automatically generated round the selected song and presented to the participant. Then, a listening and judgment task for the given playlist started. When participants had completed a task, the interactive system was automatically shut down.

After completing each judgment task, participants were asked to rate the playlist just listened to, on a scale ranging from 0 to 10.

At the end of the experiment, a small interview was conducted.

3.3.7 Participants

Twenty participants (17 males, 3 females) took part in the experiment. They were recruited by advertisements and all got a fixed fee. All participants were frequent listeners to jazz music; for admission to the experiment, they had to be able to freely recall eight jazz musicians, rank them on personal taste and mention number of recordings (CD albums, tapes) owned for each musician. The average age of the participants was 26 years (min.: 19, max.: 39). All participants had completed higher vocational education. Sixteen participants played a musical instrument.

3.4 Results

Playlists contained 11 songs from which one was selected by the participant. This song was excluded from the data as this song was not determined by the system, leaving 10 songs per playlist to consider for analysis.

3.4.1 Precision

The results for the *precision* measure are shown in Figure 3.

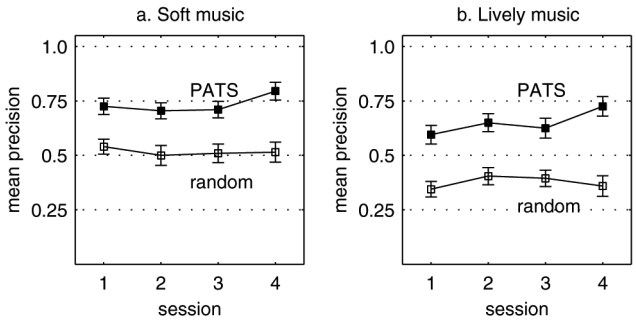

Figure 3. Mean *precision* (and standard error) of the playlists in different *contexts-of-use*. The left-hand panel (a) shows mean *precision* for both playlist generators (PATS and random) in the 'soft music' context-of-use. The right-hand panel (b) shows mean *precision* for both generators in the 'lively music' context-of-use.

A MANOVA analysis with repeated measures was conducted in which *session* (4), *context-of-use* (2), and *playlist generator* (2) were treated as within-subject independent variables. *Precision*

was dependent variable. A main effect for *playlist generator* was found to be significant ($F(1,19) = 89.766$, $p < 0.0001$). Playlists compiled by PATS contained more preferred songs than randomly assembled playlists (mean *precision*: 0.69 (PATS), 0.45 (random)). A main effect for *context-of-use* was found to be significant ($F(1,19) = 13.842$, $p < 0.005$). Playlists for the 'soft music' context-of-use contained more preferred songs (mean *precision*: 0.63 (soft music), 0.51 (lively music)). An interaction effect for *playlist generator* by *session* was just not significant ($F(3,17) = 2.675$, $p = 0.08$), whereas, in the univariate test, it was found to be significant ($F(3,57) = 2.835$, $p < 0.05$). Further analysis of this interaction effect revealed a significant difference in mean *precision* between the fourth PATS playlist and mean *precision* of preceding PATS playlists in contrast to randomly assembled playlists ($F(1,19) = 8.935$, $p < 0.01$). In other words, each fourth PATS playlist contained more preferred songs than the preceding three PATS playlists (mean *precision* of fourth PATS session: 0.76; mean *precision* of the first three PATS sessions: 0.67). No other effects were found to be significant.

3.4.2 Coverage

The results for the *coverage* measure are shown in Figure 4.

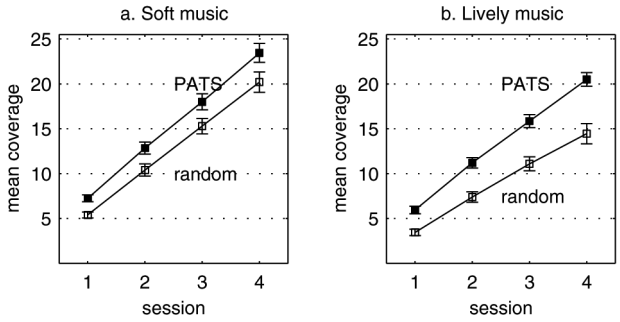

Figure 4. Mean *coverage* (and standard error) of the playlists in different contexts-of-use. Recall that *coverage* is a cumulative measure. The left-hand panel (a) shows mean *coverage* for both playlist generators (PATS and random) in the 'soft music' context-of-use. The right-hand panel (b) shows mean *coverage* for both generators in the 'lively music' context-of-use. Note the maximally achievable *coverage* in four successive playlists is 40.

