USER EVALUATION OF A NEW INTERACTIVE PLAYLIST GENERATION CONCEPT

Steffen Pauws
Philips Research
Prof. Holstlaan 4
5656 AA Eindhoven, the Netherlands
steffen.pauws@philips.com

Sander van de Wijdeven
Philips Research
Prof. Holstlaan 4
5656 AA Eindhoven, the Netherlands
sander.van.de.wijdeven@philips.com

ABSTRACT

Selecting the 'right' songs and putting them in the 'right' order are key to a great music listening or dance experience. 'SatisFly' is an interactive playlist generation system in which the user can tell what kind of songs should be contained in what order in the playlist, while she navigates through the music collection. The system uses constraint satisfaction to generate a playlist that meets all user wishes. In a user evaluation, it was found that users created high-quality playlists in a swift way and with little effort using the system, while still having complete control on their music choices. The novel interactive way of creating a playlist, while browsing through the music collection, was highly appreciated. Ease of navigation through a music collection is still an issue that needs further attention.

Keywords: playlist generation, user evaluation, constraint satisfaction.

1 INTRODUCTION

Playlist creation ranges from the laborious variant of having to select each song one-by-one to the ease of random/shuf e play and one-click playlist generation. In the latter method, a user only has to indicate a single 'seed' song and gets a complete playlist with additional songs in return by a single button press. In this paper, we describe the working and evaluation of the 'SatisFly' system that allows a user to select songs one-by-one, to ask for additional 'similar' songs based on a referent song, and to specify additional requirements that the playlist should meet.

2 SATISFLY SYSTEM

'SatisFly' is a software system for automatic playlist generation. A user speci es requirements what kind of songs should be in the playlist and what kind of songs should not. The system then uses constraint satisfaction to arrive at a list of songs that meets these requirements. The input modality of the system is a conventional remote control using the cursor keys, the 'ok' button, and the color keys. Its output modality is a visual display.

Figure 1: SatisFly playlist generation system.

Users can browse through the music collection using a two-panel visualisation, as shown in Figure 1. Movement through the panels is done by using the cursor keys on the remote control. The left-hand panel provides the current choices for the user. The right-hand panel displays the consequences of a choice. Selecting an item can be done by pressing the 'ok' button. Color keys can be used to invoke functions; in Figure 1, the user can invoke a 'reset', a 'clear' and a 'generate' function. While navigating and listening, songs can be added to or removed from a playlist. In addition, a user can select playlist requirements on, for instance,

- the number of songs or duration of a playlist,
- the variety in genres, artists, and albums,
- tempo and period of release of the songs, and
- the similarity of songs.

To this end, the system uses a database with attribute information about each song including song title, artist, album, genre, duration, year of release, and tempo.

Users can select and alter almost any combination of requirements. By pressing a single button, the system generates a playlist satisfying the current set of requirements. At all times, the content and song order of the playlist can be changed manually, leaving complete control in the hands of the user.

2.1 Constraint satisfaction

In a constraint satisfaction approach (Tsang, 1993), the playlist requirements are modelled as logical constraints of song attributes (e.g., artist name, genre, tempo) de ned over playlist positions. Each constraint limits the combinations of songs that are allowed in the playlist. Constraints can be distinguished by the number of playlist positions on which they are de ned: *unary*, *binary*, and *global* constraints. Unary constraints restrict the songs that are allowed to occur at a single playlist position. An example is that the rst song should be of a particular artist. Binary constraints represent a binary relation that has to be met between songs at two (successive) playlist positions. For instance, two successive songs should have the same tempo. Finally, global constraints are de ned on any number of positions; they can represent a set of unary or binary constraints. For instance, if we want to bound the number of occurrences of particular attribute values in a playlist, we can instantiate a *counting* constraints. Using this constraint, we can declare that we want, say, 4 to 6 Rock songs in a playlist of 10 songs, or at most, say, 3 songs of 'Prince' or 'Michael Jackson'. In the same vein, constraints are de ned for sorting songs in a playlist, for ensuring the similarity of successive songs, total duration of the playlist, etc.

