Music and choice : adaptive systems and multimodal interaction

DOI:
10.6100/IR528621

Document status and date:
Published: 01/01/2000

Document Version:
Publisher's PDF, also known as Version of Record (includes final page, issue and volume numbers)

MUSIC AND CHOICE: ADAPTIVE SYSTEMS AND MULTIMODAL INTERACTION

The work described in this thesis has been carried out in two research projects at IPO, Center for User-System Interaction, Eindhoven, the Netherlands, from November 1994 to October 1998. Writing up took more than one extra year. Part of the software implementation has been done as a visiting researcher in a five-month period at Philips Research Laboratories, Eindhoven, the Netherlands. The projects were known and administered under several names including 'Accessing large amounts of information in multimedia applications for home entertainment environments', 'Turn on the Base (ToB)' and 'Adaptive Multimodal Interaction (AMI)'. The projects were substantially funded by Philips Research Laboratories, Eindhoven, the Netherlands.

Cover design by Steffen Pauws, Henk Korteweg and Ben Mobach.
Printed by UniversiteitsDrukkerij, Eindhoven University of Technology.

CIP-DATA LIBRARY TECHNISCHE UNIVERSITEIT EINDHOVEN

Pauws, Steffen Clarence

Music and choice: Adaptive systems and multimodal interaction / by Steffen Clarence Pauws. - Eindhoven: Technische Universiteit Eindhoven, 2000.
Proefschrift. - ISBN 90-386-1441-1
NUGI 859
Trefw.: kunstmatige intelligentie / gebruikersinterfaces / audio-apparatuur
Subject headings: artificial intelligence / user interfaces / audio equipment

MUSIC AND CHOICE:
ADAPTIVE SYSTEMS AND
MULTIMODAL INTERACTION

PROEFSCHRIFT

ter verkrijging van de graad van doctor
aan de Technische Universiteit Eindhoven,
op gezag van de Rector Magnificus, prof. dr. M. Rem,
voor een commissie aangewezen door het College voor Promoties
in het openbaar te verdedigen op
woensdag 12 januari 2000 om 16.00 uur

door

Steffen Clarence Pauws

geboren te Amsterdam

Dit proefschrift is goedgekeurd door de promotoren:

prof.dr. D.G. Bouwhuis
en
prof.dr. P.M.E. de Bra

Copromotor:
dr.ir. J.H. Eggen

VOOR ELINE.

Dankbetuidingen / Acknowledgements

Op deze plaats wil een groot aantal mensen bedanken die het mogelijk maakten mijn onderzoek te doen en dit boekje te schrijven.

Mijn grootste dank heb ik gereserveerd voor Eline Stumpel, mijn liefde. Ze heeft me ondersteund, aangemoedigd en al haar liefde gegeven om dit boekje af te kunnen ronden. Zonder jou was het nooit gelukt. Ik heb je vreselijk gemist. Dit boekje is voor jou, Eline.

Graag wil ik Berry Eggen bedanken, ‘een fantastische vent’. Hij is de grote inspirator en dagelijkse begeleider van het project geweest. Het was allemaal zijn geweldige idee om PATS te realiseren en om multimodaal te denken.

Ik wil ook Don Bouwhuis bedanken voor zijn constructieve begeleiding, de cognitieve noot, het willen ordenen van mijn gedachten op papier, zijn humor en zijn antipathie voor jazz (‘ook wel djazz, jass of jas genoemd’).

I am very grateful to Peter Wavish for his RTA support overseas and for his elaborate and immediate replies on my numerous e-mails regarding RTA and situated agents.

Bedankt, Dunja Ober, Sitan van Sluis en René Scheffer voor jullie werk dat een belangrijke bijdrage was voor mijn onderzoek.

Ook wil ik twee van mijn vrienden niet vergeten alvast te bedanken voor hun paranimf-willen-zijn: Arjen te Pas en Henk Korteweg. Mijn verontschuldigingen gaan uit naar Arjen voor het moeten werken tijdens kerst en het millennium door de verschuiving van de datum. Dat wordt weer goedmaken aan de bar. Mijn speciale dank gaat uit naar Henk voor het vinden van de juiste modellering, het juiste perspectief en de juiste belichting voor de PATS-enhanced platenwisselaar, de bal met strook en de knikkerbaan.

Ik dank ook de heren waarmee ik de werkkamer heb mogen delen: Martijn Ceelen, Bart Jonkers, Leo Poll, Bert Bongers, Paul Kaufholz en Martijn Willemsen (in chronologische volgorde van kamerdelen). Ik vond het in ieder geval groots en inspirerend.

Mijn fantastische ouders en zus: Jos, Til en Serifa. Ik wil jullie bedanken voor jullie ondersteuning, aanmoedigingen en de liefde voor muziek.

I would like to thank Rachel Hoekstra for correcting my English, even one day before I sent the manuscript to the printer’s.

Natuurlijk ook al mijn proefpersonen. Hartelijk dank. Zonder jullie had ik geen waar woord kunnen schrijven. En alle IPO mensen. Bedankt!

Contents

CHAPTER 1

A profusion of music

Recent technological developments have provided music listeners with ample opportunities to select and listen to an ever-increasing number of music recordings. As a result, future interactive music players should not be touted as mere playing devices, but rather as a way of easily organising, selecting and programming music. To achieve this, these players must meet user requirements on *adaptation to the user's music choice behaviour* and *interactive information presentation*. Both requirements are often mentioned in the context of *personalising* interactive devices. It is apparent that existing CD players do not address these requirements adequately. CD jukebox players, in particular, are full of opportunities for human error, thus limiting the full experience of music selection and listening over many CDs. Among other things, it appears that convenient use of current players for music selection and programming is limited by an inadequate visual display of relevant information, an awkward method of referring to music by numbers, and a restricted set of programme editing features. None of the players have properties that are intended to adapt to the music listener. In addition, CD player users commented in a questionnaire that they do not use music programming features because they are too laborious and too time-consuming. Nevertheless, they commented that they value direct access to a large music collection, if this is done in an 'intelligent' way. As will be briefly discussed in this chapter, the central theme of this thesis is, for these reasons, how future music players can be of help to music listeners in finding preferred music in large personal music collections.

Music shapes everyday life for many people as music can be heard everywhere and at any time. People develop individual tastes in music during their life time. For instance, the more you listened to music that was popular in your adolescence and early adulthood, the more you like it now (Holbrook and Schindler, 1989; Rubin, Rahhal, and Poon, 1998). In addition, the more you listen to an unknown piece of music, the more you like it (Peretz, Gaudreau, and Bonnel, 1998). Even so, people are sometimes said to have conservative musical tastes.

Listening to music has become one of the most widely indulged leisure activities, including visits to concerts and listening to recorded music at home. As an example, in 1953, Virgil Thomson (1967), a music critic for the New York Herald Tribune, wrote that twice as many people attended classical music concerts in America compared to those who spectated at America's most popular sports game, major league baseball. Since the advent of the gramophone in 1893, the value of listening to a wide variety of recorded music was already recognised, eventually resulting in a mass production of players and music recordings (Gronow, 1989).

The recent rapid technological developments have led to an unprecedented availability of recorded music; it may be evident that the current supply of recorded music largely exceeds the demand for recorded music. In contrast, the first record catalogue in 1890 comprised only a single-page list of phonograph and graphophone cylinders (Schoennherr, 1999). Currently, recorded music is available on numerous media such as Compact Discs (CDs), audio cassettes and computer files, and can be bought in numerous ways, for example, at music retail shops, service stations, supermarkets, and by phone, mail, or Internet download and order. The International Federation of the Phonographic Industry (IFPI) reported that a total of 850 million music albums were sold in 1997 in Europe, with an annual growth of 4% (Laing, 1999). In the European Union, the music industry, which makes money by recording, promoting, performing, broadcasting and selling music, has a greater turnover (about 18.8 billion ECUs[1] in 1995) than both the video and cinema industry (Laing, 1996). Music is thus heavily embedded in a world-wide production, distribution and consumption context, and hence plays an important economic role (North and Hargreaves, 1998). But what happens when music listeners have paid to own a private copy of their favourite music and have established a sizeable personal music collection at home?

When at home, it can be generally said that music listeners prefer to listen to music that is chosen to suit their current interests or activities; they want to select music that is 'best' for a specific occasion. To select the 'best' music, music listeners may be repeatedly confronted by the need to find their way through their own collection. Selecting from a wide assortment of music can be difficult and may put so much cognitive load on the listener that it negatively influences enjoyment of the actual content (Eggen, 1995). Interactive music players should therefore support music listeners in finding their favourite music by simplifying and speeding up music selection. These devices should not put the listener off by complicated operation, by any perceived loss of control of music selection, or by an inadequate fit to music listening behaviour. Surprisingly, little attention has yet been paid to the question of how music listening intentions can best be translated into interactive means to support music selection.

This thesis is particularly concerned with music programming and compilation using interactive music players. *Music programming* is defined here as the serial selection of multiple preferred music recordings from a large to very large music collection, in order to prepare a play sequence that can be played in one go. *Music compilation* is a similar task, but is often used to refer to an automatic process carried out by an interactive device. The result of both tasks is either a personally created or a system created *music programme*, that is, the play sequence of music recordings. The term *music preference* is defined here as the temporary liking of a particular music recording which allows the use of the term *preferred music*. A more precise definition of a music preference will be given in Chapter 2. The term *music option* is reserved to refer to individual music recordings in order to emphasise that a music programming task is a choice task. The term *music listener* is used to refer to a person who uses an interactive music player to select and listen to music.

This chapter further discusses the user requirements to be met by future interactive music players and how some of these requirements are addressed in existing CD players. Future music players should not be considered mere playing devices, but rather as a way of easily organising and selecting music. They should therefore meet specific user requirements which are discussed in Section 1.1. As there is a

1. The exchange rate between national currency and the ECU (European Currency Unit) is that pertaining in 1997 (Laing, 1996). One ECU is equal to 2.2108 Dutch Guilders and is approximately one Euro.

general trend of providing an overwhelming supply of information and multimedia content (Chester, 1998), these requirements may also hold for other applications. Next, an overview of currently available single-CD players and their multi-disc versions is given in Section 1.2. The discussion concentrates on the way in which existing players help music listeners to select and program music in relation to the specific user requirements.

Relevant research fields related to the work carried out in this thesis are discussed in Section 1.3. Finally, an overview of the contents of this thesis, including a summary of all chapters, is presented in Section 1.4.

1.1 Requirements for interactive players

First, any interactive device designed for people should be easy to use and easy to learn (Nielsen, 1994). Home usage even requires that the device can immediately be used without any necessities for learning or consulting manuals, that is, *instant usability*.

In order to simplify search and selection in a large music collection, future interactive music players should be primarily focused on two interrelated user requirements, namely *interactive information presentation* and *adaptation to music choice behaviour*. These requirements are considered part of the current trend toward the *personalisation* of interactive devices. All requirements are discussed further below.

1.1.1 Interactive information presentation

Interactive information presentation is mainly concerned with when, how and what information about music is presented by an player to a music listener. Players should present information about music interactively, in such a way that it facilitates user navigation in a large music collection. This means that the information presented should be kept to the minimum needed for an adequate search for music and should be disclosed progressively at appropriate times. To achieve this, interactive information presentation deals essentially with two issues: a *reference model* for the musical content and a *navigation structure* that is imposed on the musical content.

Reference model

The music reference model used by the player should correspond to the way in which music listeners think about and give meaning to music. Therefore, an important constituent of a reference model is the music *attribute representation* used. Musical attributes and their values should allow music listeners to easily recognise, rather than recall relevant features of the recorded music. For instance, do music listeners think of recorded music in terms of the emotional expression of the recording (e.g., tempo, dynamics, contrasts and improvisation), the aesthetics of the recording (e.g., performance practice by musicians and conductors, recording quality), music styles and idioms, musical performers and instruments, or the music composition itself?

Another important constituent of a reference model is the predominant *unit of reference* that is used by the player. This unit should also correspond to the way in which music listeners prefer to refer to recorded music. For instance, do they think of recorded music in terms of physical storage media such as CDs, track numbers of

CDs, computer file names, whole albums, individual recordings, song titles or musical cues?

Navigation structure

A navigation structure defines the interactive opportunities and constraints while searching for music. This structure is implemented by an *interaction style*, which is loosely defined here as the general way in which users communicate or interact with an interactive device. A more formal definition is given elsewhere (de Bruin, de Ruyter, and de Vet, 1996). In our definition, interaction styles include console and remote control operation, direct manipulation, voice-controlled, command-driven, menu-oriented or form-filling interfaces.

An ideal interaction style for a interactive player allows music listeners to freely navigate in a music collection, without losing their bearings or goals. Free navigation means that users can go to any option for listening and information inspection, can correct slips and errors, can exit from unwanted situations, and can cancel actions performed. Thus, user control should be an essential property of the interaction style. In particular, Bouwhuis and Bunt (1994) stressed the importance of the transparency of interaction, meaning that users know where they are, know how they got there, know where they came from, know where they are heading, know the consequences of actions, and know what the alternative courses of action are to proceed.

1.1.2 Adaptation to music choice behaviour

The general aim of *adaptation to the listener's music choice behaviour* is to select music that is of current interest to the music listener. Players should adapt to the listener's typical music listening styles and purposes, so that they can provide adequate assistance to ease and speed up the music selection process by freeing the listener from complex searches and complex operation. To achieve this, a player should acknowledge that music choice follows from the use of a particular choice strategy, in which various music options are evaluated and compared. The use of this strategy is guided by choice criteria presumably pertaining to attributes of the music. This means that a valid attribute representation of music is also an important constituent for adaptation. If these criteria are known, a player can pre-select, order or recommend relevant music. However, these criteria are generally not known and cannot be observed. They have to be inferred either from explicit user feedback or from other less explicit measuring techniques. For instance, by monitoring the listener's likes and dislikes, a player can infer what options are likely to be relevant in the perception of the listener. In this way, music selection becomes a shared responsibility for both the player and the music listener.

1.1.3 Personalisation

All kinds of adaptive properties of interactive devices are often grouped in the term *personalisation*. Adaptation to choice behaviour and interactive information presentation can be interpreted as aspects of personalisation, though the use of the term also includes the automatic adjustment of interaction style features based on a user's typical actions or procedures. The general aim of personalisation is to provide only those items, that information and those interaction style features that are currently needed by the user, without requiring the user to search for items, to look for the appropriate information and format or to customise the interaction style.

1.2 Current CD players and their use

Though the CD was first judged rather sceptically at its advent in 1982, CD players are currently an indispensable and affordable component of any home and car audio hi-fi set. The IFPI estimated that in Northern and Western Europe almost three quarters of the households (or in some cases more than three quarters) have at least one CD player of some type. In 1997, every Dutch household owned one-and-a-half CD players, on average (Laing, 1999). When we consider this large market penetration of the CD player, its diverse user population and home usage, we can see that skills required to operate a player should be low.

Besides single-CD players and their portable equivalents, there are also CD changers that can hold five to ten CDs at a time, CD jukebox players that have a storage capacity varying between 50 and 301 CDs, and all kinds of solutions in between.

All types of CD players are frequently compared in product tests (Consumer Reports, 1991; 1997; 1999; Kumin, 1994; Pohlmann, 1995). From the results of these tests, we can work out which player features are designed to support music selection and programming. Some of these features of single-CD players and CD jukebox players are discussed in Section 1.2.1 and 1.2.2. One observation was that none of the analysed players have properties that are intended to automatically adapt to typical music selection and listening behaviour of the listener. This feature is therefore not further discussed in the treatment of the players. In addition, characteristics of use of some player features were elicited from users using a questionnaire. The results and discussion of this questionnaire are presented in Section 1.2.3.

1.2.1 CD player

Control elements and visual display

Standard features on current players include control elements on the console to choose, play, pause, stop, skip, repeat and program, and to review selected recordings on a CD. Programming features are called Favourite Track Selection (FTS), program play or select play (for an inventory of some basic functions of audio players see Eggen and Haakma, 1992). A remote control duplicates or extends nearly all the console control elements. A numeric keypad (or sometimes a joggle-dial) enables track numbers to be entered directly. Generally, track numbers directly correspond to physical locations of music recordings on the CD. The technical term ’track’ has become an established expression to refer to a piece of music found on a sound carrier (Eggen, Haakma and Westerink, 1996), though audio may run continuously across multiple CD tracks, or multiple short audio clips may be indexed on one and the same CD track.

Less standard features include previewing the first seconds of CD tracks, (intro-scan), continuous replaying an audio fragment (AB-repeat), playing tracks in a random order (random, shuffle or aselect play), skipping tracks in a playing sequence (delete-track), and taping features (peak level search, auto space, edit) allowing music to be copied and fitted neatly from CD onto tape.

The only source of visual feedback is generally presented on a small console display, whereas operation can take place on the console as well as by using the remote control, that is, from a large viewing distance. The console display shows the track number being played, the elapsed playing time or remaining playing time, and indicates what tracks on the CD are already played or will be played in a so-called ‘music calendar’ format. Textual information about the music is not shown.

Music programming

Programming on a CD player comes down to selecting those musical recordings on a CD, which one wants to hear, and setting them in a particular play order. By selecting the desired recordings, a sequence holding up to 32 selections can be built up, one recording at a time. Note that programming across multiple CDs requires the additional specification of a CD from which the music recordings are to be selected.

A de facto interaction style for music programming is initiated by pressing a ’program’ key on the remote control or console. Then, track numbers of musical recordings have to be entered using numeric keys. Programming is terminated by pressing the same ’program’ key. During programming, no direct music feedback is given, unless additional actions are performed. Another common way to program music is, therefore, to use the intro-scan feature, which gives a preview of the first seconds of a track, and then to press the ’program’ key to add the track to the programme. A review feature allows you to check the programme, though the programme can not usually be altered without starting all over again. Many CD players can only keep one or a very limited number of programmes in memory for the CD that is currently loaded, so the programming process has to be repeated every time the CD is swapped.

1.2.2 CD jukebox

Besides a mere playing device, a CD jukebox is also an interactive means to store and manage an entire CD music collection electronically, and to play hours of music continuously. Most jukebox models consist of two units: a control unit and a cabinet unit which holds the play mechanism and CD storage. The control unit fits into an audio hi-fi rack and the cabinet can be placed out of sight. If a collection exceeds the maximum storage capacity, some models can be extended by one or two additional cabinet units. However, maximum storage capacity is currently limited to 301 CDs: 300 CDs in the cabinets and one CD in an additional drawer. Obviously, loading and cataloguing a CD collection is a time-consuming task.

CD jukebox players have many features that are intended to help the user to select and program music. As will become clear, many of these features only provide opportunities for human error and inconvenient operation.

Control elements and visual display

The consoles and remote controls of jukebox players have many control elements, making it more difficult to learn how to operate these devices. The control elements are in general not well-organised and are not shaped or coloured differently per function group. Their label names are also hard to read or hard to understand at first.

Visual displays are too limited to adequately present all information involved in operating a jukebox. On-screen menus are only one attempt to ease operation. In general, jukeboxes have small, low-contrast displays. Information on the display is cluttered and poorly legible in dimly lit conditions or from a large viewing distance.

Music programming

Successful music programming using a jukebox player requires, among other things, an adequate presentation and use of music information, an adequate reference mechanism for music and a useful set of programme editing features.

Some features concerning information presentation are listed below.

> Music listeners can group CDs into between 8 and 15 musical genres, or in categories of artists or user-defined listening moods. This music *categorisation* is intended to focus the scope, for instance, when music listeners select music.
>
> Some models can read and use *textual information* about album names, artist names and song titles, when this data is encoded on the CD. If this data cannot be read from the CD for any reason, album titles can be generally entered using a remote control (with a keypad or jog-wheel) or a PC keyboard. Entering titles is time-consuming and usually restricted to between 8 and 18 letters per album title. Only a few players allow a CD to be searched for the first few letters of an album title. However, programming music by title is not as effective as it seems. Titles are not necessarily learnt together with the music. Consequently, music listeners are less likely to recall a title if they cannot listen to musical cues, when they are not familiar with the lyrics of the song or when the song contains no lyrics (Peynircioglu, Tekcan, Wagner, Baxter, and Shaffer, 1998).
>
> An automatic *index mechanism* for CDs is generally lacking. Some jukeboxes do not register which slots (the physical locations of CDs in the cabinet) are filled or empty. This causes errors. For instance, shuffling and replacing CDs in the cabinets is impossible without invalidating references in existing programmes. Some jukebox manufacturers have, thus, incorrectly assumed that a music listener will never change the contents of the cabinet units once they are fully loaded.

Some features concerning the reference mechanisms used are listed below.

> It is common to refer to CDs and music recordings by using their *slot numbers* and *track numbers*, respectively. Referring to recordings by numbers is awkward as music listeners are notoriously inaccurate in remembering a song as a specific track and slot number, without consulting paper-based mnemonics or learning by heart.
>
> As a music listener may forget which CD is in which slot, some jukeboxes are provided with a separate slot-numbered '*photobook*' album in which CD booklets can be mounted. Though a 'photobook' is a valuable mnemonic, it does not provide direct access to the music.
>
> CD jukeboxes adhere to the concept of a CD as the predominant *unit of reference*. For instance, only CD albums can be catalogued and selecting a CD is often prerequisite to selecting a music recording. As the CDs are physically hidden from the music listeners, it is likely that they become less familiar with the CDs and their contents. In addition, music programming usually revolves around individual recordings, which suggests that an individual music recording is a better unit of reference.

Some features concerning programme editing facilities are listed below.

> *Modifying* existing programmes incrementally is not a standard feature of jukeboxes; an entire programme sequence has to be entered at one time, while erasing a previous programme.
>
> Music programming is limited by apparent *memory limitations*. The number of programmes that can be saved (e.g., 36) and the size of a programme (e.g., 32) are restricted to a number that may be low in comparison to the amount of music available.

1.2.3 Questionnaire on CD player use

A questionnaire was designed to find out what CD player owners (users) regard as the characteristics of use of some CD player features, and what they want and need from CD player features.

Questionnaire

The questionnaire consisted of five questions. The first four questions contained fixed-response categories; the last one was an open question. In the first two questions, participants were asked about what type of CD player they owned (single-CD, changer or jukebox player) and to categorise the size of their personal CD music collection. In the next two questions, participants were asked how frequently they used programming and random play features on their player and the reason why. The last question asked what kind of features participants lacked on their player. The complete description of the questions can be found in Section III.1 of Appendix III.

Some questionnaires were completed in dialogue with the experimenter by using a structured interview; most questionnaires were completed by the users on their own.

Users

In total, 72 CD player users (59 males, 13 females) were asked to complete a questionnaire. The users also participated in experiments on music programming, reported in this thesis. The questionnaires were handed out at the start of the first experimental session. At that point, they were not informed about the purpose of the experiments. The average age of the users was 30 (min: 19, max: 57). Of the 72 users, 26 were graduate students and 8 were PhD students. Most of the other 38 people had an occupation. All users, except four, had at least completed a higher vocational education.

Results

Of the 72 users, 70 users indicated that they owned a single-CD player including portable or car versions. Two users owned a CD changer, and none of the users owned a CD jukebox player. Almost half of the users indicated that they owned a CD music collection containing less than 50 CDs. A quarter of the users owned 50 to 100 CDs. The other users owned more than 100 CDs. Many users commented that they also owned vinyl records and audio cassettes, which were often still played.

Figure 1.1. Frequency of use of program play (a) and random play (b) as responded by 72 CD player users.

The responses on frequency of use of program and random play features are shown in Figure 1.1. The results show that half of the users had never used program or random play. Non-parametric tests showed that there were no statistical associations between the size of a CD music collection and the frequency of use of program play or random play. These results suggest that music programming or random play was neither found attractive for a large collection nor found superfluous for a small collection or vice versa.

Of the 37 users who never used program play, twelve users commented that programming was not worth the effort because it was found to be too laborious or too time-consuming. Nine of them preferred the original order of the music as found on a CD album instead of a personally selected order. Eight of them found programming to be a useless feature, especially for a single loaded CD. Three of them had never learnt the operation involved.

If programming was used, it was frequently intended to skip disliked music or for recording purposes on another medium.

Of the 39 users who never used random play, fifteen users commented that they did not want to violate the original order of a CD album, and seven users did not find it to be a useful feature. Fourteen users gave no reason for not using random play.

If random play was used, most users commented that they liked the surprise effect and variation, for example, to break out of a rut of listening to the same order.

In the last question, users were asked to freely comment on the lack of features of their CD player. Fourteen users commented that they would like a multi-disc player. In particular, eleven mentioned that they would like a CD changer, two wanted a CD jukebox, and one wanted direct access to a large disc on which all music was stored. A desire to have more ’intelligence’ in their players was expressed by twelve users. This ’intelligence’ included advanced delete functions to permanently reject uninteresting music, a programme memory that is never lost, and a mood function. Seven users commented that their player was not easy to use, especially for more advanced features such as programming. Another seven users wished to see more textual information about the music on the console or remote control. Six users complained about mechanical noise and disturbances and how easy it was to damage the CD player physically.

Discussion

The questionnaire provided valuable information on CD player use and user desires and needs on CD player features. The questionnaire only had an explorative nature because respondents were not selected on a random basis and the number of respondents was small in comparison to the population that own and use a CD player. Hence, it is likely that selection bias is present in the results and that the sample parameters deviate from population parameters. These shortcomings make a valid generalisation of the results to a wider population difficult.

Although a CD player has evolved to become a common item of property and users have learnt to perform a number of fixed actions with it, such as switching to another music recording, it is unlikely that they will learn the whole repertoire of CD player features. For instance, the results showed that some users had never taken the opportunity to learn how to program.

The results also showed that users do not use programming features on single-CD players because it is simply not worth the effort. Besides the fact that programming a single CD was found to be useless, a music programming task was also found to be laborious, time-consuming, and difficult to perform. Lastly, the results indicated that users want multiple-disc players that access their music collection directly (e.g., multi-disc players) and 'smarter' player features that acknowledge their music listening behaviour and habits.

1.3 Related research fields

Adaptation to user choice has become a popular field of research. The application domain may vary in this field: written documents, books, music, films, television programmes, World Wide Web pages, USENET news, holidays or electronic mail. Some techniques that are commonly used are briefly discussed below.

1.3.1 Information retrieval

Information retrieval systems have traditionally been based on a static text database and users who use the system sparingly by formulating a query consisting of keywords (Belkin and Croft, 1992). Currently, these systems are expanding to include more dynamic domains such as still images, movies and music.

Information retrieval is mainly focused on the application of statistical techniques to retrieve written documents relevant to a user query, though the use of a variety of other techniques has evolved (Chen, 1995). In the statistical *vector space* technique, words are indexed on the basis of their frequency in texts, which allows each document to be represented as a vector of indexed words. As a user query is also represented as a vector of words, retrieval is done by measuring the *similarities* between document vectors and the query vector (usually by calculating the cosine of the angle between two vectors), and retaining the documents that are highly similar to the query. The similarity measure is adapted using relevance feedback from the user on each retrieved document. The first information retrieval system developed was the SMART system which has its origins in the late 60s (Salton and McGill, 1983; Salton, 1989). The *vector space* technique is commonly used by search engines on the Internet for finding relevant Web pages (Balabanovic, 1998).

1.3.2 Information filtering

Information filtering systems are intended to mediate between a user and incoming, dynamically changing streams of multimedia information. Users are expected to use the system repeatedly with long-term, but imprecise, goals and interests, that is, without the requirement of formulating a query. Adaptation to user choice behaviour is a prime concern for these systems (Belkin and Croft, 1992). Information filtering has adopted many statistical retrieval techniques from information retrieval research (Foltz and Dumais, 1992).

One approach of information filtering is called social or collaborative filtering, which recommends a small set of items to a single user (Malone, Grant, Turbak, Brobst, and Cohen, 1987). Social filtering is based on the concept that people want to exchange their personal experiences and that people are willing to rely on personal experiences of others for their choices (Resnick and Varian, 1997). When a user requires a recommendation, a social filtering system recommends items that were highly valued by other users with similar tastes and preferences. Social filtering is already commercialised on the World Wide Web by on-line stores for books and CD albums, for instance.

1.3.3 Data mining

Data mining or knowledge discovery is a new research field which integrates statistics, databases, machine learning and artificial intelligence to search for relationships, global patterns and trends in large databases (Holsheimer and Siebes, 1994; Fayyad, Piatetsky-Shapiro, Smyth, and Uthurusamy, 1996). Current database management systems only allow deduction mechanisms to infer information that is a logical consequence from the database structure. The basic idea of data mining is to provide future database management systems with induction mechanisms to generalise information that is not directly encoded in the database structure (Holsheimer and Siebes, 1994).

1.4 Contents of the thesis

The central theme of this thesis is how future interactive music players can be of help to music listeners in finding preferred music in a large personal music collection. Unfortunately, existing music players do not help. To help the listener, players should adapt to the music choices made by the listener and should present music information in an adequate and interactive way. This thesis seeks to explore how these user requirements can be carried out in the design and implementation of adaptive functionality and new interaction styles for music players. A number of applied studies and controlled user experiments were conducted in support of this.

In this thesis, *adaptation to music choice behaviour* has prime focus. The underlying assumption is that music listeners choose and refer to music on the basis of musical attributes. If these attributes can be identified by the system, by observing what music is accepted and what music is rejected, the system can infer what other music is likely to be relevant for the music listener.

A substantial effort has been spent in the development and evaluation of an adaptive system called PATS (Personalised Automatic Track Selection). PATS was, and still is, intended to automatically create music programmes, relieving music listeners from the task of programming music by themselves. Later on, PATS was configured as a music recommender system to suggest small sets of music options, intended to ease and speed up a music programming task carried out by a music listener. In the specialist literature, social filtering systems are also called recommender systems (Resnick and Varian, 1997). As the PATS system is a personalised system and does not take opinions of other people into account, it may be inappropriate to call the PATS system a recommender system.

In the last chapters of this thesis, the focus is on *interactive information presentation*. A *multimodal interaction style* using PATS has been designed, implemented and evaluated. In a multimodal interaction style, multiple human (or system) input and output channels are combined to convey information and feedback during interaction. A typical example is the combination of manual operation and voice control at the human output (system input) side and visual, tactual and speech feedback at the human input (system output) side.

1.4.1 Jazz as the application domain

Jazz music was chosen as the application domain to study the performance of the PATS system and human choice behaviour. It also served as test material. Besides personal interest and knowledge, two requirements for a musical genre or idiom to be chosen were defined before work commenced.

Firstly, in order to guarantee easy continuation of the work over time, the musical genre to be chosen had to be 'timeless', that is, its appreciation had to be insensitive to temporarily prevailing music cultures and movements. This requirement was motivated by the need to define and collect musical attribute data of a large music collection for experimental purposes. As music extraction from CDs and collection of attribute data is very time-consuming, it had to be a 'once and for all' activity, or at least an activity that could be carried out in a careful but carefree manner while other work progressed. In contrast to popular music, it was concluded that the appreciation of jazz music was unlikely to be influenced by any prevailing music movement during the time period in which the work was carried out.

Secondly, in order to acquire different types of participants for experimental validity, the musical genre to be chosen had to consist of a variety of well-defined styles. Different styles within a genre cover distinct music listeners. Through its history, jazz music has developed styles that can be ascribed clearly to time periods. In contrast, classical and popular music consist of a multitude of unspecific styles caused, respectively, by its historical background and its widespread admiration.

1.4.2 Overview of the chapters

Chapter 2 describes what kinds of choice strategies people generally use in response to different conditions, such as the amount of presented information, the number of options, or time constraints to choose. An overview of relevant literature on classical choice theory is given. Relationships and distinctive properties of a music programming task with respect to this theory are identified. Terms required for the study of music selection and an attribute representation for jazz music are also defined. Finally, results of a focus group study are reported to investigate on what attributes jazz music is likely to be selected. It is shown that the attribute representation chosen adequately covers the attributes used by music listeners, which provided a platform for further implementation and experimentation.

In Chapter 3, the implementation and a comparative evaluation of four inductive learning algorithms is described. Inductive learning algorithms are required to identify the covert criteria on which a music listener judges music. One of these algorithms is an essential component of the PATS system.

In Chapter 4, the complete design and implementation of the PATS system is reported. While using the PATS system, music listeners only have to select one preferred music option. The system then finds similar options to complete a music programme. Its compilation strategy adapts on the basis of what options in the programme were accepted and rejected by the music listener. Related systems and likely PATS applications are discussed.

In Chapter 5, results of computer simulations and a user test to assess performance and adaptive properties of the PATS system are reported. In the evaluations, the PATS music programmes were compared to randomly assembled music programmes. Among other things, it is shown that PATS programmes contain more preferred music and cover more preferred music that is available in the music collection than randomly assembled programmes. In addition, the PATS system adapts, though slightly, to the music preference of a music listener.

In Chapter 6, the results of two user experiments are reported. The experiments studied music listening and selection behaviour in a music programming task under different conditions. One experiment studied the extent to which a music recommender system such as the PATS system can help a music listener to perform a music programming task more efficiently, with less search effort, and under what level of user control. In the other experiment, the same research questions were posed, but with a time constraint to complete the programming task. Among other things, it is shown that using the PATS system leads to less search effort, but does not necessarily lead to a more efficient task performance or higher programme quality. Though the PATS system appears to be a highly preferred feature for music programming, using a recommender system can lead to a disturbing loss in control of music selection. The general progress of a music programming task is also identified, providing ways for a better user task integration of music recommender systems in the future. Implications for interaction style design with respect to the user requirements on *adaptation to music choice behaviour* and *interactive information presentation* are discussed.

In Chapter 7, the design and implementation of a multimodal interaction style for music programming with the PATS recommender system is reported. Findings from previous chapters led to design requirements for the interaction style. The envisaged home use requires that the interaction style can be used, instantly, also when visual inspection of information is impossible, difficult or undesirable. The latter design requirement was prompted by the fact that, for instance, existing players have inadequate visual displays and the use of a remote control can imply the use of the interaction style from a large viewing distance. A conceptual design process resulted in the use of a visual roller metaphor, the use of a force feedback trackball as the input device, and the use of tactual and auditory (speech and non-speech audio) feedback. The use of tactual and auditory feedback is primarily to compensate a possible lack of visual information.

In Chapter 8, the results of a user evaluation of the interaction style are reported. The evaluation assesses instant usability with and without the use of a visual display. Among other things, it is shown that a music programming task can be done successfully, right from the start, even using the interaction style without a visual display. Experience with a visual display is not strictly necessary. However, using the interaction style without a visual display requires more time in order to discover and actively remember procedures.

In Chapter 9, the main conclusions of the work carried out in this thesis are presented.

CHAPTER 2

Choice and music selection

In the existing literature, various choice strategies are proposed that people may employ when they are faced with a new and complex choice problem. These choice strategies allow people to behave flexibly in response to the task and context characteristics, while coping with their limitations on information processing resources. A brief survey of frequently reported choice strategies in classical choice theory is given. The effects of some relevant task and context factors on choice strategies, that are also relevant for the music domain, are then discussed. In order to study decision processes in the music domain, the terms *context-of-use, music preference, musical taste, music programming* and *attribute representation* are widely used. Definitions and the use of these terms are discussed. Besides some similarities, a music programming task has some special properties that differ from classical choice tasks. These properties pertain to both the nature of music options and the nature of the task. Finally, in a focus group study, an investigation is carried out to see what valued attributes music options can be represented by, and to what extent music preference can be described by these attribute values. Results are interpreted for further experimentation and in terms of choice strategies that music listeners may use while searching for preferred music in a large music collection.

We make choices throughout our everyday life. The central theme in choice is preference, which is the choice of an option above other options for particular reasons. Although preference seems easy to express at first sight, making a choice is notoriously difficult in new and complex situations. Factors that influence the difficulty of a choice include task characteristics, characteristics of the choice options, personal characteristics and characteristics of the social environment.

With respect to task characteristics, a choice task often consists of many options and allows little time to decide. Each option contains much information to be processed.

With respect to characteristics of the choice options, that is, the choice context, options may have conflicts in their properties which require a trade-off to resolve them, they may have too much in common or they may be too incompatible for comparison.

With respect to personal characteristics, people may feel uncertain about what they actually prefer, or they may be unable to accurately anticipate the consequences of a particular choice.

With respect to characteristics of the social environment, some decisions need to be justified to others or need to be done collaboratively, because they require a considerable investment, have a long lasting impact, or affect other people.

For many people, listening to recorded music carries an intrinsic enjoyment during leisure and relaxation. Music choice may then be a regular pursuit. If there is a wide assortment of music available, finding preferred music can be quite difficult.

This chapter gives a brief survey of classical choice theory and choice strategies that people employ in a choice task. Music choice and music programming are discussed in the context of classical choice theory. Widely used terms in the study of music choice such as *context-of-use* and *music preference* are defined. One of the research purposes reported in this chapter was to arrive at a fixed attribute representation for jazz music that adequately captures choice criteria for jazz music. An attribute representation of jazz music is formulated and validated by means of a focus group study.

2.1 Choice theory

Classical choice theory is central in psychology and economics, providing systematic knowledge on how people make choices from a set of well-defined discrete options in specified experimental choice tasks and contexts.

Empirical work on choice theory is typically done in laboratory tests by letting participants select an option from a well-defined set while conforming to specific task instructions under various conditions. Information about options is generally displayed on an option $\times$ attribute matrix board (see, for an example, Payne, Bettman and Luce, 1998). Since options are in general abstract, artificial, or imaginary, they are easy to manipulate for experimental purposes. In addition, the amount of information per option, the values of the attributes, and the task instructions can be provided in such a way as to adequately control the experiments. The choice task is then well-structured and the options are well-defined. The number of studies involving real-world artefacts is relatively small (for an example, see Dhar and Simonson, 1992; Simonson and Tversky, 1992; Pieters, Warlop and Hartog, 1997).

Research on preference and choice was initially an inquiry into normative and descriptive theories of choices, that is, rational theories of how people ought to make choices to yield optimal results. However, the research has gradually moved to theories on the psychological processes involved in decision making, rather than an account of the choices themselves. In normative terms, to get the optimal choice you need to gather all the information on *each* option and to inspect the costs and benefits of *each* option. The option with the most benefits should be chosen. Complete inspection is deemed to be impossible however, due to limitations in human information processing abilities, unless there are only a few alternatives to consider or a single option overrules all other options on all objectives.

Slovic (1995), among others, revised the assumptions in normative choice strategies by demonstrating that people do not have well-established notions of options, do not have some master list of preferred options, and do not follow a master plan to handle choices. He stressed that preference is constructed on-the-spot, instead of retrieved from memory, by employing fast rules of thumb instead of by reasoning rationally. Rules of thumb yield provisional choices instead of optimal choices; people make errors or are inconsistent when they choose. Actual choice behaviour deviates from normative descriptions; actual choice is rather a compromise between the task and the context, on the one hand, and a humans' limited information processing abilities, on the other hand (Payne, Bettman and Luce, 1998). Given these restrictions, it can be argued that provisional choices are optimal ones in a more general sense because people also attempt to jointly optimise other criteria such as minimising the required time and effort, rather than solely maximising choice quality.

The material on choice theory is too extensive to treat here comprehensively (see Payne, Bettman and Johnson, 1993; Payne, Bettman and Luce, 1998 for a complete overview on choice theory). Only areas that can be generalised for the field of music selection and programming are addressed here. For instance, a theory on risky choice is not discussed here, since music choices do not bear any risk, threat, or uncertain outcome. What both fields have in common is that options are represented as a collection of valued attributes including quantitative, ordinal, categorical, and nominal ones.

2.1.1 Searching for a preferred option

A choice task is defined here as the selection of a preferred element from a set of options. The evaluation of options to arrive at a choice can be described by elementary cognitive actions, such as examining attributes of an option, comparing options by a single attribute, eliminating an option or attributes from further consideration, and providing the next option. The order in which these elementary actions are executed is called a choice strategy. It is commonly accepted that people display flexibility by employing various choice strategies in response to a choice problem (Payne, Bettman and Johnson, 1993).

As a choice strategy is essentially an instantiation of cognitive processes, observations of differing patterns of information processing allow different choice strategies to be identified (Payne, Bettman and Johnson, 1988; Payne, Bettman and Johnson, 1993).

- *Compensatory or noncompensatory.* People can respond in two different ways when faced with conflicts, that is, the presence of desirable as well as undesirable attribute values of an option. The two ways are based on *resolving* the conflict and *avoiding* the conflict. Strategies are called *compensatory* when conflicts are resolved by compensating desirable attribute values for undesirable ones. In general, compensatory strategies require different attribute values to be transformed into a common (psychological) scale to compute an overall evaluative value of an option. Strategies are called *noncompensatory* when conflicts are avoided by evaluating only the most important attributes of options. An attribute can be so important that other attributes can never compensate for an undesirable value of that important attribute. As a result, the preferred option is superior to other options only on the most important attributes, although it may be inferior when comparing all attributes.

- *Consistency or selectivity of information processing.* People can divide their attention differently to acquire information from options and from attributes. Information can be completely examined, partially examined, or not examined at all. If information is examined *consistently*, that is, with equal amounts for all alternatives, it is assumed that a compensatory strategy is employed. In contrast, if attention to information is divided *selectively*, a noncompensatory strategy is likely to be applied. People can also process information qualitatively or quantitatively.

- *Cognitive effort or load.* People need to process a certain amount of information when looking for preferred options. Different choice strategies require the execution of different kinds or different numbers of elementary cognitive actions. Total cognitive effort in applying a choice strategy can be approximated by the number of elementary cognitive actions executed. As cognitive actions are covert, the number of physical actions in a choice task performance can be taken as a measure for cognitive effort, under the assumption that physical actions are reliably coupled with cognitive actions. The time required to perform a choice task is another measure for cognitive effort, under the assumption that the execution of a cognitive action takes a certain amount of time.

- *Alternative-based or attribute-based.* People can proceed differently with respect to options and attributes. Basically, they can evaluate multiple attributes of a single option before moving on to the next option, or they can evaluate a single attribute by jumping back and forth between options, before going on to the next attribute. The former refers to an *alternative-based* strategy, whereas the latter refers to an *attribute-based* strategy (Tversky, 1969; Russo and Dosher, 1983). An alternative-based strategy is generally implemented by computing an overall evaluative value per option.

2.1.2 Choice strategies

The general interest in choice strategies is not how they are precisely constituted, but under what circumstances they are evoked, what cognitive load they induce, and what choice quality is obtained for what effort.

In the following sections, frequently reported choice strategies are grouped and discussed.

Normative strategy

The normative strategy assumes that options have attributes that have higher or lower subjective importance expressed by weights. More specifically, each attribute is multiplied by its weight, resulting in a weighted value for each attribute. These products are summed to arrive at a single *scale value*, reflecting an overall evaluation for an option. This is done for all options. The strategy concludes by choosing the option having the highest overall evaluation. In other words, it is based on a maximisation principle.

The normative choice strategy has dominated decision research for years. It is known as the *weighted additive value* strategy (Payne, Bettman, and Johnson, 1988; Payne, Bettman, and Luce, 1998) or the *simple additive* model (Tversky, 1969). Many special cases of the normative strategy exist to allow for different decision behaviour in particular choice situations such as time constrained tasks, risky choice, the availability of new information about options (Jagacinski, 1995), and equal importance of all attributes (Einhorn and Hogarth, 1975).

Under the assumption that no errors are made, the normative strategy yields optimal choices. Normative choices can be considered to be an upper limit for choice performance in experimental settings, because actual human choice behaviour falls short of this normative prescription.

Using a normative strategy means that *all* options are examined independently (alternative-based), and that *all* attributes of each option are processed quantitatively to compute an overall evaluative value. Consequently, a deficiency of a certain attribute value can be compensated for the excess of another attribute value (compensatory strategy). Clearly, using a normative strategy puts the heaviest demands on cognitive effort.

Random strategy

The random strategy assumes that people simply choose an option at random without examining information. A random choice can be interpreted as avoiding the making of a choice; it costs minimal cognitive effort. The application of this strategy occurs in situations where people are indifferent about the consequences of the choice (probably, all options are considered equally good or bad), where other strategies were previously unsuccessful, or when information processing limitations have been met (e.g., severe time constraints, overload of options or attributes). Random choices are often used to indicate a lowest limit for choice performance in an experimental setting.

Other strategies that process little information, but that certainly do not produce random choices, are *habitual* behaviour, determined by choosing what was chosen the last time, and *affect referral* (Wright, 1975), which directly retrieves some pre-installed option from memory. Both strategies are probably based on prior examination of information.

Conjunctive strategy

The conjunctive strategy assumes that people continue the search for further options, as long as the effort does not exceed some maximum, *and* the attributes of the best possible option at hand do not exceed pre-defined reference points. Because people have no knowledge about options not yet examined, it is assumed that they set reference points or aspiration levels to decide whether or not a satisfactory option exists among the already examined options. The options are examined one at the time or in small groups in the order that they are provided.

The *satisficing* principle (Simon, 1955), which is essentially a conjunctive strategy, chooses the *first* option with all attributes exceeding their reference points; the search for further options is immediately concluded. If all options are considered and no choice could be made, people have to adjust their reference points and repeat the whole process, or just side-step to another strategy. The result of the conjunctive strategy is dependent on the order in which the options are evaluated; the selection of one out of multiple equally preferred options solely depends on whether one is considered before or after the other.

The conjunctive strategy is noncompensatory, alternative-based, and selective in its information acquisition. Using a conjunctive strategy demands less cognitive effort than using a normative strategy because it reduces information processing and processes information qualitatively.

Tournament strategy

The tournament strategy assumes that people compare options two-by-two along the attributes, that is, attribute-based. A single pair of options is considered at any one time, and only one option is retained for further comparisons. The retained options are compared in a similar fashion with other options. These pair-wise comparisons are conducted repeatedly until only a single option is retained which is then the preferred option. Although suggested by the name, the strategy is not required to follow a strict 'fair' tournament structure, that is, that 'winners' of random pairs promote to the next round, until a single overall 'winner' is left. The term 'tournament' is only used here as an umbrella term for all strategies involving pair-wise comparisons.

People can apply several rules or heuristics to choose one option from a pair-wise comparison.

> The *additive difference rule* (Tversky, 1969) considers weighted differences between each pair of attribute values of two options. These weighted quantities are summed over all attributes attaining the relative overall evaluative value of the two options. This defines the best option to choose. At least two truncated cases of the additive difference rule exist that reduce the amount of information processing, and hence reduce cognitive effort (Russo and Dosher, 1983; Bockenholt, Albert, Aschenbrenner and Schmalhofer, 1991).
>
> The *majority of confirming dimensions rule* (Russo and Dosher, 1983) retains only the signs of each difference among attribute values, while ignoring the difference magnitude. By taking each option as a reference in the comparison, signs are assigned to both options. The heuristic chooses the option that contains the largest number of attributes with a positive sign. The heuristic is uninformative when ties occur. It is a simplification of the additive difference rule; it ignores possibly valuable information. It trades-off cognitive effort with less accurate results.
>
> The *lexicographic rule* simply retains the option that has the best value for the most important attribute. If a tie occurs, the next most important attribute is considered, until a decision can be made. If, in addition, just noticeable differences (jnd) are defined on attribute values, a semiorder is imposed on the lexicographic ordering. With this semiorder, attributes are only considered to be perceivably, conceptually or reliably different when the attribute value difference between two options exceeds the jnd. Only after exceeding the jnd is an attribute subjected to a comparison, otherwise a tie occurs. In particular, the *lexicographic-semiorder rule* sometimes violates the transitivity[1] assumption in normative choice theory (Tversky, 1969; Fishburn, 1991).

Elimination strategy

The elimination strategy assumes that people eliminate options from further consideration if attribute values of options do not comply to a criterion value or pre-defined standard. The elimination-by-aspects strategy (Tversky, 1972) is essentially an elimination process governed by successive selection of a desired aspect, that is, an attribute and its criterion value. All options that do not include a desired aspect are eliminated from further consideration. The strategy continues by selecting other desired aspects until a single option remains that complies to all desired aspects. When selecting an aspect to be eliminated, attributes are sequentially considered in order of importance, and a criterion value for the attribute is established separately.

1. The transitivity assumption is central to normative choice theory, stating that if option x is chosen over option y, and option y is chosen over option z, implies that x must be chosen over z when both x and z are offered (Luce, 1959).

In the original formulation of the elimination-by-aspects strategy (Tversky, 1972), aspects were freely accessible for elimination purposes. Aspects were selected in a probabilistic manner dependent on their weights of importance. This was revised however in a more parsimonious version of the theory by externally imposing a binary *preference tree* structure on the entire set of options (Tversky and Sattath, 1979). This hierarchical tree only allows a sequentially ordered access to aspects and, hence, specifies the way in which aspects are expected to be eliminated. For instance, choosing between two compositions of Beethoven and two compositions of Miles Davis is, first of all, likely to be viewed as a choice between classical music and jazz music. Thus, options intrinsically suggest a hierarchical structure by their mutual similarities, or by the order in which they are considered (Tversky and Sattath, 1979).

2.1.3 Task and context

The use of a particular choice strategy is dependent on task and context factors (Payne, Bettman and Johnson, 1993; Payne, Bettman and Luce, 1998). Task factors are general characteristics of the decision task that are assumed to be independent of the nature of the options. Task factors include the number of options to choose from, the order in which options are provided, and the time allowed for making a choice. In contrast, context factors deal with the nature of the options such as the number of attributes per option to examine, the similarity between options, and the ranges that the attribute values can take. Factors that pertain to personal characteristics and the social environment are not further treated here.

The effect of information load

Information overload due to a large number of options or much irrelevant information per option has a negative effect on choice quality (Payne, 1976; Onken, Hastie and Revelle, 1985; Keller and Staelin, 1987; Sundström, 1987). For instance, Keller and Staelin (1987) found a decreasing choice quality and inconsistent choices over time when more information about options was displayed. However, with practice people learn how to become more selective in their information processing (Grether, Schwartz and de Wilde, 1986). It is unclear however to what extent selective information processing accounts for the extraction of relevant information and to what extent this interferes with irrelevant information (Payne, Bettman and Johnson, 1993).

Information overload also results in the use of a simple choice strategy. If people are confronted with many options, they prefer to use noncompensatory strategies or to combine several strategies (Payne, 1976; Sundström, 1987). Payne (1976) reported a shift to an elimination-by-aspects strategy, whereas Sundström (1987) observed choice behaviour consistent with a conjunctive strategy.

The effect of time constraints

When faced with time constraints, people are held back by a discrepancy between the amount of time that would be necessary to complete a given choice task successfully, and the actual amount of time that is given (Svenson and Edland, 1987). Time constraints decrease people's belief in their ability to attain satisfiable goals, that is, the *perceived level of control* (Paterson and Neufeld, 1995).

Because it is difficult to simultaneously cope with time constraints and choice quality (Payne, Bettman and Luce, 1996), time constraints often lead to inaccurate choice (Zakay and Wooler, 1984), though performance improves with practice

(Payne, Bettman, and Johnson, 1988). Time constraints even lead to preference reversals, probably because negative aspects of options become more pronounced under time pressure (Svenson and Edland, 1987). Svenson and Edland (1987) found that a majority of participants chose other apartments when under time pressure than when decision time was unlimited.

It is generally accepted that people use at least three coping mechanisms to deal with a time-constrained choice task (Miller, 1960).

> One way in which people cope with time constraints is by *accelerating their processing* (Ben Zur and Breznitz, 1981). It is argued, however, that performing time-constrained tasks in the same way as a choice task under no time constraints, only faster, is an unwise coping mechanism because it leads to inaccurate choices (Payne, Bettman, and Luce, 1996).
>
> Another way in which people cope with time constraints is by *focusing on important attributes* that score best (Edland, 1994), that are less difficult to evaluate (Garbarino and Edell, 1997), or that decrease negative impact (Ben Zur and Breznitz, 1981; Svenson and Edland, 1987).
>
> A third way in which people cope with time constraints is by *using a choice strategy that requires little cognitive effort*. Zakay (1985) found that people use noncompensatory strategies, such as a lexicographic strategy, under time pressure. Under severe time pressure, people prefer to shift to attribute-based strategies (Payne, Bettman, and Johnson, 1988) or even avoid making well-considered choices. The latter may result in random choice in an extreme case (Ben Zur and Breznitz, 1981; Miller, 1960) or 'conservative' or habitual choice in purchases (Wright, 1974) and in political voting (Hansson, Keating, and Terry, 1974).

The effect of other options

An important basis of the formalisation of normative strategies is the assumption that a particular choice is *in*dependent of the absence or presence of other options, that is, the choice context (Luce, 1959). More particularly, the choice of one option over another should not change if the choice context changes, as long as both options are still available.

However, Simonson and Tversky (1992) showed that irrelevance of a third option or context independence in choice does not hold in practice. They formulated their findings in a componential context model (Tversky and Simonson, 1993). The following example, illustrating that people do use the presence of other options to identify the most preferred option, has been adopted from Simonson and Tversky (1992). Under normative laws, the choice of a *consumer product of brand X* over a similar *consumer product of brand Y* should not change, irrespective of other similar available products. However, consider a third product that is clearly inferior to the less favoured product of brand Y, but not necessarily inferior to the favoured product of brand X. For instance, this third product may be a similar *consumer product of the same brand Y at a higher price.* If this product is added to the choice context, it appears that preference for the originally less favoured product increases.

If options have a lot in common, there is less differential information to compare and, hence, it requires more cognitive effort to reveal their differences. People are then tempted to use compensatory strategies that require more time to decide

(Biggs, Bedard, Gaber and Linsmeier, 1985; Bockenholt, Albert, Aschenbrenner and Schmalhofer, 1991). Making a choice from many similar options is thus experienced as being more difficult.

2.2 MUSIC SELECTION

For the study of decision processes in the music domain, the terms attribute representation, context-of-use, and music preference are widely used. This section first gives an outline of which attribute representation is used to represent music options. Then, definitions for context-of-use and music preferences are given, followed by an elaborate definition of a music programming task and its similarities and differences when compared to classical choice tasks.

2.2.1 Features and attribute representation of jazz music

The term musical feature is reserved to cover subjective and varied properties of music recordings that are found relevant by a music listener. In contrast, a music attribute refers to a fixed and a more objective aspect of a music recording. A music attribute with a range of fixed values may cover one or several musical feautures.

As people are used to talk about music in terms of musicians, instruments, music styles, etc., it is reasonable to represent music options as a collection of valued musical attributes. Some attributes contain information mainly for reference purposes (e.g., title, artist, recording company). Some attributes address the nature of the piece of music (e.g., music style, standard). Some attributes provide information about the particular musical performance (e.g., ensemble strength, soloists). Some attributes reflect the recording circumstances (e.g., whether or not it is recorded in front of a live audience, the place of recording). Some attributes relate to the perceptual and cognitive aspects pertaining to musical experiences (e.g., tempo, rhythmic accompaniment, harmonic progression). A set of 18 musical attributes that are thought to be key attributes for jazz music preference is presented in Appendix I. The attributes and their values were primarily extracted from CD booklets, discographies (Cook and Morton, 1994; Erlewin, Woodstra and Bogdanov, 1994), and books on jazz music training (Coker, 1964; Sabatella, 1996).

2.2.2 Music preference, musical taste and context-of-use

In common language, the terms *music preference* and *musical taste* are intuitively meaningful and apparently self-evident. In research on music therapy and education, the term music preference is also interchangeably used with the term musical taste to refer to the same concept (Leblanc, 1982). Here, a distinction must be made between music preference and musical taste.

Music preference depends on the circumstances in which music is heard. Consequently, music preference is determined by the real world environment including the mood, the current activities and the listening purposes of the music listeners, henceforth called the *context-of-use*.

It is common knowledge that *mood* is affected by music that music listeners know and have learned to love and appreciate (Cantor and Zillman, 1973). Music listeners prefer to listen to music that they were accustomed to listen to in adolescence or early adulthood, perhaps out of a sense of ‘nostalgia’ (Holbrook and Schindler, 1989; Rubin, Rahhal and Poon, 1998). However, repetitive listening increases the preference for unfamiliar music (Peretz, Gaudreau and Bonnel, 1998). The perceived emotional content of music may even induce goose pimples (Panksepp, 1995). Breckler, Allen and Koneçni (1985) found that, in order to optimise mood, the most pleasing music was listened to at the end and in the longest runs in listening sessions. In addition, people in a good mood do not feel better after hearing well-liked music, but feel worse after hearing disliked music. People in a bad mood, on the other hand, feel better after hearing well-liked music, but do not feel worse after hearing disliked music (Wheeler, 1985). Music is even an effective means for mood induction in marketing applications (Bruner, 1990) and clinical and therapeutic applications (Bunt, 1997).

Music listeners may have different *listening purposes* such as the intention to listen actively or passively. Active listening means consciously exploring and enjoying the musical content. A music listener then devotes complete attention to the main task of music listening. When listening to music actively, music listeners tend to forget about the passage of time (Palmquist, 1990). Passive listening refers to the situation in which the music plays in the background while the listener is engaged in performing a main task that is not directly related to the music experience itself. A music listener is then able to perform other tasks simultaneously, as long as the music does not attract attention or is not explicitly attended to (see Wickens, 1992; Proctor and Dutta, 1995, for a survey on multiple task paradigm). In fact, music is valued as a background for dull and repetitive work, though it may have negative effects on work that requires thought and alertness (Kroemer and Grandjean, 1997).

Music preference is defined here as an individual's temporary liking of particular musical content in a specific context-of-use. It is assumed that different contexts-of-use induce different expectations about what the music should contain. Music preference is instantaneous in nature and subordinate to musical taste, which is defined here as an individual's slowly evolving long-term commitment to a particular musical genre. This distinction between music preference and musical taste exactly follows the definitions as given by Abeles (1980). Development of musical taste is assumed to depend on the cultural environment and personal characteristics (Leblanc, 1982; Hargreaves and North, 1997). For instance, musical taste is affected by major consensus (Furman and Duke, 1988), by the approval of radio disc-jockeys, peers and music teachers (Alpert, 1982), by musical training (Geringer, 1982) and by age (Tolhurst, Hollien, and Leeper, 1984). It is very likely that purchase behaviour over time will result in a home music collection that reflects the musical taste, comprising many different music preferences, of the owner.

2.2.3 Music programming

A music programming task is defined here as the serial selection of multiple preferred music options from a music collection to arrive at a music programme that is suitable in a specified context-of-use. As multiple choices have to be made, programming one's preferred music can probably be best envisaged as an iterative, dynamic search until the selection of the music is completed.

In experiments on music programming, as in Chapter 6, participants are first instructed to concisely express a specific context-of-use at will or to imagine a given context-of-use. Then, in various experimental conditions, they are instructed to select a pre-defined number of music options from a music collection that is suitable in the given context-of-use. During the selection process, participants are allowed to listen to music at will.

A music programming task has several distinctive properties when compared to most classical choice tasks reported in the relevant literature. Some distinctions are discussed below.

> *No optimal solution* for music choice exists, whereas normative choice theory is based on the assumed existence of optimal choices. Quality of music choices is a strongly subjective notion and the quality of a music programme may not be equal to the sum of the values of its constituent music options.
>
> Music has *personal appeal* to the music listener. Music options are fixed real-world stimuli, whereas options in many empirical studies on normative choice are rather abstract, artificial or imaginary options that are easy to manipulate.
>
> Music programming deals with *multiple serial choices*, which is often underexposed in classical choice theory. Earlier choices in the programming process influence a choice at hand, resulting in choice criteria pertaining to individual options as well as to the already selected options. A given choice may even lead to rejection of other, possibly already expressed, choices. It is thus evident that the presence of other alternatives in the choice context defined by both the music collection and the music programme influences music choice behaviour.
>
> Music programming *extends over time*. Perceived similarities between music options may emerge over time by comparing or evaluating music options repeatedly on different aspects. Also, music preference may be in a state of development, which means that music listeners have to accommodate their search accordingly.
>
> Targets to search for in a music programming task are *poorly defined* at the outset of the task. Music listeners may have no master list of preferred music or they may be uncertain what music would be appropriate in the given context-of-use. Though some desired, rather abstract, musical features can be determined from the context-of-use, these attributes do not necessarily lead to a sufficient number of concrete music options to complete the task. More particularly, music preference must be developed by listening to musical cues while performing the programming task. If search targets are loosely defined, searching for preferred music is no longer concerned with the achievement of a goal per se. Rather, the activity of the task itself becomes inherently rewarding. In general, people tend to perform a task for its own sake, when there is a slightly skewed balance between challenges on the one hand and skills on the other hand; the perceived challenges must just beat the present skills (Csikszentmihalyi and Csikszentmihalyi, 1988). This captivating effect of particular tasks, referred to as the experience of flow, is not only predominant in, for instance, artistic and sports activities, (Csikszentmihalyi and Csikszentmihalyi, 1988), but also in playing computer games (Malone, 1982). A particular consequence of a task involvement is that less attention will be paid to apparent passage of time (Tsao, Wittlieb, Miller, and Wang, 1983). Likewise, listening to music tends to make music listeners forget about the time

(Palmquist, 1990). This is likely to be reinforced when music listeners are carried away by the ample opportunities for music selection and listening which result from a large music collection.

In contrast, music programming is likely to depend on the same task and context effects that are investigated in classical choice theory such as the effects of information load, the presence of other options and time constraints.

The size of the collection may affect choice behaviour due to information overload and the presence of other choice alternatives (see Section 2.1.3). A very large music collection renders a complete and thorough examination of all available music impossible because it is simply too unwieldy to do so. Consequently, music listeners have to resort to simplifying choice strategies such as habitual choice and an elimination strategy in which only part of the collection is examined.

If time constraints are imposed on a music programming task, the same type of effects on choice quality and choice strategies may also arise as discussed in Section 2.1.3. Under a time constraint, music listeners must pay additional attention to negative affects associated with not completing the task in time. It is likely that they perceive less control to access all courses of action to search for preferred music, less control to listen to music, and less control to make preferred music choices.

2.3 Focus group study

The purpose of the focus group study was to investigate what criteria are used in the expression of preference for jazz music and how these criteria can be addressed by fixed musical attributes (see Appendix I). Results were interpreted in terms of likely choice strategies used by music listeners doing a music programming task.

2.3.1 Method

Participants were instructed to listen to 22 jazz music recordings, on tape, in an active listening session at home, and to indicate their preference on a form. Preference decision was a binary choice; they were instructed to mark on the form whether they liked a music recording or not. The form only indicated the title of the music recording, the performing main artist or ensemble, and the names of the composers. After that, participants could elucidate their likes and dislikes of the recordings as indicated on the form in a semi-structured interview. During the interview, it was possible to listen to each music recording at will.

2.3.2 Material

A compilation of 22 jazz music recordings taken from 11 commercial CDs was recorded on an audio cassette tape; only a two-minute excerpt was recorded. The compilation generally contained music from well-known jazz musicians; only one relatively unfamiliar musician was added. Different performances of two titles were included, and musicians made contributions to different recordings. Three recordings had a dubious recording and production quality, but were included to assess whether sound quality is an important determinant for jazz music preference. A listing of the music recordings is presented in Section III.2 of Appendix III.

2.3.3 Participants

Five male participants took part in the experiment (average age: 33). Of the five participants, three were amateur musicians and were educated in music theory and improvisation. To check whether or not the participants were frequent jazz listeners, they were instructed to rank eight freely recalled jazz musicians on personal taste and number of recordings they owned (Geringer and McManus, 1979). Only one participant devoted his private music listening time solely to jazz music, while the other participants appeared to listen to other music idioms as well (e.g., classical, popular and rock).

2.3.4 Interview

The interview contained five questions. The first three questions were meant to obtain data about the general opinion of the listening task and the compilation, and data about general music selection and purchase behaviour.The last two questions were meant to obtain data on the use of musical attributes when judging music. Only answers on the latter two questions are presented in the results. These questions were the following (translated from Dutch).

Q1 Elaborate on a characterisation of a preferred jazz recording and a rejected jazz recording in the play list. Try to stress both the commonalities and differences between the recordings on comparable aspects.

Q2 You have deliberately judged the jazz recordings while actively listening to them, which resulted in a category of *preferred* music and a category of *rejected* music. Can you state a rationale that explains this categorisation of *preferred* and *rejected* music?

The interview was not recorded, but answers to questions were directly written down in a condensed form.

2.3.5 Results

Each interview lasted approximately 45 minutes. One participant could not categorise the music into a *preferred* and *rejected* group and was unable to answer the questions; he stated that he liked all music. Data of this participant was left out of the analysis.

First question: Q1

All participants found it difficult to justify why one given music recording was chosen over another given music recording. Instead, they justified their decisions by discussing small groups either containing preferred or rejected music, independently, without comparing preferred music and rejected music. In total, twelve groups of preferred music and six groups of rejected music were discussed in that manner.

Characterisation of the music was done by giving positive and negative comments. Qualitative comments were given, rather than quantitative ones. In total, nine distinct positive comments and fourteen distinct negative comments were collected. Comments were assigned to three categories: musical performance, musical instruments, and musical content. Table 2.1 presents a complete list of these comments, their categorisation, and the number of participants who used these comments. Some comments were used more than once by the same participant while characterising different groups of music, which is not displayed in the results. Comments mostly referred to musical performance (20), and less frequently referred to musical content (3). In the comments referring to musical instruments (7), perceived quality of the vocals was most frequently commented upon (5). Negative comments were typically characterised by an excess of a particular musical feature (e.g., ‘too sweet’, ‘too skilful’). Positive comments typically expressed the achievement of an ideal (e.g., ‘good tempo’, ’nice swing’).

Of the twenty-three comments from the first question, twelve seem to have a corresponding attribute in the attribute representation. These comments concern tempo, rhythm, melodic and harmonic development, musical instruments, musicians, and general familiarity (standard or classic piece) of a music option. The remaining eleven comments are hard to match with attributes. These comments mainly concern personal opinions on musical performance.

Table 2.1. Positive and negative comments (translated from Dutch) and number of participants who used these comments.

	Positive		**Negative**	
Musical performance	‘good improvisation’	1	’too sweet’	2
	’good arrangement’	1	’too skilful’	2
	’good tempo’	1	’too light-footed’	1
	’nice swing’	1	’too messy’	1
	’nice melody’	1	’too lively and loud’	1
	’nice rhythm’	1	’too funky’	1
	’high playing intensity’	1	’too slow’	1
			’unclear playing style’	1
			’lacking feeling’	2
			’dislike playing style of Brecker’	1
Musical instruments	‘good vocals’	3	’bad vocals’	2
			’dislike guitars’	1
			’dislike drum solos’	1
Musical content	’familiar theme’	2	’too produced’	1

Second question: Q2

All participants could easily give criteria for their categorisation of *preferred* and *rejected* music. Qualitative criteria pertained either to *preferred* or *rejected* music. Again, *preferred* and *rejected* recordings were not comparatively evaluated. It appeared that all criteria given could be easily associated with general and objective musical features. Table 2.2 shows a complete list of musical features mentioned. This table also shows the number of participants who found desirable and undesirable features in the music, and attributes in the attribute representation that are likely to correspond with the musical features.

Table 2.2. Comments explaining the categorisation of *preferred* and *rejected* music and number of participants who found desirable (+) and undesirable (-) attribute values in the music. The right-most column shows some corresponding attributes in the attribute representation.

Comments	**Number of participants**	**Attribute name**
Ensemble strength	4 (+), 1 (-)	numMusicians, *is_played_by*
Type of musical (solo) instrument	3 (+), 2(-)	*has_solo_from*, Instrument
Musicians	2 (+), 2 (-)	*has_solo_from*, **Musician**
Jazz style and era	2 (+), 1 (-)	Style, *year*
Recording quality	1 (+), 1 (-)	-
Familiar theme	1 (+)	standard, classic
Different performances	1 (+)	Title
Rhythmic / Harmonic / Melodic structure	1 (+)	*tempo*, **Rhythm**, **Melody**
Live recording	1 (+)	live

The following responses are summarised in Table 2.2.