A MANOVA analysis with repeated measures was conducted in which *session* (4), *playlist generator* (2), and *context-of-use* (2) were treated as within-subject independent variables. *Coverage* was dependent variable. A main effect for *playlist generator* was found to be significant ($F(1,19) = 63.171$, $p < 0.001$). More distinct and preferred songs were present in successive PATS playlists than in successive randomly assembled playlists (mean *coverage* at fourth session: 22.0 (PATS), 17.3 (random)). A main effect for *context-of-use* was found to be significant ($F(1,19) = 13.523$, $p < 0.005$). It appeared that playlists for the 'soft music' context-of-use contained more distinct and preferred songs (mean *coverage* at fourth session: 21.8 (soft music), 17.5 (lively music)). A main effect for *session* was found to be significant ($F(3,17) = 284.326$, $p < 0.001$). More particularly, the *coverage* curves for all conditions showed a significantly linear course over sessions ($F(1,19) = 852.268$, $p < 0.001$). Also, an interaction effect for *playlist generator* by *session* was found to be significant ($F(3,17) = 7.602$, $p < 0.005$). Successive playlists compiled by PATS contained more varied preferred songs than randomly assembled playlists. Likewise, the slopes of the coverage curves for PATS playlists appeared to be significantly higher than for randomly assembled playlists (*coverage* slope: 5.2 (PATS), 4.3 (random)). For each new playlist, PATS added five preferred songs that were not already contained in earlier playlists. For comparison, the random approach added four songs. No other effects were found to be significant.

3.4.3 Rating score

The results for the *rating score* are shown in Figure 5.

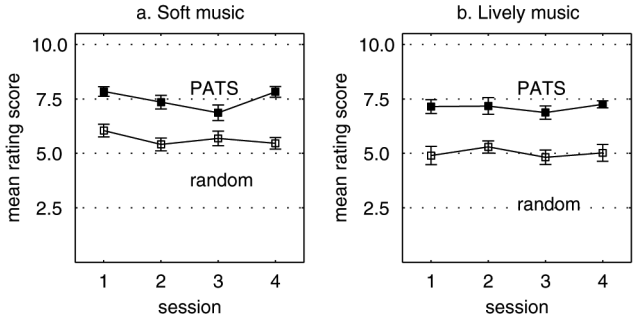

Figure 5. Mean *rating score* (and standard error) of the playlists in different contexts-of-use. The left-hand panel (a) shows mean *rating* for both playlist generators (PATS and random) in the 'soft music' context-of-use. The right-hand panel (b) shows mean *rating score* for both generators in the 'lively music' context-of-use.

A MANOVA analysis was conducted in which *playlist generator* (2), *context-of-use* (2), and *session* (4) were treated as within-subject independent variables. *Rating score* was dependent variable. A significant main effect for *playlist generator* was found ($F(1,19) = 85.085$, $p < 0.001$). Playlists compiled by PATS were rated higher than randomly assembled playlists (mean *rating score*: 7.3 (PATS), 5.3 (random)). In normative terms, PATS playlists can be characterized as 'more than fair' and randomly assembled playlists as 'almost sufficient'. A significant main effect for *context-of-use* was found ($F(1,19) = 12.574$, $p < 0.005$). Playlists for the 'soft music' context-of-use were rated higher (mean *rating score*: 6.6 (soft music), 6.1 (lively music)). No other significant effects were found.

3.4.4 Interview

The post-experiment interview yielded relevant supplementary findings about the perceived usefulness of automatic music compilation. Of the 20 participants, twelve participants (60%) told that they would appreciate and use an automatic playlist generator; they commented that it would easily acquaint them with varying music styles and artists and would be a means to adequately cover their personal music collection. Two participants explained their appraisal by referring to easy searching in an ever-increasing number of songs. The other eight participants rejected the usefulness of such a system. Their main objection was a loss of control in music selection, though one of these participants found automatic playlist generation relevant for cafe's and department stores.