When generating a playlist, songs are assigned to playlist positions in a constructive search method while guaranteeing that all constraints will be satis ed. The search method consists of *constraint propagation*, *construction*, and *backtracking* that are applied until either a complete and consistent playlist has been found or it is aborted.

Constraint propagation is the set of techniques aimed at reducing the search space by eliminating songs from which it can be determined that they can not be part of a playlist that meets all constraints. For instance, if we know that we only want Rock songs with a given tempo range to appear in a part of the playlist, we can leave out all songs that do not t this description from further consideration.

For construction, playlist positions are addressed one-by-one using the *fail- rst principle*: positions are addressed rst for which the smallest number of songs is available. For each considered position, songs are chosen based on a *constraint voting principle*: only those songs are tried from which it can be computed that most constraints will be satis ed.

If no songs can be found for the current position without violating any of the constraints, a dead end in the search space has been reached. A backtracking procedure is then required that changes the song assignment at a previous position; we use a chronological variant of backtracking. If no backtrack is possible, all possible song assignments have been evaluated without success. In other words, there is no complete and consistent solution. Fortunately, it is possible to keep track of the best possible partial solution, that is, the longest partial playlist for which all constraints are still satis ed. As an expedient for completing a playlist to its required length, we add randomly selected songs that have not been eliminated during previous constraint propagation steps.

One of the advantages of constraint satisfaction is that it strives for exact solutions by constructive search and, hence, provides means to detect that no playlist exist while performing the search process.

2.2 Related work

Without proof, we state that automatic playlist generation in the current de nition is a NP-hard problem. It is thus unlikely that a polynomial algorithm exists that computes a playlist that meets any given set of constraints.

Literature presents several approaches for the automatic playlist generation problem. Alghoniemy and Tew k (2001) formulated the problem as an integer linear programming (ILP) problem and used a standard ILP solver. This is not a time-ef cient method and hence not practical. Constraint satisfaction techniques have been also used by others (Pachet, Roy, and Cazaly, 2000), which are less inef cient than integer programming. For further scalability, local search has been proposed and realized (Aucouturier and Pachet, 2002). Unfortunately, the methods were not paired with a thorough evaluation and application to prospective users to assess the performance and user bene ts of the methods.

3 USER TEST

The user test assessed user task performance, perceived ease-of-use and usefulness, and user preference of the 'SatisFly' playlist creation system in comparison with a control system. In contrast to 'SatisFly', The control system did not constructively meet the requirements related to what genres, artists, and albums should be present in the playlist; it made use of a random selection process for addressing these requirements. However, it did meet all other wishes, for instance those related to time period/tempo range selection and ordering. Note that the user interface of both systems were xed. Test participants were asked to create a playlist using both systems twice (i.e., at two trials) for a xed, personally imagined music listening situation.

3.1 Hypotheses

Participants are given ample time for making a preferred playlist; it is expected that the quality of the playlist does not differ under various experimental conditions. However, we expect that less time and fewer actions are required to make a playlist when using the 'SatisFly' system than when using the control system. In addition, we

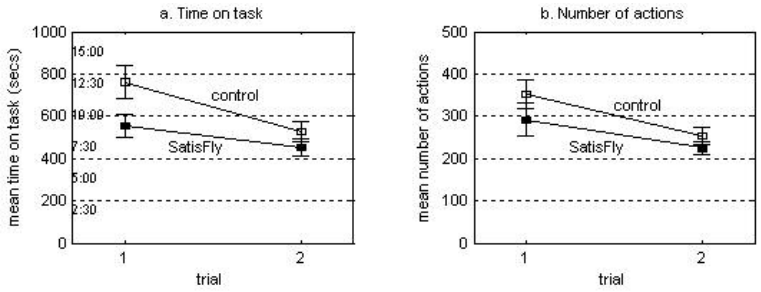

Figure 2: Panels (a) and (b) show respectively mean *time on task* and mean *number of actions* across systems and trials. Cross-bars represent standard error.

expect less time and fewer actions are required to make a playlist at the second trial.