- All four participants stated that they preferred the recordings that were performed by ensembles of an ideal ensemble strength resembling a standard jazz quartet or quintet (piano, bass, drums, trumpet and / or saxophone). One participant rejected two recordings because there were too many horn players.

- Three participants emphasised their liking of a particular *type of musical instrument* as a solo instrument (piano, saxophone). One rejected drum solos in some recordings and another rejected electric instruments and stressed the inappropriateness of an electric bass as a solo instrument.

- Two participants favoured particular *musicians* and consequently preferred recordings of these musicians. In addition, two participants rejected particular musicians.

- Two participants stated that they preferred the recordings from a particular *jazz style or era* (the periods 1950-1960 and 1955-1970). One participant stated that he rejected all recordings from a particular jazz style (jazz rock).

- One participant rejected music with a ’cleverly produced’ *recording quality*, but rather preferred ‘noisy’ recordings. Two other participants stressed that recording quality was of no concern to them. This suggests that sound quality of a recording plays only a marginal role in jazz music preference.

- One participant commented that he preferred the recordings containing an improvisation on a *familiar theme*. Another participant liked the presence of *different performances* of the same theme in one compilation. One participant was concerned with the *rhythmic and harmonic structure* of a piece of music and preferred the recordings with a ‘non-trivial rhythmic and harmonic transparency such as modal improvisation’. One participant was keen on the *live recordings* in the compilation

2.3.6 Discussion

The participants found it difficult to put a comparative judgement of a preferred and rejected music option into words (First question: Q1). A local comparison of two jazz music recordings is probably limited due to the many musical features that have to be examined in a pair-wise fashion. These abstract features were hard to verbalise or to compare in pairs in the experimental setting used, but need to be experienced during listening.

Taking one small group at a time, either containing preferred or rejected music, was found a more convenient way for discussion than comparing two recordings. In general, the appraisal of the music in a group was motivated by either a positive or negative comment. Comments were qualitatively expressed and mostly referred to musical performance, which suggests that improvisation skills and interpretation of a piece of jazz music are the most important preference criteria for jazz. This seems plausible since musical training of jazz musicians and academic study of jazz music performance also have a significant focus on improvisation and interpretation (Coker, 1964; Berliner, 1994; Järvinen, 1995; Toiviainen, 1995). The results also showed that participants seem to have threshold criteria for musical features (e.g., tempo, improvisational skills, melody). If some features fall short of these thresholds (e.g., ’too slow’, ’too skilful’), the music is rejected. Otherwise, the music is preferred (e.g., ’nice melody’).

One of the purposes of this study was to find a direct relationship between the subjective comments and the fixed attribute representation of music options in Appendix I. Roughly half of the comments from the first question seems to have a corresponding attribute in the attribute representation. The other half was hard to match with attributes, as they concerned personal opinions on musical performance. Overall, this is not a convincing match.

Putting a rationale for a personal assignment of music recordings to a *preferred* and a *rejected* category into words was considered easy (Second question: Q2). Again, participants did not explicitly compare *preferred* and *rejected* music, but constrained their qualitative preference criteria on either *preferred* or *rejected* music. It is likely that participants could take a more global perspective on their preference, when discussing the whole compilation. This eased verbalisation. As a result, features and criteria elicited from the second question were more general and more likely to be objective than the comments of the first question.

Because of their general nature, all features and criteria expressed in the second question, except for recording quality, have a direct match with attributes in the attribute representation of music options. Recording or sound quality was not included in the attribute representation, because it was found difficult to devise a valid encoding for it. Many factors contribute to the overall sound of a recording such as the production style of producers and studio engineers, the place of recording, and live or studio registration. These factors can be matched with attributes in the attribute representation. However, results seem to indicate that sound quality of a recording plays a marginal role in the preference for jazz music. Since many old jazz recordings are corrupted by noise due to inferior recording techniques at that time, it is likely that jazz music listeners are accustomed to a noisy sound quality.

2.3.7 Conclusion

The focus group study examined what types of criteria are used in preference decisions of jazz music, and how these criteria can be reflected in a fixed attribute representation of jazz music. In general, when music listeners are allowed to have a global perspective on their music preference, for instance, when asked for a rationale, music listeners verbalise their music preference by referring to musical features that are general and that can be made objective. For example, they speak in terms of ensemble strength, instrumentation, musicians and music styles. In contrast, when music listeners need to compare or evaluate a small number of music options, music listeners mainly refer to personal and qualitative opinions about a musical performance. Only part of these opinions can be made objective, to some extent. In summary, since many musical features mentioned in the focus group study are general and can be made objective for inclusion in the attribute representation, a large part of music preference is likely to be captured by the attribute representation.

A valid attribute representation of jazz music provides a platform for further experimentation on music preference, choice and programming. For instance, it facilitates the development of adaptive features that are intended to automate parts of a music programming task.

Suggested music choice strategies

Jazz music holds many abstract features that need to be considered for a preference decision. Firstly, these features are hard to recall and formulate without listening to the music or without consulting information from secondary resources (e.g., album covers or booklets). Secondly, many features have a nominal or categorical nature (e.g., musicians, music styles), which are hard to compare to find differences. Consequently, music listeners are unlikely to make pair-wise comparisons of music options by examining features one-by-one, that is, using attribute-based processing. Music listeners are more likely to qualitatively judge music options per alternative, that is, using alternative-based processing, by applying threshold criteria for features. If features fall short of these thresholds, music options are rejected.

The results of this study provide some suggestions about what kind of choice strategies are employed by music listeners when programming music.

Firstly, the apparent absence of pair-wise comparisons of music options precludes the use of tournament choice strategies in large music collections; the results showed that music options are primarily evaluated one-by-one.

Secondly, the size of the compilation used allowed the participants to listen to all the music and to evaluate each music option. It is unlikely, however, that this choice behaviour will continue if the amount of music is large. Using a normative choice strategy in a large music collection simply exceeds human cognitive abilities.

Thirdly, the apparent use of threshold criteria, a selective number of attributes, and qualitative processing suggests a variant use of conjunctive or elimination choice strategies: (a) the use of a conjunctive strategy may then result in the selection of music options when their features comply with threshold criteria and further search is too intensive; (b) the use of an elimination choice strategy may then result in iteratively eliminating music options when their features fall short of a threshold criteria, until a set of music options is retained that makes up a (pre-)final selection.

CHAPTER 3

Preferred features of music

Judging music on its suitability for a given context-of-use is defined as a temporary activity, in which music preference is developed. It is assumed that the decision to either prefer or reject a given music recording is based on criteria of desirable or undesirable musical features in the music. These criteria of a music listener are covert however. If the purpose is to learn music preference of a music listener, inductive learning algorithms can be used to uncover part of these important features. Four algorithms, namely ID3, ID3-IV, ID3-BIN and INFERULE, were implemented and comparatively evaluated. All algorithms belong to the class of top-down decision tree constructors. The questions posed at the evaluations were as follows: Which algorithm produces the highest categorisation *accuracy* on a collection of judged music by using the most *compact* tree? and which algorithm produces trees that have the highest 'preference fit'? *Accuracy* was defined as the proportion of correctly categorised music options. *Compactness* was defined as the proportion of leaves of a decision tree when compared to the least compact tree that is possible. Note that the least compact tree has a number of leaves that equals the number of options that were used in the construction of the tree. 'Preference fit' was established by ranking decision trees according to their correspondence with a given music preference. Results of the evaluations showed that INFERULE produced decision trees yielding the highest categorisation *accuracy* while utilising the most *compact* tree and with the best 'preference fit'. Among the four alternatives, INFERULE is the best choice to learn about the music preferences of a music listener and will be used in more elaborate interactive and adaptive systems intended to ease and speed up music selection and programming.

Music listeners can rely on many cues to determine their music preference. Important cues originate from the context-of-use, background knowledge of the music and the musical content. Given these cues, music listeners are able to judge what music is preferred and what music is rejected to suit a given context-of-use.

In this chapter, it is argued that preference judgement of music can be seen as a music categorisation process. This process is assumed to be driven by criteria on desirable or undesirable musical features in the music. Since these criteria are not directly observable, inductive learning algorithms were implemented that attempt to uncover these criteria. To achieve this, the inductive learning algorithms used require music that has been pre-categorised by the listener as either preferred or rejected. Uncovering preference criteria is particularly important for an interactive music player, if it is supposed to learn about music preferences and adapt to the listener's music choice behaviour. In this way, an inductive learning algorithm is an essential part of more elaborate interactive adaptive systems for music selection and programming. As only the best music preference learning algorithm should be used, four algorithms were comparatively evaluated for the music domain. It appeared that inductive learning algorithms differed considerably in their performance to uncover criteria related to music preference.

3.1 Categories, concepts and music preference

3.1.1 Common and ad hoc categories

People group things into various types of *categories* that help them to comprehend their environment (Smith, 1990; Estes, 1994). C*ommon taxonomic* categories, for instance, reflect how natural kinds and artifact kinds are organised in the world, and what features are shared by them. In most music shops, music is categorised by properties of musical performance, different musical idioms, different music styles and language of the lyrics. This should help customers to find the appropriate music to buy.

People also spontaneously develop *ad hoc* or *goal-derived* categories of things to temporarily serve a particular goal in their daily life (Barsalou, 1983; 1991). Prompted by a goal, people construct *ad hoc* categories by considering only features of things that are relevant for the goal. For instance, the goal of collecting things for a backpacking vacation requires the collection of things that can be easily carried along.

3.1.2 Concepts and music preference

An adequate description of a category is termed a *concept* of the category (Smith, 1990; Estes, 1994). The use of concepts is a way to reduce information load, to gain a better understanding of the underlying subject, and to generalise on the basis of it.

A concept of a *common taxonomic* category is formed by encountering category members and learning from them (Medin and Smith, 1981). These concepts can be characterised in two ways, either by *a list of critical features* that must be common to all category members and that are sufficient and necessary to discern category membership (Katz and Postal, 1964), or by *family resemblance* of category members in which features may vary for each individual member (Rosch and Mervis, 1975). Both ways assume that similarity between items governs the way in which concepts and categories are formed.

A concept of an *ad hoc* category is best characterised by the set of features that are relevant to the goal. These features represent *ideals* that should serve the goal optimally and are established by a prior reasoning process (Medin and Barsalou, 1987; Barsalou, 1991). However, reasoning about *ideals* does not necessarily lead to the identification of concrete category members. For instance, when going on a diet, it is wise to eat food that is low in calories and rich in fibre and vitamins, but these ideals do not precisely say what food one is allowed to eat and what food could better not eat. And when going on a backpacking vacation, it is wise to pack things that are low in weight and small in size, but a thorough search in the house is still needed to find these ideal things.

As far as music judgement is concerned, it usually serves a temporary goal: the goal to find music that is suitable for a given context-of-use. Prior reasoning involved in forming an *ad hoc* concept may include some musical features of the preferred music or some features that definitely should *not* be in the preferred music. In this way, developing music preference resembles the process of developing an *ad hoc* concept.

However, the result of a prior reasoning stage delineates only an incomplete description of preferred music. What constitutes a desirable or undesirable musical feature is subject to a wide variance and generating concrete examples of preferred music may be a difficult exercise. Upon listening to the music, desirable and undesirable musical features can be better defined and concrete examples of preferred music can be easier identified. In other words, music preference is refined while listening to music. For instance, when judging music for a dance party, music listeners may beforehand generate some desirable musical features including tempo, rhythmic structure and type of instruments. Criteria and ranges for these features will be set during music listening.

3.2 INDUCTIVE LEARNING ALGORITHMS

A context-of-use is supposed to produce constraints and opportunities for music listening, entailing specific criteria for musical features. When judging a music programme for a context-of-use, music listeners are asked to form a category of preferred music and a category of rejected music. If musical features can be adequately expressed in fixed attribute values of the music, inductive learning algorithms can be used to uncover part of these criteria in order to learn the music preference of a music listener.

Inductive learning algorithms attempt to learn concepts from pre-categorised items, that is, by supervised learning. A class of inductive learning algorithms constructs a decision tree in which concepts are summarised in terms of attribute values of the items that should be sufficient to discern category membership of items (Breiman, Friedman, Olshen and Stone, 1984; Quinlan, 1979; 1986; 1993). Originally, these algorithms were meant to disclose concrete situation-action rules of a domain expert by examining large sets of pre-categorised data in order to support the development of knowledge-based expert systems (Quinlan, 1979; 1986). The use of these algorithms here is essentially intended to uncover attribute values relevant to music preference from a judged music programme. These identified attribute values are then represented by a tree structure that can be easily inspected.

The decision tree construction process will be described in Section 3.2.2. A given music option is categorised using a decision tree by recursively walking through the tree by taking appropriate branches. This will be described first.

3.2.1 Categorisation by a decision tree

In our case, a decision tree is a knowledge structure that ideally sorts music options into the categories *preferred* and *rejected*. Figure 3.1 shows a decision tree which is consistent[1] with a set of 14 preferred or rejected jazz music options shown in Table 3.1. The attribute values of these music options are only for illustrative purposes, since some value combinations do not reflect existing music recordings.

If we want to know whether an ‘unseen’ music option (a music option that is not used in the tree construction) would be preferred or rejected, the categorisation process of the given music option is started at the root of the tree. Attribute values on the branches of the tree are compared to the value of the corresponding attribute of the music option. A branch is then taken that is appropriate to the outcome of the comparison. This comparison and branching process continues recursively until a

1. A decision tree is called *consistent* with the items it is based on, that is, its training set, if it correctly categorises all these items.

leaf is encountered, at which time the *predicted* preference category of the option is known.

Table 3.1. A set of 14 jazz music options that are either preferred (+) or rejected (-) and an 'unseen' music option for which its preference is unknown.

Option	**Title**	**Musician**	**Tempo**	**Live**	**Preference**
1	'My Funny Valentine'	Miles Davis	fast	yes	-
2	'My Funny Valentine'	Miles Davis	slow	no	+
3	'My Funny Valentine'	Michael Brecker	slow	no	+
4	'My Funny Valentine'	Michael Brecker	fast	no	-
5	'My Funny Valentine'	Chet Baker	slow	yes	+
6	'Donna Lee'	Charlie Parker	fast	yes	-
7	'Donna Lee'	Jaco Pastorius	fast	no	+
8	'Donna Lee'	Jaco Pastorius	fast	yes	-
9	'Donna Lee'	Miles Davis	fast	no	+
10	'Donna Lee'	Chet Baker	slow	yes	-
11	'Body & Soul'	Miles Davis	slow	yes	+
12	'Body & Soul'	Jaco Pastorius	slow	yes	+
13	'Body & Soul'	Charlie Parker	fast	yes	+
14	'Body & Soul'	Chet Baker	slow	no	+
'unseen'	'My Funny Valentine'	Stan Getz	slow	no	unknown

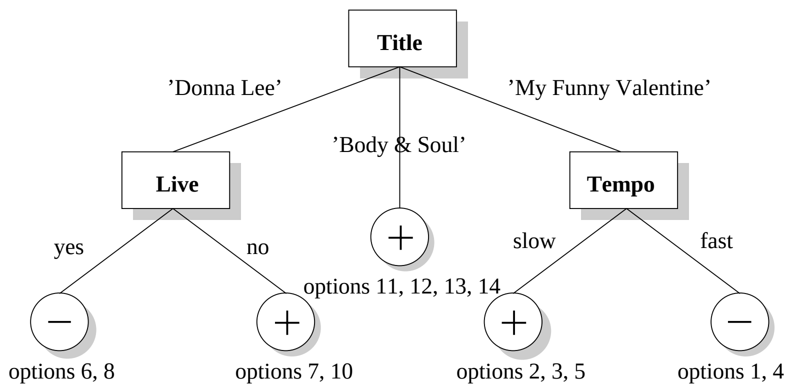

Figure 3.1. A decision tree constructed from the data in Table 3.1.

As a categorisation example, an ‘unseen’ music option in Table 3.1 is played by the musician ‘Stan Getz’; this musician was not encountered while constructing the tree. The categorisation process for this option takes successively the branches labelled as ‘My Funny Valentine (Title)’ and ‘slow (Tempo)’ and concludes by stating that this music option would be preferred.

We can quite easily read from the decision tree in Figure 3.1 that preferred music options are titled ‘Donna Lee’ that were not recorded in front of a live audience, or are titled ‘Body & Soul’, or are titled ‘My Funny Valentine’ that were performed at a slow tempo.

3.2.2 Construction of a decision tree

A decision tree is constructed from a set of pre-categorised music options, that is, a judged music programme. Options are categorised as either *rejected* or *preferred*. The number of possible decision trees that can be constructed from a judged music programme is too large to search exhaustively. Finding an *optimal* decision tree for any set of data with respect to most performance criteria cannot be determined non-deterministically in polynomial time. It has been proven to be *inherently* intractable, that is, NP-complete (Hyafil and Rivest, 1986; for theory on NP-completeness, see Garey and Johnson, 1979). Therefore, a decision tree is incrementally constructed by a greedy, non-backtracking search algorithm in which the search is directed by a heuristic function. This heuristic function is based on local information about how well an attribute partitions the set of options into categories. 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 characterised by replacing a leaf of the tree under construction by a new subtree of depth one. This subtree consists of a node, which carries an attribute that provides the best category partition on heuristic grounds, and branches that represent the partitions along the values of that attribute. This process is continued until a stopping criterion is reached (Breiman, Friedman, Olshen, and Stone, 1984; Quinlan, 1986). A general stopping criterion, which is also used in our implementation, is to stop when the partitions only contain options of one category. If the stopping criterion holds, the leaves of a decision tree carry options that are all either preferred or rejected (also see Figure 3.2). The constructed tree then contains interior nodes and branches specifying attributes and their values along which the music programme was originally partitioned into the categories *preferred* and *rejected*.

Decision tree construction algorithms differ in the type of heuristic function for attribute selection and the branching factor on each interior node. Four different algorithms were implemented: ID3 (Quinlan, 1986), ID3-IV (Quinlan, 1986), ID3-BIN and INFERULE (Spangler, Fayyad and Uthurusamy, 1989). The use of decision trees in the music domain also requires some domain-specific adjustments in both the tree construction process and the tree categorisation process.

3.2.3 Attribute selection heuristic

The heuristic to select an attribute is the most important aspect determining the tree construction process and its result (Mingers, 1989; Fayyad and Irani, 1992). Three heuristics are considered here: *information gain*, *gain ratio* and *relative goodness*. Let the set of N pre-categorised options be denoted as S. Each option $x \in S$ belongs to one of the categories in C_m, $m = 1 \ldots M$. If the set S represents a judged music programme containing only preferred and rejected music options, two categories

($M = 2$) will suffice. If an attribute A has K distinct values, a partitioning of S along all A's values yield K disjoint subsets denoted as S_k, $k = 1 \ldots K$.

Information gain

An information-theoretic heuristic is based on selecting the attribute that aims at minimising the depth of the decision tree, that is, minimising the cost to categorise an option (Quinlan, 1986). According to Quinlan, a decision tree can be interpreted as an information source that returns the message of what category an option belongs to. The expected amount of information, that is, entropy, of this message with respect to the set S can be approximated by,

$$E(S) = -\sum_{m=1}^{M} P_{S, C_m} \log_2 P_{S, C_m} \tag{3.1}$$

where P_{S, C_m} is the proportion of options in S that belong to category C_m.

Partitioning the set S along A's values yields the expected amount of information,

$$E(S, A) = \sum_{k=1}^{K} \frac{|S_k|}{N} E(S_k) \tag{3.2}$$

where $|S_k|$ denotes the number of options in the subset S_k.

The suggested selection for the attribute to extend the decision tree, is now the attribute A that has the highest *information gain*, that is, for which

$$Gain(S, A) = E(S) - E(S, A) \tag{3.3}$$

is maximal. By selecting this attribute A to partition set S, categories are least randomly distributed over the disjoint subsets S_k. An algorithm that uses *information gain* as the attribute selection heuristic is called ID3 (Quinlan, 1986).

Gain Ratio

A well-known drawback of the use of information gain is its tendency to select attributes that take on many values. An *ad hoc* solution is to penalise broad attribute value ranges (Quinlan, 1986) in the calculation of the information gain, resulting in a *gain ratio* measure,

$$Gain\ Ratio(S, A) = \frac{Gain(S, A)}{-\sum_{k=1}^{K} \frac{|S_k|}{N} \log_2 \frac{|S_k|}{N}} \tag{3.4}$$

An algorithm that uses the *gain ratio* heuristic is referred to as ID3-IV.

Relative goodness

In some domains, the available attributes are not sufficient to discern category membership for a given option. In these inconclusive cases, it is assumed to be more appropriate to use a heuristic known as the *relative goodness* of an attribute value than the *information gain* of an attribute (Spangler, Fayyad and Uthurusamy, 1989). The *relative goodness* heuristic selects an attribute value when the category distribution in the resulting subsets differs considerably from the original set. As this heuristic considers attribute values instead of attributes, a binary decision tree will result, by definition.

An estimation of the expected number of options that contains the k-th value of attribute A as well as belonging to category C_m is

$$E_{A, C_m}(k) = \frac{N_{S, C_m} \cdot N_k}{N} \tag{3.5}$$

where N_{S, C_m} is the number of options in set S that belong to category C_m and N_k is the number of options that contains the k-th value of A.

The standard error associated with these expected numbers is

$$SE_A(k) = \sum_{m=1}^{M} \frac{E_{A, C_m}(k) \cdot (N_{S, C_m} - E_{A, C_m}(k))}{N_{S, C_m}} \tag{3.6}$$

The geometrical distance between the expected and the actual (observed) number of options that contains the k-th value of attribute A as well as belonging to category C_m is

$$DI_A(k) = \sqrt{\sum_{m=1}^{M} (E_{A, C_m}(k) - N_{A, C_m}(k))^2} \tag{3.7}$$

where $N_{A, C_m}(k)$ is the actual (observed) number of options that contains the k-th value of attribute A as well as belonging to category C_m.

Now, the *relative goodness* of an attribute A and its k-th value is defined as

$$R_A(k) = \frac{SE_A(k)}{DI_A(k)} \tag{3.8}$$

The suggested selection for the attribute value to branch on is the attribute value that has the *minimal* relative goodness. All remaining values of the corresponding attribute are lumped into a separate *default* branch.

An algorithm that uses the *relative goodness* heuristic to construct a *binary* decision tree is referred to as INFERULE (Spangler, Fayyad and Uthurusamy, 1989).

3.2.4 Branching factor

If the ID3 or ID3-IV algorithm selects a nominal attribute to partition a given set of options, a branch will be created for each nominal value of that attribute. Branching on each attribute value may include branching on *irrelevant* attribute values, which may result in *overspecialisation* to the given set of options (Cheng, Fayyad, Irani and Qian, 1988). In other words, the decision tree will not generalise to categorise 'unseen' options correctly. A method to reduce the overspecialisation effect is to *retrospectively prune* the decision tree (Quinlan, 1987). Overspecialisation can also be avoided during tree construction by reducing the number of attribute values to branch on by considering only *relevant* attribute values. Attribute relevance can be based on the attribute selection heuristics mentioned (for definitions on attribute relevance and irrelevance, see John, Kohavi and Pfleger, 1994). An extreme scheme involves the creation of a binary tree with one branch for the most relevant value of the attribute, and another *default* or *otherwise* branch to collect all other values. An algorithm that uses the *information gain* heuristic to construct a *binary* decision tree is referred to as ID3-BIN (for other methods to reduce the branching factor during tree construction, see Cheng, Fayyad, Irani and Qian, 1988).

3.2.5 Adjustments for the music domain

Musical attribute domains include not only nominal or categorical attribute values for representing data such as music styles and musical artists, but also includes: attributes with multiple nominal values, for example, to represent a group of musicians; attributes with an ordinate value structure, for example, to represent different types of musical instruments; and quantitative attributes, for example, to represent tempo and year of recording. During the collection of attribute data, some musical attribute values could not be extracted from information sources. The mechanism to cope with these missing attribute values is discussed. In addition, solutions to uncertainty in the attribute selection procedure during tree construction, and uncertainty in the categorisation process are presented.

Ordinal and quantitative attributes

Although decision tree construction algorithms were originally developed for nominal attributes, they can be easily extended to deal with quantitative or ordinal data as well. The decision of whether a quantitative or ordinal attribute of a music option surpasses a given threshold or not, that is, $A \leq T$ or $A > T$, is used to partition a set of music options at a given level of the tree. The value of the threshold T needs to be determined by first sorting all options in the set in terms of increasing value of the attribute A. If N options are sorted, $N-1$ successive pairs of values define a candidate for a threshold value, for example, the mid-point of the two consecutive numerical values is used in the case of quantitative attributes. The best candidate is retained by evaluating all $N-1$ possible threshold values, taking into account the category distribution to the left and the right of the possible threshold. More specifically, the two disjoint subsets of options at the left and the right of the possible threshold are evaluated by one of the heuristics such as information gain, information ratio or relative goodness (Quinlan, 1986). The best threshold to partition the set is then used in the attribute selection procedure.

Attributes with multiple nominal values

Attributes with multiple nominal values may refer to the group of musicians or the set of musical instruments on the recording. In that case, special treatment is required to decide on what attribute value, that is, on what musician or instrument, the set of options will be partitioned. Our approach selects the value that occurs most frequently in the judged music programme; a frequently occurring attribute value in a music programme may be relevant for music preference, for example, several recordings in which a musician participates. This value is then used to compete with values of other attributes in the attribute selection procedure. Another approach, not implemented here, would be to evaluate each value separately on the partitioning it causes, but this may cause branching on irrelevant or unique attribute values.

Attributes with an ordinate value structure

An attribute may have an ordinate structure; attribute values are then superordinate to other attribute values which allows attribute values to be put in a hierarchical tree structure. For instance, musical instruments are arranged in classes of string instruments, brass instruments, woodwind instruments, percussion instruments, keyboard instruments, electronic instruments and vocal instruments. Musical instruments, for example, guitar, trumpet, saxophone, are then assigned to one of these classes. The question now is on what descriptive level the set of options will be partitioned. Both levels in the hierarchical tree are considered, starting at the upper level. The set of options is partitioned along the attribute values pertaining to a given level and the resulting subsets are evaluated by using one of the heuristics such as information gain, information ratio or relative goodness. The best level to partition the set is then retained and subjected to the attribute selection procedure.

Missing attribute values

Some attribute values are missing due to failure to record, or unfamiliarity. For example, the place and the year of recording may not be known. At least five methods are proposed to overcome this problem during decision tree construction. In the first method, options are simply ignored if they have missing attribute values (Breiman, Friedman, Olshen, and Stone, 1984). In the second method, the value of the heuristic function is reduced proportionally to the number of options with missing values for an attribute (Quinlan, 1986; 1989). In the third method, the missing value of an attribute is filled in by determining its most likely value in terms of the value of other attributes (Quinlan, 1989; Langley, 1996). In the fourth method, the missing value of an attribute is filled in by the most common value for a nominal attribute (Quinlan, 1989), the modal value (most frequently occurring one) for other qualitative attributes, and the mean value for quantitative attributes (Langley, 1996). In the fifth method, a separate value denoted as ‘missing’ is used (Langley, 1996) or if a *default* branch is used in the tree construction, this *default* branch can be used to lump all options with a missing attribute value. The last two methods were implemented and compared. The method of using a separate ‘missing’ value had the least influence on the resulting decision trees and was therefore used in the algorithms.

Attribute selection uncertainty

In the process of tree construction, partitions of a given set of pre-categorised options along different attributes (or values) may evaluate equally under an attribute selection heuristic. The tree construction process then can not select on what attribute to proceed solely on the basis of the heuristic. In that case, the attribute is chosen that has the least number of distinct values in the given set of options, that is, the lowest branching factor. This heuristic provides the smallest lower bound on the number of leaves, but does not necessarily lead to a tree with the smallest number of leaves. Obviously, it does not work for binary tree construction. No effective means were implemented to resolve the attribute selection uncertainty problem for binary trees, besides the fact that attributes are evaluated in a random order. If attributes evaluate equally under the heuristic, the attribute that was evaluated first in this random order was selected to proceed with the tree construction. This method may be improved by ordering the list of attributes on relevance to music preference categorisation, or by using domain knowledge.

Categorisation uncertainty

In the categorisation process description in Section 3.2.1, it was incorrectly assumed that the decision tree provides an appropriate branch for each attribute value at each decision point, and that the process always produces a categorisation of a given music option. It may however be the case that the tree construction process does not encounter particular attribute values. As a result, it cannot define branches for these ‘unseen’ attribute values at an interior node. The categorisation process may then ‘get stuck’ at a particular ‘inconclusive’ interior node because it encounters an ‘unseen’ attribute value. It might also be the case that the stopping criterion during tree construction cannot be achieved because the construction process runs out of available or decisive attributes. Not reaching the stopping criterion implies that the resulting tree contains ‘inconclusive’ leaves that carry the category *preferred* as well as the category *rejected*.

In both cases, for some music options, the decision tree cannot predict whether they are preferred or rejected. In other words, the decision tree is uncertain about the preference for some options. Several methods to resolve the problem of ‘inconclusive’ leaves are proposed (Breiman, Friedman, Olshen and Stone, 1984; Quinlan, 1986b). For instance, the *majority method* assigns that category to a ‘inconclusive’ leaf that has the most occurring items. This rule, though not intended for that purpose, may also be used to force a decision at an ‘inconclusive’ interior node. The majority method was implemented for both ‘inconclusive’ leaves and ‘inconclusive’ interior nodes. However, it appeared from results of an evaluation that applying this rule to our two-category problem gave undesirable results. The number of correctly categorised music options turned out to be highly dependent on the proportion of preferred options in a judged music programme, instead of the attribute values of options. It was concluded that the majority rule application should be left out of the final algorithm implementation. In other words, the categorisation process may still remain uncertain about the preference categorisation of some music options.

3.3 Evaluation of inductive learning algorithms

Performance of a tree-construction algorithm directly relates to the quality of its constructed decision trees, which can be measured in several ways. The quality of the decision trees were empirically evaluated in the music domain, in two experiments.

3.3.1 Quality of a decision tree

The common empirical method to assess the quality of an inductive learning algorithm in a given domain is to measure *accuracy* or *error rate* of constructed trees while categorising items that were not used in the tree construction (Mingers, 1989). For any type of application domain, it is hard to decide what decision tree is the most optimal one among the alternatives, and by what algorithm this decision tree is likely to be produced (Fayyad and Irani, 1990; Murphy and Pazzani, 1994). For instance, many different decision trees can be constructed that are all consistent with a given set of pre-categorised items such as a judged music programme. It is known however that the greedy and heuristic nature of the algorithms only results in suboptimal trees. In addition, the level of *accuracy* cannot be controlled during tree construction; it can only be measured after the complete tree has been constructed. These limitations make quality assessment of decision trees a tedious job lacking convincingly general findings (Fayyad and Irani, 1990).

Performance criteria pertaining to the topology of the decision tree are therefore proposed as alternative measures for tree quality (Quinlan, 1987; Cheng, Fayyad, Irani and Qian, 1988; Fayyad and Irani, 1990). These performance measures are better informers of tree quality during the construction of the tree than accuracy measures. Variants of tree construction algorithms exist that attempt to minimise the size of a tree (Van de Velde, 1990). Justified by Occam's Razor (Blumer, Ehrenfeucht, Haussler and Warmuth, 1987), the philosophy is to *maximise the tree compactness*, that is, to minimise tree size. Small decision trees are assumed to yield high accuracy on 'unseen' data. In particular, Fayyad and Irani (1990) have proved theoretically that by minimising the number of leaves of a tree, the error rate of the tree may be reduced in a *probabilistic and worst-case* sense. This theoretical finding suggests that it is a good strategy to modify algorithms to produce trees with *fewer leaves*, because it is expected that these trees are better categorisers of 'unseen' data solely on the fact that they have fewer leaves. However, if domains do not cover the theoretical assumptions, applying this theoretical finding in an *absolute and average-case* sense should be approached carefully (Murphy and Pazzani, 1994). For many of the problems investigated by Murphy and Pazzani (1994), a lower accuracy for smaller decision trees was observed than for slightly larger ones.

3.3.2 Experiment I: Accuracy and compactness of decision trees

The purpose of Experiment I was to find out what algorithm produces decision trees with the highest level of categorisation *accuracy* and the highest level of *compactness* on the same four data sets. Each data set was the same music collection; the music was however judged by another music listener.

Design

A design with two within-subject independent variables was used. The variable *type of algorithm* consisted of the four algorithms under study, namely ID3, ID3-IV, ID3-BIN and INFERULE. The other variable *training set size* was used to vary the number of judged music options to be subjected to the algorithm. A training set consisted of the set of judged music options. The following training set sizes were defined: 15 and 150 music options. The use of different training set sizes was intended to assess performance differences, but also to assess the *overspecialisation* effect of the algorithms.

Measures

Accuracy was defined as the number of music options that were correctly categorised by a decision tree as being either *preferred* or *rejected*. The categorisation process followed the description as presented in Section 3.2.1. Music options for which the decision tree was uncertain about the preference categorisation were termed as *incorrectly* categorised. The use of an *accuracy* measure provides a summation of *hits* and *correct rejections*, but ignores any information about *misses* and *false positives*. However, a bad performance in terms of the latter two aspects may have a serious impact, especially if definitely preferred music is missed or 'awful' music is considered preferred.

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 option in a separate leaf. Thus, if 150 pre-categorised options are used to construct a tree, the least compact tree will contain 150 leaves. In contrast, the most compact tree contains only two leaves, in which each leaf corresponds to either *preferred* options or *rejected* options. By definition, a high tree compactness rather counter-intuitively corresponds to a low value of the measure *compactness*, that is, just above zero, and a low tree compactness with high values, that is, nearly one. An observation of both a low *compactness* and a low *accuracy* indicates that overspecialisation to the training set has occurred.

The prime performance criterion for a decision tree to be maximised is its accuracy on *all available* music options in order to generalise music preference beyond the scope of a training set. Therefore, *accuracy* measurement is based on all available music options, instead of only those 'unseen' ones that were not used in the tree construction.

A tree's compactness should also be maximised, that is, the number of leaves (the measure *compactness*) should be minimised. Note that the number of leaves of a tree equals the number of different paths from root to leaf or categorisation rules in a tree. Each path contains a limited number of attribute values to be tested during preference categorisation.

Hence, obtaining a high *accuracy* with a highly *compact* tree suggests that music preference can then be adequately expressed by a limited set of musical attribute values that holds for many music options.

Procedure

Participants were given 20 randomly created music programmes. Each music programme contained 15 different music options. The programmes had no overlap in music options. In total, 300 different music options were presented.

Participants were instructed to listen to each music programme individually at any convenient time and pace, by using an interactive system. A small introduction on using this system was given. Since computer resources were only available at the laboratory, music listening took place at the participant's workplace. Participants, confronted with a sequence of music options in a music programme, only had to decide which option they preferred and which option they rejected (binary forced choice). In the process of listening, participants were allowed to compare music options freely in any combination and cancel any judgement already expressed. Judging a single music programme had to be completed in one extended period of time. Therefore, the task could not be stopped (or if it had to be stopped, results were not saved) if all options in a programme were not indicated as either preferred or rejected. No instruction on preference instantiation was provided. The experiment was concluded when all music programmes had been judged.

Participants

Four male participants took part in the experiment. They were all colleagues at the same research laboratory. All participants were frequent listeners of jazz music; for admission to the experiment, participants had to be able to rank eight freely recalled jazz musicians on personal taste and number of recordings owned (Geringer and McManus, 1979). The average age was 32 (min: 29, max: 35). All participants had received a musical education and played a musical instrument on a regular basis.

Test equipment and material

A music database comprising 300 one-minute excerpts of jazz music recordings (MPEG-1 Part 2 Layer II 128 Kbps stereo) from 100 commercial CD albums served as test material. The music collection contained 12 popular jazz styles. These styles cover a considerable part of the whole jazz period. Each style contained 25 music recordings. The attribute representation of the music options is shown in Appendix I. Pilot experiments showed that the brevity and sound quality of the excerpts did not influence judgement.

The test equipment for each participant consisted of a Silicon Graphics Indy (SGI) workstation and two Fostex 6301 B personal monitors (combined amplifier and loudspeaker system). A SUN Sparc-5 workstation was used as a music storage device. Music was transported as MPEG data over the Ethernet; there were no noticeable interruptions in the data stream. Real-time MPEG software decoding took place at the SGI machine.

Interactive system

For this experiment, an interactive system was implemented to listen to and judge music programmes using a standard mouse and Graphical User Interface (GUI). The control panel of the system had a textual display in which the option sequence number, the title, and the names of composers and main artists of a music option were presented. No other attribute information was displayed. Only one music option could be presented, played back and judged at a time. ‘Previous’ and ‘Next’ buttons were provided to navigate through the sequence of music options making up a programme. ‘Play’ and ‘Stop’ buttons were provided to control music play back. Two buttons were provided to indicate preference or rejection of a music option.

The interactive system was implemented in C / C++ using the OSF / Motif toolkit on top of the Silicon Graphics Indy IRIX 5.3 operating system. The GUI was created with the X-Designer GUI builder.

Analysis

All judgement data were collected for each participant resulting in four differently judged data sets. The data sets are referred to as Participant A, B, C and D. Each data set was subjected to the four algorithms in the following three-step analysis method. Firstly, a subset, henceforth a training set, was randomly drawn from a data set. Secondly, this training set was used to construct a decision tree for each algorithm. Thirdly, the *accuracy* and *compactness* was measured for each decision tree. In order to reduce random variation in the measurement, 100 iterations of this three-step analysis method were performed. For each iteration, a new training set was randomly constructed. Mean *accuracy* and mean *compactness* over all iterations were calculated.

Results

Participants preferred quite distinct proportions of the music collection ranging from 58% to 84% (see Table 3.2). Obviously, if participants prefer only a small part of the music, it is difficult for a categoriser to yield a satisfactory *accuracy* level.

Table 3.2. Number and proportion of preferred music options from a music collection containing 300 music options. In the right-most column, base-line *accuracy* of a *random* categoriser is shown.

Participant	No. preferred	Proportion preferred	Base-line *accuracy*
A	175	0.58	0.51
B	177	0.59	0.52
C	211	0.70	0.58
D	253	0.84	0.73

In order to compare the *accuracy* of decision trees to a base-line level, one can imagine the use of a *random* categoriser that randomly states that a music option is either *preferred* or *rejected*, according to a participant's proportion of preferred music but regardless of the given music option. The base-line *accuracy* for a decision tree categoriser is then determined by the yield of a *random* categoriser for two categories according to the function

$$\text{yield} = p^2 + (1 - p)^2 \tag{3.9}$$

where p is the participant's proportion of preferred music.

The function in Equation (3.9) has a minimum value at $4p - 2 = 0$ which is at $p =$ 0.5. At that point, the minimal yield of a *random* categoriser is also 0.5. Thus, a real base-line for a decision tree categoriser to obtain is determined by the 50% *accuracy* level, as any *random* categoriser simply performs better than this level. It is also clear from Equation (3.9), that the maximum yield of a *random* categoriser equals 1, if a participant prefers all music ($p = 1$) or has rejected all music ($p = 0$). The yield of a random categoriser for each participant are shown as base-line *accuracies* in Table 3.2.

No base-line level for *compactness* can be established from a *random* categoriser as it does not impose any structure on the data. However, the most compact tree that partitions two categories consists of only two leaves.

Results on mean *accuracy* and mean *compactness* of the trees are shown in Figure 3.2. Base-line *accuracies* for each participant are shown as solid lines.

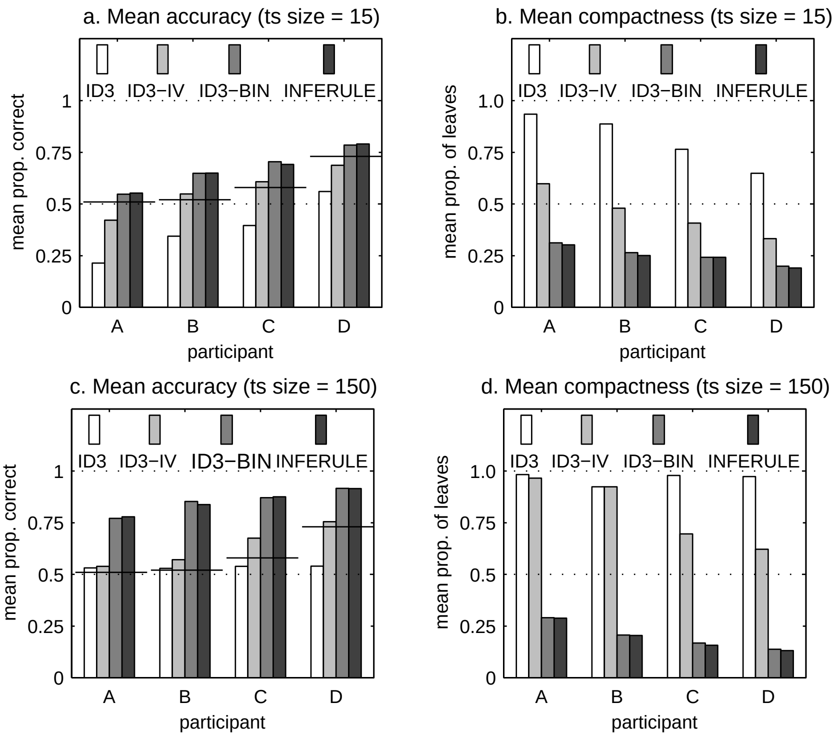

Figure 3.2. The left-hand panels (a) and (c) show mean *accuracy* (proportion correctly categorised) of decision trees constructed by four different algorithms, that is, ID3, ID3-IV, ID3-BIN and INFERULE, from a small and large training set, that is, *ts size* is 15 and 150 respectively. The training sets were randomly drawn from data sets of four participants, that is, A, B, C and D. Solid lines represent base-line *accuracy*. The right-hand panels (b) and (d) show mean *compactness* (mean proportion of leaves that would be achieved by the least compact tree) of the same trees.

As shown in the left-hand panels (a) and (c) in Figure 3.2, mean *accuracy* depended on a participant's proportion of preferred music. If more music was preferred, decision trees yielded a higher mean *accuracy*.

A MANOVA with repeated measures was conducted with *training set size* (2), and *type of algorithm* (4) as independent within-subject variables. For the analysis, the base-line *accuracy* level of each participant was subtracted from all mean *accuracy* values, because the most relevant part of *accuracy* is the part that is gained by using an inductive learning algorithm. This calculated measure is termed *gained accuracy* and was used as the dependent variable.

A significant main effect for *training set size* was found ($F(1,3) = 24.37$, $p < 0.05$). As shown in Figure 3.2, a higher mean *accuracy* (and *gained accuracy*) was observed in the case of a large training set. In the case of a small training set, decision trees obtained, on average, a lower *accuracy* level than the base-line *accuracy* levels (mean *accuracy* and mean *gained accuracy* for different training set sizes: 0.57 and -0.01 (15), 0.72 and 0.13 (150)). A trend of *gained accuracy* improvement by increasing training set size was also observed by analysing intermediate training set sizes such as 30, 60, and 100 music options. It was also observed that decision trees were increasingly less consistent for increasingly larger training sets. In other words, trees were unable to categorise correctly the options from large training sets with which the tree was constructed.

A significant main effect for *type of algorithm* was found (F(3,1) = 1179.74, $p < 0.001$). When the means were compared, it was found that there was a significant *gained accuracy* difference between ID3 and the other three algorithms (F(1,3) = 873.21, $p < 0.001$), and between ID-IV, on the one hand, and ID3-BIN and INFERULE, on the other hand (F(1,3) = 131.24, $p < 0.005$). Both ID3-BIN and INFERULE produced trees with the highest mean *gained accuracy.* Results on *accuracy* were generally poor for ID3 and ID3-IV. As shown in Figure 3.2, the *accuracy* level of ID3 trees did not even reach the base-line *accuracy* level in six of the eight cases, resulting in an overall negative *gained accuracy* (mean *gained accuracy* for all four algorithms: -0.13 (ID3), 0.02 (ID3-IV), 0.18 (ID3-BIN), 0.18 (INFERULE)).

In addition, a nearly significant interaction effect for *training set size* and *type of algorithm* was found (F(3,1) = 5.24, $p = 0.05$). When the means were compared, it was revealed that this effect was caused by a lower difference in *gained accuracy* for ID3-IV between both training set sizes than for ID3-BIN and INFERULE (F(1,3) = 23.36, $p < 0.05$). Thus, ID3-IV improved less on *accuracy* for a larger training set than ID3-BIN and INFERULE. Closer inspection of the data revealed that the ID3 and ID3-IV trees were unable to categorise options with ‘unseen’ attribute values; there were many uncertain categorisations. In contrast, as the ID3-BIN and INFERULE binary trees have a *default* branch at each interior node, the categorisation process lumps all ‘unseen’ attribute values in this *default* branch which reduces categorisation uncertainty.

A MANOVA with repeated measures was conducted with *training set size* (2), and *type of algorithm* (4) as independent within-subject variables. As *compactness* is defined here as a proportional measure with different denominators dependent on training set size, it was logit-transformed according to the formula

$$\mathrm{logit}(p) = \log\left(\frac{p}{1-p}\right) \tag{3.10}$$

where p denotes proportions. This logit-transformed proportion was used as a dependent variable.

A significant main effect for *training set size* was found (F(1,3) = 73.28, $p < 0.005$). Decision trees constructed from a small training set were more compact than trees constructed from a small training set (mean *compactness* for different training set sizes: 0.44 (15), 0.54 (150)).

A significant main effect for *type of algorithm* was found (F(3,1) = 352.01, $p < 0.05$). When the means were compared, it was revealed that all algorithms produced trees with a significantly different (logit-transformed) mean *compactness*. For instance, the mean *compactness* of INFERULE trees differed significantly from the mean *compactness* of ID3 trees (F(1,3) = 628.40, $p < 0.001$), from the mean *compactness* of ID3-IV trees (F(1,3) = 52.18, $p < 0.01$), and the mean *compactness* of ID3-BIN trees (F(1,3) = 58.80, $p < 0.01$). Thus, INFERULE produced the most compact trees, though the difference with ID3-BIN was only marginal, but significant (mean *compactness*: 0.89 (ID3), 0.63 (ID3-IV), 0.23 (ID3-BIN), 0.22 (INFERULE)).

The interaction effect for *training set size* by *type of algorithm* was not significant ($F(3,1) = 95.29$, $p = 0.075$), whereas in the univariate case, it was found to be significant ($F(3,9) = 9.60$, $p < 0.05$). ID3 and ID3-IV trees were less compact for a large training set than for a small training set, whereas this was just the other way around for ID3-BIN and INFERULE trees ($F(1,3) = 198.39$, $p > 0.005$).

Discussion

The results showed that both ID3-BIN and INFERULE produced the most accurate music preference categorisers among the four alternatives. In addition, INFERULE produced the most compact trees. ID3 is the least accurate categoriser as it did not even exceed the *accuracy* of a *random* categoriser.

Attribute selection heuristics for tree construction that are based on information entropy generally produce trees with a minimal average depth, but consequently with a large number of leaves. The results showed indeed that ID3 and ID3-IV produced broad decision trees. As both a low *compactness* and low *accuracy* were observed for ID3 and ID3-IV trees, these algorithms were liable to *overspecialisation* to the training set. In particular, ID3 tended to select unique attribute values (e.g., title of a music option) in a large training set that were too specific for the categorisation of ‘unseen’ music options.

It was also shown that categorisation *accuracy* of decision trees depends on characteristics of the training data and the attribute representation of music options. For instance, it is impossible that a small training set covers all possible attribute values in a music collection[1]. But, on the other hand, a large training set has to cope with an attribute value set that is too diverse. An inductive learning algorithm only selects a limited number of attribute values from this diverse set. In the experiment, this fact led to inaccurate categorisation of options with attribute values that were not present in the tree. For our two-category problem, a small training set of 15 music options, that was subjected to ID3 and ID3-IV, led to difficulties in achieving a level of *accuracy* that was higher than a *random* categoriser. If a training set was half of the available music (150 options), all algorithms were unable to produce a tree that was consistent with the training set. Hence, it is evident that the use of top-down decision tree constructors and the available set of musical attributes provides insufficient information to predict a unique and accurate preference categorisation for each music option at all times.

3.3.3 Experiment II: Ranking of decision trees

The purpose of Experiment II was to find out what algorithm produces decision trees that uncover attribute values from a set of music programmes that are most relevant to music preference.

Participant

One participant took part in the experiment. He had also participated in Experiment I. He was familiar with the music collection since he owned most recordings.

1. In the evaluation, the music collection contained, for example, 498 different musicians, 230 different composers, 126 different producers, 42 different musical instruments and 12 different music styles. The tempo in which the music was performed ranged from 50 bpm to 350 bpm. Different pieces of music were recorded in different time periods ranging from 1945 to 1995.

Procedure

The participant was instructed to create 14 music programmes in the form of hand-written lists. The participant was asked to keep a personally constructed context-of-use in mind for each programme individually. The task was to collect music options that would be *preferred* as well as music options that would be *rejected* according to the given music preference relevant for the given context-of-use. No instructions were given about the size of a programme and the ratio between preferred and rejected music. A list of 300 music options sorted by performing artist and title were presented on paper. No additional attribute information was provided. The participant had the possibility to listen to music options. Data collected from the participant included the music programmes as well as descriptions of the corresponding music preferences.

Listening equipment and material

The listening equipment and material were identical to those used in Experiment I (see Section 3.3.2).

Results

Of all fourteen music programmes, thirteen contained twelve music options and one contained eleven music options. The mean number of *preferred* options in a programme was 6.6 (min: 6, max: 8) and the mean number of *rejected* options in a programme was 5.3 (min: 4 max: 6).

On the basis of the descriptions of the music preferences elicited from the participant, it appeared that they could be divided into two groups. Half of the fourteen given music preferences could be best expressed by a single musical attribute value that was defined in the attribute representation of music options (see Appendix I). The following seven *simple* preference descriptions were collected from the participant (translated from Dutch):

A. 'south-american latin music',
B. 'swing jazz',
C. 'standards',
D. 'music recorded in front of a live audience',
E. 'duets / duos',
F. 'trios', and
G. 'music played or written by Miles Davis'.

The other half contained more *complex* preferences that required more musical attributes for their expression, or that were difficult to express unequivocally in musical attribute values as defined in the attribute representation. The following seven *complex* preference descriptions were collected from the participant (translated from Dutch):

H. 'quiet background music',
I. 'solid jazz rock or fusion with electric musical instruments',
J. 'hardbop / bebop stylish music, performed moderately fast',
K. 'fashionable dance music',
L. 'meditative new age music',
M. 'music with female vocals', and
N. 'quiet jazz rock or fusion'.

Decision trees that were derived from the fourteen programmes and were constructed by the four algorithms were ranked by the experimenter according to the extent that they identified attribute values relevant for music preference, that is, 'preference fit'. In particular, completely similar or logically equivalent[1] trees were considered to be tied observations. For all other trees, the number of attribute values in the tree that did and did not correspond with the given preference description was counted. If no ranking between trees could be established on this score, it was determined at what level in the tree attributes appeared. Higher level attributes were considered more decisive than lower level ones, as higher level attributes were selected on the basis of more pre-categorised material during tree construction. In this way, rank value 1 was assigned to the tree with the highest number of relevant (high-level) attribute values. Tied observations were assigned the average of the rank values that would have been assigned when no ties had occurred.

Average rank values were taken over the fourteen programmes (see Table 3.3.). The decision trees that were considered to have the best 'preference fit' are shown in Figure 3.3.

Table 3.3. Mean rank value of the four different algorithms for *simple* and *complex* music preferences. Rank value 1 corresponds to the highest 'preference fit'.

	ID3	**ID3-IV**	**ID3-BIN**	**INFERULE**
simple preferences	3.21	2.64	2.07	2.07
complex preferences	3.50	1.92	2.43	2.14
	3.36	**2.29**	**2.25**	**2.11**

In the case of *simple* music preferences, the algorithms ID3-BIN and INFERULE constructed similar trees. In general, these trees were ranked best (see Table 3.3). For six out of seven *simple* music preferences, ID3-BIN and INFERULE uncovered the attribute value that was also prominent in the participant's music preference description (see Figure 3.3 for the best results). The other algorithms also found 'immaterial' attribute values. Only in the case of the 'swing jazz' preference, did none of the algorithms uncover the music style *swing* as the most important attribute value. The participant probably had another notion of 'swing jazz' than was transcribed in the music collection.

In the case of *complex* music preferences, the trees of ID3-IV were in general ranked best (see Table 3.3). ID3 gave the poorest results, since it selected attributes with unique values that were not related to the given music preference (e.g., title, place of recording). 'Preference fit' for *complex* music preference was liable to subjective interpretation; not all decision trees could unequivocally be related to the description of a music preference.

1. Decision trees are logically equivalent if the two disjunctive normal forms obtained by taking the disjunction of the conditions on the leaves of each tree are equivalent. Two logically equivalent decision trees have exactly the same *accuracy* when used to categorise the same items (Fayyad and Irani, 1990).

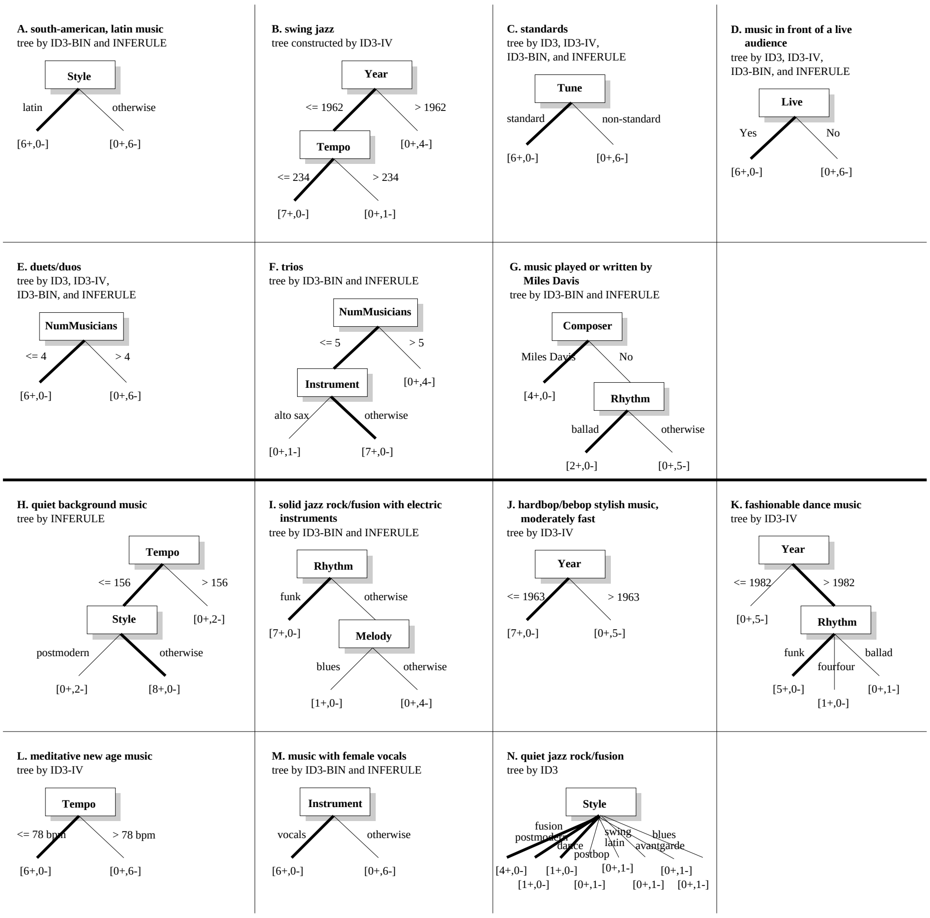

Figure 3.3. Decision trees that were ranked as having the best ‘preference fit’ for all 14 music programmes. The upper 7 decision trees were constructed from music programmes under a *simple* music preference. The lower 7 decision trees were constructed from programmes under a *complex* music preference. *Bold* branches represent categorisation paths in the tree that lead to the category of *preferred* music. [*p*+,*r*-] means *p* preferred and *r* rejected music options in a set of *p*+*r* options.

It also appeared that mismatches occurred between what the participant had in mind about a musical feature and what was actually transcribed in the music collection. For instance, duos and trios did not match with the expected number of musicians on a recording (see Figure 3.3). Also, some musical aspects in the preference descriptions, such as gender information, were not explicitly encoded in the attribute-representation of music options.

The ranking information can also be used to indicate comparative judgements of all pairs of decision trees (Guilford, 1954). Since the ranking task required the binary comparison of all decision trees involved, transitivity of ranking holds. A ranking of four different decision trees amounts then to six pair-comparison judgements. In our case, a decision tree is judged to have a higher ‘preference fit’ than trees that have a higher rank value. A tree is judged to have a lower 'preference fit' than trees that have a lower rank value. Ties in the ranking were treated as an equal 'preference fit' of the trees involved. From the pair-comparison data collected over all given music preferences, we can determine the proportion of time that each type of decision tree has a higher 'preference fit' than any other type of tree (see Table 3.4). Only half of the matrix is shown because the two halves are complementary and no data exists for the diagonal cells.

Table 3.4. Proportion matrix showing the proportion of time that decision trees constructed by an algorithm at the top were judged to have a better 'preference fit' than trees constructed by an algorithm at the side, in 14 pair-comparison judgements.

	ID3-IV	**ID3-BIN**	**INFERULE**
ID3	10 / 14	11 / 14	11.5 / 14
ID3-IV		6.5 / 14	7.5 / 14
ID3-BIN			7.5 / 14

Table 3.5. Scale value estimates and their standard error of the algorithms. Scale value of algorithm ID3 is set at zero, by definition.

	Scale value estimate	**Standard error**
ID3	0.00	0.09
ID3-IV	0.71	0.09
ID3-BIN	0.72	0.09
INFERULE	0.85	0.09

The standard way to analyse pair-comparison data is based on Thurstone's *law of comparative judgment* (Thurstone, 1927; Torgerson, 1958). Applying this law to our case, the extent to which one decision tree is judged to have a higher 'preference fit' than another is related to the difference in subjective strengths, or scale values, on 'preference fit' of the compared trees. It is assumed here that scale values, and likewise their differences, follow a normal distribution with standard deviations equal to 1. The means of these normal distributions then correspond to the scale values. The method to estimate a scale value comes down to transforming the proportions in each cell in Table 3.4 into normal deviates (z-scores). Each normal deviate then represents the difference between the scale values of all 14 decision trees constructed by two different algorithms. Collecting all six possible differences between four scale values of the algorithms results in an overdetermined set of equations.

By setting the scale value of algorithm ID3 to zero, the least-squares solution of the overdetermined set of equations yields the scale value estimates as shown in Table 3.5. The correlation between the observed data and the predictions of the least-squares solution is high (r = 0.956) which means that 91.4% of the variance is explained.

The scale value estimates in Table 3.5 represent a complete ranking of the algorithms with respect to ‘preference fit’. The low value of standard error demonstrates that the ranking is stable and that scale values of the algorithms differ significantly. It appeared that ID3 constructed decision trees with the least ‘preference fit’. The variants of the ID3 algorithm, that is, ID3-IV and ID3-BIN, were judged to have an almost equal ‘preference fit’. Among the four algorithms, INFERULE was judged to construct trees with the best ‘preference fit’.

Discussion

The results showed that the INFERULE algorithm is the best choice of the four alternatives to uncover music preference-relevant attribute values from a judged music programme. The experiment concerned a case-study with one participant and one collection of jazz music, which limits the generalisation of the results.

A well-known drawback of the *information gain* heuristic used in the ID3 algorithm is its tendency to select attributes that take on many different or unique values (Quinlan, 1986). The evaluation showed that branching on many values results in a low ‘preference fit’. Reducing the branching factor by means of the *gain ratio* heuristic in ID3-IV or enforcing the construction of a binary tree in ID3-BIN, immediately resulted in an improved ‘preference fit’.

The results also made clear that a ‘preference fit’ of a decision tree depends on the chosen attribute representation for music options. Since the attribute representation is defined externally and is fixed (see Appendix I), mismatches between what is specified in the attribute representation and what is expected by music listeners may occur. The notion of some attribute values are still subjective. For instance, it appeared that some music options were assigned to another music style than was expected by the participant. It also appeared that some criteria used in the process of compiling had no counterparts in the attribute-representation. If such mismatches occur, it is obvious that the learning algorithms are unable to identify the appropriate attribute values.

If music preferences required several musical attribute values for their expression, that is, the *complex* preferences, the algorithms uncovered only part of these attribute values in the best case. This is partly due to the greedy nature of the tree construction algorithms. The strategy for tree construction is top down and tests one attribute at a time on its effectiveness of partitioning *preferred* and *rejected* options. This favours attributes that score high locally but overlooks combinations of attributes. However, considering combinations of several attributes is computationally expensive and has not yet proven itself useful (Murthy and Salzberg, 1995). The greedy nature also imposes constraints on how attribute values should be distributed among *preferred* and *rejected* music. Since the algorithms try to partition *preferred* and *rejected* options by finding contrasting attribute values, *preferred* options should comply with all attribute values at one side of the contrast (e.g., *preferred* music should be both jazz rock and have a slow tempo), whereas rejected options should agree with only one side of the contrast (e.g., *rejected* music should either not be jazz rock or have a fast tempo).

3.4 Conclusion

A first development of music preference resembles the development of an *ad hoc* concept (Barsalou, 1983; 1991); it exists only temporarily, since it serves the particular goal at hand. While listening to music, music listeners further develop their music preference allowing music to be grouped in either a *preferred* or a *rejected* category. Although the criteria on which music is either *preferred* or *rejected* is essentially covert, it is assumed that these criteria are based on fixed musical attribute values. In subsequent chapters, an inductive learning algorithm will be used in a more elaborate interactive system on music selection and programming to learn about music preferences of listeners.

Top-down decision tree construction algorithms can be used to induce preference-related attribute values from a judged music programme. In our case, these algorithms produce decision trees that partition *preferred* music options from *rejected* ones on fixed musical attribute values. The algorithms differ in their heuristics to select an attribute in the tree construction and in the resulting tree topology. Four algorithms, namely ID3, ID3-IV, ID3-BIN and INFERULE, were implemented. They were adapted to the music domain and comparatively evaluated.

The comparative evaluations posed the following question: given four different inductive learning algorithms, which one provides the highest categorisation *accuracy* with the most *compact* decision tree, and which one uncovers the most relevant musical attribute values from a judged music programme?

The first part of this question was addressed by running all algorithms on the same four data sets. Both ID3-BIN and INFERULE trees gave more accurate categorisation results compared to ID3 and ID3-IV trees. In addition, INFERULE on average generated trees with fewer leaves. It is therefore concluded that INFERULE produces 'better' categorisers based on the fact that its trees are compact. It is known that maximising tree *compactness* will, in a probabilistic sense, increase tree categorisation *accuracy* (Fayyad and Irani, 1990).

The second part of this question was addressed by finding a ranking of 'preference fit' for all algorithms. It was found that INFERULE trees presented the best 'preference fit'. The INFERULE algorithm was originally developed for domains in which the available set of attributes are inadequate to construct a perfect categoriser for all possible items in the domain (Spangler, Fayyad, Uthurusamy, 1989), as in our music domain.

In summary, the INFERULE algorithm is the best choice from the four alternatives to learn about music preferences of music listeners based on results of categorisation *accuracy*, *compactness* and 'preference fit' of its decision trees.

3.4.1 Further research

Although results are encouraging, at least four suggestions for improvement or interrelated directions for further research can be made.

Firstly, an improvement of the musical attribute representation that enables a more accurate categorisation of music preference is urgently needed. For this, conceptual and methodological tools for music analysis are required to analyse the lacking musicological features (see, for an example, Tagg, 1982). In addition, real time extraction of acoustical events in the music may provide direct measurement of the music itself that pertains more to music perception and cognition (see, for an example, Martin, Scheirer and Vercoe, 1998). New attribute values can be easily integrated in the inductive learning algorithms.

Secondly, besides the attribute representation of music options, the used algorithms do not employ knowledge representation of the music domain. Domain knowledge includes constraints and relationships between musical attribute values at a higher abstraction level. In our jazz domain, this knowledge refers to facts such as that ballads are generally performed at a slow tempo, that progressive harmonic development is a key feature for bebop as modal harmonic development is a key feature for postbop, and that the starting periods of different jazz styles are arranged chronologically. The use of music domain knowledge is expected to improve preference categorisation of music. This knowledge may be explicitly encoded or may be discovered by a prior processing stage on the music collection. The latter may be an excellent application of unsupervised learning strategies (for a conceptual clustering application on Spanish folk songs, see Michalski and Stepp, 1983; for knowledge discovery in large databases, see Fayyad, Piatetsky-Shapiro, Smyth and Uthurusamy, 1996). An explicit encoding of knowledge about a particular musical genre will have negative consequences for the generality of the approach, however, and its use for other musical genres.

Thirdly, inspection of the results of the inductive learning algorithms showed that the algorithms capitalise on finding differences between *preferred* and *rejected* music. This leads to loss of information about what either *preferred* music or *rejected* music has in common, when considered separately. The class of single category learning algorithms (Dietterich and Michalski, 1983) tries to find a so-called *characteristic description* which capitalises on homogeneity of a given category by singling out items that belong to a given category from any other item that does not belong to that category.

Finally, judging a small music programme only allows a fraction of music preference to be learnt, which is not accurate enough to categorise ‘unseen’ music options correctly. If more judged music becomes available in time, an incremental learning approach can be undertaken. Variants of tree construction algorithms exist that revise existing decision trees based on new pre-categorised material, instead of completely building a new tree (Utgoff, 1988; 1989). Incremental learning also allows the selection of more informative training material, which results in smaller decision trees (Utgoff, 1988b).

CHAPTER 4

PATS: An automatic music compilation functionality

Selecting preferred music from a large music collection using an interactive player can take considerable time and effort, both in operating the device and in determining one’s music preference. An automatic music compilation functionality referred to as Personalised Automatic Track Selection (PATS) has been developed with the intention to speed up the process of selecting preferred music. It automatically creates music programmes that suit a particular context-of-use. Derived from the user requirement on *adaptation of music choice behaviour*, design requirements for the PATS system were formulated and guided the development process. For instance, PATS should compile programmes that are both *coherent* and *varied* using an attribute representation of music. PATS should also learn about the music preferences of its music listener and should adapt its compilation strategy accordingly. While using the PATS system, a music listener indicates, for an increasing number of music options in programmes, whether they do or do not fit the desired context-of-use. Inductive learning then enables PATS to identify musical attribute values that are relevant for music preference. PATS creates *coherent* and *varied* music programmes for different contexts-of-use using a decentralised cluster approach, in which music options are repeatedly re-grouped based on their similarity. Similarity is defined as a weighted sum of common music attribute values between music options. The weights measure the relevance of attribute-values as identified from user feedback.

Music players have become a common item of property at home. Music listeners have learnt to use simple control actions on these devices such as selecting the next music option. However, as discussed in Chapter 1, programming music on players which access a large music collection can take considerable cognitive effort and time to learn the usually awkward operations involved. In Chapter 2, it was explained that also determining one’s music preference can be difficult. For instance, it is unlikely that music listeners have a master list of preferred music for each context-of-use. The listener often has to search in the music collection to be able to judge what music is appropriate. As a complete inspection of the music collection is impractical, music listeners prefer to use local heuristics to save time and cognitive effort. Heuristics tend to ignore valuable and global information which may result in less preferred music choices. Although a somewhat disappointing programming result may not be in proportion to the ample selection and listening opportunities while exploring a music collection, a functionality that facilitates navigation in a large music collection and the selection of preferred music may be desirable.