3.5 Discussion

A user experiment examined the quality of PATS-generated playlists and randomly assembled playlists. PATS playlists appeared to contain more preferred songs and were rated higher than randomly assembled playlists in both contexts-of-use (see Hypothesis 1). In addition, PATS playlists appeared to contain more preferred songs that were not already contained in previous playlists than randomly assembled playlists (see Hypothesis 2). For each new playlist, PATS found five preferred songs that were not already contained in earlier playlists. There were no indications that PATS would deteriorate in finding new preferred music for future playlists.

In contrast to what was stated in Hypotheses 1 and 2, 'soft music' playlists appeared to contain more preferred and more varied music than 'lively music' playlists. 'Soft music' playlists were also rated higher than 'lively music' playlists. As this *context-of-use* effect both concerned PATS and randomly assembled playlists, the two most likely explanations are that (1) more 'soft music' was apparently available in the music collection than 'lively music' or (2) a preference for 'soft music' is apparently easier to satisfy than a preference for 'lively music'.

The fourth PATS playlist appeared to contain one more preferred song than the first three PATS playlists, which indicates that PATS playlists adapted to a given context-of-use (see Hypothesis 3). However, successive PATS playlists were not rated increasingly higher. This indicates that improvement of the playlists was objectively measurable, though it was too small to get noticed by the participants in the current experimental design. Participants were not told that the experiment was actually a comparison between two different playlist generation methods. It is likely that they observed the playlists as coming from one method. In addition, the two methods were alternately presented to the participants. To measure any perceived improvement, it is better to explicitly oppose the methods over time.

It was found that a more than half of the participants would use automatic music compilation, though it is evident that user control should be an essential property of any automatic feature.

4. CONCLUSION

Once music listeners have put time and effort to construct a large personal collection of music, they should be provided with means to organize their music collection to ease selection later on. By generating coherent and varied playlists for different contexts-of-use, PATS can contribute to a new and pleasant interactive means to explore and organize the ample music selection and listening opportunities of a large personal music collection. The automatic (pre-)creation and saving of playlists can also be seen as a way to organize your music collection suited to each possible listening occasion.

Music listeners may use various strategies when choosing music from a wide assortment of songs by inspecting various sources and presentations of information. Knowing on what grounds and in what ways music listeners like to organize and select their music is essential to the making of usable and viable products and services for music listening.

4.1 PATS applications

For demonstration purposes, several research prototype music systems have been implemented that have the PATS functionality inside. We will discuss three of them.

A version of the open source FreeAmp MP3 jukebox player has been extended with the PATS playlist creation feature (see Figure 6). PATS playlists can be generated (by selecting a single song and pressing a single button), adjusted and saved to establish a music organization based on the concept of context-of-use. This player also provides access to a free on-line service for meta-data of CD albums. Interactive forms for the input of additional meta-data information are implemented as well.

A multi-modal interaction style based on a slotmachine metaphor[6] presents songs on four rollers that can be manipulated by a force feedback trackball (see Figure 7). By rolling the trackball laterally, one can hop from one roller to another. By rolling the trackball forwards or backwards, one can manipulate a single roller. A press on the trackball provides spoken information about the music and the playback being toggled on or off. Double-pressing the trackball means adding or removing a song to or from a personally created playlist located at the first, left-most roller. Each time a song on the third roller is at the front, a small PATS playlist is generated on the basis of that single song and shown on the fourth, right-most roller.

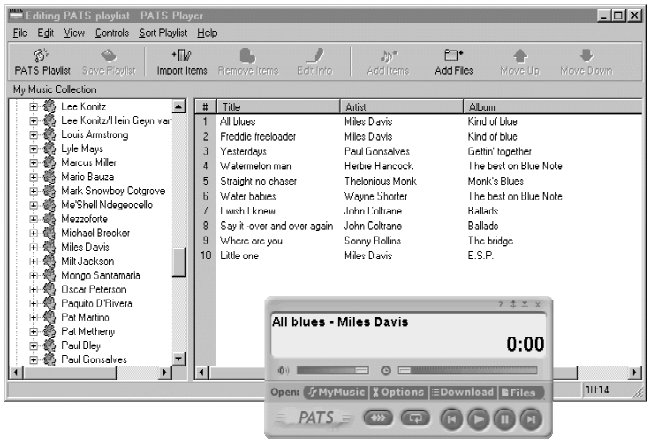

Figure 6. The PATS-enhanced FreeAmp MP3 player.