Because of its time- and effort-saving, we expect that the 'SatisFly' system will be valued as more useful and more usable, and hence more preferred, than the control system.

3.2 Participants

Twenty-four persons (14 m, 10 f, avg age: 29 yrs) participated voluntarily during normal working time. They were all colleagues or students of the research laboratory. All participants were frequent listeners to popular rock music. All participants had completed higher vocational education.

3.3 Design

A factorial within-subject design with two independent variables, named *system* and *trial*, was used. The variable *system* referred to the 'SatisFly' and the control system. *Trial* referred to the two task trials, intended to measure changes in performance, user perception, and preference as a result of experience. To compensate for any order effects, participants were randomly assigned to one of the six possible permutations of admissions to the two systems.

3.4 Test equipment and material

A music collection comprising 2248 popular music recordings from 169 CD albums from 111 different artists covering 7 different musical genres released in the period from 1963 to 2001 in MP3 format served as test material. The test equipment consisted of a PC on which the system was running. The display was directed to a Philips MatchLine television set. The remote control was tapped to control both the television set and the PC. The audio was directed to a mid-range audio ampli er and a pair of hi- loud speakers. Participants were seated in a comfortable chair in front of the television set and audio ampli - cation system.

3.5 Procedure

Participants were randomly assigned to an order of system admission in the test. They received ample instruction, practice, and time to master the system under study without need for outside help. For each trial, the system was presented with a different colour for allowing reference in the questionnaires. Obviously, participants were not told about the nature of the systems. Different 10-song playlists has to be created over four trials, but representing intentions for the same listening situation. Quality of the playlist was presented as the sole optimisation criterion. After each trial, participants completed a questionnaire. At the end of the test, participants ranked the systems according to their preference of use. Subsequently, they were asked to rate the playlist on a 0-10 scale and to indicate what songs in the playlist did not t the intended listening situation, after second listening.

3.6 Measures

3.6.1 Playlist quality

Playlist quality was measured by *precision* and a *rating score*. *Precision* was de ned as the proportion of participant indicated preferred songs in a playlist of 10 songs. The *rating score* was a participant's rating on a scale ranging from 0 to 10 (0 = extremely bad, $\cdots$, 10 = excellent).

3.6.2 Task performance

Task performance was measured by *time on task* and *number of actions*. *Time on task* measured the time elapsed from the participant performing the rst button press to the participant performing the last button press. *Number of actions* measured the number of button presses on the remote control that were performed by the participant.

3.6.3 Perceived ease of use and perceived usefulness

An (adapted) version of the Technology Acceptance Model (TAM) questionnaire (Davis, 1989) assessed perceived ease of use and perceived usefulness. Participants responded by stating to what extent they agreed with a statement in the questionnaire on a 7-point scale.

Statements assessing perceived ease of use were the following:

Q1. I find learning how to use the system easy.

Q2. I find it easy to get the system to do what I want it to do.

Q3. I find it easy to become skilful at using the system.

Q4. I find the system easy to use.

Statements assessing perceived usefulness were the following:

Q5. I find that by using the system I can make good playlists.

Q6. I find that by using the system I am able to create a playlist rapidly.

Q7. I find that by using the system I enjoy the making of a playlist.

Q8. I find this system useful at home.

3.6.4 Order of preference

Order of preference of the systems was assessed by having participants the systems rank from 1 to 4 according to their preference. Rank value 1 was assigned to the most preferred system. Indecisions resulting into ties in the ranking were treated as equal preference for the systems involved; their joint rank value was the mean of rank values that they would be assigned to.

3.7 Results

All analyses of variance (MANOVA) were conducted with repeated measures and with *system* and *trial* as within-subject independent variables.