This chapter describes the design and implementation of a functionality referred to as Personalised Automatic Track Selection (PATS), that aims at automatically compiling music programmes that suit a particular context-of-use. Eggen (1995) proposed the first conceptual ideas for an automatic music compilation functionality. A preliminary description of the PATS implementation is also reported elsewhere (Pauws and Eggen, 1996). A patent for PATS is pending (Eggen and Pauws, 1997). PATS is intended to supplement features on current interactive players to enable faster selection of preferred music from a large music collection.

The interaction style of the PATS system reflects the common console operation of current players, though the PATS system is not bounded to this interaction style. When using the current interaction style, a music listener only has to perform two small tasks. The first user task comprises the selection of a preferred music option. This music option may be the result of *habitual behaviour*, that is, choosing what you chose last time or what you have given much thought to lately, or *affect referral* (Wright, 1975), that is, retrieving a pre-installed preferred music option from memory. Music listeners usually have a well-defined music recording in mind that suits the current context-of-use. Selecting a preferred option hence requires minimal cognitive effort, at least, at the moment of selection. The PATS system then compiles music options that are similar to the preferred option and presents this set of options as a music programme to the music listener. Since PATS' compilation strategy may not provide adequate results, the second user task comprises giving preference feedback on the created music programme. For each music option in the programme, the music listener is allowed to decide whether it fits the given context-of-use or not. This task only makes a small demand on the memory processes, as the listener only has to make an accept or reject decision for a few music options. The PATS system uses this feedback to learn about the music preference of the music listener and adapts its compilation strategy to create more preferred music programmes in subsequent requests. If the system adapts well to the music preference of a listener, user feedback is no longer required. Moreover, the PATS system does not require any other control actions.

4.1 Design requirements

Ideally, the PATS system should make music choices that would have been made by the music listener as if there was no PATS system available. It is not likely that PATS will succeed in meeting this requirement at its first attempt, but it should improve its accuracy as it learns about music preferences.

In Chapter 1, it was stressed that *adaptation to music choice behaviour* should be a prime user requirement for future music players. Five main design requirements for the PATS system have been derived from this user requirement. These requirements relate to the attribute representation of music options, asymmetric music choice, programme coherence, programme variation and the use of preference feedback from the music listener.

4.1.1 Attribute representation of music options

The first design requirement is that the PATS system should use the same features for selecting preferred music as music listeners do. For that purpose, music listeners use many different musical features. It seems appropriate to represent music options in terms of a number of qualitative, that is, nominal and categorical, attribute values. A set of 18 musical attributes with fixed values was therefore defined to describe each music option (see Appendix I). These attributes and values were primarily extracted from CD booklets and discographies (Cook and Morton, 1994; Erlewine, Woodstra and Bogdanov, 1994). Results of a focus group study (see Chapter 2) indicated that music preference can be sufficiently described by the defined attribute values.

4.1.2 Asymmetric choice

The second design requirement is that the PATS system should employ asymmetric choice and asymmetric similarity judgements, as it is likely that music listeners choose asymmetrically between different music options.

Asymmetry in choice refers to the observation that an option A is chosen over option B in one context, whereas option B is chosen over option A in another context. Asymmetry in similarity judgements refers to the observation that an option A is judged as being similar to option B, whereas option B is judged as being less similar to option A (Medin, Goldstone and Markman, 1995). It has been shown that asymmetric choice between two options is produced by, among other things, the order in which both options are compared. The second option appears to act as a reference point in the comparison, which makes features that are not part of the second option of less concern to the choice (Houston, Sherman and Baker, 1989). Asymmetry in similarity judgements may be produced by the relative prominence of options (Tversky, 1977) and the direction of the comparison (Medin, Goldstone, and Gentner, 1993).

Likewise, asymmetric music choice may be affected by which music option acts as a reference point in a comparison process. Music options that are prominent by frequency, familiarity, or recentness are likely to act as such reference points. For instance, music from relatively unknown musicians may be judged to be quite similar to music of well-known musicians, whereas the converse judgement may be less true.

4.1.3 Coherence

The third design requirement is that the PATS system should create a music programme whose coherence is appreciated by the music listener. Coherence of a music programme refers to the degree of homogeneity of the music in a programme, and the extent to which individual music options are related to each other. Coherence does not solely depend on some similarity between any two music options, but also depends on all other options in a music programme and the conceptual description that a music listener can give to the music options involved. Obviously, the measure of coherence within a music programme is determined by the personal views of the music listener.

If we look at how music listeners tend to make music choices, it seems that they first try to make options comparable by eliminating some features and by retaining other features. Music listeners then make their choices between options which share some of the retained features. Coherence between music options may be based on these retained features, that is, on relevant musical attribute values.

4.1.4 Variation

The fourth design requirement is that the PATS system should create music programmes whose variation is appreciated by the music listeners. Variation refers to the degree of diversity of music options in a music programme and in subsequently created music programmes. It contradicts the other design requirement, coherence.

The most elementary requirement is that the exact same music should not be repeatedly presented for a given context-of-use. Music preference changes over time, which means that the particular value of listening to the same music is unlikely to recur in a given context-of-use. In addition, music within a programme should be varied in order to produce surprise effects for the music listener. The contribution of each additional music option in a music programme may decrease if it contains features that are already covered by other options in the programme.

Since variation produces uncertainty in the compilation results, PATS may be an adequate means to cover all music in the collection over time and to re-discover ‘forgotten’ music.

4.1.5 Preference feedback

The fifth design requirement is that the PATS system should use preference feedback from the music listener in order to improve its creation of music programmes. User feedback consists of an indication of which music options in a created music programme are preferred in a given context-of-use and which music options are rejected. However, it is unknown here what criteria were used by the music listener to categorise the options in a preferred and rejected category. Parts of these criteria that can be expressed in attribute values may be identified by using an inductive learning algorithm. In Chapter 3, an inductive learning algorithm called INFERULE was considered as the most useful among three others, namely ID3, ID3-IV and ID3-BIN, for inferring attribute values that are relevant to music preference.

4.2 COMPILATION STRATEGY OF PATS

The PATS system employs a cluster-seeking algorithm. A cluster of music options finds its way in a music programme. The clustering is governed by a similarity measure between music options. This similarity measure is asymmetric, by definition, and it gives relevant attribute values more weight in its calculation. Relevant attribute values are identified by INFERULE.

4.2.1 Interaction style

The interaction between the music listener and the PATS system is depicted by the scenario shown in Figure 4.1. The scenario specifies the steps that a user has to take to achieve a particular goal with the system in terms of pre and post-conditions, discrete user actions and episodes.

A pre-condition needs to be established by the music listener before it can pursue any action; it may also refer to a pre-set user goal. A post-condition refers to a user goal that the PATS system must satisfy after the user action has been taken.

Discrete user actions refer to particular, yet unspecified control elements in the interaction, such as button presses.

Episodes contain hidden details that are scenarios in their own right; they can be further specified.

As shown in Figure 4.1, a scenario is represented by a flow chart in which pre-conditions and post-conditions are visualised by circles, discrete user actions by a rectangle, and episodes by a stack of rectangles. In the margins, pre and post-conditions are specified. System actions may be specified between rectangles.

The scenario in Figure 4.1 assumes that a music listener wants to listen to an automatically compiled music programme in a given context-of-use; this goal is conveyed by performing an action. The process is continued by selecting a music option that is preferred in the given context-of-use from a list of available music options. Subsequently, the system compiles a music programme in which this preferred music option is contained. After the created programme is accepted, the music listener can listen to each individual music option and indicate whether it is preferred or not in the given context-of-use. If options are rejected, this will have consequences for the system when it creates a programme the next time. When the music listener has indicated to stop the listening session, the scenario concludes.

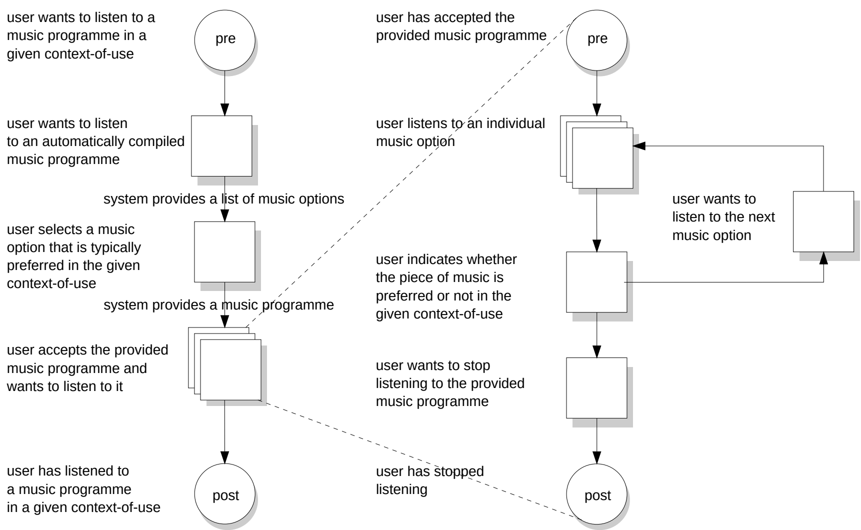

Figure 4.1. Scenario in which a music listener wants to listen to an automatically compiled music programme in a given context-of-use.

4.2.2 Similarity measure

A similarity measure between music options will be necessary to generalise music preference over music options. If it is known that a set of music options is preferred, then it is likely that preference can be generalised to other options, based solely on the fact that they are similar. Similarity must be based on attribute values of music options, as music listeners dynamically weigh positive and negative features to express their preference. However, they may use other features than those accounted for in the similarity measures, or they may use other similarity criteria dependent on the particular options being compared or the context in which options are compared.

Each music option is defined as an entity with an identity and attributes having fixed values but varying weights. Attributes are defined over different domains. A similarity measure between music options is then defined as the integration of weighted attribute values from different domains into a single similarity value.

Attribute domains

Let $\Sigma = \{x_1, x_2, \ldots, x_N\}$ denote the music collection containing the music options. Each music option $x_i \in \Sigma$ is represented by an 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 music option can thus be represented by a vector $x_i = (V_{i1}, V_{i2}, \ldots, V_{iK})$. The values of attributes are fixed in time, by definition. The domain of an attribute may be nominal, multiple-nominal, binary, categorical or numeric. For notational convenience, the value V_{ik} is defined as a vector. In most cases, this vector has a length of 1, but in the case of multiple-nominal attributes it is defined as $V_{ik} = (v_{ik1}, v_{ik2}, \ldots, v_{ikL_{ik}})$ where L_{ik} denotes the number of elements. For a nominal attribute A_k, the element of V_{ik} for a given option $x_i \in \Sigma$ is extracted from a set of possible nominal values $\Pi_p = \{\pi_{p1}, \pi_{p2}, \ldots, \pi_{pM_p}\}$. M_p denotes the number of distinct nominal values in a set.

Nominal attributes may correspond to song titles, musicians, composers or musical instruments. A multiple-nominal attribute may correspond to the set of musicians, composers, or musical instruments found on a particular music option. L_{ik} then corresponds to the number of musicians, composers or musical instruments. For a binary attribute A_k, V_{ik} can only hold one of two elements, either *true* or *false*. A binary attribute may correspond to whether a music option is recorded in front of a live audience or not, for example. For a categorical attribute A_k, V_{ik} holds a value extracted from a fixed range of categorical values as a single element. Categorical attributes may correspond to music styles, harmonic development and rhythmic accompaniment. For a numeric attribute A_k, V_{ik} has a value from a given numeric range denoted by R_k as a single element. Numeric attributes may correspond to the year of recording or the tempo of a musical performance measured in beats per minute.

Non-negative weights W_{ik} are associated with each attribute A_k and each music option $x_i \in \Sigma$. They are also defined as vectors. Weights measure the relevance of attribute-values. In most cases, the weight vector contains only one element. The weight vector is defined as $W_{ik} = (w_{ik1}, w_{ik2}, \ldots, w_{ikL_{ik}})$, except in the case of multiple-nominal attributes.

Definition

The *value similarity measure* $s(v_{ikl}, v_{jkl})$ is a numeric value ranging from 0 to 1; this measure is dependent on the domain of A_k. For nominal, binary and categorical attributes, it is either 1 if the attribute values are identical, or 0 if the values do not match. More precisely,

$$s(v_{ikl}, v_{jkl}) = \begin{cases} 1 & , v_{ikl} = v_{jkl} \\ 0 & , v_{ikl} \neq v_{jkl} \end{cases} \tag{4.1}$$

For numeric attributes, the value of $s(v_{ikl}, v_{jkl})$ is one minus the ratio between the absolute value difference and the total span of the attribute domain. More precisely,

$$s(v_{ikl}, v_{jkl}) = 1 - \frac{|v_{ikl} - v_{jkl}|}{R_k} \tag{4.2}$$

where R_k is the total span of the numeric attribute A_k.

The similarity $S(x_i, x_j)$ between the music options x_i and x_j is then essentially a normalised weighted sum of the *value similarities* of attributes. More specifically,

$$S(x_i, x_j) = \sum_{k=1}^{K} \sum_{l=1}^{L_{ik}} w_{ikl} \cdot s(v_{ikl}, v_{jkl}) \quad , \text{ with } \sum_{k=1}^{K} \sum_{l=1}^{L_{ik}} w_{ikl} = 1 \tag{4.3}$$

where K is the number of attributes, L_{ik} is the number of attribute values for attribute A_k of option x_i, and $s(v_{ikl}, v_{jkl})$ the value similarity measure of attribute A_k between option x_i and x_j. Recall that for all attributes except for the multiple nominal attributes, L_{ik} equals 1. The similarity is asymmetric ($S(x_i, x_j) \neq S(x_j, x_i)$) because the value of the weighing function corresponds to a single music option, that is, x_i, and the weights w_{ikl} corresponding to different music options may not be identical.

Other similarity measures

Although similarity may not provide all explanatory evidence, a similarity structure among choice alternatives or items is an essential component of choice behaviour and categorisation (Payne, 1982; Medin, Goldstone, and Markman, 1995; Goldstone, 1994). A brief survey of some similarity models commonly used in these fields is given, as well as some arguments of not using these models in our music domain.

In geometrical models of similarity, options are represented as points in a geometric space such that the *dis*similarities between two options x_i and x_j correspond to the metric distance $D(x_i, x_j)$ between the respective points (Shepard, 1987),

$$D(x_i, x_j) = \left[\sum_{k=1}^{K} |X_{ik} - X_{jk}|^r \right]^{1/r} \tag{4.4}$$

where K is the number of dimensions, X_{ik} is a ratio value of x_i on dimension k, and r is a constant. Often, a city-block metric (r = 1) is used for options that have easily separable dimensions, or a Euclidean metric (r = 2) for options with fused dimensions (Shepard, 1987). Crucial to the geometric approach is the assumption that psychological distance is a metric distance (just as in physical space), for which the following three axioms apply:

> *minimality,* $D(x_i, x_j) \geq D(x_i, x_i) = D(x_j, x_j) = 0$, which says that the distance between any option and itself is identical for all options, and is the minimum possible.

symmetry, $D(x_i, x_j) = D(x_j, x_i)$, which says that the distance between two options is identical regardless of which one is taken as a reference point.

triangle inequality, $D(x_i, x_j) + D(x_j, x_k) \geq D(x_i, x_k)$, which says that the shortest distance between two options is a straight line.

The use of a geometric model for similarity judgement of music options is inappropriate. Music options are hard to represent as points in a geometric space, due to the strong nominal character of musical attributes. In addition, the validity of some metric axioms is questionable. The minimality axiom holds as it is unlikely that a music option would be mistaken for another. In Section 4.1.2, it was argued that similarity between music options should not be treated as a symmetric relationship. And, the triangle inequality is anyway hard to express in terms of nominal attributes. It seems reasonable to assume the implication that if music option x_i is 'quite similar' to music option x_j, and music option x_j is 'quite similar' to music option x_k, then music options x_i and x_k cannot be 'very dissimilar'. However, as similarities between three music options may be based on distinct attribute values or arguments, the validity of the triangle inequality may be doubted.

Featural models of similarity contradict geometric models. They can be made compatible with violations of all three metric axioms (Tversky, 1977; Smith, 1990). The *contrast model* (Tversky, 1977) expresses the similarity between two options x_i and x_j as a weighted difference of the measures of common and distinctive features of x_i and x_j. Features are viewed as a result of a prior extraction process, though it is uncertain exactly what the features of an option are. Features denote binary, nominal, categorical, ordinal or cardinal attributes. Similarity is assumed to depend on the number of common features of x_i and x_j, and inversely on the number of features that belong to one but not to the other,

$$S_{\text{contrast}}(x_i, x_j) = \theta f(x_i \cap x_j) - \alpha f(x_i - x_j) - \beta f(x_j - x_i) \tag{4.5}$$

where $x_i \cap x_j$ are the features shared both by x_i and x_j, $x_i - x_j$ are the features possessed by x_i but not by x_j, and $x_j - x_i$ are the features possessed by x_j but not by x_i, f is a monotonically increasing function which measures the importance of each set of features, and θ, α and β are parameters that determine the relative contribution of the three feature sets. In a practical setting, the function f is simply the number of features present in its argument (Tversky, 1977; Smith, 1990), though it may also represent a weighing function from which the values of the weights have to be estimated. There is, however, no theory in the contrast model that indicates how the function f should measure importance of features and in what order features should be compared (Smith, 1990). The parameter values of θ, α and β also have to be estimated.

Another featural model of similarity, the *product rule* model (Estes, 1994), expresses the similarity between two objects x_i and x_j as the product of coefficients corresponding to common or distinctive features,

$$S_{\text{product-rule}}(x_i, x_j) = t^m s^{K-m} \tag{4.6}$$

where t is the coefficient for matching features, s is the coefficient for mismatching features, m is the number of matching, that is, common, features, and K is the total number of features. The value of t is often set to 1, and s has a value in the range $0 \le s \le 1$. The value of both coefficients depends on experimental considerations, however, and needs to be estimated (Estes, 1994). The log of the product-rule similarity yields the same linear dependence on common and distinctive features as the contrast model in which f denotes the number of features in its argument (Estes, 1994). The need of parameter estimation make the use of both featural models of similarity less applicable in our music domain.

Class diagram

The class diagram, as shown in Figure 4.2, describes the implementation of the similarity measure. The graphical notation is adopted from the Unified Modeling Language (UML) (Rumbaugh, Jacobsen and Booch, 1999).

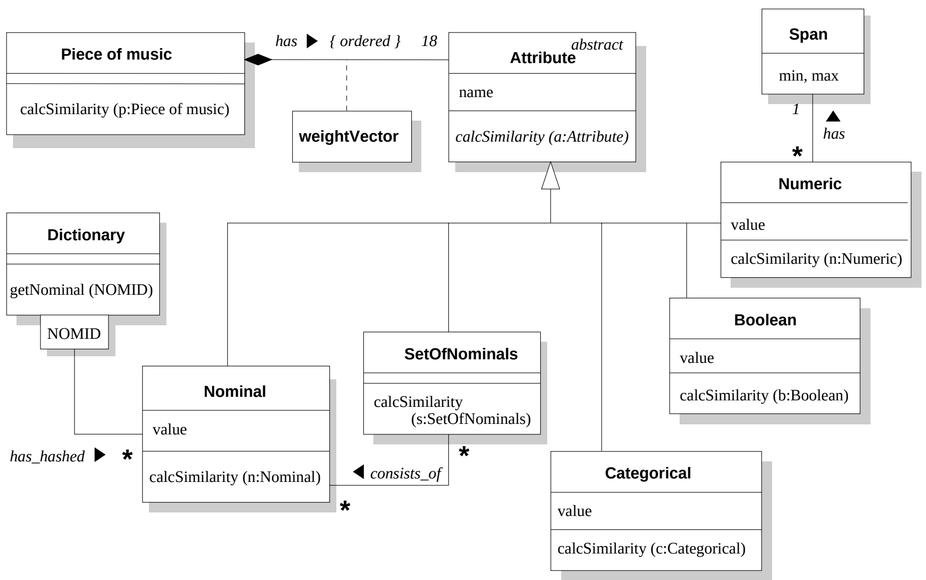

Figure 4.2. Class diagram for calculating the similarity between music options.

The class *Piece of music* has an ordered set of 18 *Attributes*. A *weight vector* is associated with each relation between an instance of *Piece of music* and an instance of *Attribute*. The *Attribute* class is actually an abstract class, from which other attribute classes are derived. The sub-classes of *Attribute* represent the different attribute domains. Numeric attributes are implemented by the class *Numeric*; instances of this class have values varying within a continuous *Span* of values. Numeric attributes include tempo and year of recording. Binary attributes are implemented by the class *Boolean*; an instance of this class may only have the value *true* or *false*. Binary attributes include live recording and standard jazz piece. Categorical attributes such as music style, harmonic development and rhythmic accompaniment are implemented by the class *Categorical*. Attributes with multiple nominal values are implemented by the class *SetOfNominals*. An instance of this class contains multiple instances of the class *Nominal*. The class *Nominal* is an implementation of nominal attribute values such as musical instrument, person (musician, composer, producer), place of recording and music title. All available nominal values are stored in *Dictionaries*. A *Dictionary* class is essentially a chained hash table, in which instances of the *Nominal* class can be retrieved using a key (NOMID). Each subclass of the class *Attribute* has an individual member function *calcSimilarity* which computes its contribution to the total similarity between two pieces of music, that is, the value similarity measure.

4.2.3 Decentralised cluster-seeking approach

A similarity measure governs the grouping of options that are preference related or that fit in the same context-of-use. This does not specify how this grouping or clustering of options is established. Many traditional cluster-seeking approaches for numeric data are based on the optimisation of a unitary performance index (Tou and Gonzalez, 1974). In these approaches, the algorithms aim to minimise the within-cluster distances or maximise the between-cluster distances. In addition, conceptual clustering for non-numeric data aims at finding presumed categories, in which items with unknown categories can be usefully grouped in the most *simple* way and with a *high fit* on the data. These algorithms generally try to circumscribe a set of items by a logical conjunction, that is, ‘ands’ of simple attribute value tests (Michalski and Stepp, 1983; Fisher, 1987; Langley, 1996). Our original objective was that clusters should be coherent by aiming at minimising within-cluster distances or maximising fit, but also varied (which tends to increase within-cluster distances or decrease fit). This objective is hard to implement in the traditional cluster-seeking approaches. Therefore, instead of an external algorithmic control optimisation, a decentralised cluster-seeking approach had to be used based on the concept of ‘swarm intelligence’.

The decentralised cluster-seeking approach has been influenced by research on collective behaviour of autonomous entities interacting in artificially socio-biological environments (Brooks and Maes, 1994; Deneubourg, Goss, Franks, Sendova-Franks, Detrain and Chretien, 1994; Drogoul and Ferber, 1994; Kuntz and Snyers, 1994; Lumer and Faieta, 1994; Kamp, 1997; Ferber, 1999). In particular, the implementation metaphor was inspired by Reynolds’ computer graphics model of the aggregate motion of flocking birds (Reynolds, 1987). The central theme in all these papers is that the combination of small local interactive behaviour specified only at the level of each individual emerges into a collaborative and clustering phenomena that can be observed at a global level.

Music options as agents

In the decentralised cluster-seeking approach, each music option is represented by an *autonomous* entity that is placed in a virtual environment such as a two-dimensional Euclidean space. This entity *continuously* perceives and reacts to other entities in the environment. The entity displays autonomy in the sense that it has a decision strategy of what kind of behaviour to exploit when perceiving changes in the environment. An entity whose actions are determined by the perception of its current situation, rather than by some internal program or plan is termed a *situated* *agent* (Agre and Chapman, 1987; 1990; Wavish and Connah, 1990). However, many other definitions exist for different kinds of agents and multi-agent systems (Ferber, 1999). Situated agents imports ideas from *situated action* which was originally described by Suchman (1987). Situated action stresses that plans are best viewed as resources, rather than prescriptions, for action and that representations for action are indexical, that is, they make sense only in the context of the current situation in which action has to take place.

Each agent has a set of behaviours that rules its interaction with other agents in the environment. The initial behaviour of each agent is to wander around in the environment; its direction and pace are determined randomly. While displaying this wandering behaviour, other behaviour enables an agent to look around for and match with other agents in its direct vicinity. The matching process consists of calculating the similarity of two music options represented by the two agents. Dependent on the similarity value, an agent may decide to follow a ‘nearest and most similar’ agent. More specifically, the similarity value is employed as a probability that the ’follow’ behaviour will actually be activated. When following, an agent adapts its wandering behaviour to the course of the other. It therefore continuously senses the location of the other and changes its velocity and direction by a small amount, as required. It must be emphasised here that any agent can be followed by multiple other agents, resulting in the emergence of clusters. An agent continually looks around, matches, and possibly follows another ’near and similar’ agent using the same strategy. In addition, periodically excited behaviour of an agent can decide to temporarily stop the ’follow’ behaviour and to resort to the initial wandering behaviour. The periods are fixed, but the decision to stop following depends on the similarity with the followed agent. If an agent adjusts its ’follow’ behaviour, this has consequences for its followers, since they persist in following their leader.

In summary, an agent is continuously moving into whatever cluster contains another highly similar agent. From a global perspective, clusters may be formed and dissolved. By definition, the similarity measure assigns selectively different weights to attribute values dependent on the music options being considered. Consequently, each cluster contains a varied set of music options that have several distinct attribute values in common. In other words, each cluster consists of a composite whole of multiple strains of *coherent* music options. However, a cluster varies continuously over time.

The use of an inductive learning algorithm

A cluster may represent a music programme that suits a particular context-of-use. The concept ‘context-of-use’ is not explicitly modelled in the PATS system, however; the system has no direct means to link clusters to contexts-of-use. Therefore, a music listener selects a preferred music option, which is initially the only information about the context-of-use for the system (see also Section 4.2.1). The system response then consists of finding a cluster which contains this preferred music option for the purpose of compiling a music programme. A music programme is constrained to a user-defined number of music options, unlike clusters. The music options in the cluster that are most similar to the preferred option are added to the music programme. Listeners, confronted with a sequence of music options making up the current music programme, only have to decide which option does not fit the desired context-of-use. This feedback on personal preferences is used by the system to adapt its clustering process.

The clustering process is adapted only by adjusting the weights of attribute values. Initially, all weights are set to 1. Before adjusting weights, the system has to uncover what attribute values are relevant to distinguish preferred music options from rejected ones. Uncovering these attribute values is accomplished using an inductive learning algorithm. In Chapter 3, it was concluded that the INFERULE algorithm is the best choice among three other algorithms, namely ID3, ID3-IV and ID3-BIN, to infer relevant attribute values. INFERULE was therefore implemented in the PATS system. The input to INFERULE is the set of music options that are assigned to *preferred* and *rejected* categories by the music listener and the attribute representation of the music options. The output is a decision tree which separates preferred and rejected music options on the basis of their attribute values. Weights of all music options in the collection are now adjusted in two stages.

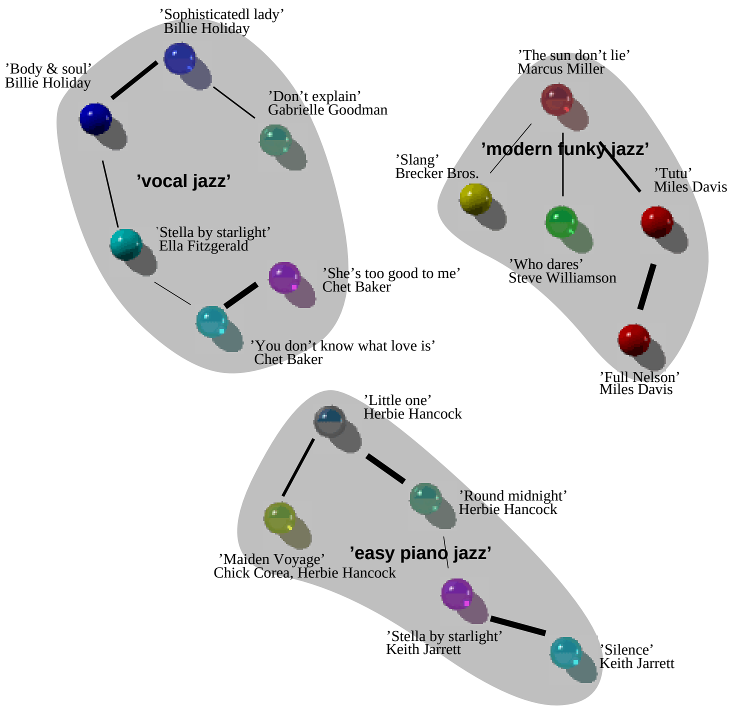

Figure 4.3. An ideal clustering result that consists of music programmes preferred in different contexts-of-use. The music options (i.e., the agents) are represented by differently coloured marbles. Similar options have similar colours. The lines connecting these marbles represent the 'follow' behaviour displayed by agents. The line width also denotes similarity between the music options.

In the first stage, the decision tree is used to categorise the *whole* music collection into the categories *preferred*, *rejected* and *undecided*. As explained in Chapter 3, the decision tree may remain uncertain about a predicted preference of some music options which explains the need of a category *undecided*.

In the second stage, weights of attribute values of music options in the category *preferred* are multiplied by an arbitrary factor. This factor is multiplied by $1/2^{l-1}$ where l represents the level of the tree at which the attribute value occurs. It is assumed that attribute values occurring at a higher level in the decision tree are more relevant than attribute values at lower levels. Weights of attribute values of music options in the category *rejected* are divided by the same factor. Weights of music options from the category *undecided* are left unchanged. Subsequently, the clustering process is re-started with the new weights, which may result in an improvement on the overall preference of the following music programmes.

While using the PATS system, the music listener decides, for increasingly more music options, whether they do or do not fit the desired context-of-use. Since clusters may have some attributes in common, clusters that find their way into programmes may eventually become suitable for different contexts-of-use. In Figure 4.3, an *ideal* state of a clustering process is shown. This ideal situation shows the formation of different music programmes for different contexts-of-use, such as 'vocal jazz', 'easy piano jazz' and 'modern funky jazz'.

Development of the agents

Three aspects in our agent software development process were relevant: the language for concurrent production rules, the deictic representation of agent behaviours and the incremental design approach.

Behaviours of an agent are represented by declarative symbolic rules in a concurrent production rule language called RTA[1]. Behaviours, in the form of rules, can be coupled with other behaviours within the same agent, with behaviours of other agents, or with behaviours of the environment; note that the environment is, by definition, also an agent. Rules execute concurrently; rules are dynamic, can be nested arbitrarily, and can be time-annotated to control the execution of other rules (Wavish, 1991b; Wavish and Graham, 1994). Essentially, the symbolic representation of behaviour establishes causality. The independent propagation of causality changes in a multi-agent system is enabled by the concurrency principle.

Although an agent has to perceive changes in the environment continuously, the number of environmental changes is too large to be represented explicitly and completely. In addition, far from all changes are of interest to the agent. Only those objects that are likely to change during the actions of an agent are of interest. Consequently, action and perception are closely intermixed in the behaviours of the agents. More importantly, the environment is not observed from an objective third-person point of view, but from a subjective first-person point of view (Wavish and Graham, 1994). Derived from the index principle in situated action (Suchman, 1987), the agent only examines aspects that are of current interest; no complete model of the environment needs to be maintained. This deictic representation is considered to be an alternative to standard logical representation (Chapman, 1991). Objects are referred to in relation to oneself, such as in 'the other agent which I am following'.

The clustering phenomenon which emerges from behaviours of individual agents was designed incrementally; the behaviour of the whole system and the individual

1. A Philips-proprietary language for designing multi-agent systems is used. The explicit representation of behaviours by declarative symbolic rules was first dealt with in ABLE (Agent Behaviour LanguagE) (Connah and Wavish, 1990; Wavish and Connah, 1990). A real-time variant of this language, used in the cluster-seeking approach, is called RTA (Real Time ABLE) (Graham and Wavish, 1990; Wavish, 1991a; Wavish and Graham, 1994). PATS is implemented using RTA 4.0.

agents was continually analysed, modified and extended as appropriate. This incremental approach can be described in terms of the Brooks' subsumption architecture (Brooks, 1986) used for mobile robots, but also in terms of human cognitive representation and control of action in action identification theory (Vallacher and Wegner, 1987). The subsumption architecture stacks small task-achieving modules in hierarchical layers, instead of ordering functional modules sequentially. Each successive layer subsumes and extends on the competences, that is, behaviours, of lower layers in the hierarchy. Behaviours from lower layers are contained in behaviours at a higher layer. Lower layer behaviours still exist, but they are used for a larger meaning or effect. For instance, the act of 'following someone' still subsumes the mechanics of 'wandering around and avoiding obstacles'. Unlike the traditional hierarchical decomposition (Dijkstra, 1971), each layer has its own input-output loop, that is, interaction with the environment. Traditionally, input and output facilities are assumed to exist only at the lowest level.

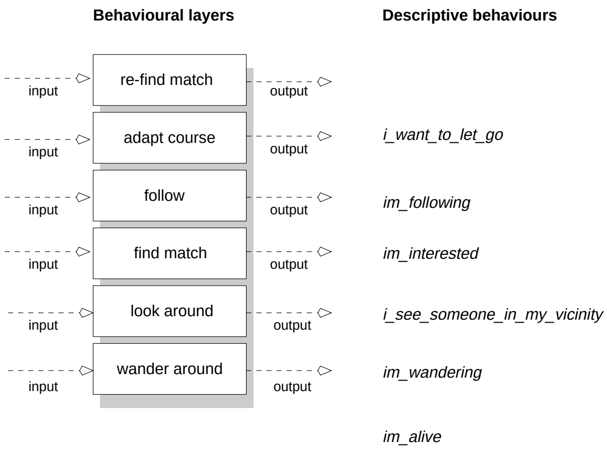

Figure 4.4. Layered decomposition of an agent and descriptive behaviours of each layer.

In our agents, a layer consists of a set of behavioural rules that fulfil a particular small task. Layers are activated if particular *descriptive* behaviours in lower layers are switched 'on'. The use of *descriptive* behaviours in a programming framework is a means to represent explicitly emergent behaviour and build higher level behaviours on them (Wavish, 1991b). In general, behaviours from a certain layer switch on behaviours on the next highest layer by using these *descriptive* behaviours. A violation to the subsumption architecture is that a higher layer may switch 'off' *descriptive* behaviours in a lower layer which de-activates all layers from the lower up to the higher. This violation is necessary because the subsumption principle does not grant behavioural decisions originating from a higher layer to be mixed with lower layer behaviour. For instance, the behaviour of 'following someone' may be stopped from a higher layer while maintaining the lowest mechanics of 'wandering around and avoiding obstacles'.

The layered decomposition of an agent is shown in Figure 4.4. Upon creation, an agent's existential *descriptive* behaviour *im_alive* is switched 'on'. When alive, an agent immediately starts to *wander around* from a randomly chosen location in the environment. Wandering around for an agent means changing its coordinate positions, confined to the two-dimensional environment, according to a randomly varied velocity and direction. This initial behaviour comprises the lowest layer in the subsumption architecture. *Wander around* puts the *descriptive* behaviour *im_wandering* into existence. On the next highest layer, behaviours are defined to *look around* for other agents. An agent repeatedly investigates if another is in its direct vicinity. If so, the *descriptive* behaviour *i_see_someone_in_my_vicinity* is switched 'on'. Next, the *find match* layer calculates the similarity and uses this value (ranging between 0 and 1) as a probability measure to determine whether the agent is *interested* in the other or not. If it is interested, the *follow* layer provides the 'follow' behaviour; it puts *im_following* into existence which triggers the next higher layer. This *adapt course* layer is essentially a refinement of the lowest *wander around* layer; velocity and direction are adapted to the course of the one that is followed. As the *descriptive* behaviour *im_wandering* still exists, an agent continues to look around for others and may likewise decide to follow others. The highest layer consists of a *re-find match* process. This layer is triggered by the *descriptive* behaviour *i_want_to_let_go* which is periodically switched 'on' and 'off' by the *adapt course* layer. The *Re-find match* layer specifies the same behaviour as the *find match* layer, but now the similarity value is used to decide continuation of the 'following behaviour'. If the decision is to continue, nothing changes. Otherwise, all *descriptive* behaviours except the *im_wandering* and *im_alive* behaviours are switched 'off', which returns the agent to the behaviour as specified in the lowest layer.

Vicinity

The agent's detection of whether another agent is in its direct vicinity is an example of deictic representation. Simulation of some form of visual perception by an agent is avoided. Information about another nearby agent is made available to an agent in the most efficient way.

In a two-dimensional Euclidean space, vicinity is defined as the area that is contained in a given circle centred at the agent's local position. The concept of vicinity thus requires an arbitrary non-negative threshold T specifying the radius of the circle. In the implementation, agents are able to retrieve the exact position, the direction and velocity of other agents. To determine what agents are in its vicinity, an agent has to compute its distance to all remaining agents and evaluate all computed distances against the given threshold T. This involves *n-1* vicinity tests in a deictic representation framework, when n agents are involved. To gain a complete picture of who is in whose direct vicinity, a total of $(n^2 - n)/2$ vicinity tests would be required. This number excludes reflexive and symmetric tests. To reduce computational overload, it is more appropriate to use a *constant time algorithm*[1] for vicinity testing, since vicinity testing is a recurrent behaviour of an agent.

1. The 'complexity' of an algorithm to solve a problem instance is generally expressed by its time requirements. A problem instance is therefore mapped onto a finite input string of symbols chosen from a finite input alphabet. This allows the use of an input length to encode a problem instance. A *time complexity function* for an algorithm expresses then the largest amount of time needed by the algorithm to solve a problem instance for any given input length. A function $f(n)$ is said to be $O(g(n))$ whenever there exists a constant c such that $|f(n)| \leq c \cdot |g(n)|$ for all values of $n \geq 0$. For instance, an algorithm is defined to have *polynomial time* requirements if its time complexity function is $O(p(n))$ for some polynomial p, where n denotes the input length (Garey and Johnson, 1979). A *constant time algorithm* is defined here as the one whose time complexity can be further bounded by a linear function for p. In our case, symbols in the input string refer to the agents involved in the vicinity test problem instance; the input length is the number of agents.

The number of tests is reduced by using a spatial elimination technique known as the sector method (Roberts, 1994). The sector method divides the two-dimensional coordinate space into a grid of equally sized sectors. Each agent is assigned to a sector according to its current position in the two-dimensional space. As agents move freely around in the two-dimensional space, the assignment of agents to sectors has to occur repeatedly after each move of the agents. Sector assignment is straightforward; the indices in the sector grid are obtained by dividing the x and y coordinates of an agent's current position by the fixed horizontal and vertical sizes of the sectors respectively. Under the assumption that agents are evenly distributed in the space, and that the number of sectors is proportional to the number of agents, sectors contain a relatively small and relatively stable number of agents. Vicinity testing is then reduced to a constant time algorithm. It is reduced to computing and evaluating the distances of agents that are assigned to the same sector as the given agent. Only the agent that is closest to the given agent and does not exceed the threshold *T* is retained and identified as the nearest one.

Class diagram

The class diagram of the PATS system is shown in Figure 4.5. The class *Agent* lives in a virtual *Environment*, that is, a two-dimensional Euclidean space. An *Agent* represents a *Piece of music*; it keeps track of what other agent it senses in its vicinity, what other agent it follows determined on the basis of similarity, that is, *CurSim*, and by what other agents it is followed. The sensing of another agent is implemented by the sector method. Therefore, a grid of sectors is superimposed on the *Environment*. An *Agent* is included in a *Sector* dependent on its current location. Only *Agents* that are located in the same sector are able to sense each other.

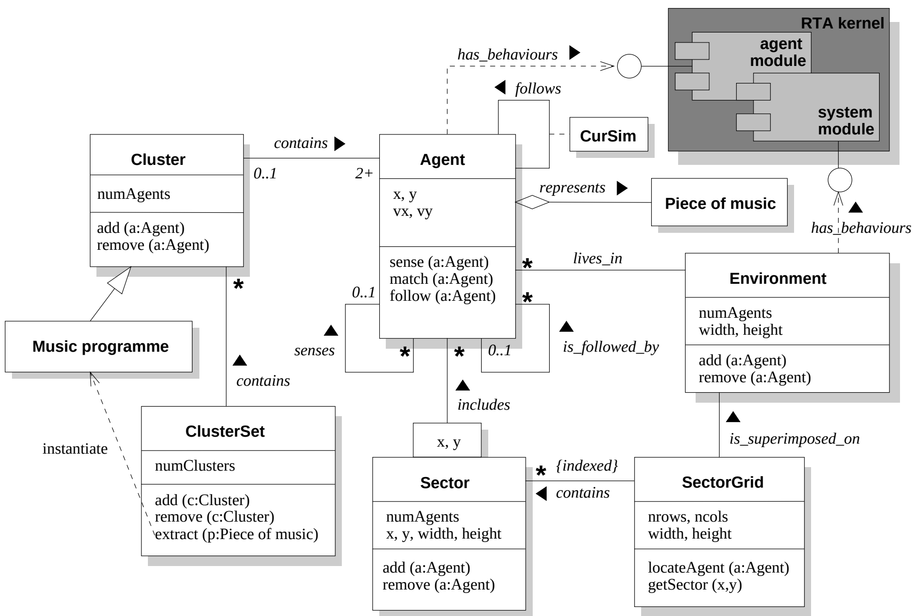

Figure 4.5. Class diagram of the PATS system.

The class *Cluster* contains at least two agents that participate in a 'follow' behaviour. The forming and dissolving of clusters is a complex dynamic process which is maintained by the class *ClusterSet*. A *Music programme* is a *Cluster* with a pre-defined number of music options in it. A *Music programme* can be extracted from a *Cluster* by specifying a *Piece of music* that must be contained in the *Music programme*.

The interactive behaviours of both an *Agent* and the *Environment* are governed by RTA behaviour rules. A bi-directional interface for communication is defined between the RTA kernel and the C / C++ programming language. RTA behaviour rules, grouped in modules, have C / C++ *structs* as their counterparts and hence can point to arbitrary data structures within an application. In that way, the state of a symbolically represented RTA behaviour can be directly used in the programming language in which the RTA kernel is embedded. Moreover, a C / C++ application is allowed to declare so-called *constraints* functions operating on RTA behaviour rules that are passed to them as arguments, such as switching them 'on' or 'off'.

RTA modules (e.g., agents) are compiled into 'asynchronous' digital circuits which are emulated on a virtual machine, that is, the RTA kernel, at runtime (Wavish and Graham, 1994). In our case, memory space usage is roughly linear to the number of modules loaded in the virtual machine. The virtual machine used (RTA 4.0) executes at over 20,000 events per second on a typical workstation which corresponds to several thousands of behavioural rule firings per second (Wavish and Graham, 1994). The required amount of processor time is mainly determined by the current activity in a loaded module; it is roughly proportional to the number of events per second (e.g., switching behaviours 'on' or 'off) multiplied by the processor time needed to process an event. Since the virtual machine maintains a timeline which is locked to real time, the number of events per second is proportional to the number of active modules loaded and to the overall real time speed at which those modules run. In practice, RTA scales well with respect to speed as adding more modules does not have an apparent effect on processor use and execution speed. If, however, the amount of processor time required equals the processor time available, speed of execution slows down sharply which happens when adding more than 1,000 agents to the PATS system. This abrupt decrease in speed is mainly due to the way in which the event queue is implemented in the virtual machine used; the time complexity of this virtual machine is $O(n^2)$ in the worst case. Fortunately, a new version of the RTA virtual machine (RTA 5.0) has a time complexity of $O(n \cdot \log n)$, since the event queue is efficiently implemented as a heap, thus providing even a better scaling for speed.

4.3 Related work

An interesting research and design question is how PATS music programmes can best be presented to a music listener, and how a music listener should interact with them in terms of selection, inspection, manipulation and preference feedback. Ideas of how these topics can be dealt with will be discussed by referring to two case studies in Section 4.3.1.

The PATS functionality can be placed in the field of recommendation systems (see also Chapter 1). Some related applications will be discussed in Section 4.3.2.

4.3.1 Innovative interaction styles for PATS

Scheffer (1996a;1996b) designed and implemented an interaction style for the PATS system intended to ease selection of a preferred music programme or cluster. PATS-generated music clusters are presented as ovals of a varied size in a two-dimensional space. An oval's size depicts the number of options in a cluster. Basically, the space is spanned by a dimension representing the extent to which options within a cluster are similar to each other, and by a second dimension representing the extent to which options within clusters are similar with respect to a given preferred music option. In that way, music listeners are given an instant overview of all available PATS programmes both in terms of the amount of music in a programme, the coherence or variation of programmes, and how much programmes cohere to their indicated music preference. A programme can be further examined on its content by selecting one of the ovals. Though user control of the interaction proceeds entirely using a standard mouse, attribute information of music options is presented in a graphical, auditory or textual format. For instance, artists, recording labels and music styles are represented by a picture or graphical icon. Tempo is represented by a tapping beat. A musical instrument is represented by a melodic sequence with a characteristic timbre of the instrument. The year of recording is presented by noise added to a music sample. And, live recordings are characterised by the sound of an applauding audience.

Van Sluis (1996) continued on a spatial presentation of PATS music programmes and implemented a prototype for a new input device designed for 3D navigation using custom-made and commercially available components. The prototype consists of an air-filled spongy balloon that can be held and squeezed by hand. Several levels of squeeze forces are measured by an air pressure sensor which are sent off digitally to a PC. A small motor from a portable audio cassette player is mounted in the balloon to provide vibro-tactile feedback. A commercially available ultrasonic tracking system is used to measure hand position in real-world space. The small transmitter unit is worn on a finger and the receiver unit is hung on the PC monitor. PATS programmes are visualised as spherical objects in a 3D space. By wearing the transmitter unit, a user controls a cursor displayed in the visualisation, which is real-time animated. A vibro-tactile cue conveys that the cursor is in the vicinity of a programme. A soft squeeze of the balloon selects and highlights the nearby programme; the programme can then be moved to another location or moved to the front. Squeezing a little harder makes attribute information of the programme to pour out of the spherical object and plays back the music. A really firm, but optional, squeeze makes the programme burst into random pieces, intended to have some control in the PATS clustering process if a programme does not contain the music that a listener is looking for.

4.3.2 Related applications

Only a small selection of related applications are presented here. In general, those applications that present original work, or reflect the music domain, or have metaphorical analogies with the PATS system are discussed. However, there are numerous other recently developed recommender systems in domains such as movies (Hill, Stead, Rosenstein and Furnas, 1995), or web sites and pages (Balabanovic and Shoham, 1997).

Book recommender system

The first system that explicitly made suggestions to users was the GRUNDY system (Rich, 1979; 1983; 1989). GRUNDY made book recommendations; it played at being a librarian. In a natural language dialogue with the user, GRUNDY solicited information about the user such as descriptive words, user commands, answers to questions and the rejection and acceptance of recommended books.

In order to make a good suggestion to the right person, users were modelled as a collection of *stereotypes*, such as a religious person, a feminist, an intellectual, or a sports-person. Though socially disputable, a stereotype contained a set of user characteristics such as those pertaining to motivations, interests, tolerance to violence, romance and education. These characteristics were assigned a value and a validity rating of that value. Stereotypes were arranged in a 'generalisation graph' to express that, for instance, Protestants and Catholics are Christians and thus religious persons. A trigger mechanism between stereotypes implemented inference. A trigger activated a stereotype and adjusted its ratings, and an activated stereotype could in turn trigger other stereotypes. The first trigger was instantiated by using information that was solicited in the dialogue with the user. The collection of all activated stereotypes represented an individual user model. Since books were attributed by the same set of characteristics as the stereotypes, books were recommended that were found to be similar to the user model. Stereotypes and their triggers were adapted based on the rejection and acceptance of recommended books.

In an experimental evaluation, it appeared that participants found significantly more GRUNDY-recommended books (72%) 'looking good' than randomly selected books (47%) (Rich, 1979; 1983).

Music recommender system

Firefly is a social filtering system. It is a commercialised descendent of the MIT Media Lab's RINGO music recommendation system, but has been expanded beyond the domain of music recommendation. The original RINGO system (Shardanand, 1994; Shardanand and Maes, 1995) collects ratings of items, such as musical artists and music albums, from users into user profiles. These profiles are used to compare the likes and dislikes of an individual user with those of others and to obtain *predicted* ratings of new items for the given user. The system only selects profiles of others that have similar likes and dislikes as the given user using a nearest neighbour clustering approach. Similarity is based on a Pearson correlation or a mean-squared difference measure between user profiles. A *predicted* rating for an item is established by a weighted average of the item ratings of these similar users.

LyricTime (Loeb, 1992) is a personalised music system which selects music from a music database to be played in a listening session. It selects music by using a set of descriptors of the music, a music listener profile and listener feedback. A profile holds the frequencies at which a music listener wants to listen to music with a given descriptor; it was also assumed here that listeners desire varied music. Different profiles can be defined for different moods; it was also recognised here that music preference is sensitive to time and context-of-use. Listener feedback about music options is used to update the profile. From this perspective, LyricTime is quite similar to the PATS system. However, its selection strategy is based on an information filtering process. This filter ensures that the actual frequency of music options with a set of descriptors in the database is transformed into a set of music options that most likely fits the desired frequency.

MusicFX (McCarthy and Anagnost, 1998) is an arbitration system that automatically selects music to best accommodate the music preferences of people who gather for an extended period of time in a shared environment. An example of a shared enviroment in which *MusicFX* has been deployed is a fitness center. The system has a database of preference ratings of fitness center members for a wide range of musical genres, a log-in mechanism for identifying who is working out at any given time, and a weighted random selection algorithm for selecting a musical genre that best suits the preferences of people who are currently working out. Essentially, this algorithm randomly selects a musical genre dependent on probability measures that express the people's preferences of various musical genres. Music from this genre is then played for a fixed period of time.

The World Wide Web

The World Wide Web (WWW) is essentially a continually expanding database of heterogeneous, loosely interconnected multimedia documents, distributed over servers in a world-wide-area network. The network communication protocol (HTTP), however, only permits documents to be retrieved from a server; there are no facilities to ask a server what documents are available or to find out what servers exist. Consequently, user navigation has evolved into a *web surfing* strategy of following interconnections between documents, that is, the hyperlinks, and retrieving relevant information in the meanwhile. Search engines based on traditional information retrieval techniques for static databases have already been implemented and are in use to help users in finding relevant documents. Users have to specify a query or a few keywords stating their interest. In general, these search engines use indexed databases of documents that are collected off-line. An off-line indexing for a distributed dynamic environment is time consuming, however. It must be done periodically, imposes a high load on network resources, and gives incomplete results.

Currently, assisted browsing tools are implemented to partly solve user search problems on the WWW. These tools adopt a *web surfing* strategy that resembles the way in users have to search for information on the WWW, but do that on-line, faster and automated (de Bra, Houben, Kornatzky and Post, 1994; Menczer, 1997; Lieberman, 1995). In general, the automated surfing strategy maps onto the classical search paradigm in Artificial Intelligence (Rich, 1983b). The WWW is therefore represented as a graph with documents as nodes and hyperlinks as edges between nodes. The search in this graph is then guided by an heuristic that should rate documents higher when they become more interesting to the user. For instance, a standard algorithm is *best-first search* which maintains a list of the highest rated documents to visit next.

A search method called the *Fish-Search* algorithm uses the metaphor of a school of fish (de Bra, Houben, Kornatzky and Post, 1994). The search is initiated by specifying a list of documents as a starting point and a user query consisting of a regular expression or a set of keywords. Each fish represents a document that is uniquely identified by a *Universal Resource Locator* (URL). Fish follow hyperlinks in a *depth-first* fashion for retrieving other documents. Depending on the relevance of retrieved documents and the network resource consumption, fish may survive, produce children to continue searching in a particular direction, or die to stop searching in a particular direction. Fish therefore have a reservoir of 'energy' or food that is adapted on the basis of document relevance. Document relevance is determined by a matching process with the user query. The output of the algorithm is a list of relevant documents retrieved by the fish.

A related search method is the *ARACHNID* algorithm, which uses a distributed, adaptive population of agents making local decisions (Menczer, 1997). The algorithm employs a combination of unsupervised learning by reinforcement for individual agents and genetic evolutionary learning to mutate whole agent populations. Here, the search is also initiated by specifying a list of documents as a starting point and a list of keywords. A population of agents is initialised at the starting documents. An agent has a representation containing ‘energy’-like parameters. Various representations of agents can be defined. The representation determines the actual mechanisms that implement agent behaviours such as selection of hyperlinks to follow, use of user relevance feedback and mutation strategy. Agents follow hyperlinks in a *best-first* fashion. Their ‘energy’ is updated by charging network resource costs and by rewarding if a document appears to be relevant. Document relevance is established by user relevance feedback as well as a matching process with the given list of keywords. An agent may also modify its ‘energy’ parameters during its life time based on prior experience; reinforcement learning allows the best hyperlinks to be predicted the next time. If an agent has accumulated sufficient ‘energy’, it may reproduce slightly mutated offspring. Agents that have insufficient ‘energy’ are killed.

Another search tool is *Letizia* that is portrayed as a user interface agent (Lieberman, 1995). Letizia attempts to anticipate documents of interest by doing a *best-first* search on the available set of hyperlinks and by making inference about user navigation behaviour. The inferences are based on heuristics pertaining to relevant user actions during navigation such as hyperlinks that are saved for later use, hyperlinks that are repeatedly visited, and hyperlinks that are ‘passed over’. Upon request, Letizia recommends documents of interest.

4.4 CONCLUSION

Music preference is affected by factors such as time, the listener’s mood, familiarity with the music, and the context-of-use in which the music is heard. Considering the many factors related to music choice, it is unlikely that music listeners will instantly and without hesitation be able to decide what music is preferred. Instead, a search in a music collection is required to develop music preference and to choose preferred music.

The PATS functionality is intended to speed up the process of selecting preferred music in large personal music collections. Its compilation strategy is based on clustering similar options to obtain coherent music programmes and, at the same time, dissolving these clusters by random perturbations to obtain varied music in programmes. Both coherence and variation of a music programme are assumed to be important criteria when music listeners program music by themselves. In particular, these criteria were used as the most important design requirements for PATS which were in turn derived from the user requirement on adaptation to music choice behaviour to be met by future music players (see also Chapter 1).

When using the interaction style for the PATS system, a music listener only has to select one preferred music option, for the PATS system to come up with a music programme containing similar music options. The listener then only has to indicate if there is any music in the programme that is inappropriate. As PATS is supposed to learn about the music preferences of the listener, it is likely that this form of preference feedback will be needed less when PATS is used long-term. An investigation should be carried out, however, to see how long the PATS system must be used before it creates preferred programmes (see Chapter 5).

4.4.1 Application domains

PATS is intended to supplement current features on interactive music players. It may provide a new and pleasant interactive means to explore the ample music selection and listening opportunities of a large music collection. Currently, recorded music is not only available on physical media but also encoded electronically in files or streamed over the Internet or other networks. PATS may also provide easy access to preferred music in a large personal assortment of downloaded music.

In addition to an in-house application, PATS may also be useful in other application domains where fast selection of preferred music is desirable and a form of listener feedback can be established. Note that PATS creates personalised music programmes, which may make it a less useful feature for multiple simultaneous listeners. Some possible applications are listed below.

- Exploration of music at retailers, public music libraries, stock music (e.g., jingles) repositories and other (on-line) music lending or purchase services. For example, to help a customer or user to find some unfamiliar music by starting from some familiar music.
- Music-on-demand at a patient's bedside in hospitals, helping to create a comfortable atmosphere for the patient.
- Automatic creation of a personalised radio programme for continuous air-play or web-play.
- Automatic creation of music listening sessions at work, helping to create a pleasant and stimulating atmosphere for an industrial worker doing repetitive work.
- 'Automated disc-jockey' in pubs and cafes, helping to create mid-morning, mid-afternoon, mid-evening and late-night cafe atmospheres.
- Continuous play-back of 'back ground music' in public places, events, health clubs, and schools for the martial arts, meant to surround one with a pleasant sound or sound to reinforce a 'complete workout'.

4.4.2 Attribute representation

A prerequisite to use PATS in a practical setting is the availability of musical attributes and their values. Although a fixed attribute representation for jazz music is chosen for the implementation, the system does not hinge on this representation. The PATS system can be easily configured to another attribute representation for another popular music idiom.

Leaving the task of entering all attribute data to the music listener is not a uniformly desirable solution. The task is too time-consuming and too cumbersome. It is expected that, in the near future, attribute information of music will be electronically encoded on music carriers and transmission. There is already a growing electronic store of music attribute information becoming readily available. Consider information on the World Wide Web (WWW) or Internet alone; a music listener can already sample information about a musician including a biography, a complete discography, instruments, music styles, record labels and related artists. On-line CD-database services automatically identify music CDs and provide artist, album, song titles, and some additional free-formatted data of the CD. The CD-databases are provided with data from Internet users, who also query the database.

An informal and open standard called ID3v2 (Nilsson, 1999) allows to tag all kinds of attribute data with a MP3[1] audio stream or file. This ID3v2 tag can hold any kind of data such as title, artist, album, style, year, tempo, musical key, (synchronised) lyrics, preferred volume, equalisation settings and still images. In addition, it maintains data about how often the music file is played and a preference rating as given by the music listener. The current Internet popularity of ID3v2 has as a desirable consequence that on-line services and Internet users distribute the tagged data among each other, thus largely freeing each individual user from entering all attribute data.

In addition, current formal technological developments and standardisations are addressing the categorisation of audio-visual content, including music. For instance, MPEG-7 (1998) aims at tagging audio-visual content with descriptive attributes. Typical attributes for MPEG-7 audio are key, onset, decay, tempo, tempo changes, and position of phenomena in sound. It is expected that these attributes can be extracted automatically or semi-automatically. An abstraction layer imposed on these attributes allow the forming of a semantic representation and description of the audio content. By relating this description with the music itself, efficient search and retrieval techniques of music that is of interest to a music listener should be possible.

1. Audio compressed in MPEG-1 Part 2 Layer III is often referred to as MP3, and is well-known to music listeners who 'surf' on the Internet.

CHAPTER 5

Comparative evaluation of strategies for compiling music programmes

Adaptive properties and effects of long-term use of an automatic music compilation strategy were evaluated in terms of music programme quality over time and under various music preferences. The automatic compilation strategy used was Personalised Automatic Track Selection (PATS), which aims at automatically compiling music programmes suiting a music preference or context-of-use of a music listener. The evaluation involved a computer simulation study and a user experiment using a music collection containing 480 and 300 music options respectively. The computer simulations compared programme quality of the PATS compilation strategy with the programme quality that would be obtained by a random compilation strategy. Programme quality was assessed by *precision*, that is, the proportion of preferred options in a programme, and *coverage*, that is, the proportion of distinct and preferred options across successive programmes. Music preference was defined by a logical expression of attribute values. It appeared that PATS programmes adapted to various music preferences in successive programmes. Factors pertaining to programme quality are discussed. In the user experiment, participants assessed the quality of both PATS-compiled and randomly assembled music programmes, in two different contexts-of-use over four experimental sessions. Participants were instructed to indicate what music options in the music programmes did not fit the given context-of-use. Programme quality was measured by *precision*, *coverage* and a *rating score*. The results demonstrated that PATS programmes contained more preferred music (higher *precision*), better covered preferred music in the collection (higher *coverage*), and were rated higher than randomly assembled programmes in both contexts-of-use. In addition, PATS programmes appeared to contain more preferred music in the fourth session; they adapted slightly to a given context-of-use over time.

When we consider the ever-increasing amount of recorded music available, little attention has as yet been paid to the question of how music preference can be translated to appropriate music programmes by music choice. The preceding chapter discussed the criteria that have to be met for preferred music programmes, and how these criteria are implemented in an automatic music compilation functionality, named PATS. The PATS system is intended to supplement current features on interactive music players for accessing a large personal music collection. Its main purpose is to enable fast selection of preferred music by compiling music programmes that suit a particular context-of-use of the music listener. This chapter describes a comparative evaluation of music programmes compiled by the PATS system and randomly assembled music programmes. The evaluation involved a computer simulation study and a user experiment. Parts of the user experiment are also reported elsewhere (Ober, 1996; Pauws, Ober, Eggen and Bouwhuis, 1996).

5.1 COMPUTER SIMULATION STUDY

A music listener selects music that suits a given context-of-use. If a music listener uses an automatic music compilation functionality for that purpose, the first requirement for the functionality would be to learn and adapt to music preferences for the context-of-use. Furthermore, a music listener is likely to be in different contexts-of-use over time, even in the course of a single day. Each context-of-use induces another music preference, one, perhaps, pertaining to dance music, another to a given musician, and yet another to piano music. Thus, each of these contexts-of-use requires the system to learn the corresponding music preference and to adapt to them, with some help from the music listener of course.

Two kind of computer simulation studies were carried out to examine the quality of PATS programmes: one in which only one music preference (context-of-use) was defined, and one in which multiple music preferences were defined, one after the other, resembling daily use. The former is referred to as Study 1 and the latter as Study 2. The results of the PATS strategy in Study 1 were compared with the results that would be obtained by randomly selecting music options from a music collection. The results of the PATS strategy in Study 2 were compared with the results of PATS in Study 1, to see whether PATS performance is affected if it has to address multiple music preferences.

The studies simulated a music listener who selects consistently, according to a well-defined music preference. In that way, computer simulations offer an easy way to examine possible adaptive properties and possible effects of long-term use of a music compilation strategy to music preferences. Simulation results are helpful to trace imperfections of the compilation strategy or to generate hypotheses for user experiments.

5.1.1 Measures

Precision and *coverage* were used to measure programme quality. The rationale of using both measures is that it is very likely that music listeners would like a single music programme to reflect adequately their music preference, as well as that several music programmes in time cover as much different music reflecting their preference as possible. The former is expressed by *precision*, whereas the latter is expressed by *coverage*. Both measures originate from the *precision* and *recall* measures in information retrieval research (Salton, 1989).

Consider a music collection C containing n music options, from which only a fraction, denoted by a set P, is preferred. This set P contains n_p preferred options. A music programme M_t is a set of m distinct music options drawn from the music collection C at time t. A music programme contains both a set P_t containing m_t preferred options and a set R_t of rejected options.

P*recision* of a music programme M_t is defined as the proportion of preferred options in M_t. More precisely,

$$precision(M_t) = \frac{|P_t|}{|M_t|} = \frac{m_t}{m} \tag{5.1}$$

where $|S|$ denotes the number of elements in a set S. Ideally, *precision* should be 1 across successive music programmes, meaning that a music programme reflects the actual music preference.

Recall is defined as the proportion of preferred options in the music collection that are selected in M_t. More precisely,

$$recall(M_t) = \frac{|P_t|}{|P|} = \frac{m_t}{n_P} \tag{5.2}$$

By varying the size of a music programme, the number of preferred options in a programme, m_t, is likely to vary accordingly. Small programmes can cover only a small part of all preferred music in the collection, that is, they have a low *recall*, whereas large programmes are able to cover a larger part, that is, they enhance *recall*. As a compilation strategy repeatedly makes music programmes of a relatively small and fixed size over time, a measure was needed which adequately assessed *recall* over time. Therefore, *coverage* is defined as the proportion of distinct and preferred options across multiple consecutive music programmes M_k, $k = 1, \ldots, t$. More precisely,

$$coverage(M_t) = \frac{\left|\bigcup_{k=1}^{t} P_k\right|}{|P|} \tag{5.3}$$

Coverage is a non-decreasing curve across multiple consecutive music programmes and ideally approaches 1, but its course is different for different compilation strategies. If *coverage* increases towards 1 over time, a compilation strategy is effective in collecting preferred music that was absent in preceding programmes and does that by nearly complete coverage of the preferred set P. The definition of coverage (see Equation 5.3), however, requires prior knowledge of the set P, which is known for computer simulations but is not directly observable with user experiments. As values for *precision* and *coverage* can be predicted for a random compilation strategy in computer simulations, only simulation experiments with the PATS compilation strategies are carried out.

Random compilation strategy

A random compilation strategy consists of randomly selecting m different music options from a music collection containing $n > m$ music options. Let X denote the number of preferred options from this random selection. X is a stochastic variable and hypergeometrically distributed,

$$P(X = k) = \frac{\binom{n_p}{k}\binom{n - n_p}{m - k}}{\binom{n}{m}} \tag{5.4}$$

where k satisfies the following inequality, $\max(0, m - n + n_p) \le k \le \min(n_p, m)$, and n_p denotes the number of preferred options in the music collection. The expected number of preferred options in a random selection, that is, EX, equals $(m \cdot n_p)/n$. By dividing the value of EX by m, the expected *precision* rate of a random compilation strategy is obtained.

The expected number of distinct and preferred options c_t across t consecutive random selections is recurrently defined as,

$$\begin{aligned} c_1 &= \mathrm{EX} \\ c_t &= c_{t-1} + \frac{m(n_p - c_{t-1})}{n} \qquad , t > 1 \end{aligned} \tag{5.5}$$

Note that c_t is a monotonically increasing function of the number of selections with a negative acceleration. By dividing the value of c_t by n_p, the expected *coverage* rate across programmes of a random compilation strategy is obtained.

As an example, if the music collection contains n = 480 music options and the number of preferred options in the collection equals n_p = 50, a random selection of m = 10 options is expected to contain approximately one preferred option (EX = 1.04, expected *precision* = 0.104). In Figure 5.1, the *precision* curve as a result of 200 successive random selections is shown.

After 10 successive random selections, it is expected that 9 to 10 distinct and preferred options are drawn ($c_{10} = 9.49$, expected *coverage* = 0.19). After 50 successive random selections, 32 to 33 distinct and preferred options are expected to be drawn ($c_{50} = 32.55$, expected *coverage* = 0.65). See Figure 5.1 for the corresponding points after 10 and 50 random selections. It is clear that the expected *coverage* curve for a random compilation strategy only approaches 1 after a long time because every item has the same probability of being included in the random selection. Also, *coverage* will exhibit the same curve for each composition of the preferred set *P*. The increase in *coverage* for successive random selections is low due to the lack of *precision*. Theoretically, PATS should adapt to the actual music preference across successive music programmes, resulting in an increasing *precision* rate. As PATS aims at varied successive music programmes, it should also cover a larger part of the preferred options in the music collection, resulting in a steeper *coverage* curve.

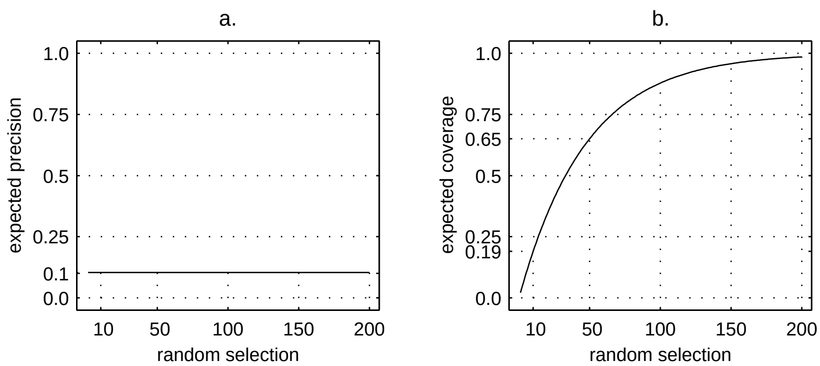

Figure 5.1. Expected *precision* (a) and expected *coverage* (b) of a random compilation strategy (n = 480, n_p = 50, m = 10).

5.1.2 Method

Test material

A music collection comprising 480 attribute representations of jazz music recordings from 160 commercial CD albums was used. The attribute representation can be found in Appendix I. The music collection covered 12 popular jazz styles.

