Diplomarbeit

Generative Music for Media Applications

ausgeführt zum Zweck der Erlangung des akademischen Grades eines
"Diplom-Ingenieurs für technisch-wissenschaftliche Berufe"
am Masterstudiengang Telekommunikation und Medien
der Fachhochschule St. Pölten

unter der Erstbetreuung von
Dr Martin Parker (University of Edinburgh)

Betreuer an der Fachhochschule St. Pölten
FH-Prof. DI Hannes Raffaseder

Zweitbegutachtung von
Mag. Michael Jaksche, MA

ausgeführt von
Julian Rubisch, tm0710262057

St. Pölten, am 26. 1. 2009 Unterschrift

Ehrenwörtliche Erklärung

Ich versichere, dass

- ich diese Diplomarbeit selbständig verfasst, andere als die angegebenen Quellen und Hilfsmittel nicht benutzt und mich auch sonst keiner unerlaubten Hilfe bedient habe.
- ich dieses Diplomarbeitsthema bisher weder im Inland noch im Ausland einem Begutachter oder einer Begutachterin zur Beurteilung oder in irgendeiner Form als Prüfungsarbeit vorgelegt habe.

Diese Arbeit stimmt mit der von den Begutachter/inne/n beurteilten Arbeit überein.

Ort, Datum	*Unterschrift*

Abstract

Finding or composing suitable media music often poses a serious challenge to media producers. Primarily low-budget productions like image videos, presentations of SMBs, TV features in regional programmes or interactive media applications like computer games often lack know-how and/or resources to have adequate media music composed for the application in question. Subsequently, in many cases royalty-free music libraries are employed as a fallback, which is why the resulting sound tracks are often characterized by poor quality and recognition value.

Therefore, a large amount of thematic relevance and economic benefit lies in the development of a tool that simplifies the automated generation of media music, in order to produce – musically – richer sound tracks while at the same time staying completely royalty-free.

Hence, this thesis investigates functions, impacts and possible taxonomies of media music and evaluates its emotional and semantic content. Subsequently, emotive functions of media music are looked into more deeply, considering them one of the most valuable factors for the categorization of this type of music. Moreover, a model for the musical representation of emotional or affective states is introduced using Russell's circumplex model of affects [Rus80] and based on state-of-the-art research on this topic [BW05, BW06, Bru07].

Within the second half of the thesis, algorithms suitable for automated composition of music are investigated, with a special focus on the ability to map the musical parameters derived from the above-mentioned emotional model onto the algorithms' input parameters. In addition, using the most applicable methods for automated composition, a prototypical application is developed in Max/MSP, implementing a user interface based on the circumplex model of affects and thus enabling smooth temporal transitions between different emotional or affective musical expressions.

Kurzfassung

Das Auffinden oder Komponieren von passender Medienmusik stellt für Medienproduzenten oftmals eine nicht zu unterschätzende Herausforderung dar. Vor allem Low-Budget-Produktionen, wie etwa Image-Videos, Präsentationen von KMUs, regionalen TV-Sendungen sowie interaktiven Medienanwendungen wie z.B. Computerspielen fehlt es oft an Know-How und/oder Ressourcen, um adäquate Medienmusik speziell für das jeweilige Produkt komponieren zu lassen. Aus diesem Grund wird oft auf lizenzfreie Musikbibliotheken zurückgegriffen, was meist zu klischeehaften Tonspuren mit geringem Qualitätsanspruch sowie Wiedererkennungswert führt.

Daraus ergibt sich die thematische und wirtschaftliche Notwendigkeit der Entwicklung eines Tools, das die automatisierte Generierung von Medienmusik ermöglicht, und in der Lage ist, musikalisch interessante, abwechslungsreiche, und dennoch lizenzfreie Tonspuren hervorzubringen.

Die vorliegende Diplomarbeit untersucht deshalb Funktionen, Wirkungen und mögliche Kategorisierungen von Medienmusik. Weiters werden emotionale Funktionen von Medienmusik genauer analysiert, sowie ein Modell der musikalischen Repräsentation von Emotionen oder Gefühlszuständen, welches auf Russells Circumplex Model of Affects [Rus80] und modernsten Forschungsergebnissen zu dieser Thematik [BW05, BW06, Bru07] basiert, vorgestellt.

In der zweiten Hälfte dieser Arbeit werden geeignete Algorithmen zur automatisierten Komposition von Musik betrachtet. Dabei wird besonderes Augenmerk auf die Eingabeparameter dieser Algorithmen, und deren Verwendbarkeit in Hinblick auf die musikalischen Parameter des zuvor vorgestellten emotionalen Modells gelegt. Des weiteren wird unter Verwendung der am geeignetsten erscheinenden Algorithmen eine prototypische Applikation in Max/MSP entwickelt, deren User Interface auf dem Circumplex Model of Affects basiert, und dadurch stufenlose Übergänge zwischen unterschiedlichen durch Musik ausgedrückten Gefühlszuständen erlaubt.

Acknowledgements

First and foremost, I want to express my sincere gratitude to Dr Martin Parker, Programme Director of the MSc Sound Design course at the University of Edinburgh, for providing guidance and scientific expertise throughout the process of writing this thesis. It was his decision to accept the supervision of this thesis which assured me that it was doable and worthwile.

Furthermore, I would like to thank FH-Prof. DI Hannes Raffaseder, Programme Director of the Telecommunications & Media Masters course at the Fachhochschule St. Pölten, and my second supervisor based in Austria, for his support and advice concerning media music theory.

I would also like to mention the professional and inspiring input of Mag. Michael Jaksche, MA and DI (FH) Matthias Husinsky, who revealed their most welcome thoughts and opinions during uncountable coffee breaks. In addition, I would like to thank Mag. Rosmarie Tomasch for proofreading parts of the thesis, and Scott Beveridge, PhD student at the School of Engineering & Computing of Glasgow Caledonian University for sharing his thoughts with me on a trip in a Swedish cab.

Special thanks go to my family, especially my parents, for providing mental support and belief in my abilities throughout the course of my studies.

Last but not least, I want to express my deep gratefulness to my beloved partner Barbara, who has been supporting me in any situation for the past 31 months, including the origination process of this thesis.

St. Pölten, Austria — Julian Rubisch

January 26, 2009

Contents

1. Introduction

1.1. Problem Definition

Creation and reception of music have altered in a drastic manner over the last 150 years. Before the invention of audio recording equipment such as the phonograph, every sound event was bound to one specific moment in time - the time it was performed. Literally, nobody was able to hear the same piece of music twice (see [Eno96]). In the following decades, recorded sound became a mass media because of the availability and popularity of recording devices and the possibilities they came up with. It spawned a whole recording and broadcasting industry, and the number of possible applications for recorded sound started growing exponentially (see [Win04], p. 3).