3.7.1 Playlist quality

With *precision* as dependent variable, a main effect for *trial* was found to be signi cant ($F(1,23) = 5.24$, $p ¡ 0.05$). On average, playlists created at the second trial contained half a preferred song more than the playlists created at the rst trial (mean *precision*: 0.83 (trial 1), 0.88 (trial 2)). With *rating score* as dependent variable, no effects were found to be signi cant. Participants rated their playlists consistently. The mean *rating score* for a playlist was 7.5.

3.7.2 Task performance

The results on *time on task* are shown in the left-hand panel (a) of Figure 2.

With *time on task* as dependent variable, a main effect for *system* was found to be signi cant ($F(1,23) = 13.78$, $p ¡ 0.001$). Making a playlist with the 'SatisFly' system took 505 seconds (8:25), on average, which was faster than with the control system which took 646 seconds (10:45).

A main effect for *trial* was found to be signi cant ($F(1,23) = 15.46$, $p ¡ 0.001$). Making a playlist for the rst time, which was 658 seconds (10:58), took more time than for the second time, which was 491 seconds (8:11). No other effects were found to be signi cant.

The results on *number of actions* are shown in the right-hand panel (b) of Figure 2. A main effect for *system* was found to be signi cant ($F(1,23) = 4.59$, $p ¡ 0.05$). Participants performed 258 actions, on average, when using the 'SatisFly' system, which was a fewer number than when using the control system (302 actions).

A main effect for *trial* was found to be signi cant ($F(1,23) = 9.87$, $p ¡ 0.01$). Participants performed more actions when working with the systems for the rst time (322 actions) than for the second time (239 actions). No other effects were found to be signi cant.

3.7.3 Perceived ease of use and usability

Responses to the adapted TAM questionnaire were subjected to a two-dimensional non-linear principal component analysis. The eight items in the questionnaire were treated as active variables and the two different systems over two trials were treated as passive variables to label the plot (i.e., SatisFly 1, SatisFly 2, control 1, control 2). The responses were treated as ordinal categories.

The visualisation of the PCA solution of the TAM questionnaire is shown in Figure 3.7.3. It displays the mean transformed item responses related to the two different systems over two trials. Also, the mean scores to the eight questionnaire items (i.e., Q1 to Q8) are displayed. The dashed lines go through the origin and the mean scores of each group of items. These lines represent the 'mean' axes along which the transformed ordinal response categories of the items (i.e., the 7-point scale of the questionnaire) are located.

A rst observation of Figure 3.7.3 tells us that the 'SatisFly' systems and the control systems are positioned at either side of the origin. However, a regression to the mean (i.e., the origin) over trials is clearly visible. Item responses were more discriminatory after working with the two different systems for the rst time (i.e., SatisFly 1, control 1), but responses were more similar after working with the two different systems for the second time (i.e., SatisFly 2, control 2).

The scores for the items Q1, Q2, Q3, and Q4 are highly correlated as well as the scores for the items Q5, Q6, Q7, and Q8, though items Q2 and Q6 are speculative. Nevertheless, the high correlations mean that the two sets of four items load on different factors that can be labelled as 'perceived ease of use' and 'perceived usefulness'. As shown in Figure 3.7.3, both groups of correlated items are displayed as clusters while their response categories are best displayed as two almost orthogonal axes (the dashed lines). In this way, the upper right-hand corner and the rst quadrant represents high 'perceived ease of use', and the lower right-hand corner of the fourth quadrant represents high 'perceived usefulness'.

The visualisation of the TAM solution suggests that working for the rst time with the 'SatisFly' system provided the highest perceived usefulness. This dropped when working for the second time with it. The usefulness of both control systems was perceived lower than both 'SatisFly' systems but it did not change over trials. It also suggests that working for the rst time with the control system provided the lowest perceived ease of use. This improved when working for the second time with it. Both 'SatisFly' systems were perceived as easier to use than the control systems.