Music preferences

Eight music preferences were defined as shown in Table 5.1. The music preferences are typical in common jazz listening. Each music preference was specified by a logical expression of attribute values. Music options that satisfied this specification were termed as *preferred*; otherwise, they were termed as *rejected*. For instance, music preference P1 ('piano with small ensemble') covered 24 preferred music options in the music collection, whereas music preference P2 ('dance music') covered 92 preferred music options. The music preferences could be separated into two groups. The first group consisted of *simple* music preferences P3, P4, and P7 which involved the learning of a single attribute value. The second group consisted of *complex* music preferences P1, P2, P5, P6 and P8, which involved the learning of a logical combination of multiple attribute values.

Procedure Study 1 - Each music preference independently

A computer simulation was carried out for each music preference. Each simulation consisted of a Monte Carlo experiment in which *precision* and *coverage* of the PATS compilation strategy was assessed across 20 trials. A trial was defined here as the compilation of 15 successive music programmes addressing a given music preference. At the start of each trial, a music option was randomly selected from the preferred options. This music option was repeatedly used to initiate the compilation. The number of music options in a music programme was fixed at 11 including the music option that initiated the compilation. Initially, 1500 clustering steps were used to start the PATS clustering process in each trial. A clustering step denoted the addition or removal of a music option to or from a cluster. Pilot simulations showed that 1500 clustering steps provided clusters with a high intra-similarity. Subsequent compilations were preceded by 750 clustering steps.

Procedure Study 2 - Three music preferences following each other

A Monte Carlo experiment was run for each combination of three music preferences. These simulations addressed a sequence of music preferences P1, P2 and P4, at that order, a sequence of music preferences P2, P7 and P8, at that order, and a sequence of music preferences P3, P4 and P5, at that order. These sequences were chosen to represent a variety of different degrees of inference between music preferences. As shown in Table 5.2, music preferences P1, P2 and P4 had a relatively small overlap in preferred options, music preferences P2, P7 and P8 had a relatively moderate overlap, and music preferences P3, P4 and P5 had a relatively large overlap. Note that besides concrete music options that music preferences may share, preferred options may also have mutually common attribute values, not necessarily occurring in the logical specification of a music preference. This possibility was not investigated. Again, *precision* and *coverage* of the PATS compilation strategy were assessed across 20 trials. This time, a trial consisted of the compilation of 45 music programmes, in which each programme addressed one of the three music preferences successively. At the start of each trial, three *different* music options were randomly selected, one from each music preference, to initiate each compilation. Also, 1500 clustering steps were used to start the clustering, and 750 clustering steps preceded each subsequent compilation.

Table 5.1. Eight music preferences that were used in the simulation referred to by a description and specified by a logical combination of attribute values. The number of options in the music collection that satisfied this specification is also presented at the bottom of each column from a total of 480 options.

Music preference	**P1 (complex)**	**P2 (complex)**	**P3 (simple)**	**P4 (simple)**
Description	piano with small ensemble	dance music	vocal jazz	Norman Granz
Specification	instrument = piano AND number of musicians ≤ 3	rhythm = funk AND (style = fusion OR style = mbase OR style = dance)	instrument = vocal	producer = Norman Granz
# Preferred options	24 out of 480	92 out of 480	77 out of 480	43 out of 480

Music preference	**P5 (complex)**	**P6 (complex)**	**P7 (simple)**	**P8 (complex)**
Description	jazz from the good old days	famous trumpet players	jazz with vocal rap	Miles Davis
Specification	year ≤ 1965 AND (tune = classic OR tune = standard)	instrument = trumpet AND ((musician = Miles Davis AND year ≤ 1960) OR musician = Dizzy Gillespie OR musician = Roy Eldridge OR musician = Fats Navarro OR musician = Chet Baker)	instrument = vocal rap	musician = Miles Davis OR composer = Miles Davis
# Preferred options	85 out of 480	52 out of 480	27 out of 480	30 out of 480

Table 5.2. Overlap of preferred options for music preferences. *p / q* denotes *p* music options out of *q* options, that is, the total number of preferred options shared by music preferences.

	P1 and P2	**P1 and P4**	**P2 and P4**	**P1, P2, and P4**
overlap	0 / 116 (0%)	1 / 66 (1.5%)	0 / 135 (0%)	0 / 158 (0%)
	P2 and P7	**P2 and P8**	**P7 and P8**	**P2, P7, and P8**
overlap	25 / 94 (26.6%)	6 / 114 (5.3%)	0 / 57 (0%)	0 / 118 (0%)
	P3 and P4	**P3 and P5**	**P4 and P5**	**P3, P4, and P5**
overlap	12 / 108 (11.1%)	22 / 140 (15.7%)	27 / 101 (26.7%)	8 / 142 (5.6%)

5.1.3 Results

Results Study 1 - Each music preference independently

Simulation results in terms of *precision* and *coverage* over 15 successive music programmes (making up a trial), for each music preference are shown in Figures 5.2 to 5.9. In these figures, means of 20 trails are shown. The results are divided up into an initiation phase (the first five programmes), a progressive phase (the intermediate five programmes), and the steady-state phase (the last five programmes). The results are displayed with the expected values for *precision* and *coverage* of a random compilation strategy. Results of the mean *precision* attained in the three phases and mean *coverage* attained with the last programme are also shown in Table 5.3.

The simulation results indicate that, for all defined music preferences, PATS programmes contained more preferred options, that is, had a higher *precision*, than randomly assembled programmes. PATS *precision* curves showed an increasing trend over successive music programmes, indicating that PATS programmes adapted in the course of time to a given music preference. From Figures 5.2 to 5.9, it is clear that PATS programmes did not adapt equally well to different music preferences. If only the steady-state phase is considered (programmes 11 to 15), PATS can in general achieve more than 80% precision. The best results on *precision* were obtained for preference P2 ('dance music'), as shown in Figure 5.3. The worse results on *precision* were obtained for preference P1 ('piano with a small ensemble'), especially in the initiation phase, as shown in Figure 5.2, and for preference P6 ('famous trumpet players'), as shown in Figure 5.7.

The simulation results also indicate that PATS was not successful in collecting all options that reflect a given music preference in 15 music programmes. All PATS coverage rates have an increasing, negatively accelerating curve tending towards a maximum. As shown in Table 5.3, this maximum is deliberately below 1 in most cases; it is unlikely that further improvement in future programmes will be observed. For music preferences that covered more than 50 music options in the collection (P2, P3, P5 and P6), PATS was only successful in presenting half of the preferred music in 15 music programmes. For music preferences that covered fewer options (P1, P4, P7 and P8), PATS retrieves three quarters of the preferred music.

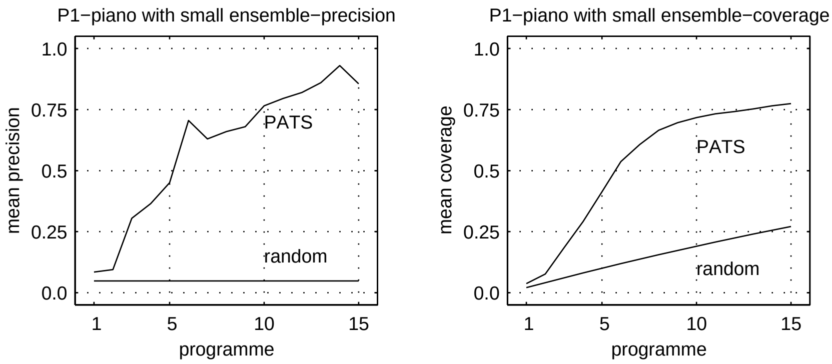

Figure 5.2. Mean *precision* (proportion of preferred options in a programme) and mean *coverage* (proportion of preferred and distinct options across programmes) for the PATS compilation strategy in a simulation and for the random compilation strategy.

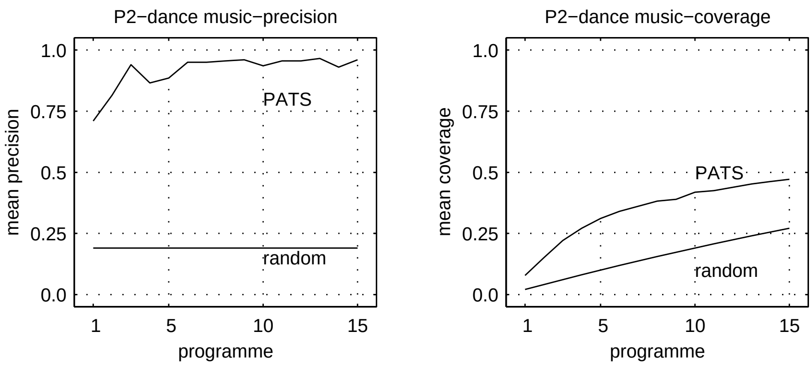

Figure 5.3. Mean *precision* (proportion of preferred options in a programme) and mean *coverage* (proportion of preferred and distinct options across programmes) for the PATS compilation strategy in a simulation and for the random compilation strategy.

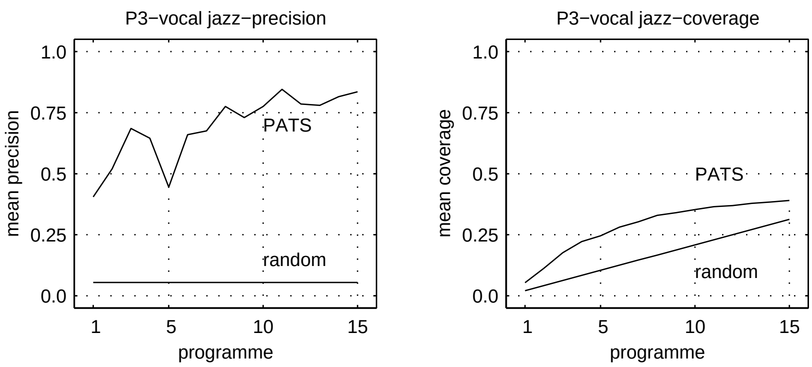

Figure 5.4. Mean *precision* (proportion of preferred options in a programme) and mean *coverage* (proportion of preferred and distinct options across programmes) for the PATS compilation strategy in a simulation and for the random compilation strategy.

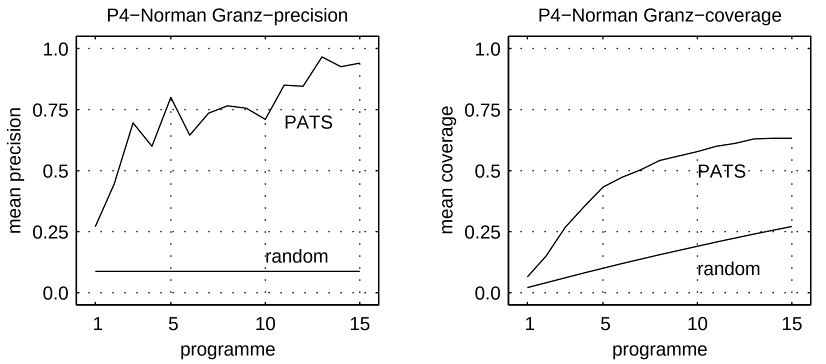

Figure 5.5. Mean *precision* (proportion of preferred options in a programme) and mean *coverage* (proportion of preferred and distinct options across programmes) for the PATS compilation strategy in a simulation and for the random compilation strategy.

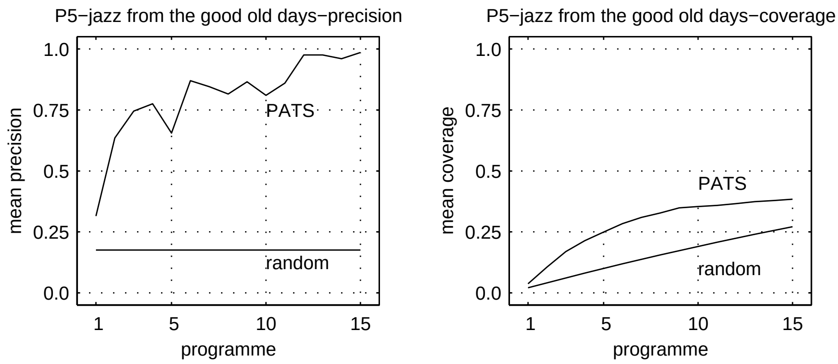

Figure 5.6. Mean *precision* (proportion of preferred options in a programme) and mean *coverage* (proportion of preferred and distinct options across programmes) for the PATS compilation strategy in a simulation and for the random compilation strategy.

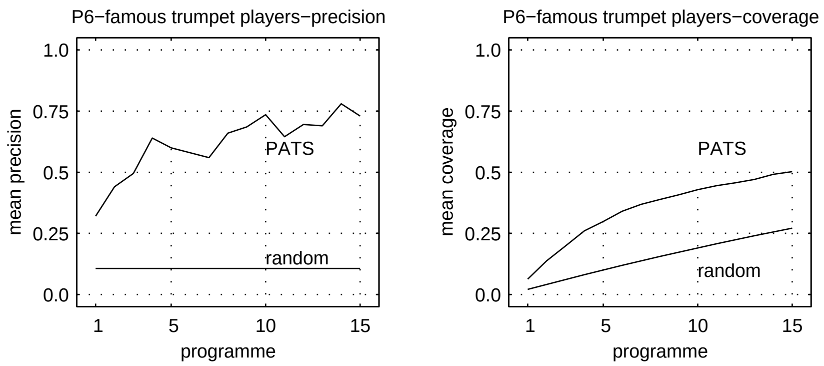

Figure 5.7. Mean *precision* (proportion of preferred options in a programme) and mean *coverage* (proportion of preferred and distinct options across programmes) for the PATS compilation strategy in a simulation and for the random compilation strategy.

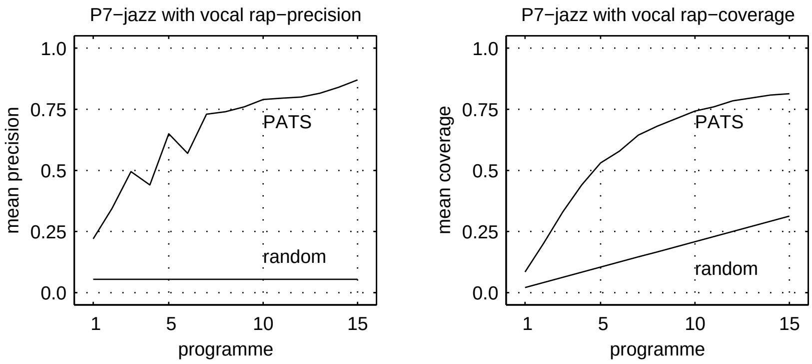

Figure 5.8. Mean *precision* (proportion of preferred options in a programme) and mean *coverage* (proportion of preferred and distinct options across programmes) for the PATS compilation strategy in a simulation and for the random compilation strategy.

.

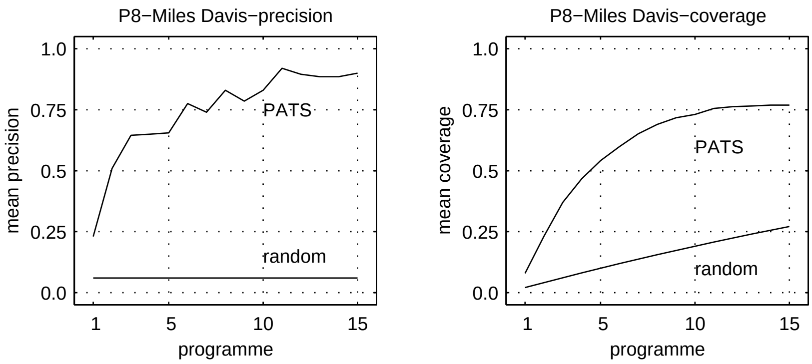

Figure 5.9. Mean *precision* (proportion of preferred options in a programme) and mean *coverage* (proportion of preferred and distinct options across programmes) for the PATS compilation strategy in a simulation and for the random compilation strategy.

Results Study 2 - Three music preferences following each other

Simulation results in terms of *precision* and *coverage* across 15 successive music programmes for three different music preferences are shown in Figures 5.10 to 5.12. The music preferences were addressed one after the other in a fixed order comprising a trial of 45 (3×15) music programmes. In Figures 5.10 to 5.12, means of 20 trails are shown. The results are divided up into an initiation phase (the first five programmes), a progressive phase (the intermediate five programmes), and the steady-state phase (the last five programmes). Results of the mean *precision as* attained in the three phases and mean *coverage* as attained with the last programme are also shown in Table 5.3.

The results for a given music preference in this study were compared to the results of Study 1 in which the corresponding music preference was addressed independently. Reported analyses were done by a two-tailed *t*-test for paired samples in which programmes were treated as pairs.

In Figure 5.10, mean *precision* and mean *coverage* of the PATS compilation strategy are shown of a simulation in which the small-overlapping music preferences P1, P2 and P4 followed each other repeatedly.

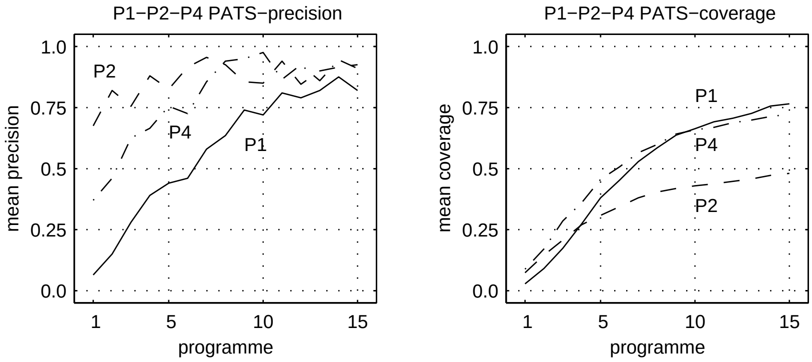

Figure 5.10. Mean *precision* (proportion of preferred options in a programme) and mean *coverage* (proportion of preferred and distinct options across programmes) for the PATS compilation strategy in a simulation in which three music preferences followed each other repeatedly.

When the mean *precision* of PATS programmes in both studies were compared, significant differences for P2 programmes ($t = 3.61$, $p < 0.005$) and P4 programmes ($t = -2.21$, $p < 0.05$) were revealed. Programmes for P2 contained one half of a preferred option *less* in this study than in Study 1. Programmes for P4 contained one half of a preferred option *more* in this study than in Study 1. No difference was found for P1 programmes.

When the mean *coverage* of PATS programmes in both studies were compared, significant differences for all preferences involved were revealed: P1 programmes ($t = 4.50$, $p < 0.001$), P2 programmes ($t = -2.40$, $p < 0.05$) and P4 programmes ($t = -7.23$, $p < 0.001$). Coverage for P1 programmes had a *slower* rate in this study than in Study 1. Coverage of P2 programmes had a *faster* rate in this study than in Study 1. However, equal coverage was obtained at the end of both studies. With respect to P4 programmes, almost 10% (= 4.3) *more* distinct and preferred options were collected at the end of this study compared to Study 1.

In Figure 5.11, mean *precision* and mean *coverage* of the PATS compilation strategy are shown of a simulation in which the moderately overlapping music preferences P2, P7 and P8 followed each other repeatedly. As shown in Table 5.2, the preferred options of P7 were almost totally contained in the preferred set of options covered by P2.

When the mean *precision* for PATS programmes in both studies were compared, significant differences for all music preferences involved were revealed: P2 programmes ($t = 6.84$, $p < 0.001$), P7 programmes ($t = -5.73$, $p < 0.001$) and P8 programmes ($t = -5.73$, $p < 0.001$). Programmes for P2 contained almost one preferred option *less* in this study than in Study 1. Programmes for P7 contained almost one preferred option *more* in this study. Programmes for P8 contained one half preferred option *less* in this study.

When the mean *coverage* for PATS programmes in both studies were compared, significant differences for all music preferences involved were revealed: P2 programmes ($t = -4.22$, $p < 0.005$), P7 programmes ($t = -6.88$, $p < 0.001$) and P8 programmes ($t = 6.27$, $p < 0.001$). For programmes of P2, on average, 3.7 *additional* preferred options were collected at the end of this study when compared to Study 1. For programmes of P8, 1.3 *fewer* options were collected at the end of this study. Coverage of P7 programmes only had a *faster* rate in this study than in Study 1.

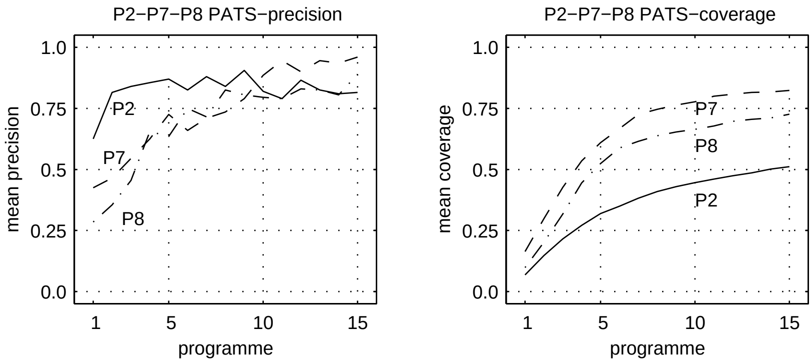

Figure 5.11. Mean *precision* (proportion of preferred options in a programme) and mean *coverage* (proportion of preferred and distinct options across programmes) for the PATS compilation strategy in a simulation in which three music preferences followed each other repeatedly.

In Figure 5.12, mean *precision* and mean *coverage* of the PATS compilation strategy are shown for a simulation in which largely overlapping music preference P3, P4 and P5 followed each other repeatedly. As shown in Table 5.2, almost 75% of the preferred options of P4 were contained in the preferred sets of P3 and P5. A union of preferences P3, P4 and P5 also had eight options.

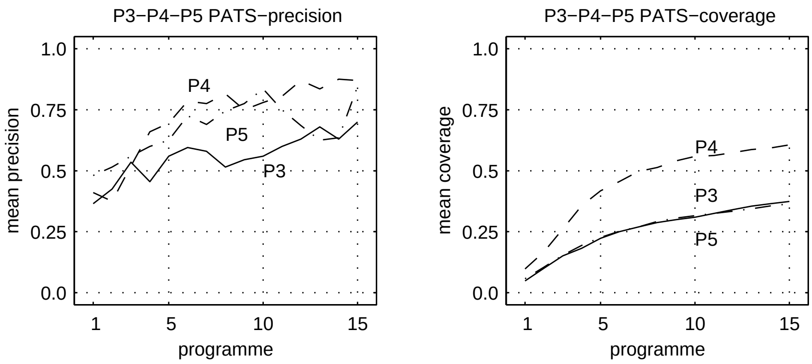

Figure 5.12. Mean *precision* (proportion of preferred options in a programme) and mean *coverage* (proportion of distinct and preferred options across programmes) for the PATS compilation strategy in a simulation in which three music preferences followed each other repeatedly.

When the mean *precision* of PATS programmes in both studies were compared, significant differences for P3 programmes ($t = 5.50$, $p < 0.001$) and P5 programmes ($t = 3.90$, $p < 0.005$) were revealed. Programmes for both P3 and P5 contained, on

Table 5.3. Average mean *precision* (proportion of preferred options in a programme) and mean *coverage* (proportion of distinct and preferred options across programmes) for the PATS compilation strategy in all simulations. Music preferences were addressed independently and in combination with others. Mean *precision* was averaged for the initiation phase (programmes 1-5), the progressive phase (programmes 6-10) and the steady-state phase (programmes 11-15). Mean *coverage* attained from the last programme is shown.

	mean precision			**mean coverage**
music preference	**prog. 1-5 (initiation phase)**	**prog. 6-10 (progressive phase)**	**prog. 11-15 (steady-state phase)**	**prog. 15**
P1	0.26	0.69	0.85	0.77
P1 combined with P2, P4	0.27	0.63	0.82	0.77
P2	0.84	0.95	0.95	0.47
P2 combined with P1, P4	0.79	0.88	0.90	0.48
P2 combined with P7, P8	0.80	0.85	0.82	0.51
P3	0.54	0.72	0.81	0.39
P3 combined with P4, P5	0.47	0.56	0.65	0.37
P4	0.56	0.72	0.90	0.63
P4 combined with P1, P2	0.58	0.90	0.90	0.72
P4 combined with P3, P5	0.53	0.78	0.85	0.61
P5	0.63	0.84	0.95	0.38
P5 combined with P3, P4	0.56	0.75	0.71	0.38
P6	0.50	0.64	0.71	0.50
P7	0.43	0.72	0.82	0.81
P7 combined with P2, P8	0.56	0.76	0.94	0.82
P8	0.54	0.79	0.90	0.76
P8 combined with P2, P7	0.48	0.78	0.83	0.73

average, 1.3 *fewer* preferred options in this study than in Study 1. No difference was found for P4 programmes.

When the mean *coverage* of PATS programmes in both studies were compared, significant differences for all music preferences involved were revealed: P3 programmes ($t = 9.23$, $p < 0.001$), P4 programmes ($t = 2.87$, $p < 0.05$), and P5 programmes ($t = 5.78$, $p < 0.001$). For all three preferences, *fewer* distinct and preferred options were collected at the end of this study when compared to Study 1.

For P3 programmes, 1.2 *fewer* options were collected. For P4 and P5 programmes, 1.1 and 1.7 *fewer* options were collected, respectively.

5.1.4 Discussion

The simulation results indicated how a PATS compilation strategy might adapt to various music preferences in successive music programmes. As shown in Study 1, the results for *precision* showed that PATS programmes adapted to a single well-defined music preference. To do this, only a single preferred option has to be indicated to set up *all* programmes and programmes have to be judged consistently. For almost all music preferences, programmes contained more than 8 out of 10 preferred options after 10 programme judgements.

The results also suggest that at least two factors influence the performance of PATS: 1) *the proportion of preferred options* in the music collection that is covered by a given music preference, and 2) *the complexity of the logical expression* of attribute values in which a given music preference is specified. If a music preference covered few options or if a music preference was complex, adaptation to this music preference in terms of *precision* over time tended to be difficult. It appeared that music preferences P1 and P6 were most difficult to adapt to. For both P1 and P6, this was most likely due to the complexity of its logical expression. In addition, music preference P1 covered a relatively low number of preferred options in the collection.

It is unclear however what precise effects might be expected on the quality of music programmes for human listeners. It is at any rate clear that when a human listener has a complex music preference and approves only a few options from the collection, any music programme will be rejected.

The results for *coverage* suggest that the PATS compilation strategy does not succeed in collecting all preferred options in a sequence of music programmes. If a music preference covered a relatively small number of preferred options (less than 10%), PATS was more successful in collecting most preferred music. If, on the other hand, more than 10% of the music collection was preferred, less than half of that amount was collected in 15 music programmes. Results indicated that no further improvement in future programmes should be expected. Referring to the clustering process of PATS as explained in Chapter 4, clusters do not dissolve or reform to new clusters sufficiently in order to add music in a programme that was not present in earlier programmes. Attaining a higher *coverage* by applying more perturbations to the clustering process can probably only be achieved by sacrificing *precision*. However, to achieve a higher *coverage* rate, human listeners can apply more variation in indicating a preferred music option to set up a programme; the effect is that programme making is then done by using other clusters that also contain preferred music. Other simulation studies indeed showed that almost complete coverage was attained when a preferred option was randomly chosen to set up each new programme (Pauws, 1996).

When addressing multiple preferences in a single trial, the simulation results indicated that the PATS compilation strategy succeeded to maintain partly learned music preferences concurrently, only when there is only a small or a moderate overlap in preferred options between the music preferences. With respect to *precision*, music programmes seem to be compiled as if music preferences were addressed independently, because the size of the effects were small. When there was a massive overlap between music preferences, negative effects on both *precision* and *coverage* were observed.

Although the simulation results seem to strongly produce the desirable behaviour for *precision* and *coverage* in a PATS compilation strategy, they have to be interpreted with caution. It is unlikely that actual music choice and judgement behave in a consistent manner as assumed in the simulations. The simulations did provide an understanding into what factors may pertain to adaptive properties and long-term use of an automatic compilation strategy, and what level of performance might anyway be expected in user experiments.

5.2 USER EXPERIMENT

Following the computer simulation study, a user experiment was run to examine the quality of PATS-compiled programmes and randomly assembled programmes in a more realistic situation. Participants judged the quality of both types of music programmes in two different contexts-of-use, over four experimental sessions. For each combination of participant and context-of-use, a different instantiation of the PATS system was used. In that way, multiple contexts-of-use, that is, music preferences, could not interfere. This was done as the simulations suggested that the PATS compilation strategy is less adequate in handling those situations. Programme quality was measured by *precision*, *coverage* and a *rating score*. In addition, participants completed post-session questionnaires to assess the perceived level of expectations, coherence, variation and suitability of music programmes. When music listeners are in a given context-of-use, it is likely that they develop expectations what the musical content of a music programme should entail before initiating a music listening task. In Chapter 4, coherence and variation were assumed to be relevant criteria for preferred music programmes. Coherence is an emergent property of a music compilation strategy; it refers to the similarity structure of a music programme, based on some sharing of relevant attribute values. Variation is a psychological requirement for continual music enjoyment by introducing new musical content and making the outcome unpredictable. Suitability refers to whether the music programme would be appropriate in the given context-of-use or not. Finally, a post-experiment interview was used to yield supplementary findings on overall quality of music programmes and personal need for an automatic music compilation functionality.

5.2.1 Hypotheses

The simulation results demonstrated that music programmes compiled by PATS were considerably better matched, expressed by *precision*, to a given music preference than randomly assembled programmes. In addition, they showed that the PATS compilation strategy has various levels of performance dependent on factors pertaining to a given music preference. However, the design objective of PATS was to perform equally well irrespective of the given context-of-use. It is therefore hypothesised that:

> (i) Music programmes compiled by PATS contain more preferred music options than randomly assembled programmes, irrespective of the given context-of-use. Similarly, PATS programmes are rated higher than randomly assembled programme, irrespective of the given context-of-use.

PATS programmes also adapted to a given music preference. Similarly, it is hypothesised that:

(ii) Successive music programmes compiled by PATS contain an increasing number of preferred music options. Similarly, successive PATS programmes are successively rated higher.

PATS programmes also contained more distinct preferred options over time than randomly assembled programmes, though PATS was unable to collect all preferred material over time. It is hypothesised that:

(iii) Successive music programmes compiled by PATS contain more distinct and preferred music options than randomly assembled music programmes.

PATS programmes should also correspond more closely to expectations of music listeners, exhibit more coherence, and be judged more suitable for a given context-of-use than randomly drawn programmes. As the outcome of a random compilation strategy is uncertain, its programmes might be preferred for their variation. It is therefore hypothesised that:

(iv) Music programmes compiled by PATS are chosen over randomly assembled programmes with respect to the level of expectation, coherence and suitability.

5.2.2 Measures

Precision

Precision was defined as the proportion of preferred music options in a music programme (see Equation 5.1). Ideally, the precision curve should approach 1, meaning adequate adaptation to a given context-of-use.

Coverage

Coverage was defined here as the number of different preferred music options that appeared in successive music programmes; it sums all preferred options in each new music programme that were not already contained in previous programmes. Equation 5.3 for *coverage* uses a normalisation procedure which can not be made in a user experiment. In spite of the absence of this normalisation, phenomena can still be explained by the course and trend of the *coverage* curve. Over successive music programmes, the *coverage* measure is a non-decreasing curve. Ideally, this curve should approach the total number of music options that were collected in the successive music programmes, meaning nearly complete coverage of preferred material given the number of music programmes.

Rating

After listening to and judging a music programme, participants were asked to rate the music programme on a scale ranging from 0 to 10 similar to the traditional report-mark in Dutch primary 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).

Questionnaire

A questionnaire was handed out at the end of each experimental session. During each session, participants had listened to a PATS and a randomly assembled programme. In four separate questions, participants indicated which of the two music programmes contained more coherence, contained more variation, met expectations more closely, and would better suit the intended context-of-use. The terms coherence, variation, expectations and suitability were not explained to the participants.

Interview

A post-experiment interview was carried out, which was intended to yield supplementary findings about the quality of the presented music programmes and the need for an automatic music compilation functionality. The questions that were posed in the interview are presented in Section III.3 of Appendix III.

5.2.3 Method

Instruction

The twenty participants were not informed about the actual purpose of the experiment; they were told that the research was aimed at eliciting what criteria people use to appraise music. They were informed about the global experimental procedures and the test material, and prepared for the relatively high demands of the experiment: listening to and judging 16 different music programmes in 8 experimental sessions on four separate days.

Participants were instructed to listen to the provided music programmes while imagining a pre-defined context-of-use. The two contexts-of-use in the experiment were described (translated from Dutch) as ‘a lively and loud atmosphere, such as dance music for a party’ and ’a soft atmosphere, such as background music at a dinner engagement’. They were allowed to imagine their personal context-of-use, that is, the general circumstances in which the music would be heard, but were asked to restrict their music listening behaviour to this instantiation during the experiment. They completed a form in which they were asked to describe what musical attributes would be appropriate in the given context-of-use. They were also asked to create a music programme using paper and pencil; they could select music from a list. Both small tasks were intended to elicit some desirable properties that the music should entail. Finally, participants had to select a music option 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 music options in the collection. The selected music option was used to set up a music programme in the experiment.

A control panel for an interactive computer system was implemented as shown in Figure 5.13. Participants were instructed how to operate the control panel. Information about interactive procedures to follow during an experimental session was readily available to the participants during the whole experiment.

Design

A factorial within-subject design with three independent variables was applied. The first independent variable *compilation strategy* 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 levels of *context-of-use* and *compilation strategy* were applied was counterbalanced. The third independent variable *session* referred to the four experimental listening sessions in which music programmes 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 programme quality and appraisal in different contexts-of-use and as a function of time.

Test material and equipment

A music database comprising 300 one-minute excerpts of jazz music recordings (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 music recordings. The attribute representation of the music options is shown in Appendix I. Pilot experiments showed that the brevity and sound quality of the excerpts did not influence judgement. 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 room. They could adjust the audio volume to a preferred level. Both the mouse pad and the monitor were positioned at a comfortable height.

Interactive system

For this experiment, an interactive system was implemented to listen to and judge music programmes using a standard mouse and Graphical User Interface (GUI). Eight buttons on the control panel were required to operate the system effectively. The control panel of the system (see Figure 5.13) consisted of a ’Program’ button to initiate and stop a programme listening task. The title, names of composers and names of artists of the currently selected music option was displayed in a designated window. Controls were provided to navigate through the music programme at will; a participant could manipulate the currently displayed music option using the ‘PREV’ and ’NEXT’ buttons. Controls for common music play features, such as playing, stopping, pausing music, rewinding and fast forward play were provided. Buttons for indicating preference in terms of ‘good’ and ‘bad’ per music option were also provided.

All performed actions pertaining to music judgement were logged for data analysis.

The interactive system was implemented in C using the XView toolkit. This toolkit facilitates building applications that conform to the OPEN LOOK GUI standard and run in the OpenWindows 3.0 environment on top of the SunOS (version 5.5.1, UNIX System V 4.0) operating system for SUN work stations. The GUI was created using the OpenWindows Developer’s Guide 3.0 (DevGuide) GUI builder.

Task

The task was to listen to a set of 11 music recordings that made up a music programme, while imagining a fixed and pre-defined context-of-use. Due to the size of a music programme, judgements of music options were collected by presenting the different music options in series. Participants, confronted with a sequence of music options in a music programme, only had to decide which option did not fit the desired context-of-use (binary forced choice). In the process of listening, participants were allowed to compare music options freely in any combination and cancel any judgement already expressed. There were no time restrictions.

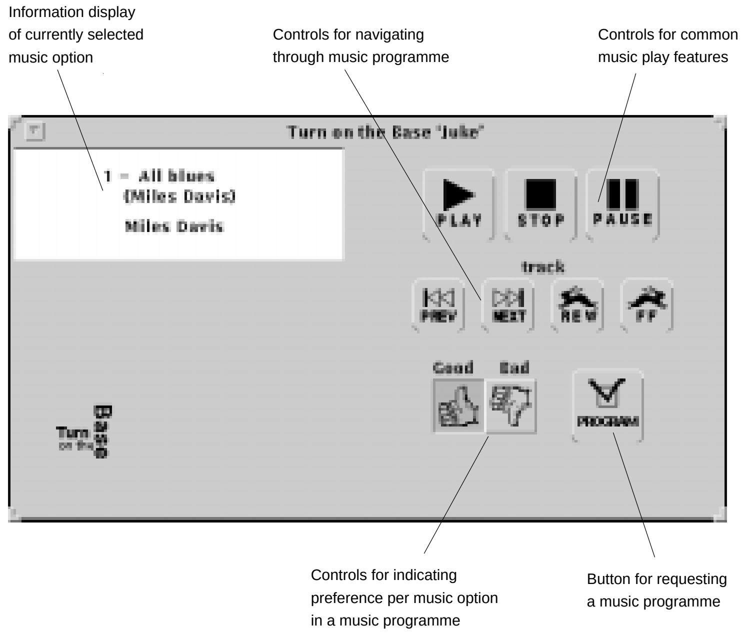

Figure 5.13. Visual display of the interactive system's control panel.

Procedure

Participants took part in eight experimental sessions on four separate days. The first session started with instructions and a questionnaire to record personal data and attributes. Use of the interactive system was explained and demonstrated. During each session, participants were alternately presented a PATS and a randomly assembled programme, 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 (see Instruction). In addition, they were asked to select a music option from the music collection that they definitely would listen to in the given context-of-use. Both this music option and the context-of-use had to be recalled each time a new experimental session started. A PATS and a randomly assembled programme were automatically generated around the selected music option and presented to the participant. Then, a listening and judgement task for the given music programme started. When participants had completed a task, the interactive system was automatically shut down.

After completing each task, participants were asked to rate the music programme just listened to, on a scale ranging from 0 to 10. After completing two tasks in an experimental session, participants completed a post-session questionnaire. At the end of the experiment, a small concluding interview was conducted.

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 of jazz music; for admission to the experiment, they had to be able to rank eight freely recalled jazz musicians on personal taste and number of recordings owned (Geringer and McManus, 1979). The average age of the participants was 26 (min.: 19, max.: 39). All participants had at least completed higher vocational education. Sixteen participants played a musical instrument.

5.2.4 Results

Music programmes contained eleven music recordings from which one was selected by the participant as the one that he / she would definitely want to listen to in the given context-of-use; it was used to set up a music programme. This option was excluded from the data as it was not determined by the system, leaving ten music options per programme to consider for analysis.

Precision

The results for the *precision* measure are shown in Figure 5.14.

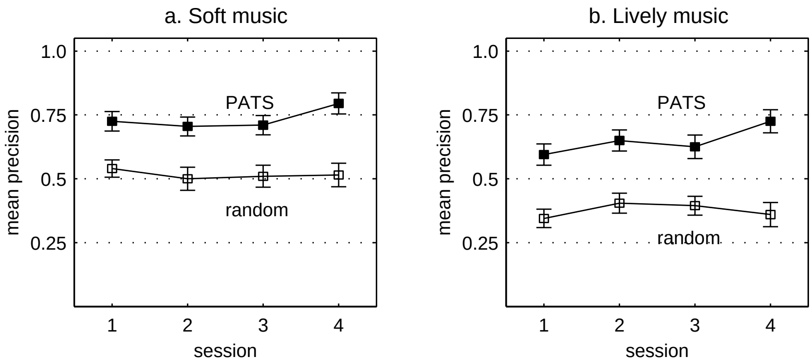

Figure 5.14. Mean *precision* (proportion of preferred music options) of the music programmes in different *contexts-of-use*. The left-hand panel (a) shows mean *precision* for both compilation strategies, that is, PATS and random, in the 'soft music' context-of-use. The right-hand panel (b) shows mean *precision* for both compilation strategies in the 'lively music' context-of-use. The cross-bars represent the standard error of the mean.

A MANOVA analysis with repeated measures was conducted in which *session* (4), *context-of-use* (2), and *compilation strategy* (2) were treated as within-subject independent variables. *Precision* was a dependent variable. A significant main effect for *compilation strategy* was found ($F(1,19) = 89.77$, $p < 0.0001$). Music programmes compiled by PATS contained more preferred music options than randomly assembled programmes (mean *precision*: 0.69 (PATS), 0.45 (random)). A significant main effect for *context-of-use* was found ($F(1,19) = 13.84$, $p < 0.005$). Music programmes for the 'soft music' context-of-use contained more preferred options (mean *precision*: 0.63 (soft music), 0.51 (lively music)). An interaction effect for *compilation strategy* by *session* was just not significant ($F(3,17) = 2.68$, $p = 0.08$), whereas, in the univariate test, it was significant ($F(3,57) = 2.84$, $p < 0.05$). Further analysis revealed a significant difference in mean *precision* between the fourth PATS programme and mean *precision* of preceding PATS programmes, in contrast to randomly assembled music programmes ($F(1,19) = 8.94$, $p < 0.01$). In other words, each fourth PATS programme contained more preferred options than the preceding three PATS programmes (mean *precision* of fourth PATS session: 0.76; mean *precision* of the first three PATS sessions: 0.67). No other significant effects were found.

Coverage

The results for the *coverage* measure are shown in Figure 5.15.

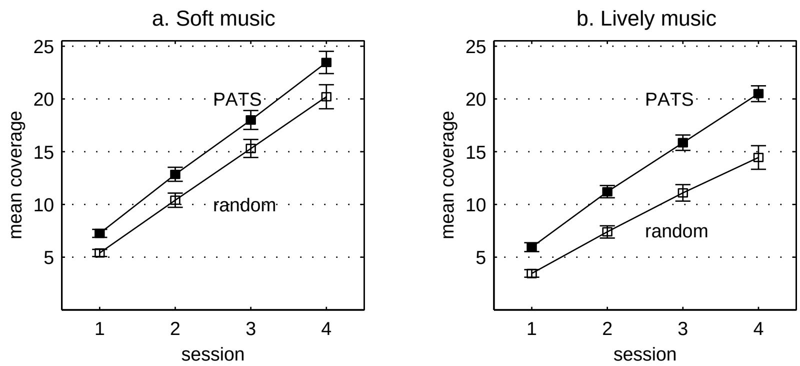

Figure 5.15. Mean *coverage* (number of distinct and preferred music options) of the music programmes in different contexts-of-use. The left-hand panel (a) shows mean *coverage* for both compilation strategies, that is, PATS and random, in the 'soft music' context-of-use. The right-hand panel (b) show mean *coverage* for both compilation strategies in the 'lively music' context-of-use. Note the maximally achievable *coverage* in four successive programmes is 40. The cross-bars represent the standard error of the mean.

A MANOVA analysis with repeated measures was conducted in which *session* (4), *compilation strategy* (2), and *context-of-use* (2) were treated as within-subject independent variables. *Coverage* was a dependent variable. A significant main effect for *compilation strategy* was found ($F(1,19) = 63.17$, $p < 0.001$). More distinct and preferred pieces of music were found in successive PATS programmes than in successive randomly assembled programmes (mean *coverage* at fourth *session*: 22.0 (PATS), 17.3 (random)). A significant main effect for *context-of-use* was found ($F(1,19) = 13.52$, $p < 0.005$). It appeared that music programmes for the 'soft music' context-of-use contained more distinct and preferred options (mean *coverage* at fourth *session*: 21.8 (soft music), 17.5 (lively music)). A significant main effect for *session* was found ($F(3,17) = 284.32$, $p < 0.001$). More particularly, the *coverage* curves for all conditions showed a significantly linear course over sessions ($F(1,19) = 852.27$, $p < 0.001$). A significant interaction effect for *compilation strategy* by *session* was also found ($F(3,17) = 7.60$, $p < 0.005$). Successive programmes compiled by PATS contained more varied preferred music options than randomly assembled programmes. Likewise, the slopes of the coverage curves for PATS programmes appeared to be significantly higher than for randomly assembled programmes (*coverage* slope: 5.2 (PATS), 4.3 (random)). Approximately one more distinct preferred music option was, on average, added to each successive PATS programme. No other significant effects were found.

Rating

The results of *rating* the music programmes are shown in Figure 5.16.

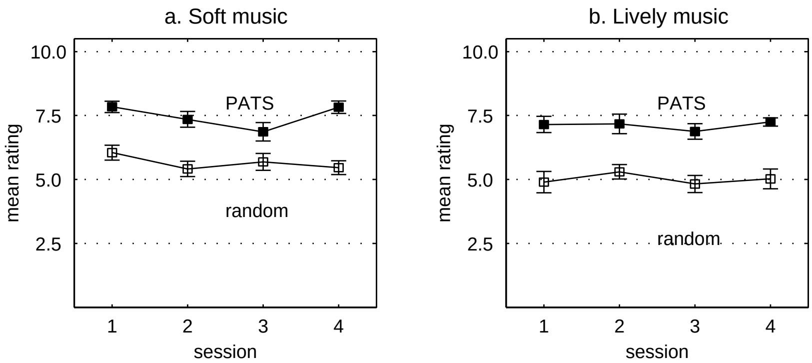

Figure 5.16. Mean *rating* of the music programmes in different contexts-of-use. The left-hand panel (a) shows mean *rating* for both compilation strategies, that is, PATS and random, in the 'soft music' context-of-use. The right-hand panel (b) show mean *rating* for both compilation strategies in the 'lively music' context-of-use. The cross-bars represent the standard error of the mean.

A MANOVA analysis was conducted in which *compilation strategy* (2), *context-of-use* (2), and *session* (4) were treated as within-subject independent variables. The *rating* was dependent variable. A significant main effect for *compilation strategy* was found ($F(1,19) = 85.09$, $p < 0.001$). Music programmes compiled by PATS were rated higher than randomly assembled programmes (mean *rating*: 7.3 (PATS), 5.3 (random)). A significant main effect for *context-of-use* was found ($F(1,19) = 12.57$, $p < 0.005$). Music programmes for the 'soft music' context-of-use were rated higher (mean *rating*: 6.6 (soft music), 6.1 (lively music)). No other significant effects were found.

Questionnaire

After each experimental session, participants were asked, in four separate questions, which of two music programmes they chose with regard to coherence, variation, expectation and suitability to the given context-of-use. The results are shown in Figure 5.17.

Binomial tests showed that, irrespective of the context-of-use, PATS programmes were chosen significantly more often than randomly assembled programmes for their coherence (133 out of 160, 83%), their level of expectancy (137 out of 160, 86%) and their suitability for the given context-of-use (138 out of 160, 86%). On the other hand, randomly assembled programmes were chosen significantly more often for their variation (117 out of 160, 73%). It appeared that participants made consistent choices across experimental sessions as Cochran Q tests (Siegel, 1956) did not determine any effect by *session* ($k = 4$) on the participants' choices ($N = 20$) between PATS and randomly assembled music programmes.

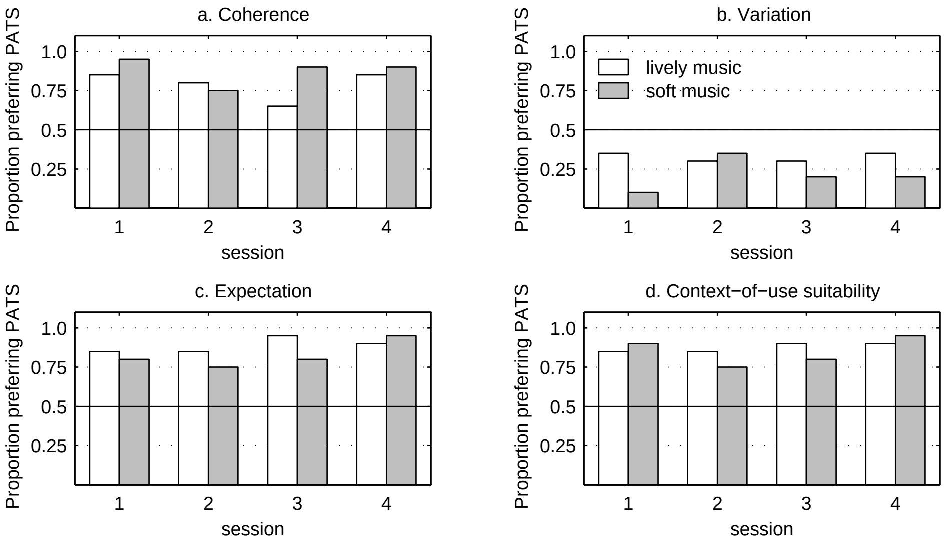

Figure 5.17. Results of the post-session questionnaire. The white bars represent preferences in the ‘lively music’ *context-of-use*, and the grey bars represent preferences in the ‘soft music’ *context-of-use*. Panel (a) shows the fraction of participants preferring PATS programmes for their coherence in different contexts-of-use across sessions. Panel (b) shows their PATS programme preference for variation. Panel (c) shows their preference for level of expectation. Panel (d) shows their preference for suitability.

Interview

The post-experiment interview yielded relevant supplementary findings about the quality of the music programmes and the need for an automatic music compilation system (for the questions posed, see Section III.3 of Appendix III). Note that no distinction between PATS and randomly assembled programmes could be made by the participants, as the participants were unaware of the purpose and the design of the experiment. Consequently, the findings of the interview apply to both types of programmes.

It appeared that fifteen of the twenty (75%) participants judged the music programmes to be predominantly positive. The other five participants found the quality of music programmes insufficient or inconstant.

Seven of the twenty (35%) participants found music programmes for the ‘soft music’ context-of-use more adequate than music programmes for the ‘lively music’ context-of-use. Two participants had the opposite opinion and eleven participants had not observed any difference in music programme quality between the contexts-of-use.

Eight of the twenty (40%) participants commented that they found a quality difference between the two music programmes in a experimental listening session; in general, they commented that there was always a ‘good’ and a ‘bad’ music programme in a listening session.

Nine of the twenty participants (45%) stated that they had observed a trend in which music programmes became more adequate to the given context-of-use over successive sessions.

Twelve of the twenty (60%) participants commented that they found a personal need for an automatic music compilation system; 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 need by referring to easily way-finding in an ever-increasing number of music options. The other eight participants rejected the usefulness of such a system. Their main objection was a loss of control in music selection and the disbelief that a system would be able to compile preferred music. However, one of these participants found an automatic music compilation system only relevant for cafes and department stores.

5.2.5 Discussion

A user experiment examined the quality of four successive PATS-compiled music programmes and four successive randomly assembled music programmes in two different contexts-of-use. The two contexts-of-use were described (translated from Dutch) as ‘a lively and loud atmosphere such as dance music for a party’ and ’a soft atmosphere such as background music at a dinner engagement’. Participants decided which music options in a music programme did not fit the given context-of-use.

According to Hypothesis (i), PATS music programmes should contain more preferred music options and should be rated higher than randomly assembled programmes, irrespective of the given context-of-use. The results showed that PATS programmes contained more preferred music options, that is, had a higher *precision*, and were rated higher than randomly assembled programmes, in both contexts-of-use. On the basis of these results, Hypothesis (i) cannot be rejected.

According to Hypothesis (ii), successive PATS programmes should contain an increasing number of preferred music options. In addition, successive PATS programmes should be rated increasingly higher. The results showed that the fourth PATS programme contained one more preferred option than the first three PATS programmes, which indicated that PATS programmes adapted slightly to a given context-of-use. However, successive PATS programmes were not rated increasingly higher. On the basis of these results, the first part of Hypothesis (ii) cannot be rejected, though the second part must be rejected.

According to Hypothesis (iii), successive PATS programmes should contain more distinct and preferred music options than randomly assembled programmes. The results showed that PATS programmes contained more preferred music options that were not already contained in previous programmes, that is, had a higher *coverage*, than randomly assembled programmes. In particular, the PATS compilation strategy consistently found five new preferred music option, while the random compilation strategy found four new preferred options, for each successive programme. There were no indications that the PATS compilation strategy would deteriorate in finding new preferred music for future programmes. On the basis of these results, Hypothesis (iii) cannot be rejected.

The results also showed that programmes for a ’soft atmosphere’ contained more preferred and more varied music than programmes for a ’lively atmosphere’. ‘Soft music' programmes were also rated higher and seven participants found that 'soft music' programmes were in general better suited than 'lively music' programmes. As this *context-of-use* effect both concerned PATS and randomly assembled programmes, the most likely explanation is that more 'soft' music was available in the music collection than 'lively' music.

According to Hypothesis (iv), PATS programmes should be chosen over randomly assembled programmes with respect to the level of expectation, coherence and suitability. The results showed that PATS programmes were preferred on the basis of these three criteria, while randomly assembled programmes were preferred for their variation. On the basis of these results, Hypothesis (iv) cannot be rejected.

It was also found that a personal need for an automatic music compilation functionality was not uniform. In particular, twelve of the twenty participants found it useful, while the other eight participants rejected its usefulness. The arguments for need were the easy way in which music covering varying music styles and musicians could be listened to. The arguments for rejection were the perceived loss of control in music selection.

5.3 Conclusion

A computer simulation study and a user experiment have examined the quality of PATS music programmes when compared with the quality of randomly assembled music programmes. The main purpose was to investigate to what extent the PATS system enables on-the-fly music selection by automatic compilation of preferred music programmes.

Computer simulations pretend that music listeners judge music consistently according to a well-defined music preference, and hence offer a convenient way to examine adaptive properties of the PATS system. Under simulation, the PATS system adapts to various music preferences in successive music programmes, yielding more than 80% precision. However, the rate of adaptation differs across preferences and the PATS system does not succeed in collecting all preferred music in the collection. To achieve complete coverage, a human listener should apply more variation in selecting a preferred option to set up a programme. Adaptation was generally influenced by the *proportion of preferred music* in the collection and the *complexity of the given music preference*. If, in the extreme case, only a small portion of the music in the collection is preferred, and preference is typified by many attribute values, adaptation will obviously be limited. Only if there is a massive overlap of preferred music between multiple music preferences, simultaneous adaptation to these preferences is worse than the adaptation to each preference independently. The PATS system finds difficulties in discerning massively overlapping music preferences.

Given the simulation results, a set of hypotheses regarding quality and adaptivity of PATS programmes was developed for the user experiment. Three desirable properties of the PATS compilation strategy in comparison to a random compilation strategy were observed in the user experiment. Firstly, PATS programmes are more preferred than randomly drawn programmes. Secondly, successive PATS programmes cover preferred music better in the collection than randomly drawn programmes. And thirdly, successive PATS programmes adapt slightly to a given context-of-use.

Only jazz music and jazz listeners were tested in the user experiment, which makes generalisation of the results to other music idioms, or a wider population, difficult. Also, only two different contexts-of-use were used in the experiment, whereas PATS should perform equally well in all circumstances. The PATS system was not tested in conditions in which it had to cope with multiple music preferences simultaneously. In daily use, however, it is very likely that different (and more erratic) music preferences would be common. This aspect of long-term use was not considered in the experiment, although it is a valuable variable to study.

However, automatic music compilation was not generally perceived to be useful. Music listeners question the possibility of losing control over their music choice, which will be further investigated in the next chapter.

5.3.1 Trade-off between precision and coverage

In the PATS compilation strategy, there is a trade-off between *precision* and *coverage*. Results of the computer simulations indicated that complete *coverage* and high *precision* are hard to achieve simultaneously. Results of the user experiment showed that successive PATS programmes did not reach full *precision*, while *coverage* increased steadily. Likewise, Information Retrieval (IR) systems often find a difficulty in improving both the measures *precision* and *recall*: improving *precision* typically results in *recall* becoming worse and vice versa (Salton, 1989).

To understand this trade-off, note that an automatic music compilation system has essentially two strategies in composing a preferred music programme. The first strategy consists of collecting music options for which the listener has already expressed a preference. As this strategy is restricted to preferred music that the music listener has already listened to, it *exploits* the music preference that a listener is known to have. When applying this strategy, *precision* is likely to increase, but *coverage* will not be enhanced. The second strategy consists of collecting options for which the listener has not yet expressed a preference, but which are similar to preferred options. As this strategy is focused on finding new music that the music listener has not yet listened to, it *explores* other aspects of the music preference that a listener might have. It is not known beforehand whether or not the new music will be appreciated; *coverage* will increase, if the new music is appreciated. *Precision* will only increase when the new music contains more preferred music than the music that it replaces. PATS does not explicitly acknowledge a mix of both strategies, but is heavily focused on the exploratory strategy; it tries to find new music to create varied programmes over time. In the user experiment, it appeared that PATS exchanged, on average, 8.2 music options in one programme containing 10 options with other options in a next programme. Of these 8.2 options, 5.7 options were preferred. In contrast, the random compilation strategy collected 9.7 other music options in each successive programme, of which 4.5 options were preferred. It may be desirable to devise a user control mechanism that adjusts the extent to which both strategies within PATS are combined and, thus, controls the level of variation between successive PATS programmes. With respect to this issue, Balabanovic (1998) already investigated the adaptive properties of a combined exploitation and exploration recommendation strategy for a web recommender system through a series of computer simulation studies. His findings suggested that by varying the mix of exploratory and exploitative strategies, the user is given a predicted and well-behaved control of the recommendations.

CHAPTER 6

The effects of music recommendations on music programming

Two experiments were carried out to examine required effort, music listening behaviour, and music selection behaviour in programming preferred music, with and without a music recommender system and with and without a time constraint. Music recommender systems provided randomly drawn recommendations or algorithmically determined (PATS) recommendations. The general hypothesis is that music listeners use music recommendations during their search for preferred music, because it reduces navigation and search effort and it improves task performance and the quality of their music choices. Task performance measures included *time on task*, *number of actions* and *proportion of navigational actions.* Music listening measures included *number of options listened to* and *average listening time* per option. Programme quality was assessed by *precision*, that is, the proportion of preferred options in a programme. Results showed that half of the chosen music was recommended by the system. When given sufficient programming time, participants included more PATS recommendations in their programmes than randomly drawn ones. Thus, participants used music recommendations during their search for preferred music. When using music recommendations, participants performed proportionally fewer navigational actions. Thus, navigation and search effort was reduced when using recommendations. Differences in task performance and programme quality across the programming systems were, in general, too small to produce significant effects. Thus, in contrast to the general hypothesis, the use of recommendations did not significantly improve task performance and the quality of music choices. It is likely that participants were carried away by the ample music selection and listening opportunities, which makes them less sensitive to task efficiency and programme quality. An undesirable property of the use of music recommendations is that it can cause music listeners to be less able to select and listen to music at will. A music recommender system, especially the PATS system, repeatedly confronted participants with music they had already heard. Consequently, participants listened less selectively to more of the same options. This perceived loss of control in music selection should be avoided in future recommender systems. However, the PATS recommender system was the most preferred and most useful feature for a music programming task when compared to other systems. It therefore seems that task efficiency, programme quality and perceived loss of control are far *less* decisive factors for determining a preferred way of music programming than appreciation of a functionality and enjoyment of a music programming task. Implications for interaction style design pertaining to *adaptation to user choice behaviour* and *interactive information presentation* of music programming systems are discussed.

The previous chapter demonstrated that automatic music compilation would be an adequate means to enable faster selection of preferred music. It is unclear whether this outweighs a perceived loss of control in music selection that would probably be associated with it. This chapter describes the design and results of two experiments investigating the effects of a music recommender system on a music programming task. More particularly, the experiments examined required effort, music listening behaviour and music selection behaviour in music programming. Tasks had to be carried out with and without assistance of music recommendations, and with and without a time constraint. The general hypothesis is that music listeners prefer to use recommended music in their music choice strategy because of its perceived usefulness; using a music recommender system reduces effort in navigation and search and improves task performance and music choice quality. A part of the first experiment is also reported elsewhere (Pauws, Eggen and Bouwhuis, 1997).

6.1 CHOICE AND MUSIC PROGRAMMING

Characteristic differences and similarities between classical choice tasks and a music programming task were already discussed in Chapter 2.

In short, a music programming task differs from classical choice tasks because it has no optimal solution, has personal appeal, consists of multiple serial choices, extends over time, and is a search for poorly defined targets. It is also obvious that it is impractical to examine a large music collection completely and thoroughly, which forces music listeners to apply a simplifying choice strategy and to consider only part of the available music. Finally, time constraints make music listeners pay additional attention to completing the task on time. It is likely that they perceive this constraint as making them less in control when searching for preferred music, when listening to music and when making satisfactory music choices. On the other hand, if programming time is unlimited, music listeners are likely to be carried away by the ample listening and selection opportunities of a large music collection, and may forget about the apparent passage of time.

6.1.1 A default choice strategy for music programming

In the experiments described in this chapter, music programming proceeded entirely by using an interactive system. A detailed description of the systems can be found in Section 6.2.3 and 6.4.3. It is emphasised here that the use of an interactive system for music programming leads to possibilities and restrictions to conduct a music choice strategy, henceforth referred to as, a *default* choice strategy. A default choice strategy covers essentially a personal search for preferred music without outside assistance of recommendations. Therefore, this strategy describes the order in which actions are performed to navigate through a music collection.

The interactive systems present music options in a hierarchical structure on music style, music artist and music titles, allowing music listeners to browse through music styles, music artists and music titles. In particular, music listeners can pre-select music that meets a particular music style, music artist or music title, and temporarily eliminate other options for further consideration. These pre-selected options are presented in an alphabetically ordered list by music artist and music title; other musical attribute information is not shown. It is unlikely that music listeners go down the entire list and listen to each option sequentially. Music listeners are more likely to be triggered into briefly listening to particular options after seeing the cues about the presented attribute information. If music appeals to the listener, further listening is opted. Consequently, music listeners divide their time and attention for music listening at will. If they spread their listening time unevenly across options (listening to one option for a longer time and to another for a shorter time), it is said that they *selectively* listen to music and have control on their music listening behaviour. After listening, music listeners decide whether the music option should be included in the programme or not. Searching for other options is continued until the programming task is completed.

6.2 Experiment I: Programming with no time constraint

Experiment I was concerned with assessing the required effort, the required time and music listening behaviour, when the sole task instruction was to collect preferred music options using an interactive programming system. Experimental conditions were defined in which participants received no outside assistance or were assisted by a music recommender system.

6.2.1 Hypotheses

Given ample time, music listeners are assumed to be able to collect preferred music; the quality of a music programme must not differ under various experimental conditions. Using music recommendations should enable faster programming. It is therefore hypothesised that:

(i) A music recommender system reduces the time that music listeners spend on programming preferred music.

Prior experience with the music collection and task is expected to produce learning effects. It is therefore hypothesised that:

(ii) Over several sessions, music listeners spend less time and fewer actions to perform a music programming task.

While programming music, music listeners perform actions to navigate and search in a music collection. If music recommendations are provided, the listeners need to perform additional actions to control the recommender system. However, when a recommendation is added to the programme, this does not need any navigational actions. If a significant portion of compiled options are recommendations, this means that music listeners will need to perform fewer actions for navigation and search. It is hypothesised that:

(iii) A music recommender system causes the music listener to perform a smaller proportion of the navigational actions.

It might be the case that a music recommender system results in the music listeners listening to fewer *distinct* music options. Part of the recommended music may be already familiar to the music listener or may be provided repeatedly, which increases the total amount of music listened to. Familiar music generally does not need further listening. Consequently, music listeners are assumed to listen to music for a shorter time and less selectively. If, in addition, music recommendations are provided without direct user control, even more music options that are probably already familiar need to be listened to. This reduces listening time per option further. In summary, it is hypothesised that:

(iv) A music recommender system causes music listeners to listen to fewer *distinct* options, to more music options more than one, and to more options in general. Providing recommendations without direct user control results in listening to even more options in general.
(v) A music recommender system causes music listeners to devote less listening time to each music option, and to listen to music options less selectively. Providing recommendations without direct user control results in even shorter listening

times and even less selective listening behaviour.

The user experiment in the previous chapter found that PATS music programmes contained more preferred music than randomly assembled programmes. Hence, it is assumed that randomly drawn recommendations are less preferred than algorithmically determined (PATS) recommendations. It is therefore hypothesised that:

(vi) Fewer randomly drawn recommendations are included in a music programme than PATS recommendations.

Users who control a music recommender system must divide their attention between monitoring visual information on various displays and performing additional control actions, which do not have any direct bearing on the music programming task. Music listeners may have to learn more skills and may experience less control of their music selection process. It is therefore expected that the usability of a programming system with music recommendations will be valued lower than a programming system without music recommendations. However, the functionality of a music recommender system may be appreciated. It is therefore expected that the usefulness of a programming system with a recommender facility will be valued higher than a programming system without a recommender. If we compare randomly drawn recommendations to algorithmically determined (PATS) recommendations, it is expected that the usefulness of a PATS music recommender system will be most appreciated. In summary, it is hypothesised that:

(vii) The usability of a programming system with a recommender is valued lower than a system without a recommender.
(viii) The usefulness of a programming system with a PATS recommender is valued higher than other programming systems.
(ix) A programming system with a PATS recommender is preferred when compared with other programming systems.

6.2.2 Measures

Task performance measures

The quality of a music programme was measured by *precision*. *Precision* was defined as the proportion of preferred music options in a music programme containing 10 music options (see also Section 5.1.1). Preferred music options were options that fit the intended context-of-use. The *proportion of recommendations included* in a music programme was also measured.

To assess the amount of required programming effort, three measures were used. *Time on task* measured the time that elapsed between the time at which the first music option was added and the time at which the last music option was added to the music programme. *Number of actions* measured the number of actions that were performed in the time span specified by *time on task*. *Proportion of navigational actions* was calculated to measure the proportion of actions that was associated with a default choice strategy pertaining to navigation and search in the music collection.

Six measures were used to assess systematicity and selectivity of music listening. *Number of options listened to* measured the total number of music options listened to in the time span specified by *time on task*. Listening to music in this time span could involve repeatedly listening to identical options and listening to music recommendations. Therefore, *number of distinct music options listened to, proportion of* *music recommendations listened to* and *proportion of music options listened to more than once* were measured. The latter measure was calculated by the ratio of the number of options that were listened to more than once and the *number of distinct options listened to*. *Average listening time* per music option was calculated to measure the time ascribed to music listening in the above mentioned time span. *Standard deviation of average listening time* was used to assess the selectivity of music listening; lower values indicate that listening time per music option was evenly spread during the task, whereas higher values indicate a more selective listening behaviour.

Perceived ease of use and perceived usefulness

The Technology Acceptance Model (TAM) (Davis, 1989) defines two subjective terms related to usability and usefulness of an interactive system. In our experimental setting, the term *Perceived Ease of Use* (PEU) refers to the extent to which a user finds a programming system easy to learn and use. The term *Perceived Usefulness* (PU) refers to the extent to which a user finds a programming system to be an aid for a music programming task. A questionnaire recommended by TAM to assess both terms (Wiedenbeck and Davis, 1997) was slightly modified to reflect the music programming domain. Each term was assessed by posing four positive statements. Participants responded by stating to what extent they agreed with each statement in the questionnaire on a 7-point scale (1 = strongly agree,..., 4 = neutral,..., 7 = strongly disagree).

The statements assessing *perceived ease of use* were the following (translated from Dutch):

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.

The statements assessing *perceived usefulness* were the following (translated from Dutch):

Q5 I find that by using the system the quality of a music programme improves.

Q6 I find that by using the system I am able to compile a music programme more rapidly.

Q7 I find that by using the system the enjoyment of music programming is enhanced.

Q8 I find this system useful at home.

Order of preference

Order of preference of the music programming systems was assessed by having participants rank the systems. Rank value 1 was assigned to the most preferred system for music programming.

Task descriptions and comments

After completion of a music programming task, participants were asked to write down how they had pursued the task, including their thoughts and decision strategies. Also, they were asked to write down what kind of system properties they would like see improved, but also what kind of system properties were well-designed.