However, with the advent of electronic music and sound synthesizers in the second half of the 20^{th} century, the habits of creating and consuming music underwent another critical transformation. Suddenly, using techniques from human-computer-interaction, the listener is in the position to directly influence the outcome of the piece he is paying attention to. The composer of such a piece of music acts more like a supervisor, only predefining certain structures and limitations to his work of art, thus opening it to the audience and including the listener in the process of finalizing the artifact (see [Ess91]). In this context, Atau Tanaka ([Tan06], p. 289) states that

> "[...] interactive systems and networks are technologies that exhibit this expressive, instrumental potential. The creative process is completed when the listener enters the loop. It is only then that expression takes place, as the sum total of the satisfaction lived out by artist, instrument, and listener."

Indeed computing power nowadays isn't a serious problem anymore, and with upcoming novel mobile devices, many implementations of generative approaches are imaginable. For instance, sensory inputs such as accelerometers, temperature sensors and related devices originate the development of interactive generative audio applications such as *RjDj*[1] or Brian Eno's *Bloom*[2].

At the same time as the boom times of digital media took place, a whole new industry engaging in the utilization of these new possibiltites was born. While it took the classic media like film, radio, or television years to adapt to these conditions, other communities, such as the world wide web, the computer gaming industry or multimedia artists managed to pick them up and adopt them for their needs rather quickly.

Consequently, generative music approaches seem to slowly begin finding their way into those so-called *new media*. Unfortunately, the attempts to adapt algorithmic methods for their needs have been highly experimental thus far. Therefore, it seems reasonable that research and development in this field have to be intensified in order to result in true benefit for the media industry.

1.2. Motivation

Current research and development in the fields of generative music and algorithmic composition indicate a great potential for these rather experimental forms of musical expression to be used as the basis of an interactive tool for generating sound tracks for media appliances. The need for such a tool derives from many media producers' every day struggle – producing content at high quality and low cost at the same time. While it seems to be common practice to employ royalty-free music libraries for purposes such as overdubbing an image video, creating musical background for a museum installation or generating a sound track for public presentations, the outcome of such a project is often poor in quality and recognition value.

Particularly where sound design for interactive media is concerned, until now in most cases audio loops are employed to populate the sound track. Often this happens out of lack of computer memory or because of outdated design patterns (or those adopted from time-based media such as film or television without adaptation). Nevertheless, mostly the results sound repetitive, and consequently boring, as the

[1] http://rjdj.me/
[2] http://theappleblog.com/2008/10/08/godfather-of-ambient-creates-iphone-app/

central concepts of dramaturgy – such as rises and falls of tension, or climaxes – cannot be applied to the product in an adequate manner (see [Raf02], p. 295). The ability to control small-scale as well as large-scale musicality would add considerable individuality to such a multimedia product.

1.3. Relevance

In contemporary literature on this topic, it is often remarked that individual musical experience is suffering from the ubiquitous ability to access and exploit music (see [Tan06], p. 271). Similarly, it has been noted that due to the excess of music omnipresent in the mass media, especially children's understanding of music is more and more shaped early by the influence of the musical mainstream conveyed by this mass culture, and thus narrowed to a passive stage of pure consumption (see [Bur98], pp. 249f). Recent game audio studies ([CGH06], p. 9) even conclude that

> "the audio factors in games tend to act as background fillers."

Among other approaches, adaptive or generative concepts are one means to address this problem, as they involve the listener in the creation of the auditory content he or she is confronted with.

Additionally, an even larger problem has to be solved: The huge amount of content produced, not only in the computer gaming industry (see [Far07], p. 76), but also by legions of media producers and/or artists working on a low-budget basis around the world, still needs a decent sound track.

1.4. Thesis Objectives

The following chapter focuses on media music, the fundamental principles of its perception, functions and impacts, possible applications, as well as common design patterns. By reviewing state-of-the-art research on the topic and analyzing relevant literature, a possible generalized taxonomy for the categorization and description of media music is developed.

In the third chapter, emotive functions of music are investigated more profoundly, since they are considered to be one of the most valuable factors for the description of music and contribute to the narrative content and style of media products to a large extent. Useful parameters for emotional classification of musical material are identified; furthermore, a model for the musical representation of affective states or moods is developed which thereafter leads to the determination of prerequisites for a prototype implementing generative techniques.

Afterward, several common algorithms suitable for the automated generation of music are examined, focusing on the ability to map the musical parameters discovered in the preceding chapter onto the input parameters of one of those algorithms, or a combination of them. A special emphasis is put on the possibility to realize smooth temporal transitions by altering the afore-mentioned input parameters.

Eventually, after reviewing related work in this research field, a prototype implementing an optimized user interface based on Russell's circumplex model of affects [Rus80] is developed in Max/MSP[3]. It is meant to enable the user to intuitively control the musical outcome in terms of its emotional or affective content. The intention here is to devise a prototype that is capable of producing structural audio information such as MIDI or OSC as an output, which can for instance be imported into any digital audio workstation (DAW) – such as Digidesign ProTools[4], Apple Logic[5] or Steinberg Cubase[6] – or played back by any sound engine capable of doing so, whereas it is not the aim of this thesis to construct a sound synthesizer that produces audio output, e.g. in WAV or AIFF format, on its own.

The music created by the prototype is analyzed with respect to the question if it fits the specified musical parameters. The aesthetical value of this music is then evaluated by a subjective critique as well as expert interviews, so as to clarify whether the musical output indeed represents the emotions or moods it is intended to. The thesis concludes with a closing evaluation of the research questions introduced in this chapter, as well as remarks on future work that is to be performed on this topic.

[3] http://www.cycling74.com/products/max5
[4] http://www.digidesign.com/index.cfm?navid=349&langid=100&itemid=33116
[5] http://www.apple.com/de/logicstudio/
[6] http://www.steinberg.net/de/products/musicproduction/cubase4_product.html

2. Music for Media Applications

2.1. Applications

To encompass the field of media music, and to provide the background for a more detailed examination of perceptional concepts, functions and design patterns, it seems advisable to begin this chapter with an outline of its possible applications. Characteristics and design issues of those applications are investigated, as well as possible starting points for generative approaches.

2.1.1. Auditory User Interfaces and Displays

Auditory user interfaces are human computer interfaces which employ audio signals to provide feedback and user guidance. Quite similar, auditory displays are devices which primarily use sonic events to communicate relevant content (see [Raf02], pp. 288f). It has to be remarked, though, that both terms are used ambiguously in literature, and many concepts seem to coincide.

Both appliances use *Auditory icons and Earcons*[1] to transport user information. Especially earcons could benefit from the additional value of emotionally designed content (see [Raf02], p. 289). Thus, for example it could be ensured that auditory displays do not sound enervating when they should in fact calm down the user, but at the same time are clearly perceivable (see [Raf02], pp. 292f). Furthermore, reactive approaches to auditory user interfaces could raise the user's immersion and participation in the respective application (see [Win98], p. 297).

2.1.2. Infotainment / Advertainment / Edutainment

In the past two decades, multimedia systems have achieved general acceptance as a platform for the distribution of information, be it in a commercial, educational or

[1] see glossary entries

cultural context. Numerous examples of image videos, museum installations, interactive gaming appliances (e.g. iFUN[2]) and multimedia-based product presentations display the potential of audio-visual techniques to efficiently communicate whatever content in an entertaining way.

Generative approaches could add an additional feeling of variation to the musical sound track of such products in order to avoid repeating sequences, and to add to their perceived realism (see [Win98], pp. 296f). A disadvantage could be the fact that by doing so, the informational content could be corrupted or subjectified.

2.1.3. Reactive and Interactive Media Applications

In the recent past, reactive and interactive media applications have gained popularity by being used as public platforms for info-/advertainment (see above), or personal entertainment systems, such as *RjDj*[3].

The world wide web, even though it could be considered the prototypical interactive media, lacks a coherent system of audio design patterns. Many websites or web applications feature a sound or musical design which can be turned off by the user. Even though this seems to be a usable feature, it also indicates that the audio design used in such applications isn't as elaborate as it could be (see [Raf02], p. 296).

Computer games, though a comparatively old application of interactive media techniques, are also worth being studied concerning their formal dramaturgical design, which of course differs widely from that of traditional, linear screen-based media (see [Raf02], pp. 294f).

In such non-linear media, mostly audio loops are used to form the sound track out of lack of computer memory (see [Raf02], pp. 295f). In the special case of music this approach entails the danger of leading to monotonous and repetetive tracks. On the other hand, an adaptive approach towards music production can result in dramaturgically more sophisticated applications (see [Raf02], p. 297 and [BH07], p. 142).

[2] http://www.ifun.at/
[3] http://rjdj.me

2.2. Perception of Media Sound Tracks

When reflecting on functions and impacts of products designed for auditory (or audio-visual) consumption, it seems critical not to omit the possibilities and limitations that the human sense of hearing brings about. The following introductory observations concerning state-of-the-art research on auditory perception are intended to serve as a basis for further contemplations on the structure of media sound tracks and the resulting consequences for media music.

2.2.1. Relevant Qualities of Human Auditory Perception

Observing the perception of musical content, it is inevitable to study the basic principles of auditory perception and the consequences that derive from them. The human ear, quite contrary to the eye, is a passive sense organ, which itself is immovable, barely directional and cannot be closed (see [Sch97], p. 31). As a direct consequence, a large part of auditory perception takes place rather unconsciously (see [Raf02], p. 249), even during the sleeping period. According to film sound designer Walter Murch (see [Mur05b] and [Sch97], p. 34), the human sense of hearing is even the first one to be activated in an unborn, embryonal state.

Furthermore, the ear is directly linked with parts of the diencephalon: the thalamus and the limbic system, both responsible for the evocation of feelings and emotions (see [Sch97], p. 31), which is why acoustic impressions can also directly cause somatic reactions (see [Raf02], p. 250 and [Chi94], p. 34). Considering prehistorical circumstances, this does not come as a surprise, since hearing was the only sense capable of warning you against the threat of an animal attacking from behind.

Moreover, sound events are perceived in a multi-dimensional manner, which has drastic consequences for media products, because several sound tracks can be superimposed upon each other without becoming unintelligible: movie characters' dialogue can be understood while music is played in the background and street ambience defines their surroundings (see [Raf02], pp. 350f). Regarding music as a special audio event, it is for instance possible to express several different emotional states, or moods, in one piece of music (see [Sch97], p. 32).

Taking all these factors into account, it almost stands to reason that hearing leads to an emotional and unconscious comprehension of the surrounding world. Thus, auditory perception conveys an image of the inner (i.e. emotional or material) state of lifeless as well as alive objects (see [Sch97], p. 31). The primal nature of hearing is also the reason why human beings instinctively rely on aural impressions as opposed to visual ones when their contents differ from each other. In the hands of a sound designer or composer, this becomes a powerful tool, because the meaning of certain images can be directly influenced by sound (see [Chi94], p. 34), particularly by music (see [Sch97], p. 34).

2.2.2. Auditory Perception Modes

As mentioned above, auditory perception occurs anywhere and anytime, actively and passively, multi-dimensionally, emotionally and rationally. In contemporary literature on this topic, there have been several efforts to combine the relevant aspects into *modes*, of which the most notable ones are outlined in this section.

Listening Modes (Schaeffer / Chion)

The way in which human beings listen to sonic events, or rather how they afterwards describe what they have heard, strongly depends on the intellectual immersion with which the subject is contemplating the sound at hand.

In this context, Michel Chion ([Chi94], p. 25) postulates that

> "[...] there are at least three modes of listening, each of which addresses different objects."

This section will include an overview of those three modes, and an evaluation regarding the impacts on the special case of music.

Causal Listening The purpose of listening causally to a sound is, in the first place, to obtain information about the source and its composition. In many cases, the characteristics of a sound produced by a well-known source can provide additional details necessary to interpret it correctly (see [Chi94], pp. 25f).

However, this type of listening underlies some considerable limitations: Objects that sound similar or belong to one class of objects - such as dogs, or vacuum cleaners - are grouped unconsciously and merged into categories; a typical listener may even not be able to tell a bulldog's bark from that of a poodle anymore. Thus, in identifying the cause of a sound, we tend to abstract from the unique to the general (see [Chi94], p. 26). In very ambiguous cases, we even settle for only connecting a sound to the general nature of its cause, e.g. *animal sounds*, *human sounds*, *mechanical sounds*, and so on.

Furthermore, a sound mostly consists of more than one single source. Regarding musical instruments, for example, there is mostly an exciting part and an oscillating part involved in the process of generating sound (see [Chi94], p. 28). Thus, the process of identification becomes multi-dimensional, involving the recognition of each part of the source.

An important implication of these features is that especially in movies there is a large potential of manipulating the causal listening mode under the limitations that the phenomenon of *Synchresis*[4] brings about (see [Chi94], p. 28). There, in many cases, we identify causes of sounds that actually are not their causes – the most obvious example for that would be that of science-fiction movies, where we are often dealing with objects that do not even exist, like space ships or laser guns, yet possess a typical sound.

Semantic Listening Whenever knowledge of a code or language is necessary to interpret an acoustic event, semantic listening takes place (e.g. speech, Morse code or similar forms of communication). The most characteristic attribute of this listening mode is that the interpretation of sonic events happens differentially. That is to say, when receiving and deciphering a coded message, the specific acoustic properties are largely ignored, only differences between sounds are examined and used to identify for example a certain word (see [Chi94], p. 28). Semantic listening applies to music regarding cultural differences: different cultures have developed various *languages* of music (see [Mur05a]), as have different media applications. In radio, television and advertising, for example, different musical gestures evolved that carry a certain meaning (e.g. radio jingles, audio logos, series signations and so on). Certainly, such structures can only emerge by educating the audience in order to enable them to decipher the

[4] see glossary entry

semantic code. Any human being who isn't familiar with the common codes included in the music of television or radio commercials, for example, will not hear any meaning enclosed in it. Because of the intention to design a prototypical generative application suitable for any kind of media music, semantic listening considerations are not taken into account in the following chapters.

Reduced Listening In the last listening mode proposed by Chion, the attention of the listener is drawn upon the characteristics of the sound itself, instead of its cause or its meaning. The sound is thus reduced to its signal properties, and suddenly most listeners find it hard to describe what they hear without employing its source, meaning or effect (see [Chi94], p. 29). Indeed, it seems evident that verbal taxonomies used for the classification of sounds either refer to their causes or impacts, or use musical terminologies that aren't apt for this kind of categorization (see [Chi94], p. 31).

Therefore, to support efforts of strengthening reduced listening as a technique for analyzing auditory content, a novel type of terminology is required, emphasizing on sound-inherent qualities. What pertains to music, or sound art in general, is that the aesthetic as well as emotional value of a sound is closely linked to its timbre and texture rather than to the causal description attached to it (see [Chi94], p. 31).

Emotional vs. Rational Perception

One of the central questions of sound design and music is, how many different high-level sonic structures can be perceived and interpreted simultaneously. At first, one might think of an acoustic event's signal spectrum as a meaningful parameter, as it defines a sound's timbre and makes it identifiable. However, human beings are able to distinguish between several sound sources that have similar spectral characteristics. For example, different instruments that have a similar tonal range (e.g. violins and flutes) can easily be identified by a human listener. On the other hand, when there are more than two such structures involved, as in an orchestra, they merge into one higher-level unit (see [Raf02], p. 259). Film sound designer Walter Murch therefore stated that approximately 2.5 high-level sonic structures, called *streams*, are discriminable under normal circumstances (see [Mur05a]).

In this context, Murch proposes an alternative classification of sonic events into *encoded* and *embodied* sounds (see [Mur05a]). Encoded sounds have to be decoded rationally after their reception, they don't convey any message unless the receiver knows how to decipher the encoding[5] – think of speech, for example, with the grammar and vocabulary attached to a specific language (see [Raf02], p. 259). The meaning of embodied sounds (for example, music), on the other hand, can be understood directly and intuitively, by using emotional categories.

The majority of acoustic events is of course situated somewhere between these two poles. Apart from entirely neutral sounds that can be regarded as their central point, there are many hybrid structures surrounding us in everyday life. Some sonic events do carry a certain coded message, such as the ringing of a telephone or the sound of a siren, whereas others, like animal sounds or church bells, are perceived more instinctively and interpreted emotionally (see [Raf02], p. 260).

An important consequence from these findings is, that when the sound tracks of a media product are evenly spread between embodied and encoded sounds, the maximum amount of discriminable streams can be extended to approximately 5 (see [Mur05a]). As might be expected, this contributes to an effect that we experience in movies every day: We are able to listen to dialogue, sound effects, ambient sounds and background music at the same time, without any difficulty to understand and intepret these different levels of acoustic communication.

2.2.3. Acousmatic Sound Perception

Another intrinsic feature of sound accompanying almost any media product, is that of *acousmatic* sounds and how they are perceived and interpreted. The term acousmatic is meant to define a sonic event whose source you cannot see (see [Sch67], pp. 91-99), which of course relates to most audio-visual artifacts. Media such as radio, compact disc or telephone are all regarded as being acousmatic (see [Chi94], p. 71), as the originating cause of sound cannot be seen by the recipient. Regarding film or television, every sound whose source is acting off-screen, is acousmatic by definition, which obviously also includes music underscoring what can be seen on the screen. Only very rarely, there is a visible source of music; nevertheless film music

[5]The concept of encoded sounds is of course closely related to Chion's above-mentioned semantic listening mode, and all its consequences.

is completely accepted by the audience as an inherent part of the movie that can have a significant impact to the film's narrative, even though it does not belong to the diegesis[6].

Many functions of media music that are described in this chapter derive the power they exert on the listener from this phenomenon. Music accompanying a motion picture according to Chion ([Chi94], p. 81) is

> "a little freer of barriers of time and space than the other sound and visual elements."

Music can even switch from the non-diegetic to the diegetic realm at a moment's notice (imagine a piano player appearing on the screen after he or she has started playing) without any difficulty for the audience to integrate this experience with the ongoing storyline (see [Chi94], p. 81). Eventually it can be stated that it is the atypical characteristics adhering to acousmatic sound perception that allow for several specific effects of media music which will be elaborated on in the following sections.

2.3. Value Added by Music

What value, or in what way value can be added to an audio-visual product is defined by how much the music layer correlates with the visual layer. Music can, for example, take on a scene's tempo and phrasing, thus emphasizing what can be seen on the screen; that is to say, it can act *empathetically* (see [Chi94], p. 8). In a sense, this type of music is tightly attached to the picture, hence perceived rather unconsciously and interpreted instinctively (see [Sch90], p. 79).

On the other hand, music can also take on a different rhythm and emotional meaning than the scene it accompanies; in that sense, it acts *anempathetically* (see [Chi94], p. 8) and needs more intellectual involvement with what is being heard, in order to understand the meaning(s) transported by it (see [Sch90], p. 80).

[6] *Diegetic sounds* are sounds whose cause belongs to the context of the storyline of the movie, whereas *non-diegetic sounds* are those who cannot be explained directly from what is happening on the screen, but comment on what what is seen or communicate emotions (see [Raf02], p. 262)

These two forms of usage of media music delineate the general field of music dramaturgy in which media composers work. In this section, the general methods of applying this concept to add value to a media product are investigated.

2.3.1. Unification

Sound in film or television acts unifyingly, as the motion picture is by definition disrupted by the editing process of putting in cuts and transitions. Especially non-diegetic music can contribute to the unity of an audio-visual artifact by providing an over-all, homogeneous motif which enables identification of this particular work (see [Chi94], p. 47).

This effect directly corresponds with, or is supported by the above mentioned concept of audio streams, representing high-level sonic structures that are perceived as a continuous flow of auditory information. The human sense of hearing constantly tries to identify such streams which structure and arrange his auditory surroundings (see [Raf02], p. 254). Hence, when certain design patterns that are described later on in this chapter are fulfilled, the process of unification takes place automatically.

2.3.2. Punctuation

Here, the term *punctuation* is not used in the linguistic sense of the word, but to express the technique of adding a certain flavor to an audio-visual product, that is, underscoring or contradicting the visual content of certain scenes. Whether there is a difference between what is seen on the screen or not determines the amount of information delivered by the sound track. Hence, punctuative use of music becomes an important design tool so as to add value to a multimedia product (see [Raf02], p. 266).

In contemporary literature, several attempts have been made to introduce a taxonomy for the classification of punctuative media music (see [Sch97], p. 79). For this thesis, the following model is regarded as the most relevant one.

Paraphrase

By far, the practice of directly translating the picture's content into corresponding sound or musical expressions, called *paraphrasing*, is the most prevalent one in contemporary film and television productions (see [Raf02], p. 267).

One of the key reasons for this has remained the same since the very beginnings of sound film: The clarification of the picture's contents for the audience. When sound in films first came up, the recipients were unfamiliar with the new medium, and untrained in interpreting the auditory content. This fact, and the technical shortcomings of optical sound made it necessary to double-code objects and plots by paraphrasing them (see [Flü01], p. 136) by means of sound design and film music.

This redundancy of information confirms the recipient's anticipation (see [Raf02], p. 267 and [Chi94], pp. 55f) and is nowadays often utilized in television series. Because television has more and more become a secondary medium which is viewed alongside other everyday activities, such as reading, cooking etc., musical paraphrasing of visual content has gained importance again. More precisely, the viewer/listener has to be kept informed about the ongoing storyline even though he or she may not be paying full attention (see [Sch97], p. 25).

Yet even in feature films, there are occasions when the use of paraphrases is not only justified, but necessary: When, especially at the climax of a movie, the intention of the story is to spark one strong emotion, and all the other dramaturgic means (acting, photography, editing etc.) point towards it, musical paraphrasing is of course useful to support this movement (see [Sch97], p. 25).

Finally, paraphrasing is sometimes used to caricature or satirize certain scenes (see [Sch97], p. 25). This tradition is well known from cartoon movies, when, for example, Tom the cat is chasing Jerry the mouse and the background music is mimicking their running exaggeratedly. This, inter alia, has led to the tradition of *mickey mousing* in animated films, where music and movement are welded together by several sync points (see [Sch97], p. 71) in a way that is typical for animated films, but nowadays also common in non-animated cinema (see [Chi94], pp. 121f).

Polarization

Neutral or ambivalent image contents attain higher-order semantic meaning when they are polarized by the underlying music or sound track (see [Raf02], p. 267). Thus, even inanimate elements of a setting, such as objects, can gain a certain significance: A door can appear to be hopeful, or menacing, depending – among other things – on the music accompanying the picture (see [Sch97], p. 24). In other words, if there is a discrepancy between the visual characteristics of an otherwise neutral object or setting, and the emotional or semantic qualities that are intended to be conveyed, polarization helps bridging this gap (see [Flü01], p. 144).

Another example are clips from archive videos which normally would only make sense in their original context. By applying polarization to them, they can be interpreted by the audience in a totally different context (see [Raf02], p. 267).

Dissonance / Counterpoint

The concept of *audiovisual dissonance* or *counterpoint* dates from a figure of speech named *oxymoron*, where a linguistic construct of contradictory meanings creates a tension that is used to call the reader's attention (see [Sch97], p. 26) – for example "deafening silence". In a similar manner, when the visual and the sound layer of a multimedia product are incongruous, this is likely to wake the viewer's/listener's curiosity, he or she has to get involved with the situation and draw his or her own conclusions (see [Raf02], pp. 267f and [Sch97], pp. 25f).

Dissonance can transport additional information by utilizing the difference between visual and auditory content, which has to be solved by the recipient using his or her own experiences and fantasy (see [Raf02], pp. 267f). The musical layer then has to be conciously perceived and interpreted, which is why this audio-visual relation is often regarded as the most valuable in an artistic sense (see [Sch97], p. 25). Chion ([Chi94], p. 56) concludes that

> "it is often more interesting when the expectation is subverted."

However, there are drawbacks included in this stylistic device which have to be considered when using it. First of all, the information that is intended to be transported can only be precisely evaluated in the overall dramaturgic context of the artifact (see [Sch97], p. 27), otherwise false conclusions are bound to occur. In fact, when the contradiction implied by audiovisual dissonance turns out to be too radical, an associative connection between the two layers becomes impossible (see [Raf02], p. 268), and their meanings remain separated.

Another form of criticism deals with the “binary logic” ([Chi94], p. 38) that some applications of audiovisual counterpoint impose on the recipient, because in order for it to take effect, the visual and auditory elements involved have to be associated with certain, often stereotyped significances (see [Chi94], p. 38). Consequently, when used excessively, the resulting products are likely to focus on some special target group because of the amount of previous knowledge necessary to decode the inherent contradictory messages (see [Raf02], p. 268).

As a concluding remark it has to be mentioned that either of these three concepts alone would be dramaturgically useless (see [Sch90], p. 90). Quite contrary, the possibilities these different approaches offer are only taken advantage of when they are combined in a dramaturgically sensible manner.

2.3.3. Aural Perspective

The emotional dramaturgy of a motion picture or video is often determined by the perspective from which the characters aurally perceive their surroundings. Similar situations are experienced differently according to the emotional state of the protagonist (see [Raf02], p. 261); his or her physical and mental conditions are reflected by what he or she hears in a specific moment (see [Raf02], p. 262). Such a subjectifying sound track, for example, would remove most background sound effects and noises and focus on music (see [Raf02], p. 263). This is because in emotional moments, the surroundings become less important, and the auditory contact with the environment disappears more and more (see [Flü01], p. 397). Thereby, relations between the dramatis personae and the audience are established (see [Raf02], p. 263); especially the protagonist’s emotional world becomes interesting in this context, because it can promote the audience’s identification with him or her (see [Sch97], p. 15).

It seems crucial, though, that this subjective, emotional aural perspective be used only carefully and in turn with a more objective, documentary perspective (see [Raf02], p. 263), because with its focus on the outward reality, the latter serves as a vessel for the movie’s authenticity and plausibility, which must not be neglected, after all (see [Sch97], p. 17). Again, it is vital to understand the aural perspective of an audio-visual artifact as a dramaturgic means to underline the characters' emotions and explain their trains of thought to the audience. It is, however, equally important to describe a movie's setting in a more factual, informative manner so as to clarify that whatever is happening on the screen could have happened in reality likewise.

2.4. Functions & Impacts of Media Music

Assembling all the major observations from the preceding chapters, they imply that perceptional principles and intercultural concepts pertaining to music for media applications lead to certain functions it embodies. While functions of film music are well documented in literature (see, for example, [Win04], [Sch97], [Sch90]), and some studies adapt them for game audio (see [Jør06], [ENF06]), the functions of music in non-linear, interactive media seem to be overlooked.

To structure the existing, exhaustive enumerations of media music functions (see [Sch97], pp. 67f, [Sch90], pp. 89ff, [Raf02], p. 274), the Swedish film composer and musicologist Johnny Wingstedt proposed a scheme to organize narrative functions of film music into classes and categories (see [Win04], pp. 3f). This model is compared to other relevant studies and evaluated as for its appropriateness concerning non-linear, interactive media.

It seems obvious that regarding novel communication forms such as the Internet or computer games, music evolves from a special social and cultural context which is largely influenced by the degree of interactivity and immersion with the medium (see [Win04], pp. 3f). A clearly defined context for music production and reception leads to a distinct musical message (see [Win04], p. 4). For example, film is nowadays a very common experience, which is why film music is often used to transport well-defined meanings; the same applies to television and commercials (see [Win04], p. 4). As for the *new media*, the sociocultural context isn't nearly as established, which is why many musical functions have ambiguous character. This section tries to identify those functions and propose methods to circumvent their weaknesses.

Wingstedt ([Win04], pp. 6ff) suggests the following six classes of narrative functions of music.

2.4.1. Emotive Class

As argued in section 2.2.2 (p. 10), music is always perceived and interpreted emotionally, which indicates that emotive functions are inherent in all kinds of music. Regarding functions in media products, it can be divided into mood induction (the influence music has on the audience's emotional experience) and communication of emotion (where the feelings of a certain movie or game character are expressed) (see [Win04], p. 6).

In computer game applications, emotive fumctions exhibit even larger significance, since they can directly influence the player's behavior, and vice versa (see [Jør06], p. 50). The same applies to interactive media, even though the importance of emotional immersion of the listeners has to be evaluated from case to case. That is to say, for example in a multimedia installation for a museum or an exhibition, informative functions may be considered more essential than emotional ones, while for interactive product presentations or commercials, it may be the other way round.

2.4.2. Informative Class

Quite contrary to the preceding class, informative functions of music rely on the recipient to be able to decode and understand the message it is meant to deliver. Nevertheless, the transmission of information by certain musical structures seems very common in many media contexts such as film or game audio (see [Win04], p. 6f, [Jør06], p. 50).

Recurrent usages involve the communication of a certain meaning or certain values, or the establishing of recognition (see [Win04], pp. 6f). For example, music can be used to evoke a certain time period or cultural setting, even though this function should be used carefully, as it is prone to result in stereotypical sound tracks (see [Sch90], p. 97 and 98).

In game audio, and many other settings involving auditory displays, informative functions are used in *Earcons*[7] which serve as carriers for notifications (messages that inform about a status or progress and do not require an immediate reaction by the recipient) or warnings (messages that indicate danger or other circumstances requiring an immediate reaction) (see [Jør06], p. 50).

Nowadays, the possibilites of music to transport informations and create recognition value are well explored, since they contribute largely to the success of television formats, commercials etc. by audio branding (see, for example, [Bro07]). An often employed design pattern to achieve such a result is the *Leitmotif*[8] (see [Win04], p. 7).

Although the use of informative functions of music in interactive settings is an intrinsic part of many applications (e.g. information data terminals or museum installations), it should be used carefully as it strongly depends on the sociocultural background of the recipients/users. Since messages may not be immediately decodable by individuals not familiar with the environment in question, such functions have to be introduced slowly until they are commonly accepted.

2.4.3. Descriptive Class

This class is comparable to the informative class, with the difference that here the physical world is described more actively by the music rather than by passively conveying certain informations (see [Win04], p. 7). In movies, it is used to establish a certain setting (e.g. time of day, season or a certain location) or to accompany a physical activity, such as movement (see [Win04], p. 7).

Nowadays, the second main category in this class – also known as *mickey mousing*, as mentioned in section 2.3.2 (p. 14) – is only used in a satirical, paroding manner, or to link a certain character to a certain music (see [Sch90], p. 93), and is otherwise frowned upon.

In computer games, there is definitely a necessity to describe areas or locations by means of music (see [Jør06], p. 50), so as to make them distinguishable from each other. However, descriptive functions of music are dependent on a visual represen-

[7] see glossary entry
[8] see glossary entry

tation of something to be described, which is why their use may not be appropriate to every possible application.

2.4.4. Guiding Class

Guiding functions of media music include indicative ones, which direct the recipient's attention to a certain object or element (see [Win04], p. 7 and [Sch90], p. 101), and masking ones, which are used to overlay e.g. bad acting or other conditions that may be worth concealing, such as weak commercial slogans which can be upvalued by music (see [Win04], pp. 7f).

As might be expected, indicative functions prove to be very useful in the new media context, since they offer a great potential to be implemented as auditory navigation interfaces or user-guidance systems (see [Win04], p. 7).

2.4.5. Temporal Class

Just like emotive functions, temporal functions are an integral part of music, because it always structures and illustrates time (see [Win04], p. 8). Not surprisingly, time-related functions of music seem to be widely accepted and studied.

First of all, music provides continuity through the process of unification (see section 2.3.1, p. 13), which is also very important in interactive media, especially computer games, because otherwise the multimodal perception would become disrupted (see [Win04], p. 8).

Secondly, music defines structure and form; it has a great potential to influence the perception of time and speed (see [Win04], p. 8). By using its natural possibilities, it is able to shorten or expand periods of time in the ear of a listener. Music can even assist in bridging large distances of space or time in the diegesis (see [Sch90], p. 104 and [Chi94], p. 82).

Because of their non-linearity, interactive media are also affected by, and can profit from this category (see [Win04], p. 8). Periods of time without any user interaction could be contracted by music, as well as the interaction itself could be commented or facilitated.

2.4.6. Rhetorical Class

Rhetorical functions of media music introduce another layer of the narrative, because music thereby gets in the position to comment on or make a statement about the ongoing story (see [Win04], p. 8). Sometimes music is even used to caricature a specific scene, by exaggerating the visual content (see [Sch90], pp. 99f). It involves the viewer/listener in such a way that he or she is enforced to engage in the interpretation of this new layer (see [Sch90], p. 100).

In non-linear media the use of such rhetorical functions becomes dangerous, because a clear narrative concept is needed which can be commented on. However, an innate quality of interactive media is that the storyline – if there is one, such as e.g. in role playing games – is not predetermined, the user himself is responsible for how the plot develops. Other types of applications, for example artistic audio-visual installations, do not incorporate a narrative function at all, which is why a rhetorical approach to comment on what is seen must fail.

2.5. Design Principles of Media Music Composition

The following explanations try to shed a light on a few methods and approaches that are worth considering when creating music for media applications. The enumeration at hand is by no means intended to display every existing design pattern, but meant to outline the most important ones from the author's point of view.

2.5.1. Gestalt Criteria

As indicated in sections 2.2 and 2.3, sonic events are organized into higher-level structures by the human brain, in musical terms for example melodies, rhythms, harmonies, motifs and so on (see [Raf02], p. 254). This so-called Gestalt quality is even preserved when the individual elements are altered; e.g. when a melody is transposed to a higher or lower pitch, it is nonetheless recognized as the same melody (see [Raf02], p. 254). The requirements that have to be fulfilled in order for the formation of higher-level structures to occur are called Gestalt criteria or laws (see [Raf02], pp. 254f); the most important ones are outlined in this section. Prior to that, it has to be remarked that in acoustic terms, these criteria not only evolve in the frequency and space domains, but also in the time domain (see [Raf02], p. 255).

Proximity

When two sonic events are closely located to each other in the frequency, space or time domain, they are combined to a higher-level structure (e.g. chords). To maintain transparency of different streams, proximitiy is often avoided deliberately, or diminished by re-arranging instrumental timbres or distributing sources in the stereo or surround panorama (see [Raf02], pp. 255f).

Similarity

Similar elements showing the same proximity form a higher-level structure; for instance, a line of adjacent notes is perceived as a melody, whereas distant notes on the same instrument could be interpreted as two or more melodies – or none at all. Notably, timbre also plays an important role here: The same note played on two different instruments is more unlikely to be perceived as a Gestalt than two adjacent notes played on the same instrument (see [Raf02], p. 256).

Good Continuation

Consecutive elements are regarded as belonging together when they show a common development of loudness, pitch or tempo (see [Raf02], p. 256).

Closure

Another important quality of perception of higher-level structures is that parts of a structure which is known but incomplete can anticipate this structure, and lead to its completion in the listener's imagination. Therefore, fragments of rhythms or melodies can be used instead of the original pattern, allowing the composer to vary it according to this rule (see [Raf02], p. 257).

Common Fate

An element can only belong to one structure, which is expressed by the law of common fate. Once a pattern is established, it is preserved until another event alters the perception or destroys the structure (see [Raf02], p. 257).

Figure 2.1 displays a visualization of four Gestalt laws.

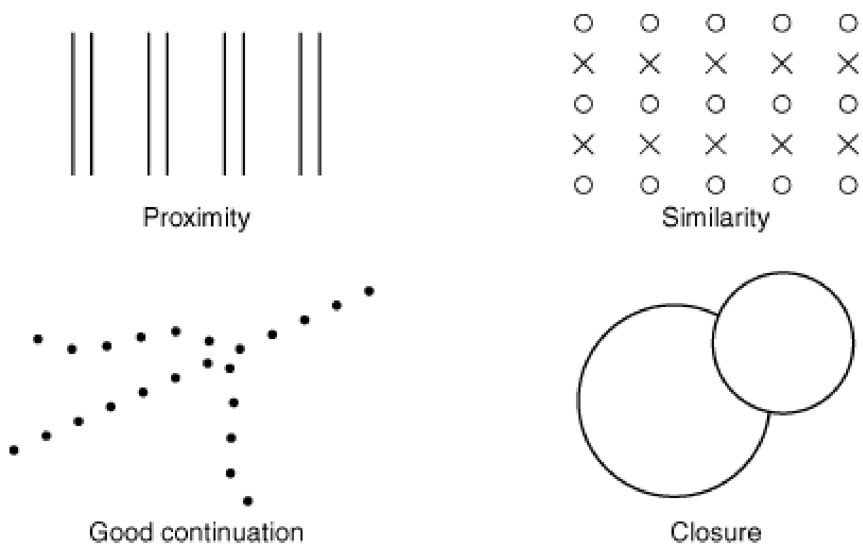

Figure 2.1.: Visualizations of the Gestalt Laws of Proximity, Similarity, Good continuation and Closure [Bla01]

2.5.2. Temporal Structures

In sound-related terms, temporal structures can be divided into three *time domains*, the *microscopic, transient* and *macroscopic* time domain (see [Raf02], p. 41), each of which involves its own design patterns and will be illustrated here.

Microscopic Time Domain

This domain is situated at the level of time where the actual acoustic oscillations take place, that is between 0.05 ms (milliseconds) and 50 ms. The processes occuring here are best described by the spectral characteristics of the signal (see [Raf02], p. 41), which includes the *timbre* and *harmonicity* of a sound.

From a technical point of view, timbre corresponds to the signal spectrum of an audio event (see [Raf02], p. 104). The envelope of a signal's spectrum also determines its harmonicity: harmonic sounds feature clear peaks at integer multiples of a fundamental frequency, whereas noisy or inharmonic sounds possess a continuous spectrum. Furthermore, timbre is the single sound quality that makes two sounds of equal loudness and pitch distinguishable (see [Raf02], p. 104). Especially in music applications, *Formants*[9] are a characteristic attribute of an instrument's tim-

[9] see glossary entry

bre, since they are completely independent of the sound's fundamental frequency, or pitch, and play an important role concerning the recognizability of the instrument (see [Raf02], pp. 104f).

However, the relevance for audio design as well as music is immense. First of all, timbre is only rarely designed deliberately at all (see [Raf07], p. 102), even though it possesses a high content of information on a semantic and emotional level (see [Raf02], p. 107). Secondly, the perception of timbre mostly takes place unconsciously, and thus triggers emotional reactions (see [Raf07], p. 102). The importance of a careful design of timbre also becomes evident when regarding that timbral features of a certain sound can remain in memory over large stretches of time (see [Raf07], p. 109).

Nowadays, with the aid of synthesizing and sampling techniques, it isn't uncommon to employ sounds whose harmonicity is residing somewhere between harmonic and noisy (see [Sch97], p. 188). However, by using novel technologies like *Physical Modeling* or *Granular Synthesis*[10] it is also possible to think of new timbres that are composed of the timbral features of several different instruments – e.g. half violin, half piano (see [Raf02], p. 109). Examples of approaches using physical modeling are Apple's sculpture instrument[11], or the works of Dr Stefan Bilbao[12]. A nice example of granular synthesis in combination with a generative evolutionary approach is Miranda's Chaosynth (see [Mir02] for details). Clearly, because of the mostly unused semantic and emotional potential, and the technical possibilities that have been coming up over the last years, timbre will gain more and more importance in media-related contexts.

Transient Time Domain

Amplitudes and frequencies of acoustic signals are seldom static, but underlie certain fluctuations in the transient time domain, which is located between 50 ms and 150 ms (see [Raf02], p. 41). Sound onset and decay, which are valuable characteristics for identifying sound sources such as musical instruments, belong to this time domain (see [Raf02], p. 41). A sound's *envelope* is situated between this domain and the

[10] see glossary entries
[11] http://www.apple.com/logicstudio/instruments/#sculpture
[12] http://www.music.ed.ac.uk/staff/academicprofile/StefanBilbaoAcademicProfileSoundfiles1.html

macroscopic one, which is why the related design patterns are evaluated in the following section.

Macroscopic Time Domain

Larger temporal structures (>150 ms) are combined in the macroscopic time domain – in a musical sense this includes structures such as *envelope, tempo, meter, rhythm* and *melody* (see [Raf02], p. 41).

Envelope The temporal composition of most sonic events can be crudely divided into the attack, decay, sustain and release (ADSR) phases, which of course represents a simplification of real conditions (see [Raf02], p.41). The attack phase is influenced by the onset characteristics of an instrument – a wind instrument for example has a longer attack time than a stringed instrument (see [Raf02], pp. 41f and [Sch97], p. 207). In the following decay phase, the natural resonance frequencies of an instrument die away. Therefore, an instrument whose vibrations are not permanently excited – like bowed instruments – has only a two-phased envelope consisting of attack and decay (see [Raf02], p. 42).

The length and level of the third, sustain phase are determined by the kind of excitation taking place (see [Raf02], p. 42 and [Sch97], p. 207). Lastly, when the external excitation has ended, the length of the release phase (called release time) is determined by the attenuation exerted by the instrument (see [Raf02], p. 42). When dealing with computer-generated, synthetic or sampled sounds, these are all parameters which can be controlled and taken into compositional considerations. Additionally, a sound's envelope can have significant impact on its perceived timbre (see [Raf02], p. 105). Figure 2.2 illustrates the four envelope phases.

Tempo Musical tempo is defined as the number of pulses per unit of time (e.g. beats per minute, *bpm*), or using qualitative notations (such as *allegro*, *adagio* and so on) (see [Raf02], p. 277 and [Sch97], p. 145). Variations of tempo (e.g. *rubato*) are an underestimated and rarely used stylistic means (see [Raf02], p. 277).

A *pulse*, on the other hand, is regarded as the smallest musically meaningful unit and is determined by the temporal distance between two elements (see

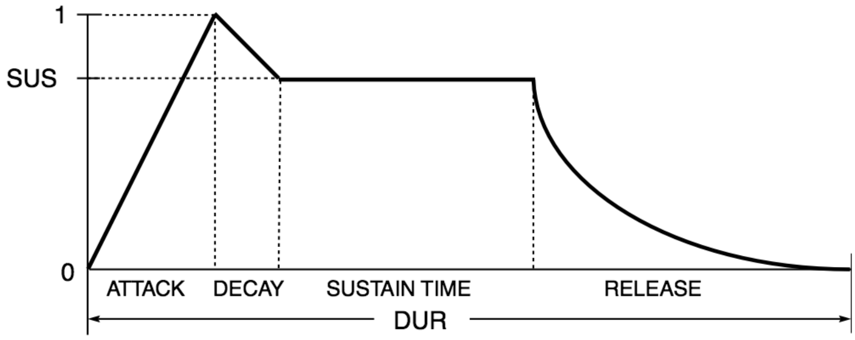

Figure 2.2.: ADSR envelope [Bea96]

[Raf02], p. 276). It is a useful term for the definition of larger structures, such as the following ones.

Meter A musical meter is constructed when musical pulse is split up into groups (bars), along with the formation of accented and unaccented pulses according to the amount of pulses in a bar (see [Raf02], p. 277 and [Sch97], p. 143). The most common meter in Western music is 4/4, but other, asymmetric meters such as 3/4, 5/4, or 7/4 are also known and used (see [Raf02], pp. 277f.)

The superposition of several meters is called polymeter, which can lead to more complex, repeating temporal structures. For instance, a 3/4 meter interfering with a 4/4 meter leads to a repeating pattern of 12 pulses (see [Raf02], p. 278). Other, more complex polymetrics can even result in rationally unintellegible structures (see [Sch97], pp. 143f).

Rhythm A musical rhythm is a macroscopic temporal structure with reiterating nature, whose anatomy can range from simple to complex (see [Sch97], pp. 142f). Its pattern is mostly composed of accented and unaccented beats of different duration (see [Raf02], p. 278). Once established, it can easily be recognized by the listener and can thus be utilized to orginate other, larger structures (see [Raf02], p. 278). It is important to regard a rhythmic structure in the context of the underlying meter, since the position within a bar or between several bars can directly influence its impact (see [Raf02], p. 278).

Melody A melody is composed of a progression of notes that differ in duration and/or pitch, and form a high-level structure according to the above mentioned Gestalt criteria (see [Raf02], p. 279). Melodies allow for a temporal organization of music, by incorporating a self-similar, characteristic structure based on symmetry, rhythm and integrity (see [Raf02], p. 280 and [Sch97], p. 163).

In the tradition of Western music, melodies are often shaped by symmetrical, arc-like structures (typically 8 or 12 bars) that anticipate a certain point in time where the structure concludes (see [Raf02], p. 280 and [Sch97], p. 164).

Thus, melodies can unconsciously influence the listener's perception of time, e.g. if this anticipation remains unfulfilled for a longer period of time, because the melodic arc isn't finished, a feeling of timelessness can be established (see [Raf02], p. 280).

Similar to that, *minimal music* tries to avoid a large-scale temporal structuring of music by the repetition of small-scaled patterns which are continuously, but slowly altered (see [Raf02], p. 280), which can also have an impact on the listener's perception of time.

2.5.3. Arrangement / Instrumentation

In classical composition theory, arrangement represents the organization of instruments or timbres (see [Sch97], p. 192). However, unlike in the baroque or classicist musical eras, nowadays formal structures of ensembles have almost dissappeared. Moreover, the media composer has to get involved with the growing mass of electronic timbres coming up (see [Sch97], p. 193). It is crucial for him or her to know what instrument to employ to achieve a certain timbre or pitch range, or which combination of instruments will have a desired emotional or semantic impact (see [Sch97], pp. 193f).

Until the era of the First Viennese School, the instrumentation of orchestral ensembles had been rather static – harmonization was regarded as the primary compositional discipline, and it was considered unimaginable to alter a piece's instrumentation until its ending (see [Sch97], p. 194). In the early years of the 19$^{\text{th}}$ century, these static structures were broken up, and the instrumental combinations of orchestras were gradually expanded – in pitch range as well as in size (see [Sch97], pp. 197f).

In the 20th century, timbre was finally regarded as one of the key aspects of composition (see e.g. works of György Ligeti). With the development of sound synthesizers and samplers, an inexhaustible pool of timbres has become available, allowing for unheard complex progressions of timbre (see [Sch97], p. 207). The impacts of instrumentation – and those of design of timbres respectively – on mood representation will be discussed in chapter 3.

2.6. Taxonomy

Almost every existing music taxonomy relies on the classification of music into genres (see e.g. [PC00]). However, in a media context, music is mostly either custom composed for the application in question, or it is selected in such a way as to express a certain meaning or mood. Therefore, a genre-based categorization of media music appears to be a futile undertaking. Instead, a possible approach could use existing media taxonomies and try to analyze what types of music are used for each medium – if any. Moreover, media producers have developed a special categorization scheme of music according to its possible field of application, or to the meaning that is to be conveyed. Both possibilities will be investigated in this section, and enhanced by comments on their applicability regarding the objectives of this thesis.

2.6.1. Media Taxonomies

A frequent observation found in theoretical studies towards taxonomizing