3.7.4 Order of preference

Participants were asked to rank the combination of a system in a trial according to their preference. Rank value 1 was assigned to the most preferred combination; similar ranking of combinations was allowed. Fifteen (out of 24) participants ranked a 'SatisFly' as their most preferred system. Five participants ranked a control system as their most preferred system. Four participants ranked either a 'SatisFly' or a control system as their preferred one.

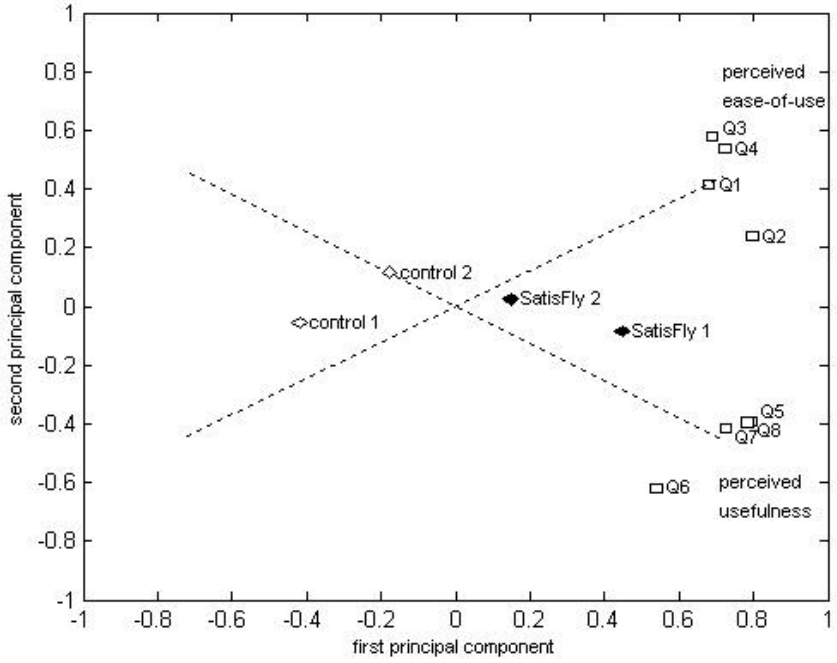

Figure 3: The non-linear principal component solution for the eight items of the TAM questionnaire loaded on the terms 'perceived ease-of-use' and 'perceived usefulness'.

For a more detailed analysis, the ranking data can also be used to indicate comparative judgements of all pairs of systems. The task of ranking four systems requires, in essence, the comparisons of six pairs of systems to tell their relative preferences. Ties in the ranking were treated as equal preference of the systems involved. Using this mode of thought, we can determine the proportion of the time that a system is more preferred than any other system. These proportions are shown in Table 1.

Table 1: Proportion of times that a system in a trial at the top was chosen over a system in a trial at the side.

	SatisFly 1	SatisFly 2	control 1
SatisFly 2	14 / 24		
control 1	20 / 24	15.5 / 24	
control 2	18 / 24	15 / 24	8.5 / 24

The standard way to analyse pair-comparison data is based on Thurstone's law of comparative judgment. In our context, this law assumes that a mean psychological value is attached to each system by users. Now, the extent to which one system is judged to be more preferred than another is related to the difference in these values of the compared systems. We refer to these psychological values as scale values. To go from proportional data to scale value in a least-squares problem sense, we refer to Guilford (1954).

By setting the scale value of system 'control 1' to zero (which happened to be the least preferred system), the least-squares solution of the over-determined set of equations yields the scale value estimates as shown in Table 2. The standard error of the estimates was 0.18. The correlation between the observed z-scores and the predicted z-score (from the least-squares solution) is high (r = 0.941) which means that 88.7% of the variance is explained.

Table 2: Scale values of the four systems.