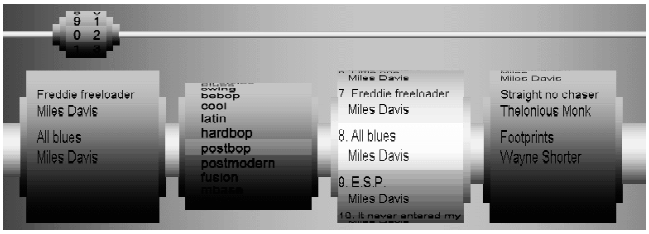

Figure 7. The PATS slotmachine jukebox. The PATS generated playlists are shown on the right-hand roller on the basis of the currently selected song on the high-lighted roller.

A Philips Pronto remote control device with a modified touch screen interface provides direct and remote access to a music server. This server incorporates PATS, essential features for music playback and spoken information feedback about the music by using text-to-speech and language generation from the music meta-database (see Figure 8).

Figure 8. The PATS pronto device.

5. ACKNOWLEDGEMENTS

Thanks go to Dunja Ober for running the experiment and to all participants in the experiment.

6. REFERENCES

[1] Abeles, H.F. (1980). Responses to music. In: Hodges, D.A. (Ed.), *Handbook of music psychology*, Lawrence, KS: National Association of Music Therapy, 105-140.

[2] Fayyad, U., and Irani, K. (1990). What should be minimized in a decision tree? In: Dietterich, T., and Swartout, W. (Eds.), *Proceedings of the Eighth National Conference on Artificial Intelligence, Volume 2, AAAI-90, Boston, Massachussets, USA, July 29 – August 3, 1990*, Menlo Park: AAAI Press / MIT Press, 749-754.

[3] Furman, C.E., and Duke, R.A. (1988). Effects of majority consensus on preferences for recorded orchestral and popular music. *Journal of Research in Music Education*, 36, 4, 220-231.

[4] Geringer, J.M. (1982). Verbal and operant music listening in relationship to age and musical training, *Psychology of music (special issue)*, 47-50.

[5] Holbrook, M.B., and Schindler, R.M. (1989). Some exploratory findings on the development of musical tastes. *Journal of Consumer Research, 20*, 119-124.

[6] Pauws, S., Bouwhuis, D., and Eggen, B. (2000). Programming and enjoying music with your eyes closed. In: Turner, T., Szwillus, G., Czerwinski, M., and Paterno, F (Eds.), *CHI 2000 Conference Proceedings*, 1-6 April, 2000, the Hague, the Netherlands, 376-383.

[7] Pauws, S.C., and Eggen, J.H. (1996). New functionality for accessing digital media: Personalised Automatic Track Selection. In: Blandford, A., and Timbleby, H., (Eds.), *HCI'96, Industry day & Adjunct Proceedings*, London, UK, August 20-23, 1996, London: Middlesex University, 127-133.

[8] Peynircioglu, Z.F., Tekcan, A.I., Wagner. J.L., Baxter, T.L., and Shaffer, S.D. (1998). Name or hum that tune: Feeling of knowing for music. *Memory & Cognition, 26, 6,* 1131-1137.

[9] Quinlan, J.R. (1986). Induction of decision trees. *Machine Learning, 1*, 81-106.

[10] North, A.C., and Hargreaves, D.J. (1996). Situational influences on reported musical preferences. *Psychomusicology, 15*, 30-45.

[11] Rubin, D.C., Rahhal, T.A., and Poon, L.W. (1998). Things learned in early adulthood are remembered best. *Memory & Cognition, 26, 1*, 3-19.

[12] Spangler, S., Fayyad, A.M., and Uthurusamy, R. (1989). Induction of decision trees from inconclusive data. In: Segre, A.M. (Ed.), *Proceedings of the Sixth International Workshop on Machine Learning, Ithaca, New York, USA, June 26-27, 1989*, San Mateo, CA: Morgan Kaufmann, 146-150.

[13] Tversky, A. (1972). Elimination by aspects: a theory of choice. *Psychological Review, 76*, 31-48.