6.2.3 Method

Design

Five interactive music programming systems were used denoted by A, B, C, D, and E, defining an independent variable *system*. Each system represented a combination of two other independent variables: *recommendation* and *locus of control*. Three levels for *recommendation* were defined as: 1) *none*, no recommendations, 2) *random*, randomly drawn recommendations, and 3) *PATS*, algorithmically determined recommendations. Three levels were defined for *locus of control*: 1) *self*, participants had to compile a programme without any outside assistance; 2) *user control*, participants could request recommendations by pressing a button; 3) *system control*, the system provided recommendations at fixed instants determined by particular actions of the participants, but beyond their direct control. Note that both user-controlled and system-controlled recommendations were immediately played back in order to speed up operation. Obviously, it would be impossible to implement nine different systems representing all possible combination of the variables *recommendation* and *locus of control*. Therefore, as shown in the left-hand panel of Figure 6.1, the experimental design was not factorial; only five cells in the design were filled by a programming system. Three successive experimental sessions made up the last independent variable named *session*. These sessions were intended to measure changes in task performance, perceived system usability, perceived system usefulness and system preference as a result of experience.

recommendation \ *locus of control*	self	user control	system control
none	A	-	-
random	-	B	C
PATS	-	D	E

Block	Conditions	Block	Conditions
(1)	A B C	(6)	A B D
(2)	A B E	(7)	A C D
(3)	A D E	(8)	A C E
(4)	B C D	(9)	B C E
(5)	C D E	(10)	B D E

Figure 6.1. Balanced incomplete block design and notation system of Experiment I.

A balanced incomplete block design from Winer (1962) was used, as it was not experimentally feasible to assign each participant be assigned to all five systems. As shown in the right-hand panel of Figure 6.1, participants were randomly assigned to a block containing three interactive systems instead of all five. Each block was repeatedly carried out in a separate session. To facilitate reference to interactive systems in post-task inquiries, the three different systems in a block were randomly presented in the colours white, grey or black. A *balanced* incomplete block design requires that every block must contain an equal number of systems, that every system must appear in an equal number of blocks, and that every system must appear an equal number of times with every other system (Winer, 1962). To compensate for order effects, the three sessions contained different permutations of the systems in a block (e.g., three sessions addressing block (1) had the following permutations: A B C, B C A, and C A B). The order in which these permutations were carried out was counterbalanced.

In summary, the balanced incomplete block design was designed to compare 5 programming systems, 3 systems per block, by a required number of 30 participants invited for 3 experimental sessions (excluding a familiarisation session). Each system was used by 18 different participants. At each session, each system was tested against any other system by 9 different participants.

Test material and equipment

A music collection comprising 480 first-minute excerpts of jazz music recordings (MPEG-1 Part 2 Layer II 128 Kbps, stereo) from 160 commercial CD albums served as test material. Pilot experiments made clear that the brevity and sound quality of the excerpts did not influence preference. The collection comprised 12 popular jazz styles. These styles cover a considerable part of the whole jazz period. Each style contained 40 recordings. The test equipment consisted of a Silicon Graphics Indy (SGI), on which the interactive programming system was implemented, a SUN Sparc-5 workstation, which was mainly used as a music storage device, a mid-range audio amplifier (Philips Integrated Digital Amplifier DFA888), and a pair of high quality loud speakers (Philips 9818 multi-linear 4-way). Music was transported as MPEG data over the Ethernet; there were no noticeable interruptions in the data stream. Real-time MPEG decoding by software took place at the SGI machine.

Participants were seated, in a comfortable chair, in front of a 17-inch monitor (Philips Brilliance 17A) in a sound studio, originally designed for speech recordings. They could adjust the audio volume to a preferred level. Both the mouse pad and the monitor were positioned at a comfortable level.

The experimenter was seated in a control room next to the studio and was provided with an additional monitor linked up to the monitor of the participant, allowing real-time observation and supplementary data collection of the task progress.

Interactive system

For this experiment, five interactive music programming systems were implemented using a standard mouse and a Graphical User Interface (GUI). The systems were denoted by A, B, C, D and E in the experiment. Besides controls to compile a music programme, four systems offered randomly drawn or PATS music recommendations in a user control mode or a system control mode. Recommendations were automatically generated while participants conducted the task.The music recommendation algorithm used was Personalised Automatic Track Selection (PATS), as described in Chapter 4. The PATS recommender system basically monitored what music was listened to, accepted (added to the programme or marked as *good*), and rejected (removed from the recommendation or programme list or marked as *bad*). Music that was accepted or rejected in the course of programming was subjected to an inductive learning algorithm to infer relevant attribute values (see Chapter 3). Knowledge of these attribute values was used to set up new clusters to link similar music (see Chapter 4). Options that appeared in the same cluster were used as mutual recommendations. However, rejected and already compiled options were not contained in future recommendations. Music that had already been listened to had a lower probability of occurring in future recommendations.

Three slightly different control panels were implemented for each level of *locus of control*. As shown in Figure 6.2, the control panel was divided into three sections with displays containing from left to right the music collection, the music programme and the music recommendations.

A section on the left-hand side presented the controls to navigate through the music collection. Music options (tracks) were displayed by title and artist in a scrolled list. Music options that were included in the music programme, not or only listened to, or rejected from the music programme during the course of the programming task were indicated as *good*, *neither good nor bad* and *bad* respectively. Controls were provided to set preference (*good*, *neither good nor bad* and *bad*) for a music option explicitly by the participant. Music options that were listened to were marked by a tick. A music option was selected by positioning the mouse pointer and pressing the left mouse button; only one option could be selected at a time and a selected option was highlighted. Controls were provided to show a different subset of the music collection such as music options from a particular music style or music options whose titles or whose artist surnames started with a particular letter. This subset was alphabetically ordered by titles or by artist names. In this way, participants could choose to either browse through the complete collection referred to as ’all styles’, through a particular music style, through a set of music artists, through a set of titles, or through a combination of style and artist or title. For experimental purposes, a button was placed at the bottom of the control panel to start or stop the music programming task.

The middle section of the control panel displayed a list of music options added to the music programme. Beneath this list, buttons for common music playing features were placed. To play music, participants selected a particular music option in any of the lists and pressed on the ’play’ button. The system continuously played the appropriate music while participants were selecting other options, until music playing was switched off.

The right-hand section was not presented in system A, and differed for systems B, D and systems C, E. This section displayed a small list holding three music recommendations. In the systems B and D, participants could personally request music recommendations (user control). To do this, they selected a music option in one of the other two lists and pressed a button displayed on the right-hand side of the control panel (the ‘Recommendation’ button in Figure 6.2). The system then displayed three recommendations and immediately selected and played back a recommendation at the bottom of the list. Immediate playback of recommendations was intended to speed up operation and emphasise external system control. In the systems C and E, participants had no control over the presentation of recommendations (system control). No button was present. Recommendations were displayed or updated each time a new music option was added to the music programme, or each time an option from the recommendation list was removed. As in the previous case, the system selected and immediately played back a recommendation at the bottom of the list when the list was updated.

Actions for navigation through the music collection were carried out in the left-hand section of the control panel. The actions included scrolling a list, selecting a music style or selecting a music option, and were considered to be part of the *default* choice strategy (see Section 6.1.1). Actions for collecting options in a music programme were implemented by a ‘drag-and-drop’ mechanism; music options could be picked up from any list, transported to another list and released. In the

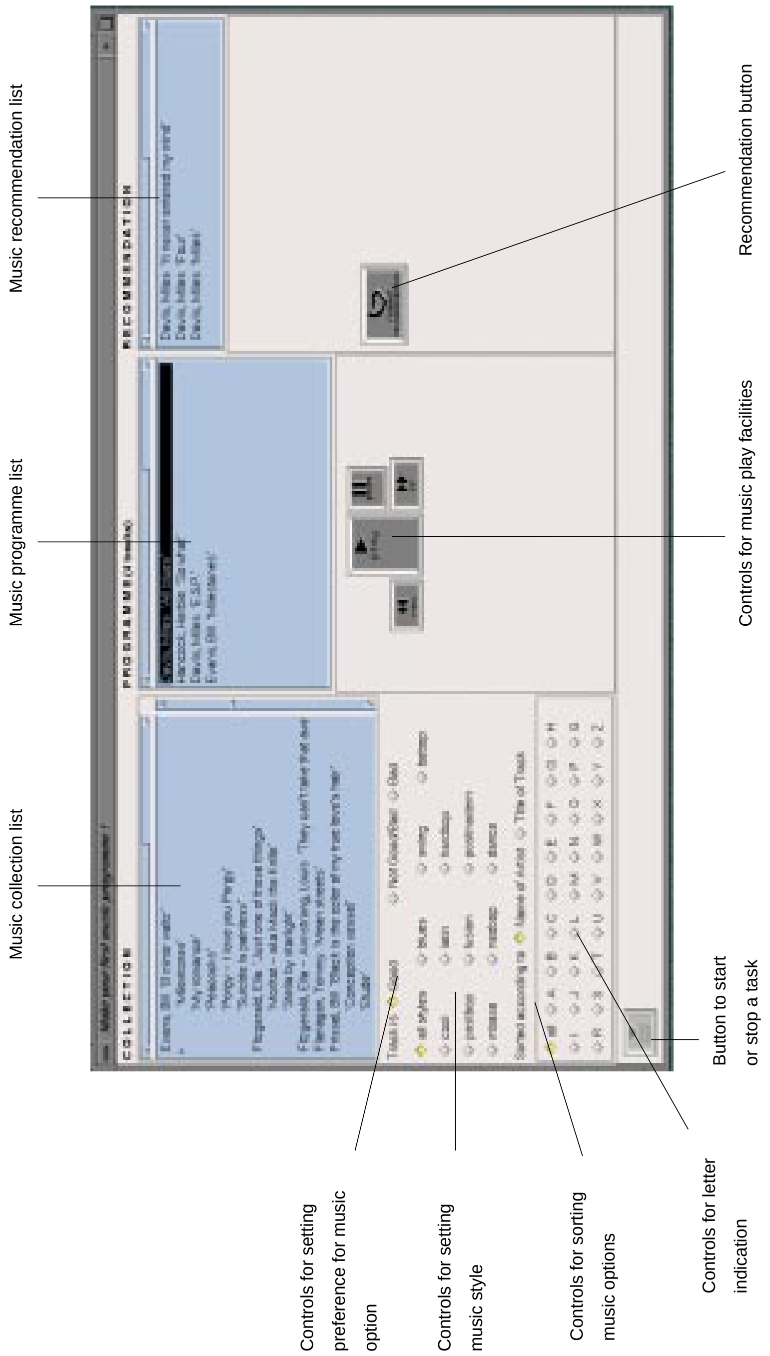

Figure 6.2. Visual display of the interactive programming system B and D providing user-controlled recommendations. Four music options are added in the music programme, and three recommendations of the highlighted option in the music programme are displayed.

same way, the order of music options in a music programme could be changed. Releasing a music option into the music programme list included that option in the programme; the option was then regarded by the system as being preferred (*good*). If a music option, originating from the music recommendation list or music programme list, was released into the music collection, that option was returned to the music collection; the option was then regarded by the system as being rejected (*bad*). Music recommendations that were not listened to were regarded by the system as being *neither good nor bad.*

All actions that were performed by participants were logged. A time indication and an indication of on which music options and control elements the action was performed were also kept.

The interactive system was implemented in C / C++ using the OSF / Motif toolkit on top of the Silicon Graphics Indy IRIX 5.3 operating system. The GUI was created with the X-Designer GUI builder.

Procedure

Thirty participants returned for four experimental sessions on four separate days. The last three sessions were used to acquire experimental data; these are referred to as the experimental sessions.The first session was intended to acquaint participants with the music collection and to assess their musical taste. During this so-called familiarisation session, participants were informed about the experimental context. Participants were then asked to indicate seven jazz styles from a list of twelve they were accustomed to listening to. The selection of seven jazz styles, which was assumed to cover their musical taste, was facilitated by listening to randomly selected music options from each jazz style. Subsequently, they were given the opportunity to browse and listen at will to all 280 music pieces within the seven jazz styles. Only music options in the seven preferred jazz styles were used in the experimental sessions.

At the beginning of each experimental session, participants received brief instructions on use of all three interactive system they were supposed to work with during that session. The instructions included a demonstration by the experimenter and a practice round by the participant, if desired. From the second experimental session onwards, participants commented that instruction was not required. The participants were then asked to concisely express a specific context-of-use for which they would like to create a music programme; they were allowed to define a context-of-use at will. Some examples were 'music while cooking', 'music while studying at night' and 'music for a birthday party'. Subsequently, participants completed the creation of three music programmes by means of three different programming systems. They retained the same imagined context-of-use throughout the session. Each music programme had to contain *10 distinct preferred music options* while repetitions across the three tasks in a session had to be avoided as much as possible. Music options could be listened to as many times as participants desired. No clues were given on how the task should proceed, or how music options should be examined and evaluated. Music options could be added or removed individually to or from the music programme under construction; in the process of compilation, participants were allowed to cancel any option already compiled, and change the order of options within the programme. Time to perform the task was unlimited and speed of operation was not presented as a criterion of success. Quality of the music programme was presented as the sole optimisation criterion.

After each programming task, participants completed the questionnaire assessing *perceived ease of use* and *perceived usefulness* of the interactive system. Participants also ranked the three interactive systems with rank value 1 assigned to the most preferred interactive system for music programming. They were also asked to write down how they had pursued the task and to comment on desirable and undesirable system properties. At the end of each session, participants were presented with the music programmes, they had created just moments earlier. They were asked to judge what music options in the programme did not fit the intended context-of-use while listening to them for the second time.

Participants

Thirty participants (24 male, 6 female) were recruited by advertisements and all got a fixed fee. All participants were frequent listeners to jazz music; for admission to the experiment, participants had to be able to rank eight freely recalled jazz musicians on personal taste and number of recordings owned (Geringer and McManus, 1979). The average age of the participants was 31 (min: 19, max: 57). Of these participants, 18 had received a musical education or played a musical instrument on a regular basis. All participants, except four, had at least completed a higher vocational education.

6.2.4 Results

As the actions associated with changing the order of music options within a music programme were not defined as a part of the music programming task, time and actions associated with this ordering were removed from the data. Time measures, such as *time on task*, and action measures, such as *number of actions*, were adjusted accordingly.

Preliminary analysis revealed that data on *time on task* contained extreme values for one participant. The mean *time on task* under all conditions was 2144 seconds (35 min. 44 sec.) for this participant, whereas the mean *time on task* for all other participants was 710 seconds (11 min. 50 sec.). The participant stated that he had experienced no difficulties in completing the task, but argued that a thorough examination of the whole music collection was necessary each time a preferred music programme had to be made. Though time performance was not a success criterion in the experimental instructions, the participant's time performance deviated severely from the time performance of others. Therefore, task performance data of this participant were not subjected to the analyses. Data on other measures were left unchanged.

Data of proportions with different denominators were logit-transformed according to the formula

$$\text{logit}(p) = \log\left(\frac{p}{1-p}\right) \tag{6.1}$$

where p denotes proportions.

As some of the cells in the design contained missing data because participants did not execute tasks in all conditions, missing values were replaced by the mean value of measures of each participant. This is a conservative procedure to cope with missing data, but it does not introduce spurious effects. Effectively, it decreases relatively high means in conditions and increases relatively low means. Consequently, it decreases the effect size of existing effects between conditions or may even hide possible effects between conditions. Tables and figures present statistics of the original data. All reported analyses of variance (MANOVAs) with repeated measures were conducted with the within-subject independent variables *session* and *system*, unless stated otherwise.

Task performance measures

In the familiarisation session, participants selected and listened to more than half of the options, and one participant listened to all options (mean *number of options listened to*: 148.1, min: 31, max: 280).

As a control mechanism, we must first check whether programme quality, in terms of *precision*, differed under various experimental conditions. A MANOVA with repeated measures was used with *precision* as a dependent variable. No effects were found. Participants preferred their music programmes to a great extent (mean *precision* = 0.95; note that a programme contained 10 options).

Hypotheses (i) and (ii)

According to Hypothesis (i), a music recommender system should reduce compilation time, that is, *time on task*. According to Hypothesis (ii), less time and fewer actions for programming should be required for each successive session. Therefore, two MANOVAs with repeated measures were conducted. In the first analysis, *time on task* was used as a dependent variable; in the second, *number of actions* was used. The results on *time on task* are shown in Figure 6.3.

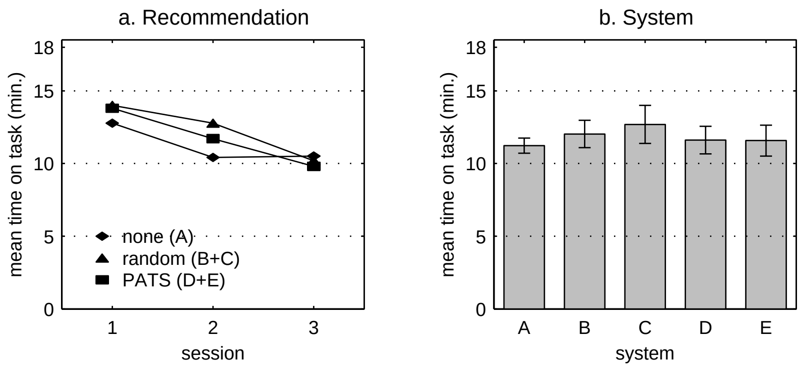

Figure 6.3. The left-hand panel (a) shows the mean *time on task* across the variable *recommendation* over three sessions. The right-hand panel (b) shows the mean *time on task* for the programming *systems*. Crossbars represent the standard error of the mean.

In the analysis of *time on task*, a significant main effect for *session* was found (F(2,27) = 9.58, $p < 0.005$). When the means were compared, a significant difference was revealed between the first and the last two sessions (F(1,28) = 18.20, $p < 0.001$), and between the second and last session (F(1,18) = 8.20, $p < 0.01$). As shown in the left-hand panel of Figure 6.3, participants spent less time on compiling a programme in each successive session (mean *time on task* over three sessions: 13 min. 35 sec., 11 min. 45 sec., 10 min. 11 sec.). A significant main effect for *system* was also found (F(2,27) = 3.15, $p < 0.05$). When the means were compared, it was revealed that variation was largely due to a significant difference between system A and system E

($F(1,28) = 10.40$, $p < 0.005$). This *time on task* difference can not be seen in the right-hand panel of Figure 6.3, because it contains the mean values over all thirty participants who worked with systems in any combination, and only nine participants both worked with system A and E. These nine participants spent more time on compiling a programme when using system E than when using system A (mean *time on task* for participants who used both systems A and E: 9 min. 42 sec. (system A), 10 min. 34 sec. (system E)).

In the analysis of *number of actions*, a main effect for *session* was just not significant ($F(2,27) = 3.12$, $p = 0.06$), whereas the univariate test revealed a significant main effect for *session* ($F2,56) = 4.52$, $p < 0.05$). When the means were compared, it was revealed that there was a significant difference between the first and the last two sessions ($F(1,28) = 5.07$, $p < 0.05$); participants performed fewer actions in the last two sessions than in the first session (mean *number of actions* over three sessions: 135, 123, 115). No other effects were found.

Hypothesis (iii)

According to Hypothesis (iii), a music recommender system should lead to the execution of a smaller proportion of navigational actions. Therefore, a MANOVA with repeated measures was conducted in which logit-transformed *proportion of navigational actions* was a dependent variable. Only a significant main effect for *system* was found ($F(4,25) = 12.99$, $p < 0.001$). When the means were compared, it was found that most variation was due to a significant difference between system A and the other four systems ($F(1,28) = 47.32$, $p < 0.001$); participants performed a higher proportion of navigational actions when there was no recommender system available (mean *proportion of navigational actions*: 0.71 (system A), 0.39 (system B), 0.38 (system C), 0.46 (system D), and 0.37 (system E)).

Hypothesis (iv)

According to Hypothesis (iv), a music recommender system should cause music listeners to listen to fewer *distinct* music options, to listen to more music more than once, and to more music options in general. Effects are expected to be bigger when music recommendations are not under direct control of the user. Four MANOVAs with repeated measures were performed. The dependent variables in these analyses were respectively: *number of options listened to, number of distinct options listened to, proportion of recommendations listened to* and *proportion of options listened to more than once.* The results on *number of options listened to* and *number of distinct options listened to* are shown in Figure 6.4.

In the analysis of *number of options listened to*, a significant main effect for *system* was found ($F(4,25) = 7.81$, $p < 0.001$). When the means were compared, it was found that variation was due to a significant difference between system A and system C ($F(1,28) = 26.48$, $p < 0.001$), and a significant difference between system A and system E ($F(1,28) = 9.74$, $p < 0.005$). As shown in the left-hand panel of Figure 6.4, participants listened to more music options when music recommendations were provided that the user could not control (mean *number of options listened to*: 52.6 (system A), 56.4 (system B), 90.0 (system C), 59.4 (system D), and 66.0 (system E)).

In the analysis of *number of distinct options listened to*, no effects were found. As shown in the right-hand panel of Figure 6.4, the number of distinct options listened to did not vary much under various conditions (mean: 48.5). On average, participants listened to five distinct music options for each option included in the programme.

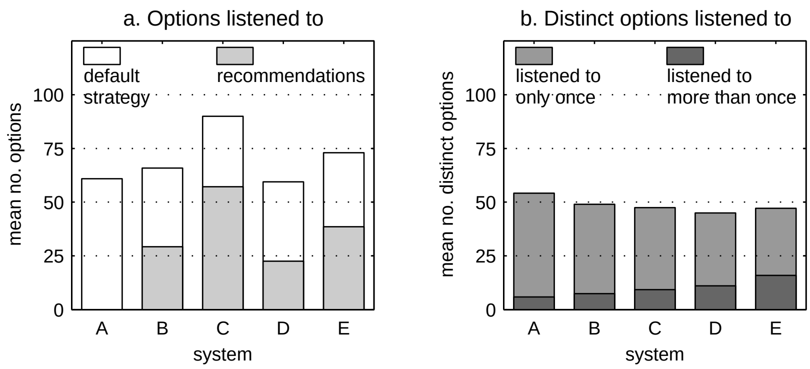

Figure 6.4. The left-hand panel (a) shows the mean *number of options listened to* across programming systems. A distinction has been made between music that was listened to while conducting a default music choice strategy and music that was recommended. The right-hand panel (b) shows the mean *number of distinct options listened to* across programming systems. A distinction has been made between music that was listened to only once and music that was listened to more than once.

As shown in the left-hand panel of Figure 6.4, more than half of the listened to music was recommended, if a music recommender system was used. A MANOVA with repeated measures was performed with *session*, *recommendation* (random, PATS) and *locus of control* (user control, system control) as within-subject independent variables and logit-transformed *proportion of recommendations listened to* as a dependent variable. A significant main effect for *locus of control* was only found ($F(1,28) = 12.23$, $p < 0.005$). Participants listened to a higher proportion of music recommendations if they could not control them (mean *proportion of recommendations listened to*: 0.48 (direct control: system B and D), and 0.61 (system control: system C and E)).

As several music options were listened to more than once, a MANOVA with repeated measures was performed with logit-transformed *proportion of options listened to more than once* as a dependent variable. A significant main effect for *system* was found ($F(4,25) = 10.61$, $p < 0.001$). When the means were compared, it was found that variation was largely due to a significant difference between system A and the other systems ($F(1,28) = 13.54$, $p < 0.005$), a significant difference between system B and D on the one hand and systems C and E on the other hand ($F(1,28) = 7.58$, $p < 0.05$), and a significant difference between systems B and C on the one hand and systems D and E on the other hand ($F(1,28) = 21.16$, $p < 0.001$). As shown in the right-hand panel of Figure 6.4, participants repeatedly listened to the largest number of music options when using a PATS system; this effect was even more pronounced when recommendations were provided that the user could not control. As options repeatedly occurred in the PATS recommendations, some participants commented that it gave the impression of a cyclic operation (mean *proportion of options listened to more than once*: 0.08 (system A), 0.14 (system B), 0.21 (system C), 0.28 (system D), and 0.41 (system E)). An interaction effect for *system* by *session* was nearly significant ($F(8,21) = 2.21$, $p = 0.08$), whereas the univariate test did find a significant interaction effect ($F(8,224) = 2.61$, $p < 0.001$). Investigating this interaction effect revealed a significantly increasing difference between system A and the other systems for the first session, compared with the last two sessions ($F(1,28) = 6.86$, $p < 0.05$). Participants repeatedly listened to increasingly more options over sessions if a music recommender system was used. If no recommender system was used, they repeatedly listened to increasingly fewer options over sessions.

Hypothesis (v)

According to Hypothesis (v), listening to more options should result in less listening time per option and a less selective music listening behaviour. Effects are expected to be bigger when music recommendations are not under direct control of the user. Two MANOVAs with repeated measures were performed. In the first analysis, *average listening time* was a dependent variable; in the second, *standard deviation of listening time* was a dependent variable.

In the analysis of *average listening time*, a significant main effect for *system* was found ($F(4,25) = 3.69$, $p < 0.05$). When the means were compared, it was found that variation was due to a significant difference between system A and the other systems ($F(1,28) = 8.92$, $p < 0.01$), a significant difference between systems B and D on the one hand and systems C and E on the other hand ($F(1,28) = 7.05$, $p < 0.05$), and a significant difference between systems C and E ($F(1,28) = 5.13$, $p < 0.05$). Participants spent most time listening to each music option when using system A. They spent a moderate amount of time listening to each option when using user-controlled recommendations, that is, systems B and D. They spent the least amount of listening time when using system-controlled recommendations, that is, systems C and E (mean *average listening time*: 15.1 sec. (system A), 11.9 sec. (system B), 7.2 sec. (system C), 11.4 sec. (system D), and 9.4 sec. (system E)). In summary, the use of a music recommender system decreased average listening time per music option; if, in addition, recommendations were provided in a way that the user could not control, average listening time per option decreased even further.

In the analysis of *standard deviation of listening time*, a main effect for *system* was just not significant ($F(4,26) = 9.84$, $p = 0.072$), whereas its univariate test did find a significant effect ($F(4,112) = 5.17$, $p < 0.005$). When the means were compared, it was found that most variation was due to a significant difference between system A and the other systems ($F(1,28) = 9.79$, $p < 0.005$). Some variation was also explained by a difference between system B and D on the one hand and systems C and E on the other hand ($F(1,18) = 3.51$, $p = 0.071$). Participants exhibited a more selective music listening behaviour when using system A than when using the other systems. Use of a recommender system resulted in a more unevenly spread listening time. Even less selective listening behaviour was observed when recommendations were not under direct control of the user (mean *standard deviation of listening time*: 12.2 sec. (system A), 10.0 sec. (system B), 8.2 sec. (system C), 10.3 sec. (system D), and 8.6 sec. (system E)).

Hypothesis (vi)

According to Hypothesis (vi), fewer randomly drawn recommendations than PATS recommendations should be added to a music programme. Results on *proportion of recommendations included* are shown in Figure 6.5.

A MANOVA with repeated measures was performed with *session*, *recommendation* (random, PATS), and *locus of control* (user control, system control) as within-subject independent variables and p*roportion of recommendations included* as a dependent variable. A significant main effect for *recommendation* was found (F(1,28) = 10.79, p < 0.005). The left-hand panel of Figure 6.5 shows that participants added (proportionally) more PATS recommendations than randomly drawn recommendations to their programmes (mean *proportion of recommendations included*: 0.36 (random), 0.47 (PATS)). Also, an interaction effect for *recommendation* by *locus of control* was found (F(1,28) = 4.28, p < 0.05). The right-hand panel of Figure 6.5 shows that participants included (proportionally) more PATS recommendations than randomly drawn recommendations in their programmes when these recommendations were provided under system control. No other effects were found.

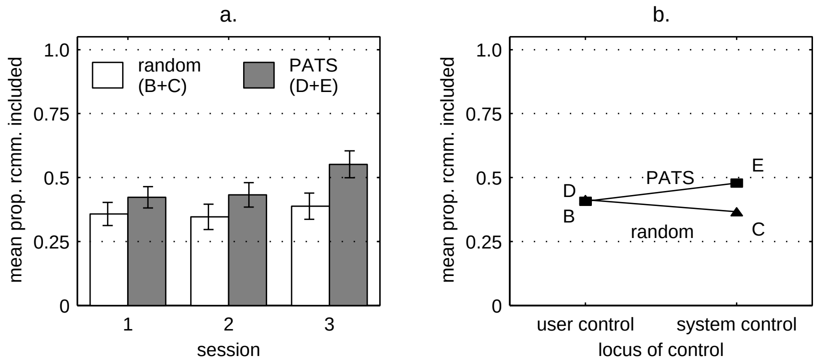

Figure 6.5. The left-hand panel (a) shows the mean *proportion of recommendations included* in a music programme for random and PATS systems across *session*s. Crossbars represent the standard error of the mean. The right-hand panel (b) shows the mean *proportion of recommendations included* for both random and PATS systems with different *loci of control*. Note that a music programme contained 10 music options.

Perceived ease of use and perceived usefulness

According to Hypotheses (vii) and (viii), the usability of a programming system with a recommender should be valued lower than a system without a recommender, while the usefulness of a programming system with a PATS recommender should be valued higher than other programming systems. The TAM questionnaire was designed to address both hypotheses.

Responses to the TAM questionnaire were subjected to a two-dimensional non-linear Principal Component Analysis (PCA) (van de Geer, 1993a; 1993b). In the analysis procedure used (GIFI, 1985), the eight statements in the questionnaire were treated as active variables, and the five programming systems were treated as passive variables to label the PCA plot. The responses to the statements were treated as ordinal categories; only the order on the 7-point scale was considered important. Consequently, the data matrix consisted of rows with responses to the eight statements per participant, per session and per interactive system. Responses collected across sessions were thus treated as new independent rows in the matrix; in that way, longitudinal data were analysed by a cross-sectional technique. This technique has a side effect that the first principal component represents the response changes over the sessions. The mean transformed responses of the questionnaire related to the five programming systems over the three sessions, are displayed in the centre of Figure 6.6. The arrows connecting the mean scores represent the course over sessions.

The scores for statements Q1, Q2, Q3 and Q4 are highly correlated as well as the scores for statements Q5, Q6, Q7 and Q8. The high correlations reinforces the fact that the first and second set of statements indeed measure *perceived ease-of-use* and *perceived usefulness* respectively (Davis,1989). Thus, the upper right-hand corner of the first quadrant represents high *perceived usefulness*, and the lower right-hand corner of the fourth quadrant represents high *perceived ease of use*. As shown in Figure 6.6, it appeared that systems B and C were perceived as least easy to use, whereas the systems A and E were perceived as most easy to use. The PATS programming systems, that is, D and E, were perceived as increasingly more useful over sessions. The PATS programming system E was perceived as most useful among all other programming systems.

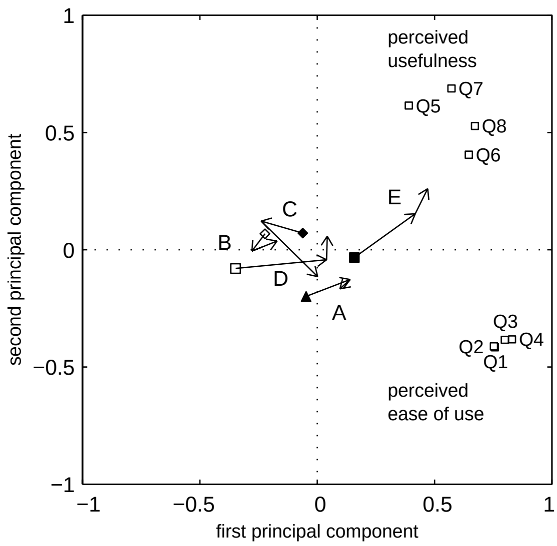

Figure 6.6. The PCA solution of the eight statements of the TAM questionnaire loaded on the terms *perceived ease-of-use* and *perceived usefulness*.

Order of preference

According to Hypothesis (ix), a programming system with a PATS recommender should be preferred when compared with other programming systems. To assess this hypothesis, participants were asked to rank three interactive systems they had worked with; rank value 1 was assigned to the most preferred system for music programming.

A Durbin rank test for incomplete block design (Conover, 1971) was conducted for each session separately. All three Durbin test statistics (respectively 12.64, 11.17 and 15.77) exceeded the 0.95 quantile of a χ^2 random variable with 4 degrees of freedom (9.49), from which it is inferred that, at each session, at least one programming system was preferred over others.

The ranking data can also be used to indicate comparative judgements of all pairs of systems (Guilford, 1954). The task of ranking three systems requires, in essence, the comparison of all systems involved, which enforces transitivity of rankings. As only three systems were ranked, transitivity may not hold when considering all five systems. A ranking of three systems then amounts to three pair-comparison judgements. In our case, a system with the lowest rank value is judged more preferred than all systems that have a higher rank value. Ties in the ranking were treated as equal preference of the systems involved. From repeated judgements in all three sessions and over all participants, we can determine the proportion of the time that each system is more preferred than every other system (see Table 6.1). Only half of the matrix is shown because the two halves are complementary and no data exists for the diagonal cells.

Table 6.1. Proportion matrix showing the proportion of the time that a system at the top is more preferred than a system at the side in 27 preference judgements.

	A	**B**	**C**	**D**
B	7 / 27			
C	5 / 27	12 / 27		
D	4 / 27	11.5 / 27	13 / 27	
E	8 / 27	7.5 / 27	8 / 27	11 / 27

The standard way to analyse pair-comparison data is based on Thurstone's *law of comparative judgment* (Thurstone, 1927; Torgerson, 1958). When applying this law to our case, the extent to which one programming system is judged to be more preferred than another is related to the difference in subjective strengths, or scale values, of the compared systems with respect to some variable of interest; in this case, music programming. As the result of a judgement process may vary slightly from time to time and between participants, it is assumed here that scale values, and likewise their differences, follow a normal distribution with standard deviations equal to 1. The means of these normal distributions correspond to the scale values. The method to estimate a scale value for each system comes down to transforming the proportions in each cell in Table 6.1 into normal deviates (z-scores). Each normal deviate then represents the difference between the scale values of two systems. Collecting all ten possible differences between five scale values of the systems results in an overdetermined set of equations.

Table 6.2. Scale value estimates and their standard deviation of the programming systems. Scale value of system A is set at zero, by definition.

	Scale value estimate	**Standard error**
A	0.00	0.16
B	0.57	0.16
C	0.72	0.16
D	0.83	0.16
E	1.00	0.16

By setting the scale value of system A to zero, the least-squares solution of the overdetermined set of equations yields the scale value estimates as shown in Table 6.2. The correlation between the observed data and the predictions of the least-squares solution is high (r = 0.886) which means that 78.6% of the variance is explained. An internal consistency check proposed by Mosteller (1951) found that discrepancies between the proportions that would be expected from the estimated scale values and the observed proportions is primarily due to sampling error ($\chi^2 = 5.74$, df = 6, non-significant). Table 6.2 shows that all programming systems with music recommendations were more preferred than system A, as their scale value estimates are all positive. The scale value estimates and the low values of standard error also indicate that all five programming systems can be ranked consistently in the order A, B, C, D and E, according to increasing preference of use for music programming. Both PATS systems, that is, D and E, were the most preferred systems for music programming.

Task descriptions

Task descriptions elicited from participants who used a music recommender system while programming are summarised in a generic task model as shown in Figure 6.7. In total, 72 task descriptions were collected, though many descriptions referred to descriptions that had been written earlier, when participants found that there was no change in how they had pursued the task. The graphical notation in Figure 6.7 is adopted from Structured Diagram Notation (Lim and Long, 1994). Tasks and sub-tasks are represented by boxes and are hierarchically ordered; a superordinate task consists of a strict sequence of subordinate tasks, or a selection over subordinate tasks (indicated by '0'), or an iteration of subordinate tasks (indicated by '*').

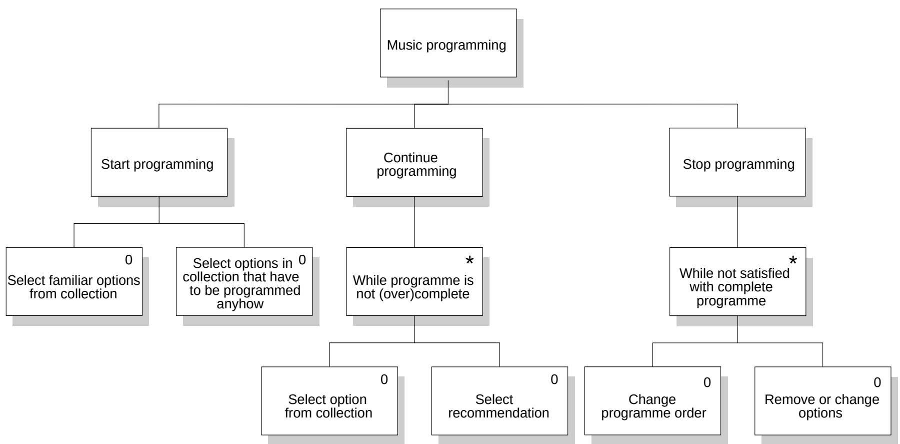

Figure 6.7. Generic task model summarising task descriptions.

From the task descriptions, it appeared that participants generally pursued the task in three phases: 'start programming', 'continue programming' and 'stop programming' in Figure 6.7. In the first phase, they started a music programming task by finding one or more music options that were familiar to them by title or artist, or that they anyway wanted in their programme. Then, in the second phase, as long as the required size of the programme was not attained, participants selected options alternately from the music collection and the recommendation list. One strategy in this phase consisted of collecting more than the required number of options in the music programme; the surplus of compiled options was then removed in a final third phase.

The way in which participants alternate their search in the second phase using the collection and recommendation list differed. In general, it was found that participants least frequently searched on music titles, moderately on music artists and most often on music styles. Participants could base their selection entirely on a search in the collection and shifted to the recommendations only if they got stuck. Participants could also be guided entirely by the recommendations, mainly out of curiosity. Another way was to select an option from the collection and incidentally inspect the recommendations that came along with the selected option. If suitable, the recommendations were also added to the programme. Besides suggestions to be included in the programme, recommendations also acted to provoke new ideas to search in the music collection. In general, if recommendations turned out to be inappropriate, the search tended to be focused on the collection.

In the third concluding phase, the programme was fine-tuned by changing the order of the options in a programme, or by removing a surplus of options from the programme, or by exchanging less preferred options with others by doing a second pass of the listening.

Comments of participants

In general, participants were enthusiastic about the direct and electronic access to a music collection as carried out in the experiments; they were keen on the fast music access times while sitting in a comfortable chair and controlling only a standard mouse. Only one participant, who was a disk-jockey in his spare time, preferred a personally arranged physical organisation of music that could be easily taken along in a suitcase to allow mobility and full control of the music selection process.

Comments about undesirable system properties generally concerned the interaction with music recommendations and the effort required in using the interactive system. It appeared that participants liked to spend the least possible effort in terms of time and action for collecting music and for obtaining recommendations. Of the thirty participants, seven (23%) commented that there were, in general, too many mouse movements and actions involved for collecting music in a programme. Twelve (40% of 30) participants commented that they preferred an automatic appearance of recommendations in which the recommendations were directly at hand; eight of them disliked the ‘recommendation’ button because it took additional control actions. The three other participants, on the other hand, found a button a more appropriate means to survey the music collection. However, immediate play back of recommendations was disliked by ten (33% of 30) participants. Automatic playback was found to be disruptive to the programming task, especially when recommendations were considered inappropriate. Four (13% of 30) participants commented that the visual display was hardly legible from a large viewing distance and after extensive use; too much textual information with too small a letter size was displayed on the monitor. Four (13% of 30) participants commented that they lacked the information or graphical design found in CD booklets. One participant commented that unintended actions such as rejecting a wrong option could not be easily undone; he also suggested implementing a means to explore all recommendations that were made during the whole course of the task.

6.2.5 Discussion

Participants were given sufficient time to complete a music programming task. The conservative procedure to cope with missing data in the analysis decreased the effect size of existing effects between conditions, though there was no indication that possible effects between conditions had been missed. With ample programming time, it was assumed that participants are able to collect preferred music, irrespective of the experimental condition. The results showed that participants made equally and highly preferred music programmes in various experimental conditions.

According to Hypothesis (i), the use of music recommendations should reduce the time spent on music programming. The results showed however that differences on task performance between the programming systems were, in general, too small to produce significant effects. On the basis of these results, Hypothesis (i) must be rejected; the use of a music recommender system did not speed up music programming and, in addition, did not reduce the number of actions performed.

According to Hypothesis (ii), less time and fewer actions for music programming should be required for successive sessions. The results showed that indeed less time and fewer actions were required with practice. On the basis of these results, Hypothesis (ii) cannot be rejected.

According to Hypothesis (iii), using music recommendations should cause music listeners to perform a lower proportion of navigational actions. The results showed that participants performed a lower proportion of navigational actions when music recommendations were available. On the basis of these results, Hypothesis (iii) cannot be rejected; the use of a music recommendations led to less effort for navigation and search in the music collection.

According to Hypothesis (iv), the use of music recommendations should cause music listeners to listen to fewer *distinct* music options, to more options more than once, and to more options in general. The results showed that more music options were listened to, only when music recommendations were provided in a way that the user could not control. Of the total amount of music listened to, 61% were recommendations. In addition, the results showed that, especially when using PATS music recommendations, more options that were already familiar to the participant were listened to. If these recommendations were also provided beyond user control, even more familiar music was listened to. On the basis of these results, Hypothesis (iv) cannot be rejected. A music recommender system, especially the PATS systems, frequently made participants listen to music they had already heard. This gave some participants the impression of cyclic operation.

According to Hypothesis (v), the use of music recommendations should cause music listeners to devote less listening time per music option and to listen to music less selectively. Effects were expected to be bigger when recommendations were provided beyond user control. Indeed, the results showed that participants spent most listening time per option and exhibited a relatively high selective listening behaviour when no music recommender system was used. They spent a moderate amount of listening time per option and exhibited a less selective listening behaviour when using user-controlled recommendations. And they spent the least amount of listening time per option and exhibited an even less selective listening behaviour when using system-controlled recommendations. On the basis of these results, Hypothesis (v) cannot be rejected.

According to Hypothesis (vi), fewer randomly drawn recommendations should be included to the music programme than PATS recommendations. The results showed that more PATS recommendations were added to the programme, especially if these recommendations were provided beyond user control. On the basis of these results, Hypothesis (vi) cannot be rejected.

According to Hypothesis (vii), the usability of a programming system with a recommender should be valued lower than a system without a recommender. The results of the TAM questionnaire showed that a programming system with a random recommender was perceived as being least usable, while a programming system without a recommender and with a system-controlled PATS recommender were perceived as being most usable. On the basis of these results, Hypothesis (vii) cannot be rejected.

According to Hypothesis (viii), the usefulness of a PATS recommender while programming should be valued higher than all other programming systems. The results of the TAM questionnaire showed that PATS recommender systems were perceived as increasingly more useful. The system-controlled PATS recommender was perceived as being most useful among all other systems. On the basis of these results, Hypothesis (vii) cannot be rejected.

According to Hypothesis (ix), a programming system with a PATS recommender should be more preferred than other systems. The results of the system ranking test showed that both PATS systems were the most preferred systems for music programming. On the basis of these results, Hypothesis (ix) cannot be rejected.

As an additional observation, it was apparent that participants listened to almost the same number of distinct music options each time they programmed music, irrespective of the programming system used. On average, they listened to five distinct music options for each option included in the programme. It seems that participants build a personal choice context, consisting of a fixed number of distinct music options from which they can determine their music preference.

Another additional observation was that, when participants had control over the recommendations, a recommender system provided almost half of the total amount of music listened to. In this case, the total amount of music listened to almost equalled the amount of music listened to when programming without the assistance of a recommender system. Hence, a user-controlled recommender system provided music that substituted a part of the music that was otherwise found in a personal search (the *default* choice strategy). When, on the other hand, recommendations were provided beyond user control, a recommender system added (thus, did not substitute) almost the same amount of music listened to that was otherwise listened to in a personal search.

Some undesirable properties of a music recommender system were also observed. It can be said from the results of the TAM questionnaire that a programming system with a recommender system was, in general, found to be less easy to use than a programming system without a recommender. This was probably due to the need to perform additional control actions, to divide attention between three instead of two information displays, but more importantly to a perceived loss of control of music listening and selection. Especially a PATS system providing recommendations beyond user control frequently made participants listen to already familiar music. Listening to more music made participants listen for a shorter time and less selectively to individual options. In contrast, while using the system without recommendations, participants could decide, at will, what amount of listening time was devoted to what music. Note that all recommender systems immediately selected and played back recommended music which was intended to speed up operation, to emphasise external system control and, thus, to give less control to the user. If participants found the music inappropriate, they could only react by switching off or by selecting another option. Both immediate playback of recommendations and having to listen to familiar music turned out to be undesirable properties of a recommender system, because they take away the initiative of the user. It is therefore desirable to make a more effective control mechanism to reduce the amount of listening to music that is already familiar to the listener.

In summary, a PATS music recommender system was found to be a preferred feature for programming music. This appreciation for the PATS system could be based on several measures: more PATS recommendations were included in a music programme than randomly drawn ones, the PATS system was perceived as most useful when compared to other systems, and the preference ranking of all systems indicated a preference for the PATS system. In addition, the use of a random or PATS music recommender reduced the need to personally search for preferred music. Negative affects associated with having to listen to more of the same music did not negatively influence a final preference judgement. On the contrary, the PATS system which provided recommendations beyond user control and made participants listen to familiar music most frequently was indicated as the most preferred system for music programming.

6.3 Experiment II: Programming under a time constraint

Experiment II assessed task performance and music listening behaviour involved with a time-constrained music programming task, with and without the assistance of a music recommender system. Pilot experiments with two participants ensured that the time constraint (6 minutes for completing a preferred music programme) was moderate, only to enforce rapid decision making without hindering a successful completion of the task. This arbitrary time constraint was set by halving the mean *time on task* as observed in Experiment I (11 min. 50 sec.).

6.3.1 Hypotheses

By giving music listeners limited time, it is assumed that it will be more difficult to attain satisfactory results by using only a *default* choice strategy. Simultaneously coping with time constraints and programming preferred music is assumed to be difficult (for simultaneously dealing with multiple goals in a choice task, see Payne, Bettman and Johnson, 1993; Payne, Bettman and Luce, 1996 and Chapter 2). Consequently, the quality of a music programme is assumed to decrease in a time-constrained programming task. Mechanisms to cope with a restricted time frame include accelerated processing and ‘conservative’ choice, that is, executing more actions per time unit and repeating music choices from earlier tasks. It is expected that a music recommender system will cause music listeners to perform fewer actions and to make less conservative and more preferred music choices. It is therefore hypothesised that:

(i) Music programmes compiled with the assistance of a music recommender system under a time constraint contain more preferred music options than when there is no recommender system available.
(ii) Successive music programmes compiled with the assistance of a music recommender system under a time constraint contain more distinct preferred music options than when there is no recommender system available.
(iii) A music recommender system causes music listeners to perform fewer actions to complete a time-constrained music programming task.

Time constraints are assumed to have effects on the performance of a *default* choice strategy: less time is available to listen to each music option, which makes music listeners less selective in their listening behaviour. It is expected that using a recommender system reinforces these effects, and thus reproduce some of the effects as observed in Experiment I. It is therefore hypothesised that:

(iv) A music recommender system causes music listeners to perform a smaller proportion of the navigational actions, to listen to more options more than once, to devote less listening time to each option, and to listen less selectively to music in a time-constrained music programming task.

6.3.2 Measures

Task performance measures

Task performance measures were similar to those used in Experiment I (see Section 6.2.2). The quality of a music programme was measured by *precision*, that is, the proportion of preferred music options in a music programme containing 10 music options. The proportion of recommendations that was added to a music programme was measured by the *proportion of recommendations included*. An additional measure, *coverage*, was defined as the number of distinct and preferred music options that appeared in successive music programmes (see also Chapter 5). *Coverage* is essentially a summation of all preferred options in each new music programme that were not already contained in previous programmes.

Number of actions, *proportion of navigational actions*, *number of options listened to*, *number of distinct options listened to*, *proportion of recommendations listened to*, *average listening time*, *standard deviation of listening time*, and *proportion of options listened to more than once* were measured in a similar way as in Experiment I.

6.3.3 Method

Design

The experimental design differed from the one used in Experiment I. A design with two within-subject independent variables was used. The design included a within-subject independent variable *system*, similar to the one used in Experiment I, defined as: 1) *none*, no recommendations, 2) *random*, randomly drawn recommendations, and 3) *PATS*, algorithmically determined recommendations. Each level was represented by an interactive system. Level *none* referred to interactive system A which was identical to the one used in Experiment I. The levels *random* and *PATS* were represented by systems F and G; system F was a combination of the systems B and C in Experiment I, system G was a combination of the systems D and E. The other independent variable was *task repetition*; participants performed three consecutive tasks in an experimental session. Task repetitions were intended to measure changes in task performance caused by practice.

To assist the participant in decision making, another variable *context-of-use* was defined as: 1) *soft music*, a context-of-use with soft music, 2) *lively music*, a context-of-use for loud music, and 3) *free*, music for a personally defined context-of-use. As this variable was intended for instructional purposes only, it was not factorially combined with the other two independent variables. A factorial combination would anyway make the experimental design too complex. As shown in Figure 6.8, nine participants were randomly assigned to one of nine blocks consisting of three experimental sessions in which, in each session, three *task repetition*s were represented in a combination of *system* (3) and *context-of-use* (3). The variable *context-of-use* was not considered in the analysis. Note that by using this experimental design, each system was tested against any other system by nine different participants, which was also the case in Experiment I.

Block	*Context-of-use*			Block	*Context-of-use*			Block	*Context-of-use*		
	soft music	lively music	free		lively music	free	soft music		free	soft music	lively music
(1)	A A A	F F F	G G G	(4)	A A A	F F F	G G G	(7)	A A A	F F F	G G G
(2)	F F F	G G G	A A A	(5)	F F F	G G G	A A A	(8)	F F F	G G G	A A A
(3)	G G G	A A A	F F F	(6)	G G G	A A A	F F F	(9)	G G G	A A A	F F F

Figure 6.8. Experimental design of Experiment II.

Test equipment and material

The test equipment and material were identical to those used in Experiment I (see Section 6.2.3).

Interactive system

The interactive systems resembled those used in Experiment I (see Section 6.2.3) but were adapted to meet the new experimental requirements. The left-hand and middle section of the control panel, containing the music collection list and the music programme list respectively, were identical to those used in Experiment I (see Figure 6.2). The right-hand section, if present, was modified such that participants could choose, at will, whether music recommendations were controlled by themselves or by the system. One additional control element was added with which participants could indicate the preferred mode; the default mode was system control and participants could repeatedly switch between modes. The recommendation button, as shown in Figure 6.2, was still present. In addition, a ‘digital’ clock was placed at the lower left-hand corner of the control panel, which counted down the 6 minutes of a programming task. The last 10 seconds were marked by audible beeps played every second. When participants initiated the task by pressing the start button, the clock started counting. When the time expired, the system was made inaccessible and results were saved.

All actions that were performed by participants were logged in the same way as in Experiment I.

Procedure

Nine participants returned for four experimental sessions on four separate days. As in Experiment I, the last three sessions were used to acquire experimental data; these are referred to as the experimental sessions. The first session was intended to acquaint participants with the music collection and to assess the personal musical taste of participants following the same procedure as in Experiment I.

At the outset of each experimental session, participants received brief instructions on use of the interactive system they were supposed to work with during that session. The instructions included a demonstration by the experimenter and a practice round by the participant, if desired. Participants were then asked to verbalise a concrete situation of the context-of-use for which they were instructed to compile a music programme (soft music, lively music, free). Some examples for soft music were 'music for a dinner party' or 'music for when parents come to visit'. Subsequently, participants completed three music programming tasks using the given interactive system. They were allowed to pause between successive tasks. In all three tasks in a session, the same imagined context-of-use was employed. A music programme had to contain *10 distinct preferred music options*; besides the distinctness of music options within a given music programme, no requirements were stated about music options across music programmes. Further instructions were identical to the procedure in Experiment I. Time to perform the task was presented as being limited to 6 minutes. The instruction mentioned both the quality of the music programme and task completion within the given time frame as optimisation criteria.

At the end of each session, participants were presented with the music programmes they had compiled. They were asked to judge what options in the programme did not fit the given context-of-use while listening to them for the second time. Unlike Experiment I, no questionnaires were handed out.

Participants

Nine participants (7 males, 2 females) were recruited by advertisements and all got a fixed fee. All participants were frequent listeners to jazz music; for admission to the experiment, participants had to be able to rank eight freely recalled jazz musicians on personal taste and number of recordings owned (Geringer and McManus, 1979). The average age of the subjects was 29 (min: 23, max: 42). None of the participants had received musical training or played a musical instrument. All participants had at least completed a higher vocational education.

6.3.4 Results

In the familiarisation session, participants selected and listened to more than half of the options (mean *number of options listened to*: 143.9, min: 58, max: 278).

Of the 81 performed tasks in the three experimental sessions, four were not completed. Three programmes lacked one option, and one programme lacked four options. These missing music options in the programmes were defined as rejected options in the *precision* and *coverage* measures. For other measures, it was immaterial whether there were missing options or not.

Of the 81 tasks, 57 (70%) were completed before the interactive system warned the participant about the near time expiration, that is, 57 tasks were already completed

10 seconds before the given 6 minutes expired. Participants needed, on average, 315 seconds (5 min 15 secs) to compile a music programme.

Proportional data with different denominators were logit-transformed using Equation 6.1. All reported analyses of variance (MANOVAs) with repeated measures were conducted with *system* and *task repetition* as within-subject independent variables, but with different dependent variables.

Hypotheses (i) and (ii)

According to Hypotheses (i) and (ii), music programmes created with the assistance of a music recommender system should contain more preferred and more distinct music options. Therefore, two MANOVAs with repeated measures were conducted. In the first analysis, *precision* was used as a dependent variable; in the second, *coverage* was used.

In the analysis of *precision*, no significant effects were found. Though differences in mean *precision* across systems were observed in the expected directions, these differences were not significant (mean *precision* across systems: 0.84 (none, system A), 0.86 (random, system F), 0.90 (PATS, system G)).

In the analysis of *coverage*, a significant main effect was only found for *task repetition* ($F(2,7) = 268.76$, $p < 0.001$). A linear course over successive task repetitions was significant ($F(1,8) = 466.67$, $p < 0.001$). Participants compiled an equal amount of preferred music that was not present in previous programmes, irrespective of the programming system used (mean *coverage* across task repetitions: 8.6 (first), 16.6 (second), 23.9 (third)).

When using a music recommender system, almost half of the options included in the programmes were recommended. In a MANOVA with repeated measures, with *proportion of recommendations included* as a dependent variable, no effects were found between the two music recommender *systems* (F and G) or *task repetitions*. Participants included almost a fixed number of music recommendations in their programme (mean *proportion of recommendations included*: 0.47).

Hypothesis (iii)

According to Hypothesis (iii), a music recommender system should cause music listeners to perform fewer actions. The results on *number of actions* are shown in the left-hand panel of Figure 6.9. A MANOVA with repeated measures was conducted with *number of actions* as a dependent variable. No significant effects were found. Though the system with no recommendations received generally the highest number of actions, as shown in the left-hand panel of Figure 6.9, the differences with the other systems were too small to produce significant effects (mean *number of actions* across systems: 96.3 (none, system A), 79.2 (random, system F), 87.1 (PATS, system G)).

Hypothesis (iv)

Hypothesis (iv) summarises the effects as found in Experiment I. First, a music recommender system should cause music listeners to perform a lower proportion of navigational actions. The results on the *proportion of navigational actions* are shown in the right-hand panel of Figure 6.9.

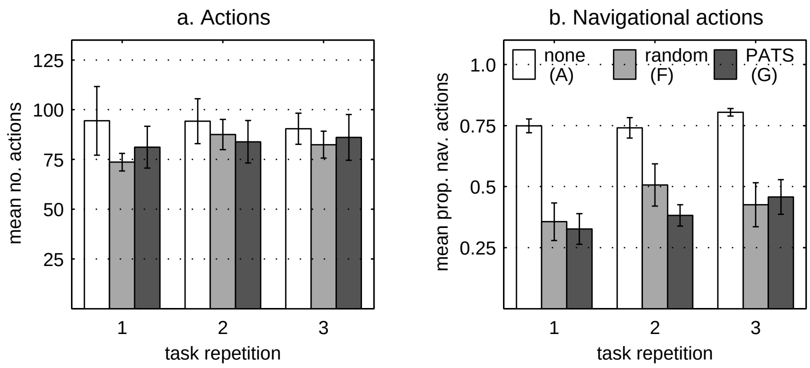

Figure 6.9. The left-hand panel (a) shows the mean *number of actions* across systems and over task repetitions. The right-hand panel (b) shows the mean *proportion of navigational actions* across systems and over task repetitions. Cross-bars represent the standard error of the mean.

A MANOVA with repeated measures was conducted with logit-transformed *proportion of navigational actions* as a dependent variable. A significant main effect for *system* was found ($F(2,7) = 11.74$, $p < 0.01$). When means were compared, it was found that most variation was due to a significant difference between system A and the other systems ($F(1,8) = 21.49$, $p < 0.005$). Participants performed a lower proportion of navigational actions when using a music recommender system (mean *proportion of navigational actions* across systems: 0.77 (none, system A), 0.43 (random, system F), 0.38 (PATS, system G)). A main effect for *task repetition* was just not significant ($F(2,7) = 4.50$, $p = 0.55$), whereas the univariate test was ($F(2.16) = 6.04$, $p < 0.05$). Participants performed a higher proportion of navigational actions over successive tasks; this increasing trend was linear over task repetitions ($F(1,8) = 10.09$, $p < 0.05$). A *system* by *task repetition* interaction effect was just not significant ($F(4,5) = 4.26$, $p = 0.072$), whereas the univariate test did find a significant interaction effect ($F(4,32) = 2.99$, $p < 0.05$). When means were compared, it was found that most variation was due to a cross-over effect for systems F and G at the second and third task ($F(1,8) = 6.55$, $p < 0.05$) (see the right-hand panel of Figure 6.9).

Further, according to Hypothesis (iv), music listeners should listen to more music options, but fewer *distinct* music options when using music recommendations. Listening to more music is due to recommended music that participants had already heard. Results on the *number of options listened to*, *number of distinct options listened to*, and *proportion of options listened to more than once* are shown in Figure 6.10. Four MANOVAs with repeated measures were performed to successively assess parts of Hypothesis (iv). The dependent variables in the four analyses were respectively: *number of options listened to*, *number of distinct options listened to*, logit-transformed *proportion of recommendations listened to* and logit-transformed *proportion of options listened to more than once*.

In the analysis of *number of options listened to*, a main effect for *system* was just not significant ($F(2,7) = 4.28$, $p = 0.061$), whereas the univariate test did find a significant main effect for *system* ($F(2,16) = 6.13$, $p < 0.05$). When means were compared, it was found that variation was due to a difference between the *number of options listened to* when using system A and the *number of options listened to* when using system F or G

(F(1,8) = 9.60, $p < 0.05$). As shown in the left-hand panel of Figure 6.10, participants listened to *more* music options when using a music recommender system (mean *number of options listened to* across systems: 42.4 (none, system A), 62.4 (random, system F), 57.3 (PATS, system G)). No other effects were found.

In the analysis of *number of distinct options listened to*, no effects were found. Participants listened to almost the same number of distinct options when creating a programme in various conditions (mean *number of distinct options listened to*: 39.2). On average, participants listened to four distinct options for each option included in the programme.

When using a music recommender system, 65% of the music listened to was recommended. In the analysis of *proportion of recommendations listened to*, no effects were found as a result of *system* F and G or *task repetition*. Participants listened to an almost equal number of recommendations when using system F or G. Note that most of the time, participants had switched the mode on in which recommendations were provided beyond their control. As shown in the left-hand panel of Figure 6.10, the number of music recommendations approximately equalled the number of music options listened to when using system A.

.

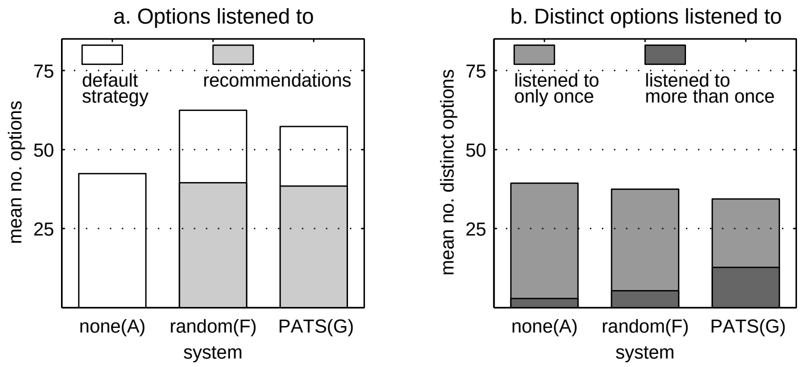

Figure 6.10. The left-hand panel (a) shows the mean *number of options listened to* across programming systems. A distinction has been made between music that was listened to while conducting a default choice strategy and music that was recommended. The left-hand panel (b) shows the mean *number of distinct options listened to* across programming systems. A distinction has been made between music that was listened to only once and music that was listened to more than once.

In the analysis of *proportion of options listened to more than once*, a significant main effect for *system* was found (F(2,7) = 24.07, $p < 0.005$). When means were compared, it was found that all system means differed significantly from each other; system A differed significantly from system F and G (F(1,8) = 25.52, $p < 0.005$) and system F differed significantly from system G (F(1,8) = 33.57, $p < 0.001$). As shown in the right-hand panel of Figure 6.10, participants without a recommender system re-listened least often to music, and re-listened most often to music when using the PATS system (mean *proportion of options listened to more than once* across systems: 0.07 (none, system A), 0.14 (random, system F), 0.41 (PATS, system G)). Also, a significant *system* by *task repetition* interaction effect was found (F(4,5) = 6.17, $p < 0.05$). This interaction effect was due to a proportional increase over task repetitions when using system A or F, and a proportional decrease over task repetitions when using system G (F(1,8) = 9.33, $p < 0.05$).

Finally, according to Hypothesis (iv), music listeners should devote less listening time per music option and should listen to music less selectively when using a music recommender system. Two MANOVAs with repeated measures were conducted. In the first analysis, *average listening time* was dependent variable; in the second, *standard deviation of listening time* was a dependent variable.

In the analysis of *average listening time*, a main effect for *system* was just not significant ($F(2,7) = 3.93$, $p = 0.072$), whereas the univariate test found a significant main effect for *system* ($F(2,16) = 6.71$, $p < 0.01$). When means were compared, it was found that variation was due to a difference between system A and the other two systems ($F(1,8) = 8.51$, $p < 0.05$). Participants listened two-and-a-half seconds longer to music options when using no music recommender system (mean *average listening time* across systems: 7.8 sec. (none, system A), 5.2 sec. (random, system F), 5.4 sec. (PATS, system G)). Also, in the univariate test, a significant interaction effect for *system* by *task repetition* was found ($F(4,32) = 3.70$, $p < 0.05$). This interaction effect is explained by the fact that *average listening time* was higher in the first task than in the other two tasks when using system A, whereas *average listening time* was almost constant for all tasks when using system F or G ($F(1,6) = 5.63$, $p < 0.05$).

In the analysis of *standard deviation of listening time*, no effects were found. Though participants exhibited a less selective music listening behaviour when using the system without recommendations, differences between systems were too small to produce significant effects (mean *standard deviation of listening time* across systems: 6.5 sec. (none, system A), 5.4 sec. (random, system F), 5.3 sec. (PATS, system G)).

6.3.5 Discussion

Participants were given limited time to complete a music programming task. The use of time constraints was intended to prevent participants from getting carried away by the ample selection possibilities and listening opportunities, and thus to speed up operation. Almost all participants completed the tasks successfully. The majority of the tasks were finished 10 seconds before time expiration, and the number of actions was almost the same across successive tasks. Hence, participants were able to cope with the limited time frame from the first task onwards.

It was assumed that a time-constrained music programming task would cause accelerated processing and result in fewer preferred and more ‘conservative’ music choices being made in a default choice strategy. Whether these time constraint effects really occurred or not could not be inferred from the results, as this experiment did not manipulate levels of time constraints (see Section 6.5.1 for some possible time constraint effects in the music programming tasks).

According to Hypothesis (i), time-constrained music programming with the assistance of a music recommender system should result in more preferred music choices than when there is no recommender system available. The results showed that programme quality in terms of *precision* improved slightly when using a recommender system. However, the difference in *precision* between systems was too small to produce significant effects. As the majority of tasks were already completed well before time expiration, it is likely that the available 6 minutes did not significantly affect the ease of programming preferred music. On the basis of these results, Hypothesis (i) must be rejected.

According to Hypothesis (ii), time-constrained music programming with the assistance of a recommender system should result in more distinct music choices in successive programming tasks than when there is no recommender system available. The results showed that participants included an equal amount of preferred music in each programme that was not already contained in previous programmes, irrespective of the programming system used. On the basis of these results, Hypothesis (ii) must be rejected.

According to Hypothesis (iii), using a recommender system in a time-constrained music programming task should cause music listeners to perform fewer actions. The results showed that though the use of a recommender system lowered the number of actions performed, this difference was too small to produce significant effects. On the basis of these results, Hypothesis (iii) must be rejected.

Hypothesis (iv) summarised the effects as found in Experiment I. Most of the effects were also reproduced here. Similar to Experiment I, the results showed that participants performed a lower proportion of navigational actions when using a recommender system. In other words, they used a default choice strategy less frequently for finding preferred music, but used music recommendations instead. Similar to Experiment I, more music was listened to when using a recommender system. Of the total amount of listened to music, 65% was recommended by the system. This resulted in a decrease in listening time per music option. Similar to Experiment I, the PATS system made participants listen to music that was already familiar to them. Under a time constraint, participants already had difficulties in freely dividing their attention for music listening. In contrast to Experiment I, differences in selective music listening were therefore not observed.

As an additional observation, it was found that almost half of the compiled music options were music recommendations. In contrast to Experiment I, an approximately equal number of randomly drawn and PATS recommendations were included in the programmes. The time constraint probably made participants less sensitive to the assumed quality difference between PATS and random recommendations.

In addition, it was apparent that, even in a time-constrained programming task, participants listened to an almost fixed number of distinct options each time they programmed music. On average, they listened to four distinct options for each option included in the programme. It seems that even under a time constraint, participants build a fixed-sized choice context for determining their music preference. Another explanation is that participants just listened to as many options as possible until time expired.

The immediate playback feature of recommendations was also an undesirable system property here, as expected. It made participants listen to too much music that was already familiar to them. It would once again be a good idea, to provide a more effective control mechanism for mitigating this property.

6.4 Conclusion

Two experiments examined the required effort, music listening behaviour and music selection behaviour in programming preferred music with and without a music recommender system, and with and without a time constraint. It was generally expected that music listeners would use music recommendations in their search to preferred music because it reduces effort pertaining to navigation and search, it leads to a more efficient task performance, and it improves the quality of music choices.

A music recommender system reduces navigational effort by almost half because it provides almost a half of the chosen music. Some undesirable properties of the recommender systems made music listeners experience a loss of control of music selection. These properties were immediate playback of recommendations and repeatedly being confronted with the same music. The combination of these properties results in a less selective listening behaviour, which demonstrates that music listeners are less able to listen to music at will. Though music recommendations should be readily available, they should not take away the initiative from the music listener.

A music recommender system does not lead to improved task performance, when listeners have ample time to program, and does not lead to improved programme quality, when listeners have limited time to program. It is very likely that music listeners are carried away by the ample music selection and listening opportunities, which makes them indifferent to task efficiency and less concerned with programme quality. Rather, if they spend more time on a music programming task, they may enjoy the task even more. It is well-known that involvement in particular 'challenging' tasks induces engrossment known as 'the experience of flow' (Csikszentmihalyi and Csikszentmihalyi, 1988; see also Chapter 2). In general, deep task involvement makes people less sensitive to the apparent passage of time (Tsao, Wittlieb, Miller, and Wang, 1983) which also applies to the task of music listening (Palmquist, 1990). As a PATS recommender system is a highly preferred and a highly useful feature for music programming, it seems that task efficiency and programme quality are far less decisive factors for determining a preferred way of music programming than the appreciated usefulness of a functionality and the enjoyment of a music programming task.

As search targets are poorly defined at the outset of a music programming task, it is likely that music listeners need to build a choice context, which contains a limited number of music options, to determine their music preference. Building this choice context is done implicitly, however, and coincidentally while evaluating music options during a search. Searching for preferred music is not done by carrying out a complete or exhaustive choice strategy in the sense that all music in the collection is evaluated. In general, music listeners rather adopt a choice strategy consisting of three phases. In the first phase, music, familiar by the title and name of the main artist, is sought that should anyhow be programmed. In the second phase, less familiar music is sought and selected to complement the initial selection. In the concluding phase, the music programme is fine-tuned. In particular, the second phase can be effectively supported by a music recommender system, which links other music with already chosen music.

6.4.1 Time constraint effects

Previous research has shown that making choices under time constraints results in coping mechanisms such as accelerated processing (Ben Zur and Breznitz, 1981; Payne, Bettman, and Luce, 1996). Though both experiments were originally not designed to investigate time constraint effects on information processing, programming system A was used in both experiments which allows a comparison. It appeared that, under a time constraint, participants listened almost half as long to music options ($F(1,24) = 6.49$, $p < 0.05$), (mean *average listening time*: 15.0 secs (Experiment I), 7.8 secs (Experiment II)) but performed almost twice as many actions per second ($F(1,24) = 12.67$, $p < 0.005$), (mean: 0.19 actions / sec (Experiment I), 0.31 actions / sec (Experiment II)). These results suggest that accelerated processing actually occurred to yield the same result as under the unconstrained time conditions. This is in terms of average listening time per music option and actions performed per second.

Payne, Bettman, and Luce (1996) argue that performing time-constrained choice tasks leads to inaccurate choices when accelerated processing is used as the only coping mechanism. Comparing the number of preferred options, that is, *precision*, in the music programmes that were made with programming system A in both experiments, revealed that music programmes made under a time constraint contained one preferred option less than in an unconstrained situation ($F(1,24) = 8.82$, $p < 0.01$), (mean *precision*: 9.5 (Experiment I), 8.4 (Experiment II)). This suggests that time-constrained music programming does lead to less preferred programmes, probably due to accelerated processing. As shown in Experiment II, 'easing' a time-constrained programming task by means of a music recommender system does improve programme quality, but not significantly.

6.4.2 Some implications for interaction style design

Some concrete recommendations for the design of interactive systems aimed at easy navigation in large music collections can be drawn from the present results. These recommendations pertain to the user requirements mentioned in Chapter 1: *adaptation to music choice behaviour* and *interactive information presentation*.

In general, the comments of participants made clear that direct, easy and electronic access to music is a preferred way of dealing with a large music collection. This is an encouraging finding for the development and market acceptance of interactive players which organise a large music collection electronically.

Adaptation to music choice behaviour

It has been shown that the second phase in a choice strategy for music programming mainly concerns finding music that contributes to previous selections, for instance, made in the first phase. The second phase, in particular, can be effectively supported by a music recommender system which links already selected music to other music in the collection, based on preferred musical attribute values. Music recommendations can be used in numerous ways by music listeners, such as adding them to the programme and getting new ideas for further searching.

How you arrive at preferred attribute values from user feedback given to a recommender system is an entirely different topic. In the experiments, measurements on what music was (only) listened to, accepted and rejected was fed back to the PATS system to infer relevant attribute values and to set up new recommendations for each option. Rejected and already compiled music was also excluded in future recommendations. It appeared in the experiments, however, that participants used the controls related to music acceptance and rejection sparingly; some commented that they were unaware or uncertain about the consequences. Too little feedback about music preference was provided to the system so that it could not prevent inappropriate music from occurring in subsequent recommendations. Eliciting user feedback is a common problem to most recommender systems. Having to give explicit preference feedback about content is often seen as inconvenient and separate from searching and experiencing the content (Balabanovic, 1998). Therefore, more effective user control and system monitoring mechanisms should be developed to obtain preference feedback in a more implicit but accurate way, if possible, and users should be informed about the possible consequences of accepting and rejecting music.

Interactive information presentation

While searching for preferred music, music listeners may rely on many cues. These cues originate from the context-of-use, from the musical content or from music attribute information. Any interactive programming system leads to possibilities and restrictions on using these cues in a music selection process. A general design requirement is, therefore, that a programming system should support all courses of action and all phases in a music choice strategy that a music listener may pursue while using these cues. This system support should be transparent to the music selection process, and its use should require the least possible user effort in terms of actions and time. This would allow music listeners to entirely focus on their cues for programming instead of on the learning of procedures or the control of system behaviour.

It has been shown that the first phase in a choice strategy for music programming mainly consists of finding familiar music (well-defined targets) that is likely to be directly cued by the context-of-use. In this respect, familiar music refers to music from which music listeners know attributes such as the title and names of musicians. If music listeners are not familiar with these attributes, however, searching on these attributes is simply impossible. In Experiment I, participants searched within titles and names of the main artist less frequently than within music styles. One explanation for this behaviour is that they were not sufficiently accustomed to the experimental music collection as it was not their own. Another, more general, explanation is that music listeners are bad in recalling titles without the help of musical cues, as music and titles are neither interchangeable nor primarily learnt in an associative manner. In daily life, music listeners may listen to music without taking the opportunity to look up the title, or may know a song title without ever having heard the music associated with it. Evidence for this explanation is that people recall more titles or have a higher *feeling of knowing* the title from instrumental musical cues than vice versa, though this asymmetry reverses for song with lyrics (Peynircioglu, Tekcan, Wagner, Baxter and Schaffer, 1998). Jazz music is mainly instrumental, however. Therefore, one set of actions should address fast retrieval of familiar music while skipping irrelevant music, which may be based on song titles and names of musicians.

When the first phase is concluded, searching for additional, less familiar music (poorly defined targets) needs other search methods pertaining to recognition of musical cues. Searching for additional music is a dynamic process in which many options are evaluated, compared and even re-evaluated. Mistakes can be easily made. An interactive system should therefore allow easy storage and retrieval of search results, and easy recovery from unintended actions. The experimental systems were not designed with this requirement in mind; the systems were solely designed for experimental purposes, that is, to observe music choice behaviour under different conditions. Locations of interesting music in the collection had to be memorised and had to be re-located if a new examination was found necessary, or if music was accidentally ‘thrown away’. Some participants found an effective and efficient coping mechanism to temporarily store interesting music, however; they first collected all interesting music in the music programme and then removed part of this music in a subsequent examination pass.

Information presentation in current players mainly support finding well-defined targets. As an example, many existing CD jukebox players need a specification of a slot number or a title of the CD and a track number to get a reference to a music recording. In general, interactive information presentation of music should support an elimination-like choice strategy, different orders of presentation, and should provide sufficient information about the music. All three topics are discussed below by referring to the experimental systems.

Elimination-like choice strategy. It is likely that music listeners temporarily eliminate options that do not meet particular features while doing a programming task. The use of this elimination strategy narrows the scope of the music collection while searching, makes options more compatible for inspection and comparison, and hence eases the choice task. The experimental systems allowed music listeners to divide the music collection into music styles, main artist, and titles. Real-time observation of the participants by the experimenter suggested that participants were sometimes lost in the music collection. Firstly, the fixed assignment of music recordings to music styles may not always be the assignment that music listeners have in mind. Therefore, it should be possible to adapt this default assignment by users. Secondly, using a combination of a music style with one of the other two (main artist and title) sometimes resulted in unexpected elimination of options, since these criteria may interact. Some participants commented that the three elimination criteria were too few in number; it restricted exploration. Other criteria may be soloing musical instruments, recording labels or more personal criteria such as mood. The concept of a music style may be extended by a combination of time periods and the year of recording.

Order of presentation. The experimental systems displayed the music collection as a list of music options, alphabetically ordered by title or artist within a music style. Appropriateness of this order was not based on empirical findings and may not support fast retrieval of preferred or familiar music. Other orders may be based on year of recording, tempo of the music performance, or more personal criteria such as date of purchase, usually listened to, and last listened to. The conduct of visual search tasks or small-scale choice tasks are adequate means to discover what order in this domain is usually appropriate under what conditions.

Attribute information. The experimental systems only displayed the names of the main artist or ensemble and the song title of each music option. The music style to which a recording was assigned to could also be traced back. The presentation of additional information was only left out to scale down experimental requirements. However, since CD booklet information was sometimes reported as lacking by participants, presentation of more attribute information (such as defined in Appendix I) is relevant in guiding music choice.

CHAPTER 7

A multimodal interaction style for music programming

Music programming is becoming an increasingly desirable property for interactive players that provide access to large music collections. However, instant access to music is limited by the inconvenient operation of current players due to a large number of control elements and the inadequate representation of visual information. This chapter describes the design and implementation of a multimodal interaction style for music programming intended for players accessing a large music collection. Design requirements were captured from a profile of a target user group, from knowledge on how music listeners search for preferred music, from an analysis of existing players, and from usability standards. The envisaged home use requires *instant usability* and *optional use of a visual display*. A conceptual design process implemented these requirements which resulted in the use of a visual roller metaphor. User control of the interaction style is done entirely by manipulating a force feedback trackball. Actions on the trackball were directly mapped onto behaviour of the roller; a user can turn a roller by rotating the trackball. This provides user navigation for listening to and selecting music. Tactual and auditory (speech and non-speech audio) cues were incorporated to strengthen the impression of rollers, as well as to support effective use without a visual display. Implementation of the interaction style was based on a component software architecture; each component addressed a particular modality. A formative user test uncovered the most prominent problems on instant usability without the use of a visual display, which were then solved in the implementation.

Music listeners are already tempted to organise their music electronically and prefer extended play facilities without the need to handle physical storage media. For instance, music jukeboxes and on-line music from the Web are becoming increasingly popular. When we consider the wide assortment of music available, instant access to a large music collection is becoming increasingly important. In particular, music programming over many discs, recordings or computer files is becoming a desirable player feature. However, music is essentially a linear streaming medium; it takes time to explore a music collection when you have only sequential access to the music. In addition, current CD jukebox players are inconvenient to operate (see Chapter 1). They generally contain numerous control elements that are not well grouped. Many of them have inadequate visual displays that lack relevant information required for music programming, or are poorly legible in dimly lit situations or from a large viewing distance. Their design is based on the erroneous assumption that users can associate numbers or titles with music recordings easily. Therefore, multimodality, as a solution to the many usability problems mentioned, may enhance interaction with music.

This chapter describes the design and implementation of a multimodal interaction style for music programming with the aid of the PATS recommender system. Findings from previous chapters have led to design requirements to be met by the interaction style. The envisaged home use requires that the interaction style can immediately be used without the need for visual inspection of information, that is, provides *instant usability* and *optional use of a visual display.*

7.1 DESIGN REQUIREMENTS

One way of eliciting user desires, problems and needs, and to achieve a better understanding of user requirements, is to observe how existing music players are used or are constructed (Lim and Long, 1994). Therefore, current CD players and, in particular, CD jukeboxes were analysed. The results of this activity were presented in Chapter 1. In addition, ideas for design were based on a profile of a target user group and usability standards.

7.1.1 Target user group

The interaction style is primarily intended for interactive music players accessing a large music collection, such as a jukebox player. Therefore, the target user group consists of music listeners who prefer or intend to store and manage their entire music collection electronically at home. These users are likely to use the device from a distance (using a remote control), when relaxing or in dimly lit conditions (e.g., while having a party). This makes inspection of visual information difficult. Home usage also requires that users do not need to be experienced computer users, by definition. It is actually preferable that the user becomes a specialist in using the device, immediately. The target user group is expected to use the device intermittently, instead of systematically or frequently, though listening sessions may extend over hours. A typical use would be a brief interaction to make a music programme addressing a particular music preference.

7.1.2 Existing systems

Most current music players have a feature to allow users to programme preferred music or to create play lists. How programming is facilitated on current players has been discussed in Chapter 1. A music programming system has been implemented for experimental purposes only, as described in Chapter 6. The purpose was to investigate the effects of music recommendations on a music programming task. Findings from current players, as well as the experimental results, were also used to establish design requirements.

7.1.3 Instant usability with and without a visual display

An important standard for the work presented here, ISO 9241-11 (1998), has the following definition for *usability.*

> *Usability* is the extent to which a product can be used by specified users to achieve specified goals with effectiveness, efficiency and satisfaction in a specified context-of-use.
>
> *Effectiveness* is defined as the accuracy and completeness with which users achieve specified goals.
>
> *Efficiency* is defined as the expenditure of physical and psychological resources with which users achieve specified goals at a given effectiveness level.
>
> *Satisfaction* is defined as the comfort and acceptability of using a product.

Other similar standards and guidelines exist for the development of interactive multimedia services (ETSI, 1998), for software quality (ISO / IEC DIS 9126-1, 1991) and for software product evaluation (ISO / IEC 14598-1, 1999). The latter two standards stress the importance of *quality in use*, which has a similar definition as the ISO usability standard.

The ISO usability standard only provides guidance in evaluating usability of products in the context of *professional* office work involving visual displays. As the focus is entirely on evaluation, guidelines for design are lacking. Other parts of the standard do deliver guidelines on which design decisions can be based, however. But more importantly, requirements for home use of products have some different properties from those for professional use. For instance, how easy it is to learn to use a device and the locus of control are attributes of usability which are not present in the ISO standard, but are relevant to the home use of interactive devices.

Effectiveness and efficiency

Various requirements for *effectiveness* and *efficiency* were identified. Translated to the music programming domain, *effectiveness* relates to the question whether or not music listeners are able to complete a music programming task successfully. *Efficiency* concerns the time, and physical and cognitive effort that needs to be spent on a successful completion of a music programming task.

For reasons of *effectiveness*, an interactive music programming system should support the way music listeners prefer to search for preferred music while programming music. This search can be described by basic actions, such as listening to a music option, including an option in the programme, rejecting an option, eliminating options from further consideration, and going on to other options. The order in which these basic actions are executed is called a choice strategy. In Chapter 6, it was concluded that music listeners generally adopt a choice strategy which starts with a first selection phase in which familiar music is sought that they anyway want to select, and continued by a second phase in which less familiar music is selected to complement the initial selection. A concluding phase is used to change the programme order or to remove less preferred options from the programme.

Providing direct feedback is another means to improve *effectiveness*. It allows an adequate monitoring of the search task and the consequences of actions performed. It is remarkable that most current players do not even make music audible during programming (though there is an intro-scan feature, see Chapter 1).

For reasons of *efficiency*, minimal cognitive and physical effort should be spent on the music selection process. In the case of current CD jukebox players, CDs are in general arranged in musical genres or other descriptive features. Music options have to be indicated by their track numbers on the CD. In the best case, CDs can be referred to by knowing their title or a part of the title. Some players even require the indication of the physical location (a slot number) of a CD. However, people are notoriously inaccurate in remembering what song has which track number or title, or which CD corresponds to which slot number. Looking up titles, track numbers or slot numbers is anyway a cumbersome and time-consuming activity. They may not be known or have been simply forgotten. If music can be directly pointed at, then no cognitive effort is lost on translating music-related features into numbers or titles and vice versa.

It is evident that performing a music choice strategy by using an interactive device takes time and physical actions. Music listeners generally prefer to spend the least possible effort in terms of time and action for collecting music.

Learnability and instant usability

Usually, first-time users of consumer devices attempt to operate the device immediately without the aid of instructions. Consulting a user manual is often perceived as too time consuming or the user manual is simply lost. Since users of home devices have no opportunities for training or are not willing to take them, *learnability* is considered a fundamental usability criterion for home devices (Eggen, Westerink, and Haakma, 1996). Learnability is defined as the resources expended before a specified level of proficiency in terms of effectiveness or efficiency is attained (Nielsen, 1994), or as the speed at which proficiency increases over time (Eggen, Westerink and Haakma, 1996). Therefore, an interaction style should be as transparent, intuitive or self-explanatory as possible, meaning that users are able to perceive the most effective and efficient ways to program music at a glance without procedural instructions. In addition, users should be able to return to the interaction style after a period of not using it, without the burden of re-learning it. In summary, rather than learnability, a home device should allow *instant usability*.

Methods to design for instant usability are consistency of operation, minimality of features, the use of a small number of control elements, and the use of a conceptual metaphor for interaction. Consistency of operation (or similarity of protocols, see Eggen, Westerink and Haakma, 1996) means that the same pattern of actions can be used in different situations, which allows users to learn such a pattern only once. Minimality of features means that infrequently used or more complex functions are implemented in such a way that they do not interfere with initial learning. A small number of control elements allows users to learn the meaning of only a small set of controls or actions. The use of a conceptual metaphor may form a starting point to understand an interaction style, as will be explained in Section 7.2.1.

Locus of control

Locus of control specifies which communication partner (the user or the system) takes initiative and control during the course of interaction. Highly interactive computer games have an interplay of external control and user control called a mixed locus of control, primarily intended to motivate and actively engage users (Gentner, 1992). However, user control conveys the feeling to users that they are able to achieve goals using their own devices; it is an essential property for an interactive home device. Thus, though a music recommender system is a preferred assistance for a music programming task, taking away the initiative of the user is undesirable.

Optional use of a visual display

Current jukebox players have an inadequate presentation of visual information. Some visual displays lack the presentation of relevant information or may be cluttered with irrelevant information. They are often too small or do not have enough contrast to be legible in dim light or from a large viewing distance. The need for visual inspection of information can even be less desirable, for instance, when relaxing while going through the collection. It is for these reasons that the use of a visual display should be *optional* in a music programming task, without sacrificing instant usability.

7.2 DESIGN

The requirements treated above were used as input for the design process. The use of a visual metaphor as a conceptual navigation model was inspired by the principles of conceptual design (Norman, 1986). Conceptual design is based on the assumption that a user constructs a *user model* while working with an interactive device. An appropriate user model explains the device to a user, helps a user to reason about operational procedures and to perform tasks successfully. The goal of conceptual design is, then, to construct a *design model* of the system that will be communicated by the user interface and which allows users to develop an appropriate user model.

7.2.1 The use of a metaphor for instant usability

It is commonly agreed that at least two types of knowledge are relevant to learn how to operate a device: *declarative* knowledge and *procedural* knowledge (Anderson, 1993). Declarative knowledge refers to knowledge of knowing *that*, in the sense that domain objects and relations between them are known. Procedural knowledge refers to knowledge of knowing *how*, in the sense that actions and procedures required to attain a specific goal are known.

The concept of music programming for the purpose of selecting preferred music from a collection is generally self-evident. This means that the objective of a music programming task is well-known, irrespective of whether it is a request or carried out at will. However, novices often find difficulties in transferring this declarative domain knowledge and task objectives into procedural knowledge on *how* to achieve the desired result, though they generally like to know *how-to-use-a-device* first. The task and action space are perceived as infinitely large if the interaction style comprises procedures of many different tasks. Without procedural instructions, the initial problem that first-time users have to overcome is to discover what exactly constitutes an action, what consequences can be expected from an action, and whether or not this result is effective with respect to their task objective (Newell and Simon, 1972; Shrager and Klahr, 1986).

Familiar analogies and metaphors may form a point from which a user can start to understand an interaction style. The use of a metaphor may aid in learning domain knowledge (Lodewijks, 1981; van der Veer, 1991). A metaphor can also act as a conceptual model of how to perform actions and what consequences can be expected from these actions. For instance, the direct manipulation metaphor of graphical user interfaces allows users to act directly in a world of interest; it seems that users do not need to wonder about the consequences of an action and how an action should be performed. This so-called ’direct engagement’ is assumed to underlie the apparent usability of this metaphor (Hutchins, 1989). In addition, metaphorical instruction helps users to learn procedures quicker and to execute them in less time, but it also helps them to recover from errors and to learn new tasks (Kieras and Bovair, 1984). Metaphorical instruction about a command-driven device also results in an improved retention of the command language and better learning of distinctive procedures of command use (Payne, 1988).

During the iterative design process, several metaphors starting from a spherical object were considered as a conceptual navigation model in the music programming domain. The spherical object was prompted by the chosen input device (see Section 7.2.2). In our case, the music programming domain consists of a music programme to be created, a music collection with music options from various music styles, and a recommender system that links 'similar' music options to each other. While adding actions essential to navigation in this domain, the conceptual model was assessed on appropriateness using a pencil-and-paper method. The main criteria for assessment were consistency of operation, a minimum number of actions required and compliance with the envisaged metaphor. Secondary criteria were implementation feasibility and predicted computational resources required. The final navigation model was based on a fruit machine consisting of four rollers, on which the music programme, the music styles, the music collection and the music recommendations were subsequently projected.

Figure 7.1. Implemented visual display of the conceptual navigation model. The music programme contains two music options. The currently selected music style is 'postbop' and a famous piece of postbop jazz music of Miles Davis is playing. Three visible recommendations are linked to this recording.

The implemented visual representation is shown in Figure 7.1. The roller which has input focus is high-lighted. The title and artist of each music recording are projected on the rollers, allowing direct search and the learning of these attributes while programming. The left-hand side represents the roller for the music programme. A counter is positioned over this roller and displays the number of options added to the programme. The next roller represents the music styles in the collection. Music styles are arranged in a chronological order, that is, in the order in which the music styles have emerged in time. Next, the music collection roller displays all options in the collection or just the subset of recordings that belongs to a particular music style. Music options on the music collection roller are first alphabetically ordered by artist. Recordings of the same artist are grouped by album in the order in which these recordings came out on the album. Finally, albums are chronologically ordered by year of publication. The numbers displayed on the music collection roller are primarily to present the number of options available in a music style or in the total music collection. The right-hand side roller contains a list of music recommendations that correspond to the music option that is at the front of the music collection roller; scrolling to another option on the music collection roller immediately changes the list of music recommendations.

7.2.2 A trackball with force feedback as input device

The IPO two-DOF trackball device with tactual force feedback was used as a pointing input device. If it is possible to define a set of expressive user actions for this trackball, as will be explained, this device can be deployed as the sole control element in an interaction style (for expressiveness of input devices, see Buxton, 1983; Mackinlay, Card and Robertson, 1990).

The IPO force feedback trackball is a ground-based input device in which user input is mediated by manipulating a small hard-plastic ball, which rotates freely in a housing. Force feedback is mediated by two motor-driven wheels which touch the ball at its x and y axes. The force can be made dependent on the current context of interaction, which the user only perceives by manipulating the ball. The mechanic and electronic structure of the trackball device used is presented in Appendix II.

User control of the trackball in the interaction style

In general, working with a pointing device requires continuous attention to a visual display rather than attention to the wrist, hand and fingers that control the device. A user continuously has to inspect the positions and states of the cursor and the task objects displayed on the screen. Obviously, visual inspection is difficult at a large viewing distance and impossible if no visual display is used.

In order to establish a cursor position, physical displacement of the trackball is used *relative* to an initial reference position; it operates in relative mode (Douglas and Mithal, 1998). Moving a cursor back and forth in a complex navigation task is an effortful activity if a trackball is used. In particular, a trackball is the most inefficient input device to perform target acquisition tasks in comparison to a mouse and a tablet (MacKenzie, Sellen and Buxton, 1991). An *absolute* pointing device such as a tablet is more efficient for target acquisition because it directly maps a physical device position onto cursor position. This provides a more direct translation of finger, hand and arm control to cursor movement.

Both the need for visual inspection and inefficient operation are reduced by providing direct tactual feedback and by directly mapping user actions onto consequences in the visual metaphor.

If a user performs an action using the interaction style, this can be directly felt. Since tactual feedback can be varied along the context of interaction, it may be adequate to guide users in performing an action, probably even without a visual display. It has been demonstrated that the addition of tactile or tactual feedback makes people respond faster with their hand (Nelson, McCandlish and Douglas, 1990) or perform finger movements more rapidly (Akamatsu, 1991). With respect to a mouse, it leads to a more efficient performance of target acquisition tasks (Akamatsu and Sato, 1994; Akamatsu, MacKenzie and Hasbrouc, 1995; Akamatsu and MacKenzie, 1996). With respect to the trackball used, added tactual force feedback leads to a more efficient target acquisition (Engel, Goossens and Haakma, 1994; Keyson, 1997) and a more accurate and faster response to a cued directional movement (Keyson, 1996).

If a user performs an action using the interaction style, it has an immediate impact on the behaviour of the visual rollers, thus providing direct meaning to an action. The latter is essential to reduce cognitive effort in user-system interaction (Hutchins, 1989). For instance, rolling the ball back or forth directly corresponds to a similar proportional rotation of a roller. Lateral roll movements correspond to hopping from one roller to the next.

As a result, a user may experience the trackball device rather as a means to act on rollers by gestures such as instantaneous hand-finger movements, than a means to roll and point to rollers.

A set of user actions

A set of ten expressive user actions pertaining to navigation and selection was defined using ball movements and ball presses. Actions to control music playback (e.g., stopping, pausing music) were left out of the design. These control actions overload the use of the trackball as a single control element and hence may negatively affect initial learning of the interaction style. Therefore, music plays automatically when the ball is located on the three rollers containing music options, and stops when it enters the music styles roller.

Essentially, two directions were defined for moving the trackball, that is, forwards / backwards and laterally. Using forward and backward ball movements, a roller can be set in motion to bring another item, such as a music option or music style, to the front. A small hand movement brings the next item to the front. A somewhat faster hand-stroke, covering a somewhat larger distance, skips the next two items. This skip was limited to only two items to allow users to easily see the gradual roller change from the start position to the end position. These two kinds of roll movements were intended to support the first two phases of the preferred choice strategy for music programming. Users are able to either quickly go to familiar items by skipping irrelevant items, or proceed sequentially along the list of items to examine less familiar material.

Using lateral trackball movements, a user can hop from one roller to another. Again, a small hand movement makes a hop to the next roller, and a somewhat faster hand-stroke skips one roller. Using movements in both directions, a user can first choose a particular music style on the music styles roller, and continue searching for music options within that style by going to the music collection and recommendation rollers.

A single press on the trackball provides spoken information of the current music style. Double-pressing the trackball results in adding or removing a music option to or from the music programme.

7.2.3 Multimodal interaction style

In addition to visual and tactual feedback, the interaction style uses synthetic speech and non-speech audio feedback. Combining one human output channel and various human input channels of information results in a multimodal interaction style. The output channel is addressed by manual operation only, but perception comes through the visual, auditory and tactual (tactile / kinesthetic) senses. Note that the auditory modality is fed by three different audio streams: synthetic speech, non-speech audio and music audio.

There seems to be a current trend towards designing multimodal interaction styles in which the auditory and tactual modality are used as alternative or supplementary means to the visual modality to convey information between the user and the system. For instance, the use of this form of multimodality has already been explored in home entertainment systems (Bongers, Eggen, Keyson and Pauws, 1997; 1998). Adding extra information by tactile and tactual feedback to user-system interaction may reduce visual load (Akamatsu, 1991) or may be useful in situations where the human visual system is likely to be heavily in demand (Keyson, 1996). The addition of auditory and speech cues to visual interfaces is a prerequisite to provide computer access for the visually disabled (Poll, 1996; Mynatt, 1997).

Besides a reduction in visual load, the general rationale for the development of multimodal interaction seems to be based on the assumption that people can use quite independent attentional resources for each sensory modality. This should allow them to perform multiple tasks concurrently. To some extent, people are able to divide attention quite efficiently across modalities to process information that comes through the senses (Wickens, 1984; 1992; Proctor and Dutta, 1995). For instance, if a stimulus is equally likely to occur across all modalities, then people can concurrently monitor multiple sensory modalities for this stimulus as effectively as they could monitor only one of these modalities (Spence and Driver, 1997). Performance decreases when people have to shift to another modality unexpectedly to perceive or respond to a stimulus (Boulter, 1977; Post and Chapman, 1991; Spence and Driver, 1997b). Hence, cross-modality limits do exist and can negatively affect task performance, which argues against the notion of independent attentional resources for each modality.

However, people attend to events in the real world instead of individual sensory sensations (Gibson, 1979). The same holds true for user-system interaction; users await system events to happen or the consequences of their actions, though they should be represented in an expected format or modality. If multimodal interaction styles preserve the way in how sensory sensations and events are correlated in the real world, they may be more usable than unimodal interaction styles. In this way, they will match user expectations on perception and action and extend user imagery. Systematic studies that address the extent to which multimodal interaction styles are more usable than their unimodal variants, or what impact cross-modality limits have on usability, are rare, however (for some user evaluations on the combined use of voice control and manual operation, see Gourdol, Nigay, Salber and Coutaz, 1992; Hauptmann and McAvinney, 1993; Robbe, Carbonell and Valot, 1997).

In the interaction style, simultaneous auditory and tactual feedback is used to strengthen the impression of the visual roller metaphor, to enhance efficient operation, and to support the optional use of a visual display. Tactual force feedback was set up to convey the feeling of setting rollers into motion or hopping from one roller to the other. Non-speech sound feedback was set up to convey the sound of manipulating rollers in various ways. Synthetic speech feedback was set up to inform the user about specific states of the interaction. In this way, a modality provides redundant information that is already encoded in some way by the visual modality. It is assumed that this form of redundancy relieves the user from explicitly shifting attention to other modalities and gets round the aspect of cross-modality limits, if the different modalities map well in time, space and function.

The use of non-speech audio

One of the approaches to implement non-speech audio in user-system interaction is by using *auditory icons* (for other approaches, see Gaver, 1997). Auditory icons are probably best described as caricatures of naturally occurring sounds. They are central to an ecological approach to sound perception, formulated in Gaver's framework of *everyday listening (*Gaver 1986; 1993). In this framework, sound perception is understood to be the notion that people attend to events in the world rather than sensations per se. For instance, if a block of material is struck, people can reliably tell what the block is made of by the sound that it makes (Gaver, 1993), the hardness of a mallet striking it (Freed, 1990), or whether a bottle has bounced or broken (Warren and Verbrugge, 1984). Thus, a benefit of using auditory icons is that they may have an intuitive appeal to the listener.

Auditory icons have been implemented in computer interfaces (Gaver, 1989; Buxton, Gaver and Bly, 1992), are tested in learning television programme categories in a consumer device context (van de Sluis, Eggen and Rypkema, 1997), and are particularly suitable for providing access to graphical user interfaces for blind users (Poll, 1996; Mynatt, 1997).

In the interaction style, the consequence of a user action is conveyed by a special class of auditory icons, namely impact sounds of material being struck. Each roller in the visual metaphor has its own set of impact sounds as produced by a different material being struck. The music styles roller is represented by sounds that are probably best described as sounds produced when a metal-like object is struck by another solid object. The music programme roller is represented by glass-like sounds, the music collection roller produces wood-like sounds and the recommendations roller sounds like rubber. The choice for these materials was prompted by the results of an explorative study which indicated that people label these materials reliably to different synthetic impact sounds (Hermes, 1998). It was not intended here that users should be able to tell what the roller would be made of by the sound that it makes. Instead, it was intended that the rollers can be distinguished by sound alone while manipulating them. In addition, the type of action needed to be distinguished by the type of sound. Therefore, manipulating a roller by the trackball step-wise produces a single click. Rolling a roller up ‘clicks’ different from rolling it down. Rolling a roller more vigorously first produces a spectrally broader sound suggesting a greater impact, and then a rattling sound to represent the visual scroll of the roller. Adding a music option to a music programme produces a ’creaking glass’ sound, while the roller is visually turned to present the newly added option at the front of the roller. Removing a music option from the music programme produces a ’bouncing glass’ sound, while the roller bounces visually to a new position.

The use of synthesised speech

Speech output is used to convey information about the current state of interaction, primarily intended to allow the interaction style to be used without a visual display. For instance, after hopping from one roller to another, spoken feedback indicates to which roller was moved to. Also, when entering the music programme roller, it indicates how many music options have been added to the programme. When the user rolls through the music styles roller, it tells the user which music style is left and which is entered (e.g., ‘from bebop to hardbop’). By pressing the ball once, the user can ask what the currently selected music style is or to what music style a selected music option belongs.

The following phrases are used:

‘Programme’, ‘Styles’, ‘Collection’ and ‘Recommendations’ to indicate a roller;

‘all styles’, and the names of 12 jazz music styles as found in the music collection;

‘from ..... to ..... ‘, in which the gaps are filled by names of music styles to indicate what music style is left for another;

‘..... track’, in which the gap is filled by ‘no’ or ‘one’, and ‘..... tracks’, in which the gap is filled by the numerals ‘two’ up to ‘twelve’, or by the phrase ‘more than twelve’ to indicate how many music options have been added to the music programme.

7.3 IMPLEMENTATION

The implementation of the interaction style was carried out on a Windows 95/98 PC platform. A component-based software architecture based on Microsoft ActiveX technology was used in order to separate the development of the different input and output modalities and to support easy change, exchange, and reuse of the software (for component-based software design, see Rogerson, 1997; Szyperski, 1998). The results of a formative user test, which concluded the implementation, are presented in Section 7.3.2.

7.3.1 Component-based architecture

As shown in Figure 7.2, the implementation of the interaction style is a composite software system. The graphical notation used in Figure 7.2 is adopted from the Unified Modeling Language (Rumbaugh, Jacobson and Booch, 1999). Each component is briefly discussed in separate sections. A detailed description of the software components is documented elsewhere (Pauws, 1998).

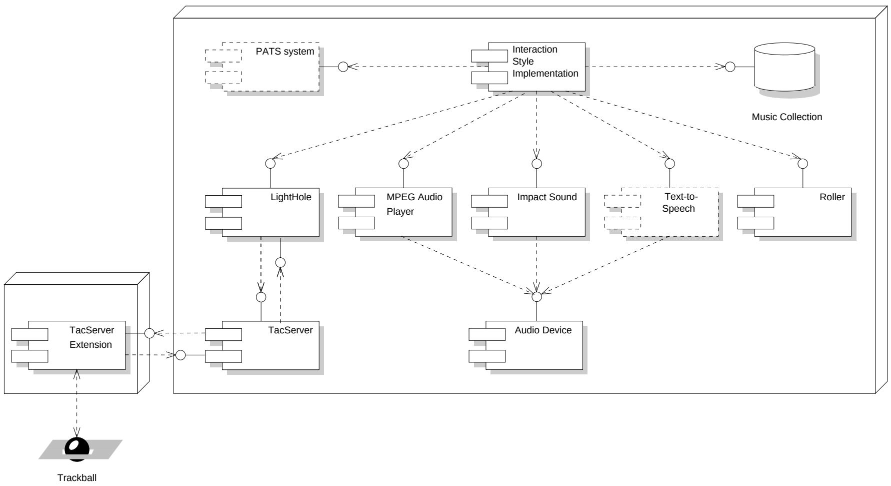

Figure 7.2. Software components of the interaction style implementation. The ‘lollipops’ represent software interfaces.

The interaction style implementation is a container application written in Microsoft Visual Basic, which controls multiple instances of the software components. The software components were written in Microsoft Visual C++. The PATS system was used in order to generate music recommendations (see Chapter 4). The system was not ported or connected to the PC platform. Instead, music recommendations were pre-generated and read in from a file. Data of the music collection is contained in a database component (see Appendix I). The manual input and tactual output of the

IPO force feedback trackball is controlled by three components: *LightHole*, *TacServer* and *TacServer Extension* (see Appendix II). As shown in Figure 7.2, the software control of the trackball is distributed over two serially connected PC platforms. Music is played back by an *MPEG Audio Player*. Non-speech audio is generated by a component called *Impact Sound*. Text-to-speech synthesis was not fully implemented. Currently, the preliminary *Text-to-speech* component concatenates pre-recorded synthetic speech. It can be easily replaced by third-party software. Audio streaming and mixing is implemented in an *Audio Device* component. The visual representation of the rollers is implemented in the *Roller* component. All components start their own threads which can be interrupted to some extent, when other services are asked from them. Threads between components are not synchronised which means that the different output modalities are also not synchronised.

Lighthole component

The addition of force feedback in an interaction style is based on a spatial arrangement of tactual objects. A tactual object is essentially an instance of the *Lighthole* component. At a lower implementation level, it represents a two-dimensional force field map that is described by geometrical formulas. The shape of these formulas have prompted names for different types of tactual objects such as 'hole', 'peak', 'hill' and 'path'. A tactual object evokes a force vector dependent on the current cursor (ball) position and its force field map. All evoked force vectors are summed and mediated by a motor-driven ball rotation. For instance, the force field map of a circular 'hole' evokes a directional pulling-force towards its force field centre, when the cursor is moved into the region of the 'hole'. This feels like being captured in a region when rolling the ball, and needs some additional hand force to leave this region. Definitions of some force field maps are described elsewhere (Keyson and Tang, 1995; Keyson, 1996; Keyson and van Stuivenberg, 1997).

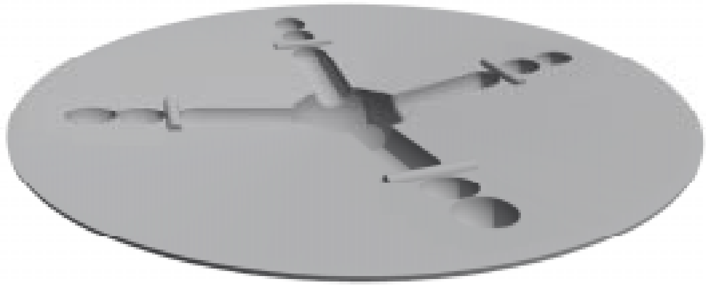

Figure 7.3. Spatial arrangement of force fields in the interaction style.

The layout of the tactual objects in the interaction style is shown in Figure 7.3. The ball's freedom of movement is constrained by software such that it is not possible to move the ball diagonally or on the plateau. To achieve this, rectangular force fields, placed in a cross, capture the ball and guide ball movement along a straight line. Each user action starts with the cursor at the cross' centre. Note that the cursor is not shown on the visual display. A circular force field pushes the ball slightly towards the centre of the cross; it requires some hand-force to leave this centre. The ends of the cross are marked by raised edges; these edges slightly hold back a continuous ball movement. When moving across an edge, the ball will be captured by a small hole. This capture event typifies the conclusion of a roll movement covering 1/16th part of the trackball's circumference (approximately 1.5 cm at the surface). Immediately, the cursor is re-positioned at the cross' centre. Rolling the trackball in a straight line and pushing through a small barrier mimics the sensation felt by a 'mechanical click'[1]. If the ball is moved across an edge with some more hand-force, the cursor will miss the first hole and will end up in the next hole. This capture event typifies the conclusion of a more vigorous roll movement.

The instance of a *LightHole* component sends ball movements and presses as mouse events to the interaction style implementation allowing this application to react and add meaning to these user actions. For instance, each forward and backward movement corresponds to a scroll of a roller, and each lateral movement corresponds to hopping from one roller to another.

Text-to-speech component

The speech output is based on the concatenation of pre-synthesised speech. Speech was synthesised by a third-party text-to-speech engine that was pointed out as being the best with respect to segmental and prosodic quality and intelligibility, among four engines by personal observations and after discussion with speech researchers at the laboratory.

Playback could be interrupted and stopped when the speech content was no longer valid within the context of the interaction, that is, when the user proceeded by performing another action.

Impact Sound component

A common software technique to generate impact sounds is by means of Constrained Additive Synthesis (Gaver, 1993), though other techniques exist as well. An in-house developed implementation of this synthesis technique made up the *Impact Sound* component. When sound is synthesised rather than pre-recorded and sampled, it eliminates storage requirements and provides a larger flexibility in producing sound in real-time.

Constrained Additive Synthesis is based on the addition of sine waveforms (pure tones) in the time domain, since it is known in Fourier theory that any complex sound can be decomposed into an indefinitely long list of sine waveform (Rossing, 1990). In order to speed up the synthesis, other elementary signal components such as block waveforms, sawtooth waveforms or triangle waveforms are also used. Each elementary signal component is described by a set of parameters such as centre frequency, initial amplitude, duration, phase shift, onset and decay. In this way, a signal component is assumed to correspond to a characteristic resonant mode of an impact conveying properties such as mass, size and hardness of the material being struck and force and proximity of the impact event (Gaver, 1993). The addition of these signal components results in a finite signal form, which can be

1. A force value (86 grams) that could be reliably felt as a 'mechanical click' in all four directions was determined by doubling the force value that converges at the 70.7% correct level, obtained by a two-down, one-up adaptive-staircase method (Levitt, 1970). Participants, admitted to a Two-Alternative Forced Choice test (2AFC) (Green and Swets, 1966), almost completely correctly discerned this force value from no force value.

excited by different types of pulse trains to obtain one or multiple copies of this signal form in time.

The synthesis of the metal-like, glass-like, wood-like and rubber-like sounds in the interaction style was based on the results of an explorative study (Hermes, 1998). In this study, different values for the parameters centre frequency and decay time appeared to encode the perception of these materials accurately. Single impacts such as a click or stroke were generated by a single pulse excitation. Rattling impacts were generated by a finite pulse train excitation. Repetitive impacts that convey the act of bouncing were generated by a pulse train with exponentially decaying intervals and amplitudes. The act of creaking was imitated by reversing the latter pulse train.

Audio Device component

In the interaction style, simultaneous streaming and mixing of audio from different sources was required while guaranteeing low latency and maximum control. Therefore, an *Audio Device* component for all audio output streaming services has been implemented which encapsulates Microsoft's DirectSound technology (Bargen and Donnelly, 1998). Required functions lacking in DirectSound 5.0 were also implemented such as audio flushing and sample rate conversion for mixing audio streams into one high-quality stereo format.

Roller component

In the interaction style, items such as music styles and music options are visually presented in a list projected on a roller. Since any rotation of the trackball had to coincide with a proportional turn of the roller on the screen, the list of items was virtually tapered around a ball (the trackball), analogous to wrapping a sheet of paper around a cylindric object (see Figure 7.4). In that configuration, a forward ball rotation turns a roller clock-wise or upward which was considered a more direct and 'natural' effect. For experienced mouse-users, however, it may be 'counter-intuitive' at first, since a forward mouse movement generally causes a downward scroll of visually presented information in a document editor, for instance.

Figure 7.4. Tapering an indefinitely long list of music options around a ball.

The *Roller* component generates a bitmap that contains textual information of the items in the list that are visible on the roller, from the user's point of view. For instance, the bitmap may present song titles and names of artists that appear at the front of the roller. This bitmap is altered by a warp function to mimic a cylindric projection and is treated by a median filter to avoid anti-aliasing. An illumination technique adds the roundness to the roller. Control and visualisation are tightly coupled since list manipulations, such as adding and removing items or changing list position, have a direct effect on the visualisation.

7.3.2 Formative user test

Implementation of the interaction style was concluded by a formative user test. The purpose of this test was to uncover prominent problems concerning instant usability of the interaction style without the use of a visual display.

Participants were assigned in blocks of two. They were instructed to freely explore the interaction style without a visual display for five minutes, and to discover as many features as possible without being given any procedural instructions at the outset (Shrager and Klahr, 1986). After this discovery learning phase, participants were instructed to create a music programme containing 10 distinct music options. If neither of the two participants in a block was able to perform a given music programming task successfully, it was concluded that instant usability was insufficient. Participants spontaneously made suggestions for improvement. After completion of a block, imperfections pertaining to instant usability were repaired for the next block. In the fifth block, both participants (the ninth and tenth participant) eventually mastered the interaction style; they both completed a music programming task successfully. No further important shortcomings or user complaints were reported. It was concluded that no further refinements to the interaction style were necessary. This iterative test-and-repair cycle required ten participants before the interaction style was found sufficiently usable.

7.4 Discussion

Current players with a large storage for music such as jukeboxes are poorly designed with respect to usability. In general, they contain a many badly grouped control elements and have inadequate visual displays. These undesirable features make it difficult to learn to use a player effectively and efficiently. This chapter reported the design and implementation of an interaction style that is primarily intended for the task of music programming for this kind of players. Main design requirements for this interaction style concerned *instant usability* and the *optional use of a visual display* when programming music. *Instant usability* is particularly important for interactive devices in a domestic context-of-use, since these devices are typically used intermittently without training (Eggen, Haakma, and Westerink, 1996). *Optional use of a visual display* is particularly important to compensate for the inadequate visual displays of current players, and to reduce the need for visual inspection, which may be inappropriate in the context of music selection and listening.

The multimodal interaction style deals mainly with navigation in a music collection to listen to and collect preferred music. In addition, music recommendations are presented to ease the music selection process. Requirements on instant usability limited the implementation of additional operations and functions, but these functions are necessary for a fully-fledged application. User control proceeds entirely by a single control element: a force feedback trackball. As prototypes for low-cost, hand-held force feedback trackball devices are already constructed, the interaction style may also be applicable for portable players, car audio equipment and remote controls. Results of a formative user test suggested that instant usability without the use of a visual display has been met. In Chapter 8, results of a conclusive user evaluation is presented for complete validation.

The use of a visual metaphor was inspired by the principles of conceptual design (Norman, 1986). The metaphor underlying the interaction style is a fruit machine consisting of rollers. This familiar analogy was intended to form a starting point to learn to use the interaction style. Auditory and tactual cues were incorporated to strengthen the impression of rollers, thus, extending user imagery, and to support the use of the interaction style without a visual display.

Findings from user experiments on music programming, the analysis of existing music players, a profile of a target user group and usability standards were a valuable input for setting up the design requirements. In general, input on user desires, problems and needs should be introduced early and should be continued in the design and implementation of an interactive device (Lim and Long, 1994). This input helps to find at least the initial answers of design concerning what, why, and for whom to design. Further design of the interaction style was largely guided by implementation feasibility, engineering heuristics and the results of empirical studies on human cognition and action, multimodality and user-system interaction. Most of these inputs to design lack precision as they originate from another level of abstraction or another application. Consequently, design decisions for visual, tactual, sound and speech feedback for multimodal interaction do not appear overnight, but require much formative user testing for their validation.

The multimodal interaction style was implemented using a component-based software architecture, in which each component addressed an input or output modality. If these components are not readily available, multimodal interaction styles require substantially more implementation effort than the development of traditional unimodal interaction styles. In general, the software architectural requirements for interaction styles are increasing because of new demands to integrate more input and output modalities. However, the formulation of these requirements are largely implicit or open.

In summary, further research is required to identify in which situations multimodal interaction can be best deployed, what cross-modality limits have their effects, and how design and software engineering knowledge in the new practices of multimodal interaction style design (e.g., tactual and sound design) can best be developed and deployed.

CHAPTER 8

Evaluation of a multimodal interaction style for music programming

The results of an experimental evaluation designed to assess the usability properties of a multimodal interaction style for music programming are reported. The visual metaphor underlying the interaction style is a fruit machine consisting of rollers. Tactual and auditory cues were added to the interaction style to strengthen the impression of rollers, but more importantly, to support optional use of a visual display. The experiment essentially investigated task performance and learning of procedures while performing music programming tasks in the presence or absence of a visual display, combined with a tactual and auditory interface. The participant's task was to create a music programme as quickly as possible without paying attention to music preference. Participants were instructed to complete four tasks in two experimental sessions. Task performance was measured by *time on task* and *number of actions*. Procedural knowledge was assessed by a post-task questionnaire. Results showed that participants were able to complete a music programming task successfully, right from the start, with or without a visual display. Task completion without a visual display required more time, but did not require the performance of more actions. However, participants who worked without a visual display were steep learners; they needed increasingly less time for the second and third task. When working without a visual display for the first time, additional time is required to acquire procedural and spatial knowledge of the interaction style. Indeed, participants who had worked without a visual display had learnt more procedures. Earlier experience with a visual display did not improve task performance without a visual display. It also appeared that participants who had worked with a visual display made a drawing of the interaction style that contained essentially aspects of the visual metaphor. In contrast, participants who had worked without a visual display made a reproduction containing control actions. The results showed that tactual and auditory feedback can make interaction possible in contexts-of-use in which information on a visual display is poorly legible or even absent, such as when using jukebox players, portable players, remote controls and car audio equipment.

The previous chapter described the design and implementation of a multimodal interaction style for music programming with the aid of the PATS recommender system. The results of a formative user test suggested that users are able to complete a music programme task successfully, when using the interaction style without a visual display, after five minutes of free exploration and without being given procedural instructions. This chapter presents the design and results of a conclusive experimental evaluation of the usability properties of this interaction style. In particular, the evaluation concentrated on the presence or absence of a visual display and the learning of procedures. In both visual display conditions, the interaction style was supplied with a tactual and auditory interface. Parts of the evaluation are also reported elsewhere (Pauws, 1998; Pauws, Bouwhuis and Eggen, 1998; Pauws, Bouwhuis and Eggen, accepted).

8.1 A RE-CAP OF THE MULTIMODAL INTERACTION STYLE

For this evaluation, a jazz music collection was divided into 12 jazz styles, each containing 40 music options. Each music option had a reference to three similar music options, that is, PATS music recommendations, based on attribute similarity of the musical content. The aim of the use of these recommendations was to enable fast selection of preferred music. PATS is discussed in Chapter 4. Music options both from the collection and from music recommendations could be added to a music programme.

As shown in Figure 7.1, the visual metaphor underlying the interaction style is a fruit machine consisting of four main rollers on which the music programme, the music styles, the music collection and the music recommendations are projected. User control of the interaction is done entirely by manual operation of the IPO force feedback trackball (see Appendix II). The interaction style combined three output modalities (visual, auditory and tactual) intended to strengthen the impression of rollers and to support optional use of a visual display. For instance, the tactual force feedback mediated by the trackball was set up to convey the feeling of setting rollers into motion. Procedures to control the interaction style are presented in Section 7.2.2.

8.2 INTERACTION WITH AND WITHOUT A VISUAL DISPLAY

As is emphasised in the case of locomotion without vision (Klatzky, Loomis, and Golledge, 1997), purposeful nonvisual interaction grows out of knowledge about the interaction space. As already discussed in Chapter 7, at least two types of knowledge are relevant for learning to use an interaction style: *declarative* knowledge and *procedural* knowledge (Anderson, 1993). It is obvious that working with the interaction style without a visual display also requires the acquisition of *spatial* knowledge about task objects (the rollers), in order to infer procedures.

In this study, declarative knowledge about the music programming domain is given at the outset, while no information about existing procedures is given. When no visual display is used, there is also no information given or visible about the spatial arrangement of task objects in the interaction style. Consequently, acquisition of spatial and procedural knowledge, that is, the development of a cognitive map, has to occur during the actual course of interaction (for an example of research on cognitive maps, see Denis, 1991; Sholl, 1996).

In the interaction style, tactual and auditory feedback is only provided by performing an action, and, by definition, is absent when no actions take place. Hence, spatial and procedural knowledge can only be acquired by performing actions designated for that purpose, when there is no visual display available. During first-time use, acquisition of this knowledge is likely to reduce task efficiency. If this knowledge has been acquired, task efficiency may improve with practice.

While working without a visual display, procedures and spatial information also have to be memorised actively. In contrast, a visual display does not require users to remember this knowledge actively, since the visual display simply shows the task objects and thus reminds users of procedures when necessary.

Furthermore, working without a visual display produces more uncertainty about the consequences of an action, and, consequently, about the current state of the interaction. This uncertainty induces users to resort to an inefficient *dead reckoning* process while performing a task. Dead reckoning means that each new state of the interaction must be determined by explicitly knowing the starting state and knowing the consequences of the action. This is in contrast to a more efficient *planning ahead* process of task execution, which is facilitated when a visual display is available. Visually represented objects can act as landmarks allowing a global delineation of the task across these landmarks, and facilitating the determination of required actions beforehand.

In summary, working without a visual display is likely to be less efficient than working with a visual display. However, something that is learnt while doing tasks with a visual display may be carried over ('transferred') to the same type of tasks without a visual display, which may facilitate an efficient performance of the latter tasks. The question is then whether or not users perform more efficiently without a visual display after earlier experience with a visual display, and not by mere practice in doing the same tasks.

If sufficient learning has taken place on a two-dimensional visual image, it has been shown that people are able to recall a mental image of objects accurately, which preserves the original spatial relationships between these objects on the visual image (Denis, 1991). Hence, earlier experience with a visual display may provide a mental image of the interaction space, which users can use from then on without a visual display. As shown by Dennis (1991; 1993), imagery may provide information that can be manipulated in a similar way as if this information was presented visually. For instance, users may plan their actions, instead of performing their actions one-by-one, using a dead reckoning process. It is therefore expected that earlier experience with a visual display makes task performance without a visual display more efficient, as users have accurate knowledge about the spatial arrangements of the task objects in the form of a mental image.

8.3 EXPERIMENT

8.3.1 Hypotheses

The critical factor between visual and non-visual interaction, with only tactual and auditory feedback, is human memory. In the interaction style, tactual and auditory feedback is transient, whereas visual feedback is continuous and persistent. Non-visual interaction requires additional time and actions to acquire and help remembering procedural and spatial knowledge, and to pursue a dead reckoning process of action. Hence, it is hypothesised that:

(i) *Visual* interaction is more efficient, that is, needs less time and fewer actions, than *non-visual* interaction, while the level of auditory and tactual feedback in both conditions is kept constant.

Non-visual interaction requires the explicit acquisition and remembering of procedures. Consequently, it is likely that users who interact non-visually develop a substantial body of procedural knowledge. It is therefore hypothesised that:

(ii) Users who have worked *without* a visual display have a higher score on *procedural knowledge* than users who have worked *with* a visual display, while the level of auditory and tactual feedback in both conditions is kept constant.

After users have worked with a visual display in previous tasks, it is likely that they will use a mental image of the visual metaphor to retrieve information and to plan actions when interacting non-visually. It is expected that this will facilitate efficient task performance. It is therefore hypothesised that:

> (iii) Users who worked *with* a visual display in one condition and are subsequently transferred to another condition *without* a visual display perform more efficiently, that is, they need less time and fewer actions, than users who start working *without* a visual display.

Working with a visual display induces users to adopt the visual metaphor for later re-production and explanation. On the other hand, working without a visual display forces users to take explicit account of actions and their non-visual consequences. It is therefore hypothesised that:

> (iv) Users who worked *with* a visual display draw a reproduction of the interaction style which contains more visual aspects, whereas users who have worked *without* a visual display reproduce the interaction style which contains more action-and-effect related aspects.

8.3.2 Measures

Task performance

Two task performance measures were defined: *number of actions* and *time on task.* They were measured from the first action to the last action of the task.

Transfer

The transfer was defined as the difference between task performance scores of participants who performed tasks with a particular interaction style B after they had done tasks using an interaction style A, and participants who started performing tasks using that particular interaction style B without having first worked with interaction style A.

Procedural knowledge

Procedural knowledge was measured using two versions of a 20-item questionnaire; one was handed out half-way through the experiment, the other at the end of the experiment. The two versions contained a relatively large overlap of questions (16) as well as four distinct questions. Each question first informed participants about an initial state of the interaction and a desired, final state of interaction. Then, participants were asked what sequence of actions most efficiently and successfully transformed the initial state into the desired state. Half of the questions in the questionnaire represented questions on procedures consisting of a single action (single-step interaction). The other half represented questions on procedures that could be performed by 2 or 3 actions (multiple-step interaction). The questions were presented in a random order. The order in which the two versions of the questionnaires were given was counterbalanced. A total of 10 pilot reviewers checked consistency and comprehensibility of the questions. Participants gave spoken answers to the questions. The test supervisor coded the answer, making use of a short-hand notation. Answers to items could not be withdrawn or corrected after the conclusion. Complete description of the questions can be found in Section III.4 in Appendix III.

Answers were judged correct if the responded action sequence successfully transformed the initial state of interaction into the final state of interaction and the action sequence was of minimal length, that is, most efficient. Answers were otherwise judged incorrect. All correct answers were added up to arrive at a *questionnaire score* (maximum: 20).

Drawing

Participants were asked to imagine a situation in which they had to explain the interaction style to a person, who was new to the interaction style (*teach-back* method, see van der Veer, 1994). To aid the explanation, they were instructed to produce a free hand-drawing of the interaction style, containing all necessary details to explain the interaction style. The test supervisor did not intervene while the participant was drawing the interaction style. After the drawing was completed, the participants were asked to explain the interaction style to the test supervisor by referring to their drawing. This explanation was primarily so that the experimenter could interpret the drawing correctly.

8.3.3 Method

Instruction

Participants read a short text about the music programming domain to provide them with the necessary amount of declarative knowledge. No reference was made to the interaction style; no instruction about procedures or spatial arrangements of task objects were given. Participants were asked to rephrase the given short text in their own words. Any misconception about the music programming domain was corrected by the test supervisor.

Task

At the outset of each music programming task, participants received a written task description. Again, participants were asked to rephrase the task instruction to avoid any misconceptions about the task. The task was to select 10 distinct music options, equally drawn from two pre-defined music styles, as quickly as possible while paying no attention to personal preferences or to the order of the selection process. Four music programming tasks were defined and their order of presentation was counterbalanced. The tasks were designed to be equally difficult; a successful and most efficient task completion demanded 23 actions.

Design

Four *conditions* were applied; two control and two experimental conditions. The control conditions were represented as receiving no change in the presence of absence of a visual display. In one control condition, denoted by VAT (Visual, Auditory, Tactual feedback), participants completed four tasks by using the complete multimodal interaction style. The four consecutive tasks were denoted by *task repetition*. In the other control condition, denoted by AT (Auditory, Tactual feedback), participants only worked with the interaction style without visual display for all four tasks; the monitor was physically removed. In the two experimental conditions, participants worked with both interaction styles, one after the other. In the experimental condition denoted by $\mathrm{VAT} \rightarrow \mathrm{AT}$, there was no visual display for the last two tasks. In the other experimental condition, denoted by $\mathrm{AT} \rightarrow \mathrm{VAT}$, this was reversed.

Test material and equipment

A music collection comprising 480 one-minute excerpts of jazz music recordings (MPEG-1 Part 2 Layer II 128 Kbps stereo) from 12 different music styles served as the test material. The interaction style was implemented on a PC, running under Windows 95. MPEG data was stored on the hard disk. Real-time MPEG decoding was done by software. Music was amplified by a mid-range audio amplifier (Philips Integrated Digital Amplifier DFA888) and played through a pair of high-quality loudspeakers (Philips 9818 multi-linear 4-way). A second PC was used to controlling the force feedback trackball (see also Chapter 7 and Appendix II).

Participants were seated in a comfortable chair in a sound studio, originally designed for speech recording. The visual display was a 17-inch colour monitor. They could adjust the audio volume to a preferred level. The IPO trackball was placed on a small table next to the chair, in such a way that the participant could control the trackball in a comfortable way.

The experimenter was seated in a control room next to the studio and was provided with an additional monitor linked up to the monitor of the participant, allowing real-time observation and supplementary data collection of the task progress.

Procedure

Twenty-four participants performed two experimental sessions on two separate days. They were randomly assigned to one of the four *conditions*. A Two-Alternative Forced Choice test (2AFC) (Green and Swets, 1966) and a specially devised set of simple target acquisition tasks was used in a familiarisation phase to let participants get accustomed to, respectively, the force feedback and the required fine motor skills to control the IPO force feedback trackball. Participants were instructed to place both the index and the middle finger in parallel near the top of the ball in order to provide a high level of ball movement control and to accurately feel the force feedback. This familiarisation phase took approximately 15 minutes. Subsequently, participants were informed about the music programming domain and could freely explore the interaction style (with or without a visual display) for three minutes. The second day, the session started immediately with a 3-minute exploration phase. The tasks were then executed.

The participants completed two music programming tasks during each session. At the end of each session, participants completed a version of the procedural knowledge questionnaire at a desk from which it was impossible to view the test equipment. The questionnaire was completed in a dialogue with the test supervisor. They were then asked to produce a drawing of the interaction style.

Participants

Half of the 24 participants (18 male, 6 female) were recruited by advertisements and got a fixed fee. The other half consisted of colleague researchers who participated voluntarily. Eight people had already participated in previous experiments on music programming. Two persons had participated in an experiment with the IPO force feedback trackball before. The average age of the participants was 28 (min.: 21, max.: 45). Participants were not selected based on their musical preferences or musical education. All participants had at least completed a higher vocational education.

8.3.4 Results

One participant, who was assigned to the AT → VAT condition, freely admitted having searched for preferred music when performing the first three tasks. Consequently, values for *time on task* and *number of actions* were three times as high as the mean values of other participants. As the task instruction was to program music as quickly as possible, without considering personal music preference, the data from this participant were excluded from the analyses concerning *time on task* and *number of actions*. For the other analyses, the data from this participant were considered as being still valid and were left unchanged.

Number of actions

The results of *number of actions* are shown in Figure 8.1. In order to compare the differences in *number of actions* between working with a visual display and working without a visual display (see Hypothesis (i)), a new variable *visual display condition* was created. In this variable, the performance in the VAT and VAT → AT conditions was separated from the performance in the AT and AT → VAT conditions. Only *task repetition* 1 and 2 were included.

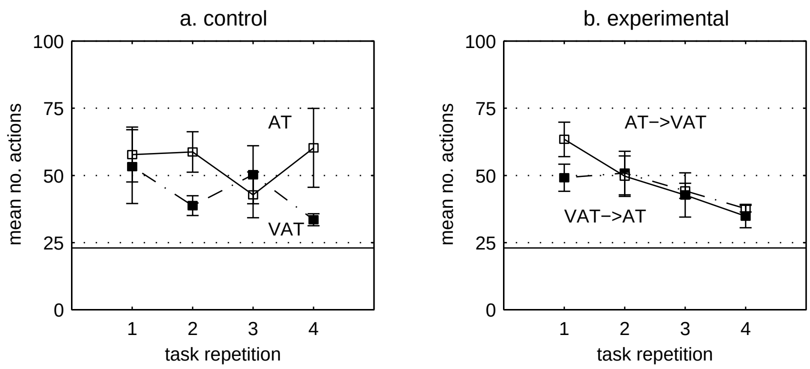

Figure 8.1. Mean *number of actions* performed. The left-hand panel (a) shows the mean *number of actions* for the four tasks in the AT and VAT control conditions. The right-hand panel (b) shows the mean *number of actions* for the four tasks in the AT → VAT and VAT → AT experimental conditions. The minimum number of actions to complete the task is 23 (horizontal line). The cross-bars represent the standard error of the mean.

An ANOVA with repeated measures was conducted in which *task repetition* (2) was a within-subject independent variable and the newly created *visual display condition* (2) was a between-subject independent variable. *Number of actions* was a dependent variable. No significant effects were found. The mean *number of actions* needed for the first two tasks with or without a visual display was 52.7. This result shows that there was no significant difference in task efficiency in terms of *number of actions* between working with a visual display and working without a visual display and between the first two tasks.

In order to compare *number of actions* across all four *task repetitions* in all four *conditions*, a MANOVA with repeated measures was conducted with *task repetition* (4) as a within-subject independent variable and *condition* (4) as a between-subject independent variable. *Number of actions* was a dependent variable. A significant main effect for *task repetition* was only found ($F(3,57) = 3.79$, $p < 0.05$). A linear trend in the data was significant ($F(1,19) = 7.92$, $p < 0.05$). Participants performed fewer numbers of actions for each successive task (mean *number of actions* across successive *task repetitions*: 55.7, 49.8, 44.6 and 40.0).

In order to assess the 'transfer' effect for *number of actions* (see Hypothesis (iii)), *transfer* was defined as a comparison between the *number of actions* performed by the eight participants doing *task repetition* 3 and 4 in the VAT → AT condition, and the *number of actions* performed by the eleven participants doing *task repetition* 1 and 2 in both the AT and AT → VAT conditions. In this way, the performance of participants who performed two tasks without a visual display, but who had done two previous tasks with a visual display, was compared to the performance of participants who started to perform two tasks without a visual display.

An ANOVA with repeated measures was conducted in which *transfer* (2) was treated as a between-subject independent variable, and *task repetition* (2) as a within-subject independent variable. *Number of actions* was a dependent variable. A main effect for *task repetition* was found to be just not significant ($F(1,17) = 4.11$, $p = 0.059$). Participants tended to perform fewer actions for the second and fourth task than for respectively the first and third task. The means for *number of actions* were 54.1 and 46.5, successively. A significant main effect for *transfer* was found ($F(1,17) = 8.81$, $p < 0.01$). It appeared that almost 17 fewer actions were required for participants who had done two previous tasks with a visual display, than for those who started to work without a visual display (mean *number of actions*: 57.2 for *task repetition* 1 and 2 in the AT and AT → VAT *condition*s, 40.9 for *task repetition* 3 and 4 in the VAT → AT *condition*). This suggests that performance when working without a visual display improved, when participants had done previous tasks with the visual display.

However, this performance improvement can also be caused by mere practice, as participants in the VAT → AT *condition* had done two more tasks than the participants in the AT and AT → VAT conditions. Therefore, the *number of actions* performed by the eight participants doing *task repetition* 3 and 4 in the VAT → AT *condition* was compared with the *number of actions* performed by the four participants doing *task repetition* 3 and 4 in the AT *condition*.

An ANOVA with repeated measures was conducted in which *task repetition* (3 and 4) was a within-subject independent variable and *condition* (AT and VAT → AT) was a between-subject independent variable. *Number of actions* was a dependent variable. There was no significant main effect for *condition* ($F(1,10) = 1.94$, $p = 0.19$). It appeared that the mean *number of actions* did not differ in the last two non-visual tasks, irrespective of whether participants had done two previous tasks with a visual display or not. Hence, performance improvement in terms of *number of actions* was more likely to be a result of practice, than a result of previous experience with the visual display. In addition, a significant *condition* by *task repetition* interaction effect was found ($F(1,10) = 8.35$, $p < 0.05$). Though participants performed almost the same *number of actions* in *task repetition* 3 (43.7 actions), the *number of action*s increased in *task repetition* 4 in the AT *condition* only (see also the left-hand panel of Figure 8.1). This interaction effect was caused by one participant in the AT condition who encountered difficulties in finding a particular music style on the roller while doing the fourth task. As a consequence, he needed twice as many actions (100 actions) to complete the task than the other three participant in the same condition (47.0 actions). No other effects were found.

Time on task

The results of *time on task* are shown in Figure 8.2. In order to compare the differences in *time on task* between working with a visual display and working without a visual display (see Hypothesis (i)), a new variable *visual display condition* was created. Similar to the analysis of *number of actions*, this variable separated the performance in the VAT and VAT → AT conditions from the performance in the AT and AT → VAT conditions, only for the first two *task repetition*s.

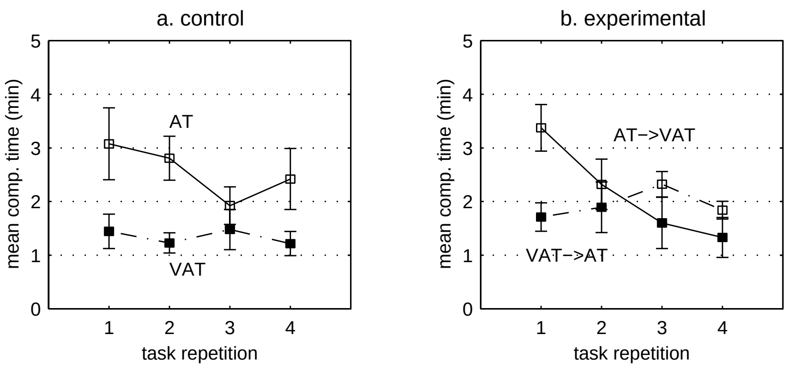

Figure 8.2. Mean *time on task* (minutes). The left-hand panel (a) shows the mean *time on task* for all four tasks of the AT and VAT control conditions. The right-hand panel (b) shows the mean *time on task* for all four tasks of the AT → VAT and VAT → AT experimental conditions. The cross-bars represent the standard error of the mean.

An ANOVA with repeated measures was conducted in which *task repetition* (2) was a within-subject independent variable and the newly created *visual display condition* (2) was a between-subject independent variable. *Time on task* was a dependent variable. A significant main effect for *task repetition* was found to be nearly significant ($F(1,21) = 4.31$, $p = 0.05$). Participants needed less time for the second task than for the first task (the means of *time on task* for *task repetition* 1 and 2 were, respectively, 2 min. 25 sec. and 2 min. 4 sec.). A significant main effect for *visual display condition* was found ($F(1,21) = 9.78$, $p < 0.01$). Participants who worked without a visual display needed almost twice as much time for the first two tasks, than participants who worked with a visual display (mean *time on task* for the first two *task repetitions*: 2 min. 53 sec. (without a visual display), 1 min. 36 sec. (with a visual display)). A significant *task repetition* by *visual display condition* interaction effect was found ($F(1,21) = 5.56$, $p < 0.05$). Participants who worked without a visual display spent 47 *fewer* seconds on the second task than on the first task (mean *time on task* in the first two *task repetition*s without a visual display were, successively, 3 min. 16 sec. and 2 min. 29 sec.). In contrast, participants who worked with a visual display needed the same amount of time for both tasks (mean *time on task* in the first two *task repetition*s with a visual display were, successively, 1 min. 37 sec. and 1 min. 40 sec.).

In addition, *time on task* was compared between the four participants in the AT and the VAT conditions for all four *task repetition*s. A MANOVA with repeated measures was conducted in which task repetition (4) was a within-subject independent variable and *condition* (AT and VAT) was a between-subject independent variable. *Time on task* was a dependent variable. A significant main effect for *condition* was found ($F(1,6) = 8.21$, $p < 0.05$). This test also confirmed that working without a visual display required more time than working with a visual display. In addition, a significant *condition* by *task repetition* interaction effect was found ($F(3,4) = 10.26$, $p < 0.05$). When comparing means, it was found that participants who worked without a visual display reduced their *time on task* for *task repetition* 3 significantly, in contrast to participants who worked with a visual display ($F(1,6) = 6.36$, $p < 0.05$). No other effects were found.

A MANOVA with repeated measures was used with *task repetition* (4) as a within-subject independent variable and *condition* (4) as a between-subject independent variable. *Time on task* was a dependent variable. A significant main effect for *task repetition* was found ($F(3,57) = 3.64$, $p < 0.05$). A linear trend in the data was significant ($F(1,19) = 5.97$, $p < 0.05$). This test also confirmed that participants needed less time to compile each successive music programme (mean *time on task* across successive tasks: 2 min. 25 sec., 2 min. 4 sec., 1 min. 53 sec. and 1 min. 41 sec.). A significant *task repetition* by *condition* interaction effect was also found ($F(9,57) = 3.26$, $p < 0.005$). When means were compared, it was found that the first task performance in the AT → VAT condition required significantly more time than the first task performances in other conditions ($F(3,19) = 4.19$, $p < 0.05$) (mean *time on task* at first task: 3 min. 23 sec. (AT → VAT), 1 min. 59 sec. (other conditions)).

Similar to the 'transfer' analysis of *number of actions*, *transfer* was defined as the comparison between *time on task* spent by the eight participants doing *task repetition* 3 and 4 in the VAT → AT *condition*, and *time on task* spent by the eleven participants doing *task repetition* 1 and 2 in the AT and AT → VAT *condition*.

An ANOVA with repeated measures was carried out in which *transfer* was treated as a between-subject independent variable, and *task repetition* (2) as a within-subject independent variable. *Time on task* was a dependent variable. A significant main effect for *task repetition* was found ($F(1,17) = 10.12$, $p < 0.01$). Participants needed less time to compile a programme for the second and fourth task than for respectively the first and third task. The means for *time on task* were 2 min. 52 sec. and 2 min. 13 sec., successively. A main effect for *transfer* was found to be just not significant ($F(1,17) = 4.20$, $p = 0.059$), however, participants who had worked with the visual display before needed 48 *fewer* seconds in a non-visual task, than participants who started to work without a visual display.

As performance improvement in terms of *time on task* can also be caused by mere practice, *time on task* was compared between the AT and VAT → AT condition for the task repetition 3 and 4.

An ANOVA with repeated measures was conducted in which *condition* (2) was a between-subject independent variable and *task repetition* (2) was a within-subject independent variable. *Time on task* was a dependent variable, in this case. No significant main effect for *condition* was found ($F(1,10) = 0.058$, $p = 0.815$). Performance improvement in terms of *time on task* was more likely to be caused by practice, than by pervious experience with the visual display. A significant interaction effect for *condition* and *task repetition* was found ($F(1,10) = 5.51$, $p < 0.05$). Participants in the AT condition spent less time (1 min. 56 sec.) than the participants in the VAT → AT condition (2 min. 19 sec.) in *task repetition* 3, while this was reversed in *task repetition* 4 (2 min. 25 sec. and 1 min. 50 sec., respectively). Also, this interaction effect was mainly caused by the participant in the AT condition who had difficulties in finding the right music style on the roller in *task repetition* 4. He needed 3 minutes and 56 seconds to complete this task, while the three others in the AT condition needed 1 minute and 55 seconds. No other effects were found.

Procedural knowledge

Data on *questionnaire score* in the four conditions were re-coded to create two different *visual display condition*s. The results are shown in Figure 8.3.

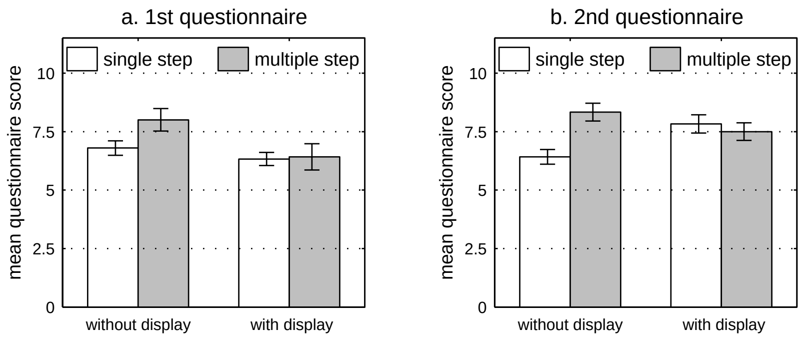

Figure 8.3. Mean *questionnaire score*. The left-hand panel (a) shows the mean *questionnaire score*, divided up into mean scores on answers concerning single step and multiple step interactions (maximum: 10 both), across visual display conditions at the end of the first experimental session. The right-hand panel (b) shows the mean *questionnaire score* at the end of the second experimental session. The cross-bars represent the standard error of the mean.

An ANOVA with repeated measures was used with *kind of question*, that is, single-step and multiple-step, as a within-subject independent variable, and *visual display condition* (2) and *experimental session* (2) as between-subject independent variables. *Questionnaire score* was a dependent variable. A significant main effect for *experimental session* was found ($F(1,44) = 5.96$, $p < 0.05$). Participants were better in completing the questionnaire for the second time (mean *questionnaire score* across successive sessions: 13.4, 15.0). A significant main effect for *kind of question* was also found ($F(1,44) = 17.76$, $p < 0.001$). Participants were better in answering the questions concerning the multiple step interactions than the questions concerning the single step interactions (mean *questionnaire score* for *kind of question*: 6.7 (single step), 7.6 (multiple step)). An interaction effect of *kind of question* by *visual display condition* was found ($F(1,44) = 23.07$, $p < 0.001$). Participants who had worked without a visual display were better in answering questions concerning multiple step interactions than participants who had worked with a visual display (mean *questionnaire score* concerning multiple step interactions for *visual display conditions*: 8.2 (without visual display), 7.0 (with visual display)).

Drawing

At the end of each experimental session, participants were instructed to produce a hand-drawing of the interaction style which would aid explanation to others who were new to the interaction style.

The drawings were analysed by judging whether or not seven aspects of the interaction style were present. The seven aspects were the following:

1 Actions (e.g., rolling, clicking and double-clicking the trackball);

2 Domain objects (e.g., ‘collection’, ‘programme’);

3 Force feedback (e.g., drawing a hole or ripple);

4 Relationships between domain objects (e.g., by drawing a structure diagram);

5 Physical devices (e.g., a trackball device, a monitor, an audio amplifier);

6 Non-speech sound (e.g., ’hearing the sound of a creaking door’);

7 Visual aspects (e.g., rollers or a graphical menu structure).

Drawings of four participants are shown in Figure 8.4. Some participants who had worked without a visual display were rather creative in developing their own metaphor (see Drawing A and B in Figure 8.4). In general, drawings differed qualitatively between participants who had worked with a visual display and participants who had worked without a visual display.

Part of the results are shown in Figure 8.5; only three aspects that were present in at least half of the drawings are shown. The other four aspects were rarely drawn. For instance, force feedback aspects were only drawn by some participants who had worked without a visual display, and a reference to non-speech audio was made only once. However, it is not obvious how one can draw aspects related to force fields or non-speech audio. It also appeared that drawing physical devices and relationships between domain objects rarely occurred; these two aspects were probably considered irrelevant to explain the interaction style.

As shown in Figure 8.5, participants in the AT and VAT conditions produced consistent drawings in both experimental sessions. This was not true for participants who received a change in visual display conditions, however, that is, in the $VAT \rightarrow AT$ and $AT \rightarrow VAT$ conditions.

Panel (a) of Figure 8.5 seems to show that participants in experimental conditions were rather inconsistent in drawing actions. However, a test on difference between non-independent proportions (McNemar, 1962; p. 52) showed that participants did not produce a change in drawing action-related aspects when experiencing a change in visual display conditions ($z = 0.63$). On the other hand, a Fisher exact probability test (Siegel, 1956; p. 96) revealed that participants who had worked without a visual display drew more action-related aspects in their first drawing than participants who had worked with a visual display ($p = 0.050$) (proportion of actions drawn in first drawing: 9/12 (without visual display), 4/12 (with visual display)).

A test on difference between non-independent proportions showed that participants changed their drawing of domain objects under a change of visual display conditions ($z = 3.46$, $p < 0.0005$). As shown in panel (b) in Figure 8.5, these participants tended to draw fewer domain objects in the second drawing (proportion of domain objects drawn: 16/16 (first drawing), 12/16 (second drawing)).

A test on difference between non-independent proportions revealed that participants changed their drawing of visual aspects under a change of visual display conditions ($z = 2.11$, $p < 0.05$). As shown in panel (c) in Figure 8.5, $VAT \rightarrow AT$ participants drew fewer visual aspects in the second drawing, whereas $AT \rightarrow VAT$ participants drew more. A Fisher exact probability test revealed that participants who had worked without a visual display produced fewer visual aspects in their first drawing ($p = 0.0014$) (proportion of visual aspects drawn in first drawing: 11 / 12 (with visual display), 3 / 12 (without visual display)).

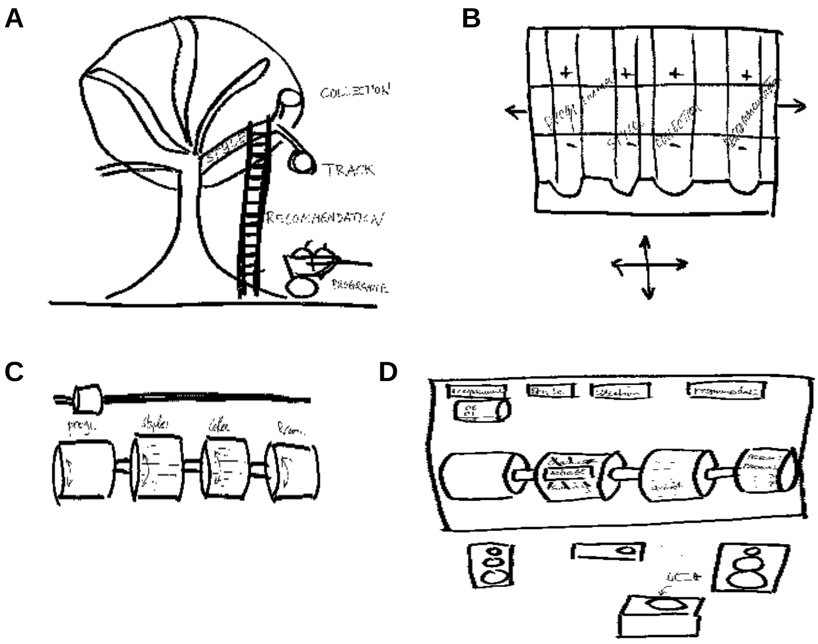

Figure 8.4. Drawings made by four different participants. Drawings A and B were made by participants, who had worked without a visual display. Drawing C and D were made by participants, who had worked with a visual display. The participant who drew drawing A explained the music collection as a fruit tree, in which the branches were music styles, and the fruit carried by these branches were music options within a music style. Compiling a programme was clarified as filling a barrow with fruit; one can take the difficult route by climbing the tree along the trunk and reaching for the fruit by shaking the branches, or one can decide to take the easy route of recommendations by climbing a ladder that directly leads to the fruit of interest. Only domain objects and their interrelations were judged to be present. The participant who drew drawing B explained the interaction style as a coin collector. Each slot in the collector, which could clearly be felt as such, was one of the concepts in the music programming domain. When in one of these slots, one can inspect and feel each coin, that is, each music option, one-by-one. Only domain objects, actions and force feedback were judged to be present. The participant who drew drawing C explained the roller metaphor accurately by telling how the rollers can be rolled, but without mentioning other output modalities. Only domain objects actions and visual aspects were judged to be present. The participant who drew drawing D also explained the roller metaphor accurately, but extended the explanation with references to the physical devices and how rolling movements of the trackball were aligned with the rotation of the visual rollers. Only domain objects, actions, physical devices and visual aspects were judged to be present.

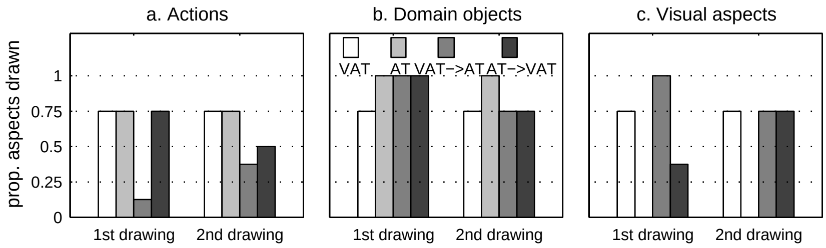

Figure 8.5. Proportion of aspects in the first and second drawing over the two control and two experimental conditions. Panel (a) shows the proportion of actions drawn. Panel (b) shows the proportion of domain objects drawn. Panel (c) shows the proportion of visual aspects drawn. Note that there were twice as many participants under experimental conditions as under control conditions.

8.3.5 Discussion

Participants were instructed to program music as efficiently as possible without considering personal music preferences using the multimodal interaction style with and without a visual display. All twenty-four participants were able to complete a music programming task successfully, right from the start, with or without a visual display. They were given a three-minute free exploration with the interaction style before the tasks started. They had received no procedural instruction on how to work with the interaction style.

According to Hypothesis (i), a music programming task should be performed more efficiently with a visual display than without a visual display. The results showed that participants who worked without a visual display needed significantly more time to complete the tasks (especially the first two tasks) than participants who worked with a visual display. Working without a visual display did not require the execution of significantly more actions. On the basis of these results, Hypothesis (i) cannot be rejected.

The results also showed that participants who worked without a visual display were steep learners; they needed increasingly less time for the second task and the third task. Without the assistance of a visual display, participants could run into local interaction problems such as finding targets during music programming (e.g., a music style on a roller). Obviously, this kind of problems also decreased task efficiency.

In order to assess what kinds of distinctive operation occurred during first-time use, task performance between the first visual task and the first non-visual task were compared on other measures. As participants who worked without a visual display needed more time, only, it is evident that programming without a visual display took more time per action (mean time per action: 3.2 sec. (without visual display), 1.9 sec. (with visual display); $t = -7.08$, $p < 0.001$). Further, more lateral trackball movements were performed in the first task without a visual display (mean number of lateral movements: 19.1 (without visual display), 7.9 (with visual display); $t = -3.11$, $p < 0.01$), whereas other types of actions on the trackball did not differ across visual display conditions. Lateral trackball movements were actions to hop from one roller to another and hence were suitable for exploring the spatial relationships between rollers. It is therefore likely that extra time and extra lateral ball movements are linked to acquisition of spatial and procedural knowledge when there is no visual display available. Acquisition of this knowledge is likely to happen by a time-consuming inspection by imagery of the current or new state of interaction. This state must be known to determine the next action to perform in a dead-reckoning process of action.

According to Hypothesis (ii), participants who have worked *without* a visual display should have a higher score on *procedural knowledge* than participants who have worked *with* a visual display. The results showed that participants who had worked without a visual display were better in answering questions about multiple-step interactions. On the basis of these results, Hypothesis (ii) cannot be rejected. Procedures had to be explicitly acquired and remembered when there was no visual display available.

According to Hypothesis (iii), participants who worked with a visual display in one condition and are subsequently transferred to another condition without a visual display should perform more efficiently than participants who start working without a visual display. The results showed that participants who had performed two preceding tasks with a visual display did not spend less time on further tasks without a visual display than participants who started to work without a visual display. They did perform significantly fewer actions. It was shown however that any improvement in task performance was caused by practice, primarily, and not by previous experience with a visual display. In other words, participants who started to work without a visual display performed less efficiently than participants who had earlier experience with a visual display, only because they had performed fewer music programming tasks. On the basis of these results, Hypothesis (iii) must be rejected.

According to Hypothesis (iv), participants who had worked with a visual display should draw a reproduction of the interaction style which contains more visual aspects, whereas those who had worked without a visual display should draw the interaction style to essentially contain more action-and-effect related aspects. The results showed that participants who had worked without a visual display made a drawing of the interaction style that contained fewer visual aspects related to the roller metaphor. These drawings contained more action-related aspects. For participants who had worked with a visual display, this was reversed. On the basis of these results, Hypothesis (iv) cannot be rejected.

In addition, qualitatively different drawings of the interaction style were made depending on whether participants were familiar with the visual metaphor or not. Once familiar with the metaphor, participants were favoured by this metaphor. When unfamiliar with the metaphor, participants tended to devise their own metaphor and analogy. In general, participants accurately reproduced the visual display of the interaction style, once they had worked with it.

8.4 CONCLUSION

The experimental evaluation has assessed the usability properties and the learning of procedures of a multimodal interaction style for music programming, in particular, on the presence or absence of a visual display combined with a tactual and auditory interface. Users were able to complete a given music programming task successfully with or without a visual display after three minutes of free exploration and without procedural instruction. Over time, they learned to perform the tasks more efficiently; less time and fewer actions were required for each successive task. As learnability is a fundamental usability criterion for interactive devices in a domestic context-of-use (Eggen, Westerink and Haakma, 1996), the observed instant usability is an encouraging property of the interaction style.

Operating the interaction style without a visual display does not impede the successful completion of a music programming task, but needs more time than operation with a visual display, in particular, during first-time use. When working without a visual display, additional time is required to acquire and remember actively procedural and spatial knowledge about the interaction space. Procedural and spatial knowledge are learnt quickly, however, as working without a visual display speeds up rapidly. In addition, knowledge about the visual display is not strictly necessary to speed up working without a visual display.

Participants who worked with both versions of the interaction style preferred a visual display, for convenience sake. Only one participant preferred the displayless interface; he commented that 'it contains all attractive features that a novel has and a movie picture lacks; one is free to make one's own interpretation and devise one's own world.' Though participants tend to favour what is on the screen when asked to make a drawing of the interaction style, it is still unclear whether or not a mental image of the visual display is used when working without a visual display.

Measuring procedural knowledge by means of a post-test questionnaire assumes that users learn to operate an interactive device explicitly, that is, that they are able to justify their actions and decisions while operating the device, and that they are able to verbalise the acquired knowledge about the device afterwards. However, the basic idea that users always learn explicitly is incorrect. Procedures of an interaction style are often learnt implicitly and incidentally. Although practice improves task performance, it does not necessarily lead to improved ability to answer questions on procedures. Likewise, procedural instructions improve the ability to answer procedural questions, but do not necessarily lead to task performance improvement (Berry and Broadbent, 1984; Berry and Dienes, 1993; Dienes and Fahey, 1998). It seems that implicit learning of procedures occurs, in particular, when all required information to act purposefully is presented on a visual display, such as in direct manipulation interfaces (Mayes, Draper, McGregor and Oatley, 1988).

In summary, it has been demonstrated that users who work with only tactual and auditory feedback are able to operate a new interaction style successfully, only less efficient in time. Tactual and auditory feedback makes interaction possible in contexts-of-use in which information on a visual display is poorly legible or even absent. Working without a visual information is not a commonly preferred method of operation, due to the need of explicitly acquiring and remembering procedures of operation. It is concluded however that the multimodal interaction style for music programming meets its requirements on *instant usability* and *optional use of a visual display*. If hand-held trackball devices with force feedback mature to become fully usable, low-cost input devices consuming only a little power, this interface to music programming may be a desirable feature on desktop computers, jukeboxes, portable players, remote controls, and car audio equipment, amongst others.

CHAPTER 9

Main conclusions

The central theme of this thesis was how future interactive music players can be of help to music listeners in selecting and programming preferred music using large personal music collections. As a starting point, two user requirements were proposed to ease selection in a wide assortment of music. The first requirement is that music players should adapt to the music choice behaviour of the listener, so that they can provide adequate assistance to ease and speed up the music selection process of the listener. The second requirement is that players should present information about music interactively, in such a way that it facilitates user navigation in a large music collection. Both requirements are part of the current trend toward the personalisation of interactive devices. These requirements has been carried out in the design and implementation of an adaptive functionality named PATS and its multimodal interaction style. Applied investigations and user experiments were conducted to support the research, including issues on performance and usability of PATS and its interaction style.

In this concluding chapter, major findings of the work carried out are presented first. Then, the characteristics of a music programming task are summarised, followed by some theoretical implications for choice theory and recommendations for future research. Next, the following subjects are discussed: general applicability of PATS in a frame work of other musical idioms, a fixed attribute representation for music, technological issues and recommendations for future research. Lastly, multimodal interaction is discussed referring to a consumer device context, some technological issues and presenting recommendations for future research.

9.1 Major findings

An adaptive system functionality called PATS (Personalised Automatic Track Selection) and a multimodal interaction style for PATS were implemented as reported in Chapter 4 and Chapter 7, respectively. Both systems were comprehensively evaluated in user tests to assess their desirable and undesirable properties.

PATS can be interpreted in two ways. PATS is essentially an automatic music compilation feature for the creation of varied music programmes that are preferred in a particular context-of-use. The creation of music programmes requires only minimal intervention from the music listener. Secondly, PATS can be used as a music recommender system, which suggests music options to a music listener while carrying out a music programming task. Music suggestions can be used as a supplement to a personally created programme or as a provoker of new ideas to continue a personal search.

The major findings in the user evaluation reported in Chapter 5 are that PATS is able to create music programmes that are highly preferred (reaching approximately 80% of preferred music in a programme in a fourth user session), that cover a varied and preferred portion of a music collection when created over time, and which tend to adapt to the music preference of a music listener.

The major finding in two user experiments reported in Chapter 6 is that PATS recommendations are a highly preferred feature for a music programming task, since they reduce search effort in a large music collection and they are found useful for a music programming task. The use of PATS recommendations does not lead to more efficient programming, when music listeners have ample time to program, and does not lead to better music programmes, when music listeners have limited time to program.

The most important finding in the user evaluation reported in Chapter 8 is that PATS' multimodal interaction style provides instant usability with and without using a visual display. Without procedural instructions, music listeners are able to complete a music programming task successfully, right from the start, though programming without a visual display takes more time. Music listeners, who work without a visual display, first have to discover and actively remember procedures. This takes time.

These desirable properties of both the PATS functionality and its multimodal interaction style makes them excellent assets for an interactive music player accessing a large music collection. This is all the more true since, as reported in Chapter 1, current CD jukebox players (one of the most likely applications for both features) provide ample opportunities for errors and inconvenient operation, all of which is very likely to negatively influence the full experience of music selection and listening.

9.2 CHARACTERISTICS OF MUSIC PROGRAMMING

Music programming deals with the achievement of a temporary goal, that is, the goal to find music that is suitable for a given context-of-use. This context-of-use is the real-world situation in which the music will be heard, including the current activities, the mood and the listening purposes of the music listeners. A context-of-use is supposed to produce constraints and opportunities for music listening, giving a first indication of a music listener's preference for music. This first indication may not be more than a couple of concrete examples of preferred music that the music listener is familiar with, such as music from which titles and main artists can be spontaneously recalled. This implies that a first indication of music preference is too incomplete to spontaneously fill a full-sized preferred music programme. In order to further develop music preference and to complete the programme, it is necessary to listen to musical cues of other options in the music collection.

As music preference is not fully developed at the start of a music programming task, most music targets to search for are poorly defined. Therefore, a common choice strategy that music listeners use is to first retrieve those familiar music options that are induced by the context-of-use, and then proceed in finding less familiar music options that suit the already selected options. In this second phase, music listeners are given cues of presented musical attribute information, which may trigger them to briefly listen to particular options at will. If music appeals to the listener, he or she can opt to listen further. This results in a selective music listening behaviour which is thus characteristic of music exploration at will and user control of music selection. Music listeners build a personal choice context which consists of the music options listened to. This choice context is essential to further develop music preference. It seems to be rather constant in size, irrespective of whether or not music listeners obtain outside assistance from a music recommender system. After the programme is complete, a concluding phase consists of determining the order of music options or still rejecting less preferred options in the music programme.

It is obvious that a large music collection makes a complete and thorough search impractical. The ample opportunities for music selection and listening implies that music listeners get distracted from the actual goal of a music programming task, that is, the making of a preferred music programme. Consequently, music listeners seem to be indifferent to task efficiency and less concerned with programme quality. In fact, a longer programming time may even be coupled with a greater enjoyment of the task. Therefore, it is plausible that programming features are not judged primarily on the task efficiency and the programme quality induced by them, but rather on the additional enjoyment and perceived usefulness they bring to the programming task. Obviously, this only holds true for programming features that are not too awkward to operate.

If music listeners are not able to listen to music at will, for instance, due to a limited time to complete a music programme, they will accelerate their information processing. This acceleration can be observed by the performance of more actions per unit of time and by listening for a shorter length of time to fewer music options than when programming time is unlimited. As coping with a time constraint and simultaneously performing a task cannot be resolved just by acceleration of information processing, programme quality decreases.

The use of a recommender system during music programming is appreciated as it reduces the search for music options. There is a danger, however, that a recommender system will put the listener off by a perceived loss of control of music selection. This loss can be observed by a less selective music listening behaviour of the listener. In other words, music listeners are then no longer able to listen to music at will.

9.2.1 Theoretical implications

Music programming is influenced by many psychological aspects. The resulting music programme has no optimal solution, has personal appeal, and is made up of multiple serial choices. In addition, music programming extends over time, consists of a search for poorly defined targets and is enjoyable. It may even be so enjoyable that music listeners forget about the apparent passage of time, which has a negative effect on the efficient performance of a music programming task.

In contrast to the psychological aspects of music programming, classical choice theory has traditionally been more concerned with how people ought to make choices rationally to yield optimal choice quality. The theory was less concerned with how people actually choose while simultaneously maximising choice quality and minimising the required time and effort to choose. Happily, research on choice theory has moved on to embrace the psychological point of view. This new approach gives valuable insight into areas such as human cognitive processes in choice strategies, especially those dealing with information overload and limited time to choose.

Choice theory is still largely focused on well-structured, one-choice problems and well-defined artificial targets. These properties inadequately fit real-world choice tasks, such as music programming, that require a dynamic search to multiple, poorly defined targets in a large music collection.

9.2.2 Future research

The given description of music programming has loose ends to be tidied up by future research. For instance, the development of a more explanatory model for music programming needs to be further pursued. To achieve this, music programming needs to be seen as a combined or phased use of choice strategies, whose effort of use and resulting programme quality are affected by factors such as the interaction style used, the size of the music collection, the similarity of music options, programming time allowed and programme size. Fundamental user experiments in which these factors are manipulated need to be carried out to construct an explanatory model for music programming. As a desirable consequence, this model will provide well-defined requirements for interaction style design, concerning adaptation to music choice behaviour and interactive information presentation.

In particular, the building of a personal choice context seems to be key to music programming; listeners need to evaluate a set of music options before they can select the preferred ones from this set. If the size of this choice context can be decreased or a personal choice context can be directly accessed using interactive and adaptive means, search effort will be substantially reduced.

Information overload is of topical interest. It leads people to make bad and inconsistent choices. In practice, they have to learn to process information selectively to avoid irrelevant information and to make better choices. It is unclear to what extent selective information processing accounts for the extraction of relevant information, however, and to what extent this interferes with irrelevant information (Payne, Bettman and Johnson, 1993). As selective information processing also hinges on user control of choice, it is a valuable dependent variable to study in the context of adaptation and interactive information presentation.

Information presentation, in its own right, is also a relevant topic to be pursued further, since it influences choice behaviour and quality (Payne, Bettman and Luce, 1998). People tend to base their decisions on information in the static way it is presented, without making any effort to transform this information into a more convenient format or to search for additional information. By moving from a static form of information presentation to an interactive and adaptive form of information presentation, people may be able to make better choices.

9.3 General applicability of PATS

As motivated in Chapter 1, jazz music was chosen as PATS' test vehicle because of the 'timeless' character of jazz and the variety of well-defined jazz styles. The experimental music collection covered a considerable part of the whole jazz period and a wide variety of musical attribute values (e.g., different musicians, composers and song titles).

A first step in finding out whether PATS is generally applicable is the question of whether or not it is useful for a jazz music collection that is different to, or extends on, the one used in the experiments. One of the main prerequisites for success is that the variety of attribute values in a music collection should not be too wide, since inferring relevant attribute values for music preference from little information is difficult and PATS compilation strategy is based on clustering music with common attribute values. However, enlarging a music collection makes a functionality for automatic music compilation even more desirable. It is therefore concluded that PATS will be useful for other jazz music collections, as long as the variety of attribute values in the music collection is not too wide.

The next step in finding out whether PATS is generally applicable is the question of whether or not it is useful in musical genres beyond jazz music. As PATS does *not* reason over an encoded knowledge representation for jazz music, in principle, it requires only an attribute representation for each new musical genre. Jazz music attributes have many similarities with attributes of other popular musical genres, so it is assumed that most attributes can be kept for these genres, though other attribute values would have to be filled in. The requirements for coherence and variation of music programmes is also likely to be appreciated by listeners to other popular genres. PATS is thus likely to be applicable in other popular musical genres. Classical music is an entirely different musical idiom with another type of listeners, however. It remains to be seen whether PATS is applicable in the classical music genre.

The last step in finding out whether PATS is generally applicable is the question of whether or not it is useful outside the laboratory and in the home with a home music collection. A home music collection is likely to cover the musical tastes of the listener. We can infer that the variety of attribute values in a home music collection is limited, as music listeners collect their music selectively to address their own musical tastes. This provides opportunities for the home applicability of PATS. It is true, however, that long-term and casual home usage will induce more erratic music preferences in poorly defined contexts-of-use than in the controlled experimental situations. In short, the issue of casual home usage needs to be further investigated.

9.3.1 Attribute representation of jazz music

Research has been focused on how musical features can best be represented by a fixed attribute representation to capture music preference using inductive learning algorithms. To a large extent, only musical features that are general and can be made objective could be included as an attribute. The attribute representation for jazz music was sufficiently capable of capturing music preference for research purposes. It may fall short of enabling the expression of all governing rules for music preference, however, as this requires a complete knowledge on musical features. The following comments can therefore be made.

> Mismatches between what is specified (or what is lacking) in the attribute representation and what is expected by music listeners may still occur. Therefore, there is an urgent need to extend on the attribute representation, especially to capture more erratic music preferences outside the laboratory and in the home.

A music preference categorisation task for inductive learning algorithms is difficult. Firstly, if universally governing rules exist for music preference, the attribute representation is inadequate to express these rules. Secondly, the real-world domain of (jazz) music is poorly structured covering many styles, many musicians, many composers and many different types of instrumentation. Thirdly, in the context of an interactive device, inductive learning algorithms can base their findings on a limited set of music options that is judged by the music listener (e.g., a music programme). This limited set does not contain sufficient attribute information to find rules capable of correctly categorising all music options in the music collection.

The class of inductive learning algorithms used, that is, top-down decision tree construction, capitalises on finding differences between preferred options and rejected options. This leads to loss of information about what either preferred music or rejected music has in common, when considered separately. Other classes of inductive learning algorithms such as single-category learners are more concerned with the commonalities within a subsumed category.

9.3.2 Technological issues

The current PATS implementation is a high-end demonstrator, designed and implemented for experimental purposes. PATS needs to be down-scaled if it must run on a current resource-constrained architecture such as a music player for the consumer market. For instance, memory usage of the current PATS implementation is proportional to the number of music options in the music collection, which is a too high cost for current resource-constrained technology. In short, the issue of down-scaling needs to be further investigated.

The PATS implementation currently runs on a UNIX platform, but can be easily ported to a PC Windows platform. A development environment and a virtual machine for the RTA language, in which PATS is partly implemented, exist for both platforms. This provides opportunities to incorporate PATS in the popular MP3 jukebox players to support music listeners who collect music computer files by ‘surfing’ on the Internet or by extracting music from other sources, but also to do long-term and casual use experimentation.

Technological standards

The problem of navigation and search in huge amounts of multimedia data is readily recognised and acknowledged by informal and formal standardization bodies and industry partners. Approved and emerging standards reflect a trend in which multimedia data such as audio and video is decomposed into its constituents that are tagged with descriptive attributes. These developments provide opportunities for the applicability of PATS.

An informal and open standard on attaching attribute data on MP3 (MPEG 1 Part 2 Layer III) audio streams and files is ID3v2 (Nilsson, 1999). The ID3v2 tag can hold any kind of data such as title, artist, album, style, year, tempo, musical key and (synchronised) lyrics of the music. It is already popular in the MP3 Internet community and is already used by MP3 jukebox players. Its Internet popularity implies that many music listeners currently tag their music files with attribute data and make this data available to others on the Internet, thus largely relieving each individual from tagging all his / her music by him / herself.

The formal standards are developed by the Moving Picture Experts Group (MPEG), a committee within the ISO / IEC standardization organization consisting of researchers and engineers from all over the world. The approved MPEG-4 standard (Koenen, 1999) employs a standard unit of representation for aural, visual, and audiovisual components in the content. The components are called *media objects* that are synchronized in time and place, which makes each component independent from the audiovisual scene in which it is contained. The MPEG-7 (1998) standard, which is still in the process of development, extends on the MPEG-4 standard by attaching descriptions to the media objects. These descriptions will rely on a standard set of descriptive features (Descriptors) and feature structures (Descriptor Schemes) that are imposed on the features to define higher levels of abstraction. Other types of information about the real-world context such as from which the content originates, links to other content, and content classification will also be included in the description. The lower-level Descriptors resemble perceptual attributes of the content. For audio material, typical features are key, onset, decay, tempo, tempo changes, and position of phenomena in sound. It is expected that most of these features will be able to be extracted automatically or semi-automatically.

9.3.3 Future research

Besides an extension of the attribute representation and the research on the use of other inductive learning algorithms, as discussed in Section 9.3.1, future research should be focused on solving some properties of PATS, as an automatic music compilation functionality and as a music recommender system.

PATS: An automatic music compilation functionality

As reported in Chapter 5, the results of both the computer simulation study and the user experiment showed that a high level of *precision* and *coverage* is hard to achieve simultaneously. Similarly, Information Retrieval (IR) systems are often limited by a trade-off between the *precision* and *recall* measures (Salton, 1989). In order to create varied music programmes over time, PATS tends to exchange the collection of preferred music that has already been listened to in previous programmes by newly found, but similar, music for which the music listener has not yet expressed a preference. Since it is not known beforehand whether the new music would be appreciated by the listener or not, the level of variation applied may be inappropriate. It may therefore be desirable to devise a mechanism in which the levels of *precision* and *coverage* of successively created music programmes can be controlled adequately.

PATS: A music recommender system

A music recommender system excels at providing appropriate music only by learning the music preferences of the music listener. Among other things, therefore, it requires preference feedback in some kind of form from the listener. In the experiments in Chapter 6, it appeared that control elements related to preference feedback were used sparingly, which limited system learning. In general, users are unaware of the possible consequences when they feed back their preference, or find it inconvenient and separate from searching and experiencing content. Future research has to be focused on how to explicitly elicit preference feedback from users, or how to monitor user navigation behaviour from which preference feedback can be derived implicitly.

As reported in Chapter 6, if a music recommender system is not adequately integrated in the progress of a music programming task, it can put off the listener by limiting music listening and selection at will. More has to be learnt about the most effective, efficient and pleasing way in which a system should initiate, present and implement the presentation of music recommendations, and how a user should have control and take action.

9.4 Multimodal interaction styles

Multimodality is currently a popular way forward in improving the usability of home entertainment systems (Bongers, Eggen, Keyson and Pauws, 1997; 1998). As an alternative or supplement to the visual and manual modality, the application of voice control, speech output, non-speech audio feedback and tactile / tactual feedback is explored in all its combinations. It provides interesting opportunities in a consumer device context, if only because of the other ways of presenting information, an extension on user imagery while interacting, or a provision of a higher perceived level of user control.

In this thesis, it has been shown that the combination of tactual and auditory feedback, in particular, can reduce the need for visual inspection, though at the expense of a higher cognitive load to discover and actively remember procedures. This brings potential gains in situations where the visual modality is already heavily under demand, visual inspection is less desirable, or visual inspection is impossible.

9.4.1 Technological issues

Similar to PATS, its multimodal interaction style is a high-end demonstrator, designed and implemented for experimental purposes. The current implementation of the interaction style uses a prototype force feedback trackball with specialised hardware, additional power supply and an additional PC. Obviously, its current technology is not suited for home use. However, low-cost hand-held prototype devices with a lower power consumption are already derived from this technology, which makes it more feasible to incorporate the interaction style in present consumer electronics.

9.4.2 Future research

The apparent trend of developing multimodal interaction styles has not yet been matched by systematic studies on their usability. For instance, the extent to which multimodal interaction styles are more usable than their unimodal variants is not known. Secondly, if multimodality is supposed to reduce the need for visual inspection, what reduction can outweigh an increase of cognitive load for learning and remembering procedures? Thirdly, research on cross-modality limits is typically focused on response times and accuracy in simple tasks, and they have never been studied in the broader context of a multimodal interaction style. All three issues should be high on the research agenda for multimodal interaction styles. Research on these issues is all the more important since the assumed usability of multimodal interaction is largely motivated by the assumption that people can freely put in attentional resources for each modality; there is as yet no uniform empirical evidence to support this.

In contrast to other engineering disciplines, multimodal interaction style design practice generally lacks the use of a design methodology, though it is embedded in a multi-disciplinary engineering context and covers new design fields such as tactual and sound design. User requirements are usually not specified at a concrete level from which requirements for design can be distilled. A design methodology should repair this imperfection by allowing active input on user desires, problems and needs, and formative user testing. It needs to be introduced early, continued and translated in the design process, and to be communicated to other engineering practices involved. In addition, a methodology should allow the extraction of design patterns and procedures that are explicitly described at a higher level than a set of design heuristics. These design patterns can then be re-used for other applications or by other designers.

References

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.

Agre, P.E., and Chapman, D. (1987). Pengi: An implementation of a theory of activity. In: Forbus, K., and Shrobe, H. (Eds.), *Proceedings of the Sixth National Conference on Artificial Intelligence, AAAI-87, Volume 1, Seattle, Washington, USA, July 13-17, 1987,* Los Altos, CA: Morgan Kaufmann, 268-272.

Agre, P.E., and Chapman, D. (1990). What are plans for?, *Robotics and autonomous systems, 6,* 17-34.

Akamutsu, M. (1991). The influence of combined visual and tactile information on finger and eye movements during shape tracing, *Ergonomics, 35*, 647-660.

Akamatsu, A., and Sato. S. (1994). A multi-modal mouse with tactile and force feedback, *International Journal of Human-Computer Studies, 40,* 443-453.

Akamatsu, A., and MacKenzie, S.I. (1996). Movement characteristics using a mouse with tactile and force feedback, *International Journal of Human-Computer Studies, 44,* 483-493.

Akamatsu, M., MacKenzie, S.I., and Hasbrouc, T. (1995). A comparison of tactile, auditory, and visual feedback in a pointing task using a mouse-type device. *Ergonomics, 38,* 816-827.

Alpert, J. (1982). The effect of disc jockey, peer, and music teacher approval of music on music selection and preference, *Journal of Research in Music Education, 30, 3,* 173-186.

Anderson, J.R. (1983). *The architecture of cognition,* Cambridge, MA: Harvard University Press.

Balabanovic, M. (1998). *Learning to surf: Multiagent systems for adaptive web page recommendation.* Doctoral Thesis, Stanford University.

Balabanovic, M. (1998). Exploring versus exploiting when learning user models for text recommendation, *User Modeling and User-Adapted Interaction, 8, 1/2,* 71-102.

Balabanovic, M., and Shoham, Y. (1997). Content-based collaborative recommendation. *Communication of the ACM, 40, 3,* 66-72.

Bargen, B., and Donnelly, P. (1998). *Inside DirectX*. Redmond, WA: Microsoft Press.

Barsalou, L.W. (1983). Ad hoc categories. *Memory & Cognition*, 11, 3, 211-227.

Barsalou, L.W. (1991). Deriving categories to achieve goals. In: Bower, G.H. (Ed.), *The psychology of learning and motivation: Advances in research and theory, 27,* New York: Academic Press, 1-64.

Beckler, S.J., Allen, R.B., and Koneçni, V.L (1985). Mood-optimizing strategies in aesthetic-choice behavior. *Music Perception, 2, 4,* 459-470.

Belkin, N.J., and Croft, B.W. (1992). Information filtering and information retrieval: Two sides of the same coin?, *Communications of the ACM, 35, 12,* 29-38.

Ben Zur, H., and Breznitz, S.J. (1981). The effects of time pressure on risky choice behavior. *Acta Psychologica, 47*, 89-104.

Berliner, P. (1994). *Thinking in Jazz: The infinite art of improvisation*. Chicago: University of Chicago Press.

Berry, D.C., and Broadbent, (1984). On the relationship between task performance and associated verbalizable knowledge. In: Henderson, L. (Ed.), *The Quarterly Journal of Experimental Psychology, Section A, Human Experimental Psychology, 36A, 2,* London: Lawrence Erlbaum, 209-231.

Berry, D.C., and Dienes, Z. (1993). *Implicit learning: Theoretical and empirical issues.* Hove: Lawrence Erlbaum.

Biggs, S.F., Bedard, J.C., Gaber, G.B., and Linsmeier, T.J. (1985). The effects of task size and similarity on the decision behavior of bank loan officers. *Management Science, 31,* 970-987.

Blumer, A., Ehrenfeucht, A., Haussler, D., and Warmuth, M. (1987). Occam's razor. *Information Processing Letters, 24,* 377-380.

Bockenholt, U., Albert, D., Aschenbrenner, M., and Schmalhofer, F. (1991). The effects of attractiveness, dominance, and attribute differences on information acquisition in multiattribute binary choice. *Organizational Behavior and Human Decision Processes, 49,* 258-281.

Bongers, A.J., Eggen, J.H., Keyson, D.V., and Pauws, S.C. (1997). Multimodal interaction styles. In: Thomas, P., and Withers, D. (Eds.), *HCI-97 Conference Companion: Adjunct Proceedings of the 12th British Computer Society Annual Conference on Human Computer Interaction, Bristol, UK, August 12-15, 1997,* 30-33.

Bongers, A.J., Eggen, J.H., Keyson, D.V., and Pauws, S.C. (1998). Multimodal interaction styles. *HCI Letters, 1, 1,* 3-5.

Boulter, L.R. (1977). Attention and reaction times to signals of uncertain modality. *Journal of Experimental Psychology: Human Perception and Performance, 3,* 379-388.

Bouwhuis, D.G., and Bunt, H. (1994). Dialogue systems and interactive literacy instruction. In: Verhoeven, L. (Ed.), *Functional literacy: Theoretical issues and educational implications,* Amsterdam: John Benjamins Publishing Company, 371-385.

de Bra, P., Houben, G.-J., Kornatzky, Y., and Post, R. (1994). Information retrieval in distributed hypertexts. In: Funck Brentano, J.-L., and Seitz, F. (Eds.), *Conference Proceedings RIAO 94, Intelligent Multimedia Information Retrieval Systems and Management, Rockefeller University, New York, NY, USA, October 11-13, 1994,* Paris: CID-CASIS, 481-491.

Breiman, L., Friedman, J.H., Olshen, R.A., and Stone, C.J. (1984). *Classification and regression trees,* Belmont: Wadsworth.

Brooks, R.A. (1986). A robust layered control system for a mobile robot. *IEEE Journal of Robotics and Automation, 2,* 14-23.

Brooks, R.A., and Maes, P. (Eds.) (1994). *Artificial life IV. Proceedings of the Fourth International Workshop on the Synthesis and Simulation of Living Systems, Cambridge, Massachussetts, USA, July 6-8, 1994,* Cambridge, MA: MIT Press.

de Bruin, H., de Ruyter, B.E.R., and de Vet, J.H.M. (1996). Reusable interaction styles in consumer electronics, *IPO Annual Progress Report, 31,* Eindhoven University of Technology, 21-29.

Bruner, G.C. (1990). Music, mood and marketing. *Journal of Marketing, 54, 4,* 94-104.

Bunt, L. (1997) Clinical and therapeutic issues of music. In: Hargreaves, D.J., and North, A.C. (Eds.), *The social psychology of music,* Oxford: Oxford University Press, 249-267.

Buxton, W. (1983). Lexical and pragmatic considerations of input structures. *Computer Graphics, 17, 1,* 31-37.

Buxton, W., Gaver, W., and Bly, S. (1992). *The use of non-speech audio at the interface.* Cambridge: Cambridge University Press.

Cantor, J.R., and Zillmann, D. (1973). The effect of affective state and emotional arousal on music appreciation. *The Journal of General Psychology, 89,* 97-108.

Chapman, D. (1991). *Vision, Instruction and Action.* Cambridge, MA: MIT Press.

Chen, H. (1995). Machine Learning for information retrieval: Neural networks, symbolic learning and genetic algorithms. *Journal of the Society for Information Science, 46, 3,* 194-216.

Cheng, J., Fayyad, U.M., Irani, K.B., and Qian, Z. (1988). Improved decision trees: A generalized version of ID3. In: Laird, J. (Ed.), *Proceedings of the Fifth International Conference on Machine Learning, Ann Arbor, Michigan, USA, June 12-14, 1988,* San Mateo, CA: Morgan Kaufmann, 100-106.

Chester, J.P. (Ed.) (1998). Towards a human information society - People issues in the implementation of the EU Framework V Programme. *USINACTS Project Consortium.* Loughborough: HUSAT Research Institute.

Coker, J. (1964). *Improvising Jazz.* Englewood Cliffs, NL: Prentice-Hall.

Collier, G.L., and Collier, J.L. (1994). An exploration of the use of tempo in jazz. *Music Perception, 11, 3,* 219-242.

Conover, W.J. (1971). *Practical nonparametric statistics.* Chichester: Wiley.

Cook, R., and Morton, B. (1994). *The Penguin Guide to Jazz on CD, LP & Cassette, New Edition,* London: Penguin.

Connah, D.M., and Wavish, P. (1990). An experiment in cooperation. In: Demazeau, Y., and Muller, J.-P. (Eds.), *Proceedings of the 1st European Workshop on Modeling an Autonomous Agent in a Multi-Agent World (MAAMAW), Cambridge, England, August 16-18, 1989,* Amsterdam: North-Holland, 197-212.

Consumer Reports (1991). CD changers, *Consumer Reports, 56, 3, March 1991*, 166-167.

Consumer Reports (1997). Jukebox time? *Consumer Reports, 62, 2, February 1997*, 34-38.

Consumer Reports (1999). Disc jockeys. *Consumer Reports, 64, 2, February 1999*, 26-29.

Csikszentmihalyi, M., and Csikszentmihalyi, I.S. (1988). *Optimal experience: Psychological studies of flow in consciousness*, Cambridge: Cambridge University Press.

Cupchik, G.C., Rickert, M., and Mendelson, J. (1982). Similarity and preference judgements of musical stimuli. *Scandinavian Journal of Psychology, 23*, 273-282.

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

Deneubourg, J.L., Goss, S., Franks, N., Sendova-Franks, A., Detrain, C., and Chretien, L. (1994). The dynamics of collective sorting of robot-like ants and ant-like robots, In: Meyer, J.-A. and Wilson, S.W. (Eds.), *From Animals to Animats 3, Proceedings of the 3rd Conference on Simulation of Adaptive Behaviour*, Paris: MIT Press, 356-363.

Denis, M. (1991). *Image and cognition*. New York: Harvester Wheatsheaf.

Denis, M. (1993). Visual images and models of described environments. In: Burger, D., Sperandio, J.C. (Eds.), *Non-Visual Human-Computer Interactions: Prospects of the visually handicapped: Proceedings of the INSERM-SETAA Conference Ministerre de la Recherche et de l'Espace, Paris, 29-30 March 1993*, London: John Libbey Eurotext, 3-12.

Dhar, R., and Simonson, I. (1992). The effect of the focus of comparison on consumer preferences. *Journal of Marketing Research, XXIX (November 1992),* 430-440.

Dienes, Z., and Fahey, R. (1998). The role of implicit memory in controlling a dynamic system. In: Berry, D.C. (Ed.), *The Quarterly Journal of Experimental Psychology, Section A: A human experimental psychology, 51A, 3,* Hants, UK: Psychology Press, 593-614.

Dietterich, T.G., and Michalski, R.S. (1983). A comparative review of selected methods for learning from examples. In: Michalski, R.S., Carbonell, J.G., and Mitchell, T.M. (Eds), *Machine Learning, an Artificial Intelligence Approach, Volume 2,* San Mateo, CA: Morgan Kaufmann, 41-81.

Dijkstra, E.W. (1971). Hierarchical ordering of sequential processes, *Acta Informatica, 1,* 115-138.

Douglas, S.A., and Mithal, A.K. (1997). *The ergonomics of computer pointing devices*. Berlin: Springer-Verlag.

Drogoul, A., and Ferber, J. (1994). Multi-agent simulation as a tool for studying emergent properties in societies, In: Gilbert, N., and Doran, J. (Eds.), *Simulating societies: the computer simulation of social phenomena,* London: UCL Press, 127-142.

Edland, A. (1994). Time pressure and the application of decision rules: Choices and judgments among multiattribute alternatives, *Scandinavian Journal of Psychology, 35, 3,* 281-291.

Eggen, J.H. (1995). Turn on the base, *Technical Note No. 3309*, Philips Research Laboratories Redhill, UK.

Eggen, J.H., and Haakma, R. (1992). Final report of DRUID project (phases 1 and 2). *IPO report no. 846*, Eindhoven University of Technology.

Eggen, J.H., Haakma, R., and Westerink, J.H.D.M. (1996). Layered Protocols: hands-on experience. *International Journal of Human-Computer Studies*, 44, 45-72.

Eggen, J.H., and Pauws, S.C. (1997). A user-oriented multimedia presentation system for multiple presentation items that each behave as an agent, *WO patent WO9733424*.

Einhorn, H.J., and Hogarth, R.M. (1975). Unit weighting schemes for decision making. *Organizational Behaviour and Human Performance, 13*, 171-192.

Engel, F.L., Goossens, P.H., and Haakma, R., (1994). Improved efficiency through I- and E-feedback: a trackball with contextual force feedback, *International Journal of Human-Computer Studies, 41,* 949-974.

Engel, F.L., Haakma, R., and van Itegem, J., (1990). Trackball with Force Feedback. Philips patent PH-N 13522, 1990.

Erlewine, M., Woodstra, C., and Bogdanov, V. (1994). *All Music Guide: the best CDs, LPs & tapes, 2nd Edition*, San Francisco, CA: Miller Freeman.

Estes, W.K. (1994). *Classification and Cognition,* Oxford: Oxford University Press.

ETSI (1998). Human Factors (HF); Human factors issues in Multimedia Information Retrieval Services (MIRS), *European Telecommunications Standards Institute (ETSI),EG 201, February, 1998*.

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.

Fayyad, U., and Irani, K. (1992). The attribute selection procedure in decision tree generation. In: Swartout, W., Rosenbloom, P., and Szolovits, P. (Eds.), *Proceedings of the Tenth National Conference on Artificial Intelligence, AAAI-92, San Jose, California, USA, July 12-16, 1992,* Menlo Park: AAAI Press / MIT Press, 104-110.

Fayyad, U.M, Piatetsky-Shapiro, G., Smyth, P., and Uthurusamy, R., (Eds.) (1996) *Advances in knowledge discovery and data mining*, London: AAAI Press / MIT Press.

Ferber, J. (1999). *Multi-agent systems: An introduction to distributed artificial intelligence*, Harlow, UK: Addison-Wesley.

Fishburn, P. (1991). Nontransitive preferences in decision theory. *Journal of Risk and Uncertainty, 4,* 113-124.

Fisher, D.H. (1987). Knowledge acquisition via incremental conceptual clustering. *Machine Learning, 2,* 139-172.

Foltz, P.W., and Dumais, T. (1992). Personalized information delivery: An analysis of information filtering techniques, *Communications of the ACM, 35, 12,* 51-60.

Freed, D. (1990). Auditory correlates of perceived mallet hardness for a set of recorded percussive sound events. *Journal of Acoustical Society of America, 87,* 311-322.

Friberg, A, and Sundström, A (1997). Preferred swing ratio in jazz as a function of tempo. *Quarterly Progress and Status Report (QPSR), 4, Kungl Tekniska Högskolan, KTH (Royal Institute of Technology), Department of Speech, Music and Hearing, Stockholm, Sweden,* 19-27.

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

Garbarino, E.C., and Edell, J.A. (1997). Cognitive effort, affect, and choice, *Journal of Consumer Research, 24, 2,* 147-158.

Garey, M.R., and Johnson, D.S. (1979). *Computers and intractability. A guide to the theory of NP-completeness.* New York: W.H. Freeman and Company.

Gaver, W. (1986). Auditory icons: Using sound in computer interfaces. *Human-Computer Interaction, 2,* 167-177.

Gaver, W. (1989). The SonicFinder: An interface that uses auditory icons. *Human-Computer Interaction, 4,* 67-94.

Gaver, W. (1993). What in the world do we hear? An ecological approach to auditory event perception. *Ecological Psychology, 5(4),* 285-313.

Gaver, W. (1997). Auditory Interfaces. In: Helander, M.G., Landauer, T.K., and Prabhu, P.V. (Eds.), *Handbook of Human-Computer Interaction, 2nd edition*, Amsterdam: North-Holland, 1003-1041.

van de Geer, J.P. (1993a). *Analysis of Categorical Data: Theory*. London: SAGE.

van de Geer, J.P. (1993b). *Analysis of Categorical Data: Applications*. London: SAGE.

Gentner, G. (1992). Interfaces for learning: Motivation and the locus of control. In: Engel, F.L., Bouwhuis, D.G., Bösser, T., and d'Ydewalle, G. (Eds.), *Cognitive modelling and interactive environments in language learning: proceedings of the NATO advanced research workshop, Mierlo, 1990*, (Volume F87 of NATO ASI Series), Berlin: Springer-Verlag, 227-237.

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

Geringer, J.M., and McManus, D. (1979). A survey of musical taste in relationship to age and musical training, *College Music Symposium, 19*, 69-76.

Gibson, J.J. (1979). *The ecological approach to visual perception.* New York: Houghton Mifflin.

GIFI, A. (1985). PRINCALS user's guide, *Report UG-85-03*, Leiden, the Netherlands: University of Leiden, Faculty of Social and Behaviorual Sciences, Data Theory Group.

Goldstone, R.L. (1994). The role of similarity in categorization: providing a groundwork. *Cognition, 52,* 125-157.

Gourdol, A., Nigay, L., Salber, D, and Coutaz, J. (1992). Two case studies of software architectures for multimodal interactive systems: VoicePaint and Voice-enabled Graphical NoteBook, In: Larson, J., and Unger, C. (Eds.), *Engineering for Human-Computer Interaction,* Amsterdam: North-Holland, 271-283.

Graham, M., and Wavish, P. (1991). Simulating and implementing agents and multiple agent systems, In: Moskilde, E. (Ed.), *Proceedings of the 1991 European Simulation Multi-Conference, Copenhagen, Denmark, June 17-19, 1991,* SCS, 226-231.

Green, D.M., and Swets, J.A. (1966). *Signal detection theory and psychophysics,* New York: Wiley.

Grether, D.M., Schwartz, A., and de Wilde, L.L. (1986). The irrelevance of information overload: An analysis of search and disclosure. *Southern California Law Review, 59,* 277-303.

Gronow, P. (1989). Sound recording. *International Encyclopedia of Communication.* New York: Oxford University Press.

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

Hansson, R.D., Keating, J.P., and Terry, C. (1974). The effects of mandatory time limits in the voting booth on liberal-conservative voting patterns, *Journal of Applied Social Psychology, 4,* 336-342.

Hargreaves, D.J., and North, A.C. (Eds.) (1997). *The social psychology of music.* Oxford: Oxford University Press.

Hauptmann, A.G., and McAvinney, P. (1993). Gestures with speech for graphic manipulation, *International Journal on Man-Machine Studies, 38,* 231-249.

Hermes, D.J. (1998). Auditory material perception. *IPO Annual Progress Report, 33,* Eindhoven University of Technology, 95-102.

Hill, W., Stead, L., Rosenstein, M., and Furnas, G. (1995). Recommending and evaluating choices in a virtual community of use. In: Katz, I., Mack, R., and Marks, L. (Eds.), *Conference proceeding of the CHI-95 Conference, volume I, May 7-11, Denver, Colarado, USA,* 194-201, Denver, CO: ACM.

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

Holsheimer, M., Siebes, A.P.J.M. (1994). Data mining: the search for knowledge in databases, *Report CS-R9406, Centrum voor Wiskunde en Informatica,* Amsterdam.

Houston, D.A., Sherman, S.J., and Baker, S.M. (1989). The influence of unique features and direction of comparison on preferences. *Journal of Experimental Social Psychology, 25,* 121-141.

Hutchins, E. (1989). Metaphors for Interface Design. In: Taylor, M.M., Néel, F., and Bouwhuis, D.G. (Eds.). *The Structure of Multimodal Dialogue,* Amsterdam: North-Holland, 11-28.

Hyafil, L. and Rivest, R.L. (1976). Constructing optimal binary decision trees is NP-complete. *Information Processing Letters, 5, 1,* 15-17.

ISO / IEC DIS 9126-1 (1991). *Information Technology - Software quality characteristics and metrics - Part 1: Quality characteristics and sub-characteristics.* International Organization for Standardization.

ISO / DIS 9241-11 (1998). *Ergonomic requirements for office work with visual display terminals (VDTs) - Part 11. Guidance on usability*. International Organization for Standardization.

ISO / IEC 14598-1 (1999). *Information technology - Software product evaluation - Part 1: General overview.* International Organization for Standardization.

Jagacinski, C.M. (1995). Distinguishing adding and averaging models in a personnel selection task: When missing information matters. *Organizational Behavior and Human Decision Processes, 61, 1,* 1-15.

Järvinen, T. (1995). Tonal hierarchies in jazz improvisation. *Music perception, 12, 4,* 415-437.

John, G.H., Kohavi, R., and Pfleger, K. (1994). Irrelevant features and the subset selection problem. In: Cohen, W.H. and Hirsh, H. (Eds.), *Proceedings of the Eleventh International Conference on Machine Learning (ICML-94),* San Francisco, CA: Morgan Kaufmann, 121-129.

Kamp, Y. (1997). Social insects with memory. *Biological Cybernetics, 77,* 447-452.

Katz, J.J., and Postal, P.M. (1964). *An integrated theory of linguistic descriptions.* Cambridge, MA: MIT Press.

Keller, K.L., and Staelin, R. (1987). Effects of quality and quantity of information on decision effectiveness. *Journal of Consumer Research, 14,* 200-213.

Keyson, D.V., (1996). *Touch in User Interface Navigation,* Doctoral Thesis, Eindhoven University of Technology.

Keyson, D.V. (1997). Dynamic cursor gain and tactual feedback in the capture of cursor movements. *Ergonomics, 40, 12,* 1287-1298.

Keyson, D.V., and Tang, H. (1995). TacTool: A tactile rapid prototyping tool for Visual Interfaces. In: Anzai, Y., Ogawa, K., and Mori, H. (Eds.), *Symbiosis of Human and Artifact, Proceedings of the Sixth International Conference on Human-Computer Interaction, Tokyo, Japan, July 9-14, 1995, Volume 1,* Amsterdam: Elsevier, 67-74.

Keyson, D.V., and van Stuivenberg, L. (1997). TacTool v2.0: An object-based multimodal interface design platform. In: Smith, M.J., Salvendy, G., and Koubek, R.J. (Eds.), *Advances in Human Factors/Ergonomics. Design of Computing Systems: Social and Ergonomic Considerations, Proceedings of HCI International '97, Volume 21B, San Francisco, CA, USA, August 24-27, 1997,* Amsterdam: Elsevier, 311-314.

Kieras, D.E., and Bovair, S. (1984). The role of a mental model in learning to operate a device. *Cognitive Science, 8,* 255-273.

Klatzky, R.L., Loomis, J.M., and Golledge, R.G. (1997). Encoding spatial representations through nonvisually guided locomotion: Test of human path integration. In: Medin, D.L. (Ed.), *The Psychology of Learning and Motivation: Advances in research and theory, 37,* 41-84, London: Academic Press.

Koenen, R. (Ed.) (1999). MPEG-4 Overview - (Seoul Version). *ISO/IEC 14496, ISO/IEC JTC1/ SC29/WG11, Seoul, South Korea, March, 1999.*

Kroemer, K.H.E., and Grandjean, E. (1997). *Fitting the task to the man: A textbook of occupational ergonomics. Fifth edition.* London: Taylor & Francis.

Krumhansl, C.L. (1990). *Cognitive foundation of musical pitch.* New York: Oxford University Press.

Krumhansl, C.L., and Kessler, E.J. (1982). Tracing the dynamic changes in perceived tonal organization in a spatial representation of musical keys. *Psychological Review, 89(4),* 334-368.

Krumhansl, C.L, and Shepard, R.N. (1979). Quantification of the hierarchy of tonal functions within a diatonic context. *Journal of Experimental Psychology: Human Perception & Perception, 5(4),* 579-594.

Kumin, D. (1994). Ch-ch-changers! *Stereo Review, 59,* 60-66.

Kuntz, P., and Snyers D. (1994). Emergent colonization and graph partitioning, In: Meyer, J.-A. and Wilson, S.W. (Eds.), *From Animals to Animats 3, Proceedings of the 3rd Conference on Simulation of Adaptive Behaviour,* Paris: MIT Press, 494-500.

Laing, D. (1996). *The economic importance of music in the European Union (Part One).* Music in Europe, European Music Office (EMO) and European Commission (DGX), September, 1996.

Laing, D. (1999). *The European music industry in 1997-1998.* European Music Observatory (EMOB), Initial report, January, 1999, see also Internet: www.imro.ie/emo.html.

Langley, P. (1996). *Elements of Machine Learning.* San Francisco, CA: Morgan Kaufmann.

Leblanc, A. (1982). An interactive theory of music preference. *Journal of Music Therapy, XIX, 1,* 28-45.

Levitt, H. (1970). Transformed up-down methods in psychoacoustics. *Journal for the Acoustical Society of America, 49, 2,* 467-477.

Lieberman, H. (1995). Letitzia: An assistent that assists web browsing. In: Mellish, C.S. (Ed.), *Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI-95), Volume 1, Montréal, Québec, Canada, August 20-25, 1995,* San Mateo, CA: Morgan Kaufmann, 924-929.

Lim, K.Y., and Long, J. (1994). *The MUSE method for usability engineering.* Cambridge, UK: Cambridge University Press.

Lodewijks, J.G.L.C. (1981). *Leerstofsequenties: van conceptueel netwerk naar cognitieve structuur.* Doctoral dissertation, Katholieke Hogeschool, Tilburg.

Loeb, S. (1992). Architecting personalized delivery of multimedia information, *Communications of the ACM, 35, 12,* 39-48.

Luce, R.D. (1959). *Individual choice behavior. A theoretical analysis,* New York: John Wiley & Sons.

Lumer E.D., and Faieta B. (1994). Diversity and adaptation in population of clustering ants, In: Meyer, J.-A. and Wilson, S.W. (Eds.), *From Animals to Animats 3, Proceedings of the 3rd Conference on Simulation of Adaptive Behaviour,* Paris: MIT Press, 501-508.

MacKenzie, I.S., Sellen, A., and Buxton, W. (1991). A comparison of input devices in elemental pointing and dragging tasks, In: Robertson, S.P., Olson, G.M., and Olson, J.S. (Eds.), *Conference proceedings of the CHI'91 Conference, New Orleans, Lousiana, USA, April 17-May 2, 1991,* New York: ACM, 161-166.

Mackinlay, J., Card, S.K., and Robertson, G.G. (1990). A semantic analysis of the design space of input devices. *Human-Computer Interaction, 5(2-3),* 145-190.

Malone, T.W. (1982). Heuristics for designing enjoyable user interfaces: Lessons from computer games, *Proceedings of the Conference on Human Factors in Computer Systems*, Gaithersburg, MD, March 15-17, 1982, Association of Computing Machinery. Reprinted in Thomas, J.C., and Schneider, M.S. (Eds.), *Human factors in computer systems,* Norwood, NJ: Ablex, 1984, 1-12.

Malone, T.W., Grant, K.R., Turbak, F.A., Brobst, S.A., and Cohen, M.D. (1987). Intelligent information sharing systems, *Communications of the ACM, 30, 5,* 309-402.

Martin, K.D., Scheirer, E.D., Vercoe, B.L. (1998). Music content analysis through models of audition. *Proceeding of ACM Multimedia 98. Workshop on Content Processing of Music for Multimedia Applications, Bristol UK, September 12, 1998*. see also Internet: sound.media.mit.edu / ~eds / paper.html.

Mayes, J.T., Draper, S.W., McGregor, A.M., and Oatley, K. (1988). Information flow in a user interface: The effect of experience and context on the recall of MacWrite screens. In: Jones, D.M., and Winder, R. (Eds.), *People and Computers IV*, Cambridge, UK: Cambridge University Press, 275-289.

McCarthy, J.F., and Anagnost, T.D. (1998). MusicFX: An arbiter of group preferences for computer supported collaborative workouts. In: Poltrock, S., and Grudin, J. (Eds.) *Conference Proceedings of the 1998 ACM Conference on Computer Supported Cooperative Work (CSCW 98), Seattle, Washington, USA, November 14-18, 1998,* New York: ACM Press.

McNemar, Q. (1962). *Psychological statistics. Third edition*. London: Wiley.

Medin, D.L., and Barsalou, L.W. (1987). Categorization processes and categorical perception. In: Harnad, S. (Ed.), *Categorical perception: The groundwork of cognition.* New York: Cambridge University Press.

Medin, D.L., Goldstone, R.L., and Gentner, D. (1993) Respects for similarity. *Psychological Review, 100,* 254-278.

Medin, D.L., Goldstone, R.L., and Markman, A.B. (1995). Comparison and choice: Relations between similarity processes and decision processes. *Psychonomic Bulletin & Review, 2 1,* 1-19.

Medin, D.L., and Smith, E.E. (1981). Strategies and classification learning. *Journal of Experimental Psychology: Learning, Memory, and Cognition, 7,* 241-253.

Menczer, F. (1997). ARACHNID: Adaptive Retrieval Agents Choosing Heuristic Neighborhoods for Information Discovery. In: Fisher, Jr, D.H. (Ed.), *Machine Learning: Proceedings of the Fourteenth International Conference (ICML-97), Nashville, Tennessee, USA, July 8-12, 1997,* San Francisco, CA: Morgan Kaufmann, 227-235.

Michalski, R.S., and Stepp, R.E. (1983). Learning from observation: Conceptual clustering, In: Michalski, R.S., Carbonell, J.G., and Mitchell, T.M. (Eds), *Machine Learning, an Artificial Intelligence Approach, Volume 2,* San Mateo, CA: Morgan Kaufmann, 331-363.

Mingers, J. (1989). An empirical comparison of selection measures for decision tree induction. *Machine Learning, 3,* 319-342.

Miller, J.P. (1960). Information input overload and psychopathology. *American Journal of Psychiatry, 116,* 695-704.

MPEG-7 (1998). MPEG-7 Context and objectives (version 10, Atlantic City), MPEG Requirements Group, *Doc. ISO/MPEG N2460, Atlantic City, USA, October 1998*.

Mosteller, F. (1951). Remarks on the method of paired comparisons: III. A test of significance for paired comparisons when equal standard deviations and equal correlations are assumed, *Psychometrika, 16,* 207-218.

Murphy, P.M., and Pazzani, M.J. (1994). Exploring the decision forest: An empirical investigation of Occam's razor in decision tree induction. *Journal of Artificial Intelligence Research, 1,* 257-275.

Murthy, S., and Salzberg, S. (1995). Lookahead and pathology in decision tree induction. In: Mellish, C.S. (Ed.), *Proceedings of the Fourteenth International Joint Conference on Artificial Intelligence (IJCAI-95), Volume 2, Montréal, Québec, Canada, August 20-25, 1995,* San Mateo, CA: Morgan Kaufmann, 1025-1031.

Mynatt, E.D. (1997). Transforming graphical interfaces into auditory interfaces for blind users, In: Oviatt, S., and Wahlster, W. (Eds.), *Human-Computer Interaction, Special Issue: Multimodal Interaction*, 12, 7-45.

Nelson, R.J., McCandlish, C.A., and Douglas, V.D. (1990). Reaction times for hand movements made in response to visual versus vibratory cues, *Somatosensory and Motor Research, 7,* 337-352.

Newell, A. and Simon, H.A. (1972). *Human Problem Solving.* Englewood Cliffs, NJ: Prentice-Hall.

Nielsen, J. (1994). *Usability Engineering, updated,* London: Academic Press.

Nilsson, M. (1999). *ID3v2. The audience is informed.* see Internet: www.id3.org.

Norman, D.A. (1986). Cognitive Engineering. In: Norman, D.A., and Draper, S.W. (Eds.), *User-Centered System Design: New perspectives in human-computer interaction,* Hillsdale, NJ: LEA, 31-61.

North, A.C., and Hargreaves, D.J. (1998). Music and consumer behaviour. In: Hargreaves, D.J., and North, A.C. (Eds.), *The social psychology of music,* Oxford: Oxford University Press, 268-289.

Ober, D. (1996). The appreciation of personalised automatic track selection: A user evaluation of music compiling functionalities. *IPO report no. 1136,* Eindhoven University of Technology.

Onken, J., Hastie, R., and Revelle, W. (1985). Individual differences in the use of simplification strategies in a complex decision-making task. *Journal of Experimental Psychology: Human Perception and Performance, 11,* 14-27.

Palmquist, J.E. (1990). Apparent time passage and music preference by music and nonmusic majors, *Journal of Research in Music Education, 38, 3,* 206-214.

Panksepp, J. (1995). The emotional sources of 'chills' induced by music. *Music Perception, 13, 2,* 171-207.

Paterson, R.J., and Neufeld, R.W.J. (1995). What are my options?: Influences of choice availability on stress and the perception of control, *Journal of Research in Personality, 29, 2,* 145-167.

Pauws, S.C. (1996). Turn on the base (Project Evaluation). *IPO report no. 1094.* Eindhoven University of Technology.

Pauws, S.C. (1998). Concept, implementation, and evaluation of a multimodal interaction style for music programming, *IPO report no. 1191,* Eindhoven University of Technology.

Pauws, S.C., Bouwhuis, D.G., and Eggen, J.H. (1998). Music programming for your hands and ears only. *IPO Annual Progress Report, 33,* Eindhoven University of Technology, 59-67.

Pauws, S.C., Bouwhuis, D.G., and Eggen, J.H. (accepted). Programming and enjoying music with your eyes closed. *Accepted for CHI 2000.*

Pauws, S.C., and Eggen, J.H. (1996). New functionality for accessing digital media: Personalised automatic track selection. In: Blandford, A., and Thimbleby, H. (Eds.), *HCI'96, Industry Day & Adjunct Proceedings, London, UK, August 20-23, 1996,* London: Middlesex University, 127-133.

Pauws, S.C., Eggen, J.H., and Bouwhuis, D.G. (1997). Explorative strategies while compiling music. *IPO Annual Progress Report, 32,* 79-88.

Pauws, S.C., Ober, D., Eggen, J.H., and Bouwhuis, D.G. (1996). A comparative evaluation of strategies for compiling music programmes. *IPO Annual Progress Report, 31,* Eindhoven University of Technology, 50-58.

Payne, J.W. (1976). Task complexity and contingent processing in decision making: An information search and protocol analysis. *Organizational Behavior and Human Performance, 16,* 366-387.

Payne, J.W. (1982). Contingent decision behaviour, *Psychological Bulletin, 92,* 382-402.

Payne, J.W., Bettman, J.R., and Johnson, E.J. (1988). Adaptive Strategy Selection in Decision Making. *Journal of Experimental Psychology: Learning, Memory, and Cognition, 14, 3,* 534-552.

Payne, J.W., Bettman, J.R., and Johnson, E.J. (1993). *The adaptive decision maker.* Cambridge University Press.

Payne, J.W., Bettman, J.R., and Luce, M.F. (1996). When time is money: Decision behavior under opportunity-cost time pressure, *Organizational Behavior and Human Decision Processes, 66, 2,* 131-152.

Payne, J.W., Bettman, J.R., and Luce, M.F. (1998). *Behavioral Decision Research: An Overview,* In: M.H. Birnbaum (Ed.), Measurement, Judgment, and Decision Making, Handbook of Perception and Cognition, 2nd edition, London: Academic Press, 303-359.

Payne, S.J. (1988). Metaphorical instruction and the early learning of an abbreviated-command computer system, *Acta Psychologica, 69,* 207-230.

Peretz, I., Gaudreau, D., and Bonnel, A.-M. (1998). Exposure effects on music preference and recognition. *Memory & Cognition, 26, 5,* 884-902.

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.

Pieters, R., Warlop, L., and Hartog, M. (1997). The effects of time pressure and task motivation in visual attention to brands. *Advances in Consumer Research, 24,* 281-287.

Pohlmann, K.C., (1995). Changer challenge. *Stereo Review, 60,* 54-62.

Poll, L.H.D. (1996). *Visualising Graphical User Interfaces for Blind Users.* Doctoral Thesis, Eindhoven University of Technology.

Post, L.J., and Chapman, C.E. (1991). The effects of cross-modal manipulations of attention on the detection of vibrotactile stimuli in humans, *Somatosensory & Motor Research, 8,* 149-157.

Proctor, R.W., and Dutta, A. (1995). *Skill acquisition and human performance.* London: Sage.

Quinlan, J.R. (1979). Discovering rules by induction from large databases: a case study. In: Michie, D. (Ed.), *Expert systems in the micro-electronic age,* Edinburgh: Edinburgh University Press, 168-201.

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

Quinlan, J.R. (1986b). The effect of noise on concept learning. In: Michalski, R.S, Carbonell, J.G., and Mitchell, T.M. (Eds.), *Machine Learning: An Artificial Intelligence Approach, Volume 2,* San Mateo, CA: Morgan Kaufmann, 149-166.

Quinlan, J.R. (1987). Simplifying decision trees. *International Journal of Man-Machine Studies, 27,* 221-234.

Quinlan, J.R. (1989). Unknown attribute values in induction. In: Segre, A.M. (Eds.), *Proceedings of the Sixth International Machine Learning Workshop, Ithaca, New York, NY, USA, June 26-27, 1989,* San Mateo, CA: Morgan Kaufmann, 164-168.

Quinlan, J.R. (1993). *C4.5: Programs for machine learning.* San Mateo, CA: Morgan Kaufmann.

Resnick, P., and Varian, H.R. (1997). Recommender Systems. *Communications of ACM, 40, 3,* 56-58.

Reynolds, C.W. (1987). Flocks, Herds, and Schools: A distributed behavioral model. *Computer Graphics, 21, 4,* 25-34.

Rich, E. (1979). User modelling via stereotypes, *Cognitive Science, 3,* 329-354.

Rich, E. (1983) Users are individuals: individualising user models, *International Journal of Man Machine Studies, 18,* 199-214.

Rich, E. (1983b). *Artificial Intelligence.* Auckland: McGraw-Hill.

Rich, E. (1989). Stereotypes and user modelling. In: Kobsa, A. and Wahlster, W. (Eds.), *User models in dialog systems,* Berlin: Springer-Verlag, 35-51.

Robbe, S., Carbonell, N., and Valot, C. (1997). Towards usable multimodal command languages: Definition and ergonomic assessment of constraints on users' spontaneous speech and gestures. In: Kokkinakis, G., Fakotakis, N., and Dermatias, E. (Eds.), *Proceedings of Eurospeech '97, ESCA 5th European Conference on Speech Communication and Technology, vol. 3, Rhodes, Greece, September 22-25, 1997,* 1655-1658.

Roberts, D. (1994). *PC Game Programming Explorer.* Scottsdale, Arizona: Coriolis Group Books.

Rogerson, D. (1997). *Inside COM. Microsoft's Component Object Model.* Redmond, WA: Microsoft Press.

Rosch, E., and Mervis, C.B. (1975). Family resemblance: studies in the internal structure of categories, *Cognitive Psychology, 7,* 573-605.
Rossing, T.D. (1990). *The science of sound, 2nd edition*, London: Addison-Wesley.
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.
Rumbaugh, J., Blaha, M., Premerlani, W., Eddy, F., and Lorensen, W. (1991). *Object-Oriented Modeling and Design*. Englewood Cliffs: Prentice-Hall International.
Rumbaugh, J., Jacobsen, I., and Booch, G. (1999). *The Unified Modeling Language Reference Manual*, Reading, MA: Addison-Wesley.
Russo, J.E., and Dosher, B.A. (1983). Strategies for multiattribute binary choice. *Journal of Experimental Psychology: Learning, Memory, and Cognition, 9,* 676-696.
Sabatella, M. (1996). *A whole approach to jazz improvisation.* Lawndale, CA: ADG Productions. See also: Sabatella, M. (1992). *A jazz improvisation primer.* Internet: www.outsideshore.com.
Salton G., and McGill, M.J. (1983). *Introduction to modern information retrieval.* London: McGraw Hill.
Salton, G. (1989). *Automatic Text Processing: The transformation, analysis, and retrieval of information by computer*, Amsterdam: Addison-Wesley.
Scheffer, R.M.M. (1996a). Interact with the Base. Contract Report. *IPO report no. 1121*. Eindhoven University of Technology.
Scheffer, R.M.M. (1996b). Interact with the Base. Progress Report: Analysis & design. *IPO report no. 1122*. Eindhoven University of Technology.
Scheirer, E.D. (1998). Tempo and beat analysis of acoustic musical signals, *Journal of the Acoustical Society of America, 103, 1,* 588-601.
Schoennherr, S. (1999). *Recording technology history: A chronology with pictures and links.* University of San Diego, see also Internet: ac.acusd.edu / History.
Shardanand, U. (1994). Social information filtering for music recommendation. *Technical report 94-04*, MIT Media Laboratory.
Shardanand, U., and Maes, P. (1995). Social information filtering: Algorithms for automating ‘Word of mouth’, *Proceedings of CHI-95: Human Factors in Computing Systems*, New York: ACM Press, 210-217.
Shepard, R.G. (1987). Towards a Universal Law of Generalization for Psychological Science, *Science, 237,* 1317-1323.
Sholl, M.J. (1996). From visual information to cognitive maps. In: Portugali, J. (Ed.), *The construction of cognitive maps,* The Hague: Kluwer, 157-186.
Shrager, J., and Klahr, D. (1986). Instructionless learning about a complex device: the paradigm and observations. *International Journal of Man-Machine Studies, 25,* 153-189.
Siegel, S., (1956). *Nonparametric statistics.* New York: McGraw-Hill.
Simon, H.A. (1955). A behavioral model of rational choice. *Quarterly Journal of Economics, 69,* 99-118.
Simonson, I., and Tversky, A. (1992). Choice in context: Trade-off contrast and extremeness aversion. *Journal of Marketing Research, 29,* 281-295.
Smith, E.E. (1990). Categorization. In: Osherson, D.N., and Smith, E.E. (Eds.), *Thinking: An invitation to cognitive science, Vol. 3,* Cambridge, MA: MIT Press, 33-53.
Slovic, P. (1995). The construction of preference. *American Psychologist, 50, 5,* 364-371.
van Sluis, S. (1996). Feel the Base. *IPO report no. 1137.* Eindhoven University of Technology.
van de Sluis, B.M., Eggen, J.H., and Rypkema, J.A. (1997). Nonspeech audio in user interfaces for TV. *IPO Annual Progress Report, 32,* Eindhoven University of Technology, 65-72.
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.
Spence, C., and Driver, J. (1997). Cross-modal links in attention between audition, vision, and touch: Implications for interface design, *International Journal of Cognitive Ergonomics, 1, 4*, 351-373.
Spence, C., and Driver, J. (1997b). On measuring selective attention to a specific sensory modality. *Perception & Psychophysics, 59*, 389-403.

Steedman, M.J., (1984). A generative grammar for jazz chord sequences. *Music Perception, 2, 1,* 52-77.

Suchman, L.A. (1987). *Plans and situated actions: The problem of human-machine communication*, Cambridge: Cambridge University Press.

Sundström, G.A, (1987). Information search and decision making: The effects of information displays. *Acta Psychologica, 65,* 165-179. Reprinted in: Montgomery, H., and Svenson, O. (Eds.) *Process and Structure in Human Decision Making,* 1989, Chichester: John Wiley & Sons, 209-233.

Svenson, O., and Edland, A. (1987). Change of preferences under time pressure: Choices and judgements. *Scandinavian Journal of Psychology, 28,* 322-330. Reprinted in: Montgomery, H., and Svenson, O. (Eds.) *Process and Structure in Human Decision Making,* 1989, Chichester: John Wiley & Sons, 209-223.

Szyperski, C. (1998). *Component software: beyond object-oriented programming.* Amsterdam: Addison-Wesley.

Tagg, P. (1982). Analysing popular music: Theory, method, and practice, *Popular Music, 2,* 37-65.

Thomson, V. (1967). Bigger than baseball. In: *Music reviewed 1940-1954,* New York: Vintage Books, 384-388.

Thurstone, L.L. (1927). A law of comparative judgment, *Psychological Review, 34,* 273-286.

Toiviainen, P. (1995). Modeling the target-note technique of bebop-style jazz improvisation: An artificial neural network approach. *Music Perception, 12, 4,* 399-413.

Tolhurst, G.C., Hollien, H., and Leeper, L. (1984). Listening preferences for music as a function of age, *Folia phoniat., 36,* 93-100.

Torgerson, W.S. (1958). *Theory and methods of scaling.* New York: John Wiley & Sons.

Tou, J.T., and Gonzalez, R.C. (1974). *Pattern Recognition Principles. Fourth printing.* Amsterdam: Addison-Wesley.

Tsao, Y.-D., Wittlieb, E., Miller, B., and Wang, T.-G. (1983). Time estimation of a secondary event. *Perceptual and Motor Skills, 57,* 1051-1055.

Tversky, A. (1969). Intransitivity of preferences. *Psychological Review, 76,* 31-48.

Tversky, A. (1972). Elimination by aspects: a theory of choice. *Psychological Review, 79, 4,* 281-299.

Tversky, A. (1977). Features of Similarity, *Psychological Review, 84, 4,* 327-352.

Tversky, A., and Sattath S. (1979). Preference trees. *Psychological Review, 86, 6,* 542-573.

Tversky, A., and Simonson, I. (1993). Context-dependent preferences. *Management Science, 39,* 1179-1189.

Utgoff, P.E. (1988). ID5: An incremental ID3. In: Laird, J. (Ed.), *Proceedings of the Fifth International Conference on Machine Learning, Ann Arbor , Michigan, USA, June 12-14, 1988,* San Mateo, CA: Morgan Kaufmann, 107-120.

Utgoff, P.E. (1988b). Improved training via incremental learning. In: Laird, J. (Ed.), *Proceedings of the Fifth International Conference on Machine Learning, Ann Arbor , Michigan, USA, June 12-14, 1988,* San Mateo, CA: Morgan Kaufmann, 362-365.

Utgoff, P.E. (1989). Incremental induction of decision trees, *Machine Learning, 4,* 161-186.

Vallacher, R.R., and Wegner, D.M. (1987). What do people think they're doing? Action identification and human behavior. *Psychological Review, 94, 1,* 3-15.

van der Veer, G.C. (1990). *Human computer interaction: learning, individual differences, and design recommendations.* Doctoral dissertation, Vrije Universiteit, Amsterdam.

van der Veer, G.C. (1994). Mental models of computer systems: visual languages in the mind. In: Tauber, M.J., Mahling, D.E., and Arefi, F. (Eds.), *Cognitive aspects of visual languages and visual interfaces,* Amsterdam: North-Holland, 3-40.

van de Velde, W. (1990). Incremental induction of topologically minimal trees. In: Porter, B., and Mooney, R. (Eds.), *Proceedings of the Seventh International Conference on Machine Learning, Austin, Texas, USA, June 21-23, 1990,* San Mateo, CA: Morgan Kaufmann, 66-74.

Warren, W., and Verbrugge, R. (1984). Auditory perception of breaking and bouncing events: A case study in ecological acoustics. *Journal of Experimental Psychology: Human Perception and Performance, 10,* 704-712.

Wavish, P. (1991a). Real Time ABLE. *Philips Research Laboratories Redhill Review of 1990, 1991.*

Wavish, P. (1991b). Exploiting Emergent Behaviour in Multi-Agent Systems. In: Demazeau, Y., and Werner, E. (Eds.), *Proceedings of 3rd European Workshop on Modeling an Autonomous Agent in a Multi-Agent World (MAAMAW), Kaiserlautern, Germany, 5-7 August, 1991,* Amsterdam: North-Holland, 297-310.

Wavish, P., and Connah, D. (1990). Representing multi-agent worlds in ABLE. *Technical Note, TN2964, Philips Research Laboratories,* Redhill, UK.

Wavish, P., and Graham, M. (1994). Roles, skills, and behaviour. In: Wooldridge, M., and Jennings, M. (Eds.), *Proceedings of the ECAI-94 workshop on Agent Theories, Architectures, and Languages, Amsterdam, the Netherlands, August, 1994,* Amsterdam: North-Holland, 371-385.

Wheeler, B.L. (1985). Relationship of personal characteristics to mood and enjoyment after hearing live and recorded music and to musical taste. *Psychology of Music, 13, 2,* 81-92.

Wickens, C.D. (1984). Processing resources in attention. In: Parasuraman, R., and Davies, D.R. (Eds.), *Varieties of attention,* New York: Academic Press, 63-102.

Wickens, C.D. (1992). *Engineering psychology and human performance,* New York: Harper Collins.

Wiedenbeck, S., and Davis, S., (1997). The influence of interaction style and experience on user perceptions of software packages, *International Journal of Human-Computer Studies, 46,* 563-588.

Winer, B.J., (1962). *Statistical Principles in Experimental Design*. London: McGraw-Hill.

Wright, P. (1974). The harassed decision maker: time pressures, distraction, and the use of evidence, *Journal of Applied Information Processing, 59,* 555-561.

Wright, P.L. (1975). Consumer choice strategies: Simplifying vs. optimizing. *Journal of Marketing Research, 11,* 60-67.

Zakay, D. (1985). Post-decisional confidence and conflict experienced in a choice process. *Acta Psychologica, 58,* 75-80.

Zakay, D., and Wooler, S. (1984). Time pressure, training, and decision effectiveness. *Ergonomics, 27,* 273-284.

APPENDIX I Attribute representation of music options

The music collection used for experimental purposes, comprised 480 jazz music recordings extracted from 160 commercial CD albums. The selection of recordings was done by two persons independently; from each album, they selected two favourite recordings and a less favourite one. The selected pieces were not recorded in their entirety but for a number of reasons only a one-minute excerpt was taken. Firstly, by taking a fixed-sized excerpt, the assembly process proceeded significantly faster. Secondly, using full-length music recordings in the experiments could be experimentally unwieldy as participants might listen for much longer. Finally, it was possible to take excerpts that were completely representative of the whole piece. The first minute of each piece was chosen as the excerpt, as this contains the main chorus. In the majority of jazz styles, musical improvisation follows a hierarchical structure composed of choruses, i.e., units of a fixed harmonic structure. A common harmonic structure for jazz choruses is a variation on a simple 12-bar blues chord progression. Another standard chorus for bebop entails a major key progression of 32 bars, divided in parts of eight bars according to the pattern AABA. Performance of a piece of jazz music consists of a main chorus carrying the musical theme, a number of improvisation choruses, and a final chorus. As the main chorus constitutes the representative core of the composition, listeners are likely to recognise the piece of music mostly after only a few seconds of listening to the excerpt. If listeners do not happen to know the recording, the full-minute excerpt should be a sufficient representation of the recording as a whole, even during the future encounters with that recording.

The complete music collection contained, for example, 498 different musicians, 230 different composers, 42 different musical instruments and 12 popular jazz styles. Each jazz style contained 40 music recordings. The tempo in which the music was performed ranges from 50 beats per minute (bpm) to 350 bpm. Different pieces of music were originally recorded in different time periods ranging from 1945 to 1995. It can thus be concluded that the collection covers a considerable part of the whole jazz period. The music excerpts were stored in a compressed form (MPEG -1 Part 2 Layer II 128 Kbps stereo). The use of Layer II in this MPEG-1 standard was a compromise between storage requirements, computational costs for real-time decompression, and acceptable loss in perceptual quality. Dependent on the musical content, compression factors higher than 10 were common for music that was originally sampled at CD quality (44.1 kHz, 16 bit precision, stereo); one minute of music amounts to approximately 1 Mb of data. In the worst case, real-time decompression absorbed 70% of the computing resources of a typical work station, which was just tolerable for the experimental systems. Though degradation of the audio quality was introduced (in particular ringing effects could be heard at higher frequencies in the drummer's ride cymbal performance during a steady swing, for instance), this was hardly noticeable in pilot tests and was not commented upon by any of the actual listeners in the experiments.

Database design starts with the specification of a data model or schema. Using an object modeling technique, data model specification comes down to discovering and describing entities in the domain. Entities are described by a list of attributes and relations with other entities.

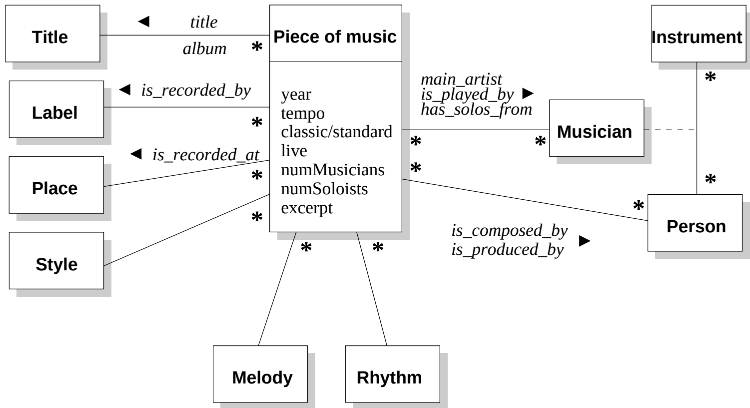

Figure I.1. Class diagram showing attribute representation of a music option.

The class diagram of a music option in the database is shown in Figure I.1; it represents the musical attribute representation for each music recording. The graphical notation is adopted from the Unified Modeling Language (UML) (Rumbaugh, Jacobson, and Booch, 1999). Ten entities pertaining to concepts in the jazz music domain are defined. If all relations and attributes are counted, a set of 18 valued attributes represents a music option; the excerpt is not included. Attributes have different domains: quantitative, nominal and categorical values. All attribute values are fixed in time. Relations in the class diagram express attributes with multiple values such as a list of producers and composers or a list of musicians.

For each entity, that is, a piece of music, a detailed list of attribute data was collected by using information found on the CD booklets, discographies such as the ‘All Music Guide’ (Erlewine, Woodstra, and Bogdanov, 1994) and the ‘Penguin Guide’ (Cook and Morton, 1994) and books on jazz music education (Coker, 1964; Sabatella, 1996). Part of the attribute data aims at capturing the improvisational nature of jazz which is highly admired by jazz music listeners. Improvisation is influenced by harmonic structure, melody of the theme, rhythmic style of the accompaniment, the musical instrument used, and the skills of the soloist (Coker, 1964; Berliner, 1994). Tempo and instruments are important determinants to judge similarity between jazz improvisations (Cupchik, Rickert and Mendelson, 1982). Data on tempo, rhythmic structure and melodic development were acquired by systematic listening.

Piece of music - a recording of a piece of jazz music, with the following attributes

> *year* - year of recording. This year can be placed in a jazz era or period. The year of recording is particularly important for music preference, as it is known that music listeners prefer music they were accustomed to in early adulthood, perhaps out of a sense of ’nostalgia’ (Holbrook and Schindler, 1989; Rubin, Rahhal and Poon), which largely explains music preference as a function of age (Tolhurst, Hollien and Leeper, 1984).

tempo - mean pace of the musical performance measured in beats per minute. Stable tempo appears to be an important component of the rhythmic foundation of jazz called ‘swing’. Collier and Collier (1994) found that both single jazz performers and jazz ensembles played with a stable tempo across different musical passages on recordings. The height of tempo also influences the ’swing ratio’ in jazz (Friberg and Sundström, 1997); we will come back to that later on. Although not used in this study, tempo measurement and beat tracking of music can be done automatically in real-time. Its performance is similar to the performance of human listeners who tap along with the music (Martin, Scheirer and Vercoe, 1998; Scheirer, 1998). Here, tempo was measured by counting the beats in a given unit of time while listening.

standard - defines a composition that is commonly played by jazz musicians but that was already popular before the heyday of jazz. The composition was thus not primarily intended for jazz musicians. Standards are jazz pieces which are generally familiar to jazz music listeners.

classic - defines a composition that is commonly played by jazz musicians and composed by a jazz musician. Classics are jazz pieces which are generally familiar to jazz music listeners. A piece of jazz music can be defined as a standard, a classic or neither of these two.

live - indicates whether the recording is recorded in front of a live audience or not. Hearing a live jazz recording, representing a live performance, is likely to induce a positive preference decision, as a live performance is intuitively more enjoyable than studio-recorded music (Wheeler, 1985).

numMusicians - number of musicians who play along on the recording. It measures ensemble strength (literally: number of instruments played on the recording). In jazz, standard ensembles exist such as a jazz quartet or quintet (e.g., piano, bass, drums, trumpet and / or saxophone).

numSoloists - number of musicians playing solo (literally: number of musicians playing the musical theme of the main chorus in unison or individually).

excerpt - reference to a first-minute excerpt of the recording.

Title - song or album title; both song and album title are included. A music recording has a unique title, though some recordings are also known under other names, when the music is used for a commercial, movie sound track or television programme, for instance. A song title can be an adequate descriptor for music preference, as it forms a unique identifier for a music recording. However, a title may be a less suitable mechanism of referring to a music recording, as it is hard to recall titles spontaneously or even with the aid of musical cues (Peynircioglu, Tekcan, Wagner, Baxter, and Shaffer, 1998).

Label - the record / distribution company of the recording. Legendary jazz record labels exist and some labels are well-known for their re-releases and compilations of previously unreleased jazz material.

Place - place of recording (e.g. studio, concert hall). Some studios are well-known for their ’natural’ registration of music, and some live events in concert halls are famous for their atmosphere which inspires musicians.

Instrument - a musical instrument played on a recording. Research on automatic musical instrument recognition and identification is on its way (Martin, Scheirer and Vercoe, 1998).

In the class diagram, musical instruments are arranged in classes of string instruments, brass instruments, woodwind instruments, percussion instruments, keyboard instruments, electronic instruments, and vocal instruments. The coding scheme of musical instruments is adopted from the 'Penguin Guide' (Cook and Morton, 1994).

Person - any person who plays a role in a recording and therefore deserves credit. A distinction has been made between musicians (by the association *is_played_by*), composers (by the association *is_composed_by*), producers and studio engineers (by the association *is_produced_by*). Hence, musicians, composers and producers (studio engineers) are known for each music option. The same person can be a musician, a composer and a producer on one recording. Musicians are likely to be preferred for their improvisational skills and interpretation, composers are likely to be preferred for their artistic quality, and producers and studio engineers are likely to preferred for their technological skills and editorial precision.

Musician - a person who plays an instrument on a recording. A distinction has been made between musicians who are the main performing artists (by the association *main_artist*), musicians who play a solo on the recording (by the association *has_solo_from*), and all musicians who play an instrument on the recording (by the association *is_played_by*). Since the class **Musician** is a combination of the classes **Person** and **Instrument**, solo instruments can be easily identified, if necessary.

Style - the jazz style or era of the recording. The twelve most popular styles of jazz are present (Erlewine, Woodstra and Bogdanov, 1994). These styles cover a considerable part of the whole jazz period. Jazz styles comprise different rhythms, tonalities, melodies, harmonies, instrumentation, musical interpretations for improvisation, and cultural expressions ranging from blues-based styles to modern classical music approaches or a combination of popular dance and rap. These general characteristics of jazz styles are important for music preference.

> <u>Blues Jazz</u> comprises rhythm and blues schemes incorporated into a swinging, mostly vocal context. Some typical musicians are: Billie Holiday, Dinah Washington and Louis Armstrong.
>
> <u>Swing</u> comprises big bands and small groups (1930-1940) that were popular for dance. Some typical musicians are: Ben Webster, Count Basie, Oscar Peterson and Toots Thielemans.
>
> <u>Bebop</u> comprises the beginning of modern jazz marked by complex harmonics, up-tempos, and high skills (1940-1950). Some typicial musicians are: Charlie Parker, Fats Navarro, Dizzy Gillespie, Stan Getz and Gene Ammons.
>
> <u>Cool Jazz</u> (or West Coast Jazz) comprises the counterpart of bebop and a reaction against its ideas. It is marked by soft melodies and less dynamics. Some typicial musicians are: Chet Baker, Gerry Mulligan, Bill Evans and Stan Getz.
>
> <u>Latin Jazz</u> comprises latin rhythms (e.g., bossa nova, samba) on which jazz melodies are superimposed. Some typicial musicians are: Stan Getz, Jobim, Paquito D'Rivera, Tania Maria and Mario Bauza.

HardBop comprises the return to more bluesy, less technically demanding melodies than the bebop but played with the same intensity (1950-1960). Some typicial musicians are: John Coltrane, Miles Davis, Sonny Rollins, Thelonious Monk, Ron Carter and Cannonball Adderley.

PostBop (Modal Jazz) comprises the innovative improvisations on open-ended harmonics rather than strict popular chord schemes (1950-1960). Some typicial musicians are: Miles Davis, Herbie Hancock, Wayne Shorter and Dexter Gordon.

Fusion (Jazz-rock) comprises the combination of rock and funk elements with jazz solo techniques. Some typicial musicians are: Brecker Brothers, Weather Report, John Scofield, Yellow Jackets and Casiopea.

PostModern (New Age) comprises the influence of world music, classical music, and folk music into jazz. The composition is far more important than the improvisation. The result can be rather esoteric and atmospheric. Some typicial musicians are: Pat Metheny, Paul Bley, John Abercrombie, Bill Frisell and Ralph Towner.

MBase (Avant Fusion) comprises the combination of complex funky rhythms with rather angular melody lines. MBase is short for Macro-Basic Array of Structured Extemporization. Some typicial musicians are: Steve Coleman, Greg Osby, Gary Thomas, Kevin Eubanks and Cassandra Wilson.

NeoBop (Neo classicism, Neo swing) comprises the new generation of young players who find their influence in the acoustic bebop and postbop eras. Some typicial musicians are: Branford / Wynton Marsalis, Joshua Redman, David Kikowski, Keith Jarret, Ray Hargrove and Christian McBride.

Dance (DooBop, Jazz-dance, Acid-Jazz) comprises dance-oriented contemporary funk, soul, and hip-hop with jazzy snips. Some typicial musicians are: Jazzmatazz, Me'Shell Ndegeocello, Marcus Miller, Miles Davis and Mezzoforte.

Rhythm - defines the rhythmic foundation or metre of the recording. The first three categories, that is, threefour, fourfour, and fivefour, define the rhythmic foundation or metre that jazz music distinguishes itself, commonly referred to as 'swing feel'. In many jazz styles, odd eight notes (on the beat) are commonly lengthened whereas even eight notes are commonly shortened. The alternating long-short duration of consecutive groups of eight notes is referred to as 'swing ratio'. This 'swing ratio' appears to be linearly related to tempo. In jazz recordings, Friberg and Sundström (1997) found that 'swing ratio' can be as high as 3.5:1 at relatively slow tempi (120 bpm) and reaches 1:1 at fast tempi (> 250 bpm). Jazz music listeners prefer a similar linear decrease in 'swing ratio' at increasing tempo (Friberg and Sundström, 1997). Note that quantitative measurement of 'swing ratio' has not been attempted here.

Besides the use of local chords, a chord progression, and a melodic motive, jazz improvisation also hinges on rhythm and metre. Improvisation is often a process of connecting two or four bars and using reference points found in a metre (Järvinen, 1995).

The following categories were extracted from books on jazz improvisation (Coker, 1964; Sabatella, 1996). Recordings were assigned to a category by systematic listening.

threefour - rhythmic foundations known as swing performed in 3/4 and 6/8 time, i.e., waltz.

fourfour - rhythmic foundation known as swing performed in 4/4 time.

fivefour - rhythmic foundation known as swing performed in 5/4 or other less common modes.

ballad -rhythmic foundation known as ballad which is played slow and rubato without a well-defined beat.

latin - rhythmic foundation known as traditional latin rhythms such as mambo, samba, and bossa nova.

rock - rhythmic foundation known as rock, comprising a steady beat.

funk - rhythmic foundation with funk and soul influences. The rhythm is 'groovy', modern dance-oriented and syncopated.

Melody - defines the melodic and harmonic development of the recording characterising the relationship of the chord progression and melody (improvisation) lines. Music listeners are able to determine relative perceived stability of the 12 chromatic tones in a tonal context of Western music (Krumhansl and Shepard, 1979; Krumhansl, 1990) and also develop a sense of key during a chord progression (Krumhansl and Kessler, 1982; Krumhansl, 1990). This suggests that music listeners are able to distinguish various tonal relationships between a melody and a harmony.

Attempts to analyse and model jazz improvisation with respect to a chord progression have been made (Järvinen, 1995; Toiviainen, 1995). Jazz chord progressions have also been computationally studied to find governing rules of their form, analogous to rules for grammar of natural languages (Steedman, 1984).

The following categorical values are an attempt to categorise various tonal relationships between a melody and a harmony. The categories were extracted from books on jazz improvisation (Coker, 1964: Sabatella, 1996). Recordings were assigned to a category by systematic listening.

Progressive represents a strong relationship between the melody (improvisation) lines and the chords and scales that hold at that moment. The notes are neatly stringed together and cover the whole chord extension. Improvisation is done on a 32-bar progression.

Blues represents a melody line (improvisation) linked closely to a former blues chord progression, that is, a 12-bar blues progression.

Modal represents a *very melodic* melody line on a small chord progression mainly stuck to basic chord notes.

Chromatic represents angular and chromatic melody lines played with some notes outside the chordal context. Angular means large unusual tone interval and chromatic means the opposite: playing notes separated by only a single interval.

Non-tonal represents the elimination of western tonality, for instance, by using non-western chord scales and key centres. Chords are not played for their resolution but merely for the overall sound.

> Free represents the elimination of a chord progression, chord scales, and key centres.

The data of all pieces of jazz music, that is, the music collection, are collected in a database. Data access for applications running under UNIX is facilitated by a small, specialised, object-oriented database management system (DBMS). This DBMS was implemented in C/C++ and comprises the class diagram in Figure I.1. For applications running under Windows, the content is available as a Microsoft Access database. As MS Access is a *relational* database management system, the complete object model is mapped onto tables by following the procedure as suggested by Rumbaugh, Blaha, Premerlani, Eddy and Lorensen (1991). In this procedure, objects are mapped onto table rows with primary keys, and binary relations between objects are mapped onto distinct tables consisting of a primary key and two foreign keys. Both the class diagram and the resulting relational database are in third normal form; as data entry extended over several months, a high degree of entity and referential integrity was required to ensure consistency in the database.

APPENDIX II IPO trackball with force feedback

The IPO force feedback prototype devices (Engel, Haakma and van Itegem, 1990; Engel, Goossens and Haakma, 1994; Keyson, 1996) were created in close cooperation between the IPO and Philips Research Laboratories in Eindhoven. The trackball used in the development and evaluation of a multimodal interaction style for music programming is a motor input and tactual output device with two Degrees Of Freedom (DOF). This trackball device is shown in Figure II.1. Motor-driven wheels touch the ball at its axes, which creates a two-dimensional force plane. In that way, force feedback is mediated by a motor-driven ball movement. The motors are controlled by software which enables the mediated force feedback to be varied depending on the current context of interaction. Trackball devices with one degree of freedom or three degrees of freedom also exist. The one-DOF devices are rotary dials which use either motor- or electromagnetic force feedback. The three-DOF device is essentially a two-DOF trackball device, but, in addition, it has a hinge mechanism with force feedback on which the whole trackball unit can be moved up and down (Keyson, 1996). Software to control a force feedback trackball was developed in-house.

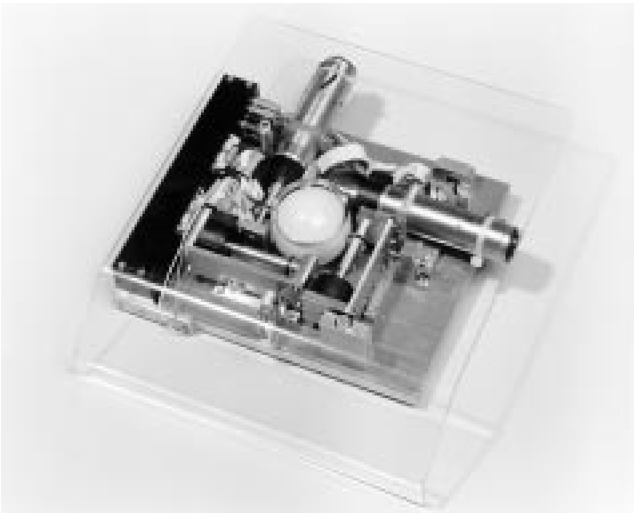

Figure II.1. The IPO two-DOF ground-based trackball with force feedback.

II.1 Mechanics and electronics

The mechanical and electronic structure of the IPO two-DOF ground-based trackball device is patented (Engel, Haakma and van Itegem, 1990). A block diagram of the mechanical and electronic components of the device is shown in Figure II.2. Some components are mounted on a solid plate resting on shock-proof rubbers, which comprises the trackball unit. This unit can be firmly placed on any solid surface. It is covered by a Plexiglass surface on which the wrist can be rested comfortably while the fingers manipulate the ball. A second unit comprises other components such as the motor amplifiers and I/O cards. The IPO trackball device has powerful motors which means that the extensive dedicated hardware has to be placed outside the trackball unit.

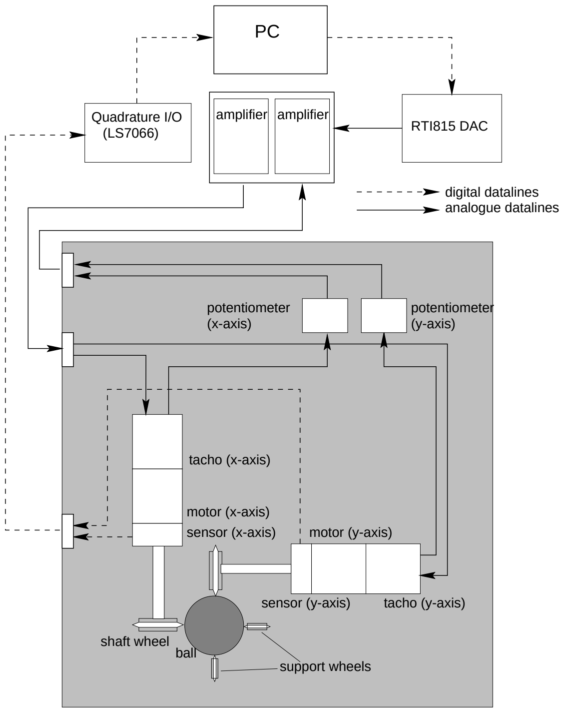

Figure II.2. The mechanical and electronic components of the IPO two-DOF ground-based trackball with force feedback.

The hard-plastic ball with a diameter of 57 mm rests on a ball-bearing, which minimises any friction. A contact switch is mounted underneath this bearing system that enables the notification of discrete push movements executed vertically to the ball, that is, depressing the ball. Another device, which was not used in the implementation and evaluation of a multimodal interaction style for music programming, has a separate button instead of a contact switch underneath the ball; this button is placed in such a way that it can be easily depressed by the thumb of a right-handed person, while manipulating the ball with the index and middle fingers. The apparatus is provided with two independent direct current motors, one for each degree of freedom, that is, the x and y component. Both rubber-rimmed shaft wheels, each driven by a motor, are fastened at the perpendicular centre lines of the ball. Two small free-rolling support wheels that are fixed at the opposite sides of each shaft wheel ensure that the ball does not wander from its bearing. The orthogonally positioned shafts enable the creation of a two-dimensional force plane on the ball. Two optical position sensors placed at the end of the motors sense the position of the ball. Another device, which was not used in the implementation and evaluation of a multimodal interaction for music programming, has its optical position sensors placed on additional wheels, thus ensuring a finer measurement of the ball position independent from the motors.

The optical sensors are connected to a Quadrature I/O card. The LS7066 chip on the card can be controlled by software. This software defines two data lines or counters that represent the x and y components of the ball position.

Two independent power amplifiers produce the voltages provided to the motors. These voltages are fixed at a constant level, when the input voltage of the amplifiers is constant. As a consequence, the torque produced by the motor, that is, the force exerted on the ball, is proportional to the input voltage of the amplifier, but also proportional to the current in the motors. To provide a constant force level on the ball, irrespective of the manual ball rotation, the currents in the motors need to be constant. The undesirable current in the motor, induced by ball rotation, is compensated by a feedback loop within the amplifiers. Voltages corresponding to ball rotation speeds are measured by tachogenerators and voltage dividers and are fed back to the amplifier to adjust the voltage provided to the motor, keeping the current level within the motor constant.

The input voltage of the amplifier is derived from a digital set point value. This set point value is a two-element vector and is interpreted by a Digital-to-Analog Converter (DAC) RTI815 card. Since one set point value and two communication paths to the motors are required, one digital input channel and two analogue output channels of the card are used. The DAC card can be controlled by software.

II.2 Software

The software environment to control force feedback trackball devices uses two serially connected PCs; one PC which mainly runs the application and one PC which controls the trackball device. Due to foreseen computational resources required for the support of various input and output modalities in interaction styles, the environment was divided across two PC platforms. The software was developed by the Philips Research Laboratories in Eindhoven, starting from preliminary software developed at the IPO Center for User-System Interaction in Eindhoven (Keyson and Tang, 1995; Keyson and van Stuivenberg, 1997). The software was adapted to meet specific requirements for the implementation of a multimodal interaction style for music programming, that is, the control of a two-DOF input device and the use of specific force feedback. A detailed description of the software can be found elsewhere (Pauws, 1998).

II.2.1 Tactual objects and workspaces

The addition of force feedback in an interaction style is based on a spatial arrangement of tactual objects in a workspace. Some of these objects evoke a local force field when their designated region is passed through by a cursor. Others evoke a global force field when other conditions hold, such as timing conditions. In both cases, the force is passed through as a motor-driven ball movement. Cursor control is entirely mediated by manipulating the trackball, though the cursor may not be seen on the visual display. Workspaces consist of a collection of tactual objects which supply a certain functional coherence. A workspace can refer to a certain physical region of a visual display or to a well-defined part of an interaction dialogue that can be called up during run-time.

Tactual objects are represented as icons and have names such as ‘hill’, ‘peak’, ’hole’, ‘path’ and ‘bump’. This representation refers to real-world objects and provokes a mental imagery of their associated tactual sensations, which facilitates easy recall and a common language among designers (Keyson and Tang, 1995; Keyson, 1996; Keyson and van Stuivenberg, 1997). Though the tactual objects are represented by graphical objects such as ellipses, circles and rectangles, they do not need to be visualised during run-time.

Each tactual object represents a parameterised two-dimensional tactual force field map, which is described by geometrical formulae. For instance, the force field map of a circular 'hole' evokes a directional pulling-force towards its force field centre, when the cursor is moved into the region of action of the 'hole'. This feels like being captured in a region when rolling the ball, and needs some additional hand force to leave this region. Definitions of some force field maps are described elsewhere (Keyson and Tang, 1995; Keyson, 1996; Keyson and van Stuivenberg, 1997).

II.2.2 Software components

As shown in Figure II.3, three software components make up the control software for the trackball: *LightHole*, *TacServer* and *TacServer Extension*. The graphical notation is adopted from the Unified Modeling Language (Rumbaugh, Jacobson, and Booch, 1999).

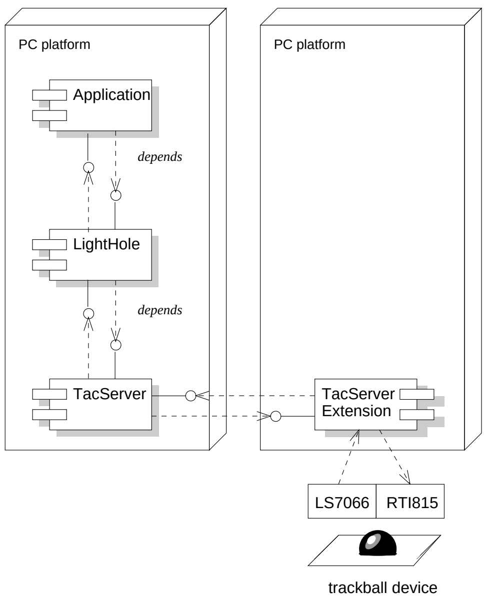

Figure II.3. Software components for controlling the IPO force feedback trackball. The 'lollipops' represent software interfaces.

LightHole is an ActiveX control component that can be directly incorporated in an (container) application without any hard-coding. This component facilitates the specification of where, when and how force feedback is evoked in the application. Being an instance of an ActiveX control, a tactual object has inherited all functionality of a window object (e.g., window event processing) plus additional functionality such as event firing and support of properties and methods. Event firing is required to notify the application of events such as entering or selecting a tactual object. Methods and properties are exposed to the application to change the state or appearance of the control.

TacServer is a static link library, which facilitates the communication between the two PCs. It therefore communicates asynchronously (currently by an RS-232 connection) with the PC that actually controls the trackball device. The protocol communicates creation requests of tactual objects, the objects' parameters, the objects' states, ball push events and ball rotation speeds. Events originating from the trackball hardware are coerced into mouse events to be captured by the *LightHole* component. Components on either side of *TacServer* can vary independently, as long as they comply to the shared interfaces.

TacServer Extension is a process that continuously runs on a separate PC dedicated to control the trackball device by its RTI815 and LS7066 interfaces. It calculates the force feedback to be mediated by the motors of the device based on the current cursor position and the tactual objects laid out in a workspace. In addition, it captures hardware events (e.g., ball rotation and presses) that are sent off to the *TacServer* component.

APPENDIX III Questionnaires, interview and music material

III.1 Questionnaire on CD player use

A questionnaire on CD player use, as reported in Chapter 1, contained the following five questions and format (translated from Dutch):

Q1 What kind of CD player do you own?

❏ Single-CD player ❏ CD changer ❏ CD jukebox player

Q2 How many CDs do you own?

❏ Less than 50 ❏ 50-100 ❏ 100-200 ❏ More than 200

Q3 Can you indicate how often and why you use the CD player feature, if present, to program music? Note that these features are called Favourite Track Selection (FTS), program play or select play dependent on the manufacturer of the player.

❏ Never ❏ Seldom ❏ Sometimes ❏ Often ❏ Always

Why do you or don’t you use this feature?..

....

Q4 Can you indicate how often and why you use the player feature, if present, to play songs in a random order? Note that these features are called random, shuffle or aselect play dependent on the manufacturer of the player.

❏ Never ❏ Seldom ❏ Sometimes ❏ Often ❏ Always

Why do you or don’t you use this feature?..

....

Q5 Can you describe what features of your player you lack or like see improved?

....

....

III.2 Music material in the focus group study

In the focus group study of Chapter 2, the following jazz music recordings were provided to the participants.

Deborah Brown - “It don’t mean a thing (If it ain’t got that swing)”, “Embraceable you” and “Bebop”, taken from the album “Jazz 4 Jazz”, Timeless Records.

The Brecker Brothers - “Skunk Funk” and “Dream Theme", taken from the album “The Brecker Brothers Collection Volume One”, Arista Records.

Chet Baker - “You don’t know what love is” and “Donna Lee”, taken from the album “Jazz Trumpet”, Jazz World Series, and “All blues” taken from the album “Chet Baker 79”, Celluloid Records.

Jaco Pastorius - “Donna Lee”, taken from the album “Jaco Pastorius”, Epic CBS Records.

Marcus Miller - “Teen Town” and “Rampage", taken from the album “The sun don’t lie”, Dreyfuss Jazz.

Miles Davis and John Coltrane - “Fran-dance”, taken from the album “Konsertthuset Stockholm March 22 1960”, Giants of Jazz.

Miles Davis - “All blues” taken from the album “Kind of blue”, CBS Records.

Michael Brecker - “Suspone”,” Scriabin” and “Don’t try this at home“, taken from the album “Don’t try this at home”, Impulse! MCA Records.

Joe Pass - “Solitude” and “Satin Doll”, taken from the album “Portraits of Duke Ellington”, Pablo Records.

Keith Jarrett - “Bye bye blackbird” and “Straight No Chaser“, taken from the album “Bye bye blackbird”, ECM Records.

Julian “Cannonball” Adderley - “Stay on it” and ”Straight No Chaser”, taken from the album “Cannonball’s sharpshooters”, Mercury Records.

III.3 Post-experiment interview

A post-experiment interview was carried out in the user experment in Chapter 5. The interview involved the following five questions (translated from Dutch).

Q1 What did you think about the presented music programmes, in general?

Q2 Did you notice a difference between the presented music programmes for one context-of-use and the presented music programmes for the other context-of-use?

Q3 Did you notice something special about the two presented music programmes in a listening session?

Q4 What did you think about the presented music programmes in successive listening sessions? Did you observe a certain trend?

Q5 Do you have a personal need for a system that compiles music programmes for you?

III.4 Questionnaire on procedural knowledge

The questionnaires to assess procedural knowledge in the user evaluation of the multimodal interaction style for music programming in Chapter 8 were compiled from a set of 24 questions. Before completion of the questionnaire started, the participants read the following instruction (translated from Dutch).

> The next questions relate to physical actions you need to perform to go from a given start situation to a desired final situation. We are only interested in the actions you need to perform using the interactive system that takes you from the start situation to the final situation *as fast as possible*. Please, try to describe the sequence of actions required as detailed and complete as possible. You are not allowed to change once-given answers, unfortunately. If you do not understand a question or feel uncertain about your answer, feel free to tell the experimenter. The experimenter will make notes of your answers.

The questions concern both single-step and multiple-step interactions. Single-step interactions can be completed by performing a single action. Multiple-step interactions can be completed by performing two or three actions. Each questionnaire contained 10 items from each type of questions. All questions are listed below (translated from Dutch).

Single-step interactions:

1. Imagine you are listening to a piece of music in the music collection (‘collection’). What action(s) do you need to perform *in order to get information about to which jazz style this piece of music belongs?*
2. Imagine you are listening to a piece of music in the music collection (‘collection’). What action(s) do you need to perform *in order to go to another (e.g., the next) piece of music within the music collection by a step size of 3?*
3. Imagine you are listening to a piece of music in the music collection (‘collection’). What action(s) do you need to perform *in order to get information about how many pieces of music you have already added to your music programme (‘programme’)?*
4. Imagine you are listening to a piece of music in the music collection (‘collection’). What action(s) do you need to perform *in order to add this piece of music to your music programme (‘programme’)?*
5. Imagine you are listening to a piece of music in the music collection (‘collection’). What action(s) do you need to perform *in order to stop the playing of music?*
6. Imagine you are listening to a piece of music in the music collection (‘collection’). What action(s) do you need to perform *in order to listen to another (e.g., the next) piece of music in the music collection?*
7. Imagine you are currently in the jazz style section (‘styles’). What action(s) do you need to perform *in order to select another jazz style by a step size of 3?*
8. Imagine you are currently in the jazz style section (‘styles’). What action(s) do you need to perform *in order to get information what jazz style is currently selected?*
9. Imagine you are currently in the music recommendation section (‘recommendations’). You have just added a piece of music from the music recommendation to your music programme. What action(s) do you need to perform *in order to listen to this piece of music in your music programme (‘programme’)?*
10. Imagine you are currently in the music collection (‘collection’). You are listening to one pair of pieces of music in a sequence; first the first piece, then the second one. What action(s) do you need to perform *in order to listen again to the first piece of music?*
11. Imagine you are listening to a piece of music in your music programme (‘programme’). What action(s) do you need to perform *in order to go to another (e.g., the next) piece of music within your music programme by a step size of 3?*
12. Imagine you are listening to a piece of music in the music recommendations (‘recommendations’). What action(s) do you need to perform *in order to listen to another (e.g., the next) piece of music in the music recommendations?*

Multiple-step interactions:

13. Imagine you are currently in the music collection (‘collection’). You have just added a piece of music from the music collection to your music programme (‘programme’). What action(s) do you need to perform *in order to add also the first music recommendation of this piece of music to your music programme?*
14. Imagine you are currently in your music programme (‘programme’). What action(s) do you need to perform *in order to arrive at the music collection (‘collection’)?*
15. Imagine you are currently in the music recommendations (‘recommendations’). What action(s) do you need to perform *in order to arrive at the jazz style section (‘styles’)?*
16. Imagine you are currently in the jazz style section (‘styles’). What action(s) do you need to perform *in order to go 2 jazz styles farther?*
17. Imagine you are currently in the music collection (‘collection’). You have just added a piece of music from the music collection to your music programme (‘programme’). What action(s) do you need to perform *in order to listen to this piece of music within your music programme?*
18. Imagine you are currently in your music programme (‘programme’). What action(s) do you need to perform *in order to add an arbitrary piece of music from the music collection (‘collection’) to your music programme?*
19. Imagine you are currently in the music collection (‘collection’). What action(s) do you need to perform *in order to remove an arbitrary piece of music from your music programme (‘programme’)?*

20. Imagine you are currently in the music collection (‘collection’). You have just added a piece of music from the music collection to your music programme (‘programme’). What action(s) do you need to perform *in order to remove this piece of music from your music programme?*
21. Imagine you are currently in the music collection (‘collection’). What action(s) do you need to perform *in order to add the fourth piece of music from the music recommendations (‘recommendations’) to your music programme (‘programme’)?*
22. Imagine you are listening to a piece of music of a particular jazz style in the music collection (‘collection’)? What action(s) do you need to perform *in order to listen to a piece of music of another (but arbitrary) jazz style within the music collection?*
23. Imagine you are listening to a piece of music in the music recommendations (‘recommendations’). What action(s) do you need to perform *in order to listen to the music recommendations of another (but arbitrary) piece of music?*
24. Imagine you are currently in the music recommendations (‘recommendations’). You have just added two different pieces of music from the music recommendations to your music programme (‘programme’). Your music programme contains two pieces of music. What action(s) do you need to perform *in order to remove the firstly added piece of music from your music programme?*

Summary

It goes without saying that music listening is one of the most widely indulged leisure activities. However, it can be difficult to choose preferred music to listen to in a given situation, especially when there is a wide assortment of music available. Unfortunately, existing music players do not help when selecting appropriate music from a large music collection.

It is for these reasons that the central theme of this thesis is how future music players can be of help to music listeners in selecting preferred music from a large personal music collection. Particular attention is paid to music programming, which is known as the selection of multiple preferred pieces of music to listen to in one go. As a starting point, the current research formulates two user requirements to be met by future players. The first requirement is that players should adapt to the music choice behaviour of the music listener, so that they can provide adequate assistance to ease and speed up the music selection process of the listener. The second requirement is that players should present information about music interactively, in such a way that it facilitates user navigation in a large music collection. In order words, a music player should be more personalised. Subsequently, the current research seeks to explore how these requirements can be carried out in the design and implementation of adaptive functionality and new interaction styles for music players. A number of applied investigations and controlled user experiments were conducted in support of this, including issues of performance and usability of a new functionality and interaction style.

The first part of the research concerns the design, implementation and evaluation of an adaptive functionality called PATS (Personalised Automatic Track Selection). PATS can be used in two ways. On the one hand, PATS is essentially an automatic music compilation feature for the creation of preferred and varied music programmes. The creation of music programmes requires only minimal intervention from the music listener, as PATS tries to learn about the music preferences of the listener. On the other hand, PATS can be used as a music recommender system, suggesting a few pieces of music to a music listener while carrying out a music programming task. Music suggestions can be used as a supplement to a personally created programme or as a provoker of new ideas to continue a personal search.

The major findings in a user experiment are that PATS is able to create music programmes over time that are highly preferred (7 to 8 out of 10 pieces of music in a programme are preferred), that cover a varied and preferred portion of a music collection, and which adapt to the music preference of a music listener.

The major findings in two other user experiments are that PATS recommendations are highly valued for music programming, because they reduce search effort in a large music collection and they are found useful for a music programming task. A music recommender system can have the undesirable property that they induce a perceived loss to user control of music selection, if recommendations are not adequately presented to the listener.

The second part of the research concerns the design, implementation and evaluation of a multimodal interaction style for music programming, aided by PATS music recommendations. The envisaged home use of the interaction style requires that it can be used, immediately, also when visual inspection of information is impossible, difficult or undesirable (think of the use of a remote control or portable players).

The visual representation of the interaction style resembles a fruit machine consisting of rollers, on which the pieces of music from the music programme, the music collection and the recommendations are presented. User control of the rollers (and, thus, user navigation for listening to and selecting music) proceeds entirely by manipulating a trackball which provides touch feedback. In this way, a music listener has the feeling that he or she really moves rollers. The performance of an action is also marked by a characteristic sound and spoken information is given, for instance, to indicate which roller you are on. The use of speech, sound and touch feedback is primarily to compensate for a possible lack of visual information.

The major findings in a user experiment are that the interaction style can be used instantly, without instructions at the outset, with and without a visual display. The use without a visual display requires more time, as music listeners first need to discover and actively remember procedures.

In general, the current research indicated that both the PATS functionality and its multimodal interaction style can be excellent assets for music players accessing a large personal music collection. The present research also led to or contributed to:

- the use and assessment of inductive learning algorithms to uncover criteria for music preferences;
- the use of situated agent technology in the context of a decentralised cluster-seeking approach, aimed at compiling preferred and varied music programmes;
- the development of re-usable software components for adaptive systems and multimodal interaction;
- the finding that incorporating empirical research by means of user experiments is essential to the design of interactive (consumer) devices.

Samenvatting

Het behoeft geen betoog dat luisteren naar muziek een vrijetijdsbesteding is waar men naar hartenlust van kan genieten. Het kiezen van geprefereerde muziek om te beluisteren in een gegeven situatie kan echter moeilijk zijn, zeker als de keuze aan muziek omvangrijk is. Bestaande muziekspelers zijn helaas niet behulpzaam in het maken van de juiste muziekkeuzes uit een grote muziekcollectie.

Om deze redenen is het centrale thema in dit proefschrift dan ook hoe toekomstige muziekspelers muziekluisteraars kunnen helpen in het selecteren van geprefereerde muziek uit een grote persoonlijke muziekcollectie. Bijzondere aandacht is geschonken aan het programmeren van muziek, wat bekend staat als het selecteren van meerdere geprefereerde muziekstukken om deze in één keer te kunnen beluisteren. Het onderzoek heeft als uitgangspunt de formulering van een tweetal gebruikerseisen waar toekomstige spelers aan moeten voldoen. De eerste eis is dat spelers zich moeten aanpassen aan het muziekkeuzegedrag van de muziekluisteraar opdat ze adequate handreikingen kunnen geven om het muziekkeuzeproces van de luisteraar te vergemakkelijken en te versnellen. De tweede eis is dat spelers muziekinformatie op een dusdanige interactieve wijze moeten presenteren zodat muziekluisteraars gemakkelijk kunnen navigeren in een grote muziekcollectie. Met andere woorden, een muziekspeler moet meer op het individu afgestemd worden (‘personalisation’). Het onderhavige onderzoek richt zich vervolgens hoe beide eisen volbracht kunnen worden in het ontwerp en de realisatie van adaptieve functionaliteit en nieuwe interactiestijlen voor muziekspelers. Een aantal toegepaste studies en gecontroleerde gebruikersexperimenten is uitgevoerd om het onderzoek te ondersteunen, onder meer op het gebied van de prestatie en de bruikbaarheid van een nieuwe functionaliteit en interactiestijl.

Het eerste deel van het onderzoek betreft het ontwerp, de realisatie en de evaluatie van een adaptieve functionaliteit genaamd PATS (Personalised Automatic Track Selection). PATS kan op een tweetal wijzen worden gebruikt. PATS is enerzijds een functionaliteit die automatisch geprefereerde en gevariëerde muziekprogramma’s compileert. De creatie van een programma vereist slechts minimale interventie van de muziekluisteraar omdat PATS tracht de muziekvoorkeuren van de luisteraar te achterhalen. PATS kan anderzijds ook gebruikt worden als een muziekaanbevelingssysteem dat een klein aantal muziekstukken aanbeveelt terwijl de muziekluisteraar zelf muziek programmeert. Muziekaanbevelingen kunnen gebruikt worden om het persoonlijk gecreëerde muziekprogramma aan te vullen of om op nieuwe ideeën te komen terwijl men zoekt.

De belangrijkste bevindingen uit een gebruikersexperiment zijn dat PATS in staat is om meerdere muziekprogramma’s over tijd te compileren die in hoge mate geprefereerd worden (7 à 8 van de 10 muziekstukken in een muziekprogramma worden geprefereerd), die een gevariëerd en geprefereerd deel van een muziekcollectie beslaan en die zich aanpassen aan de muziekvoorkeur van de muziekluisteraar.

De belangrijkste bevindingen uit een tweetal andere gebruikersexperimenten zijn dat PATS muziekaanbevelingen erg worden gewaardeerd tijdens het programmeren van muziek, omdat ze het zoeken in een grote muziekcollectie reduceren en ze als goed van pas worden ervaren tijdens het programmeren van muziek. Een muziekaanbevelingssysteem kan wel de onwenselijk eigenschap hebben dat een zeker verlies in controle over de muziekkeuzes bij de muziekluisteraar wordt teweeggebracht, indien aanbevelingen op een onjuiste wijze worden gepresenteerd aan de luisteraar.

Het tweede deel van het onderzoek betreft het ontwerp, de realisatie en de evaluatie van een multimodale interactiestijl voor het programmeren van muziek met behulp van PATS muziekaanbevelingen. Het beoogde huiselijke gebruik van de interactiestijl vereist dat het onmiddellijk gebruikt kan worden, ook indien visuele inspectie van informatie onmogelijk, moeilijk of onwenselijk is (denk daarbij aan het gebruik van een afstandsbediening of een draagbare speler).

De visuele representatie van de interactiestijl lijkt op een fruitmachine bestaande uit rollers, waarop de muziekstukken uit het muziekprogramma, de muziekcollectie en de aanbevelingen worden gepresenteerd. De bediening van deze rollers (en dus het navigeren voor het beluisteren en het selecteren van muziek) verloopt geheel door het manipuleren van een trackball die terugkoppeling geeft via de tastzin. Zodoende heeft een muziekluisteraar het gevoel dat hij of zij daadwerkelijk rollers beweegt. Daarbij wordt het uitvoeren van een handeling gemarkeerd door een karakteristiek geluid en wordt er gesproken informatie gegeven over bijvoorbeeld op welke roller men zich bevindt. Het gebruik van spraak-, geluids- en tastzinsterugkoppeling in de interactiestijl is in eerste instantie bedoeld om het mogelijke gemis aan visuele informatie te compenseren.

De belangrijkste bevinding uit een gebruikersexperiment is dat de multimodale interactiestijl onmiddellijk gebruikt kan worden, zonder instructies vooraf, met en zonder een beeldscherm. Het gebruik zonder beeldscherm vergt wel meer tijd omdat muziekluisteraars allereerst de benodigde handelingsprocedures moeten ontdekken en eigen maken.

Uit dit onderzoek is gebleken dat zowel de PATS functionaliteit als haar multimodale interactiestijl excellente aanwinsten kunnen zijn voor muziekspelers die toegang bieden tot een grote persoonlijke muziekcollectie. Het onderzoek heeft tevens geleid tot of een bijdrage geleverd aan:

- het gebruik en de evaluatie van inductief lerende algoritmen voor het onthullen van criteria voor muziekvoorkeuren;
- het gebruik van gesitueerde agenttechnologie in de context van een gedecentraliseerde clustermethode gericht op het creëren van geprefereerde en gevariëerde muziekprogramma’s;
- de ontwikkeling van herbruikbare softwarecomponenten voor adaptieve systemen en multimodale interactie;
- de bevinding dat het verweven van empirisch onderzoek in de vorm van gebruikersexperimenten binnen een ontwerpproces voor interactieve (consumenten) apparaten van essentiëel belang is.

About the author

Steffen Pauws is a computing scientist with some applied specialities and interests.

He studied and graduated in Medical Informatics at the University of Leiden from 1986 to 1991, with an emphasis on biomedical signal analysis. Subsequently, he started the post-master programme Software Technology (OOTI) at the Stan Ackermans Institute (SAI) at the Eindhoven University of Technology (EUT). He completed his graduate design project for this programme, concerning automatic speech recognition, in 1993 at IPO, Center of User-System Interaction, EUT. From 1993 to 1994, he completed his alternative national service at IPO. During that period, he researched and developed a method for automatic speech segmentation and, in the meanwhile, started to explore the field of user-system interaction. Captivated by the latter field of research and his admiration for music, he started the Ph.D. research as described in this thesis, at the end of 1994.

Currently, he researches the field of speech interfaces at IPO.