SatisFly 1	0.90	control 2	0.30
SatisFly 2	0.56	control 1	0.00

The scale values in Table 2 shows that the combinations of systems and trials can be ordered according to their preference. Participants had an overall preference for the 'SatisFly' system used at the first trial (i.e., SatisFly 1), followed by the same system at the second trial (i.e., SatisFly 2). They had the least preference for the control system that they had used in the first trial (i.e., control 1).

4 DISCUSSION

When using the 'SatisFly' system, participants needed 2 minutes and 20 seconds less time and 44 fewer actions to create a playlist than using the control system. This was all done without any decrease in quality of the playlist being created. Thus, 'SatisFly' enabled participants to create their preferred playlist in less time and fewer actions.

The test found out that participants needed almost 3 minutes (167 seconds) less time and 83 fewer actions to make a playlist at the second trial. This was all done without any decrease in quality of the playlist being created. Thus, learnability of the systems was less of a issue; in

short time, participants became skilful in creating their preferred playlist.

The TAM questionnaire indicated that the 'SatisFly' system was perceived most useful, especially when used for the first time, and that the control system was perceived least easy-to-use, especially when used for the first time.

The task to order the systems on preference found out that the 'SatisFly' system was chosen over the control system in both trials. It was remarkable that participants were consistent in finding that working with one system for the first time is different from working with the same system for the second time.

5 CONCLUSION

Easy-to-use tools to pick out the 'right' songs and to put them in the 'right' order from a daunting volume of music are attractive features of music players. Most participants (16/24) stated explicitly their appreciation of the novel way of playlist creation by selection and generation as demonstrated by the 'SatisFly' concept.

The 'SatisFly' system enables users to create high-quality playlists in a swift way and with little effort, while having still complete control on their music choices. In the test, participants needed more than 8 minutes and about 250 button presses on the remote control to create a playlist. We expect that users will spend less than 8 minutes when selecting music from their personal music collection. The large number of actions required is definitely an issue, as it is mainly caused by repetitive navigation behaviour such as going through long lists or switching between panels. In general, designs of navigation structures should focus on minimization of number of actions required.

Observations during the test made clear that the behaviour of users is not uniform. Users differ in degree; some need more than 20 minutes to select 10 songs. Others spend only 2 minutes. Users also differ in essence; some are precise in formulating their music preferences. Others just pick some songs or let the system generate some random songs. Yet others select all songs one-by-one and use the system only to order these songs on tempo or year. It might be obvious that the way in which users use an interactive system determines how they will appreciate the system.

Constraint satisfaction tries to find an exact solution by meeting all constraints in a playlist generation problem. Another way is to find only an approximate solution, which solves issues on feasibility, scalability and running time with respect to longer playlists and larger music collections. We used a simulated annealing approach to solve the problem approximately. Findings in a user evaluation made clear that playlists generated by this approximating method better reflect a set of constraints than the playlists generated by constraint satisfaction.

ACKNOWLEDGEMENTS

We thank our colleagues Vincent Buil, Gerard Hollemans, Fabio Vignoli, and all participants in the test.

References

Alghoniemy, M., and Tewfik, A.H. (2001). A network flow model for playlist generation. *Proc. ICME 2001 Japan, Aug. 2001.*

Aucouturier, M., and Pachet, F. (2002). Scaling up music playlist generation. In: *Proceedings ICME 2002 - IEEE International Conference on Multimedia and Expo,* 26-29 August 2002, Swiss Federal Institute of Technology, Lausanne, Switzerland.

Davis, F.D. (1989). Perceived usefulness, perceived ease of use, and user acceptance of information technology. *Management Information Science Quarterly, 18*, 189-211.

Guilford, J.P. (1954). *Psychometric methods, Second edition.* New York: McGraw-Hill.

Pachet, F, Roy. P., and Cazaly, D. (2000). A combinatorial approach to content-based music selection, *IEEE Multimedia, 7, 1*, March 2000, 44-51.

Tsang, E. (1993). *Foundations of constraint satisfaction.* Academic Press, 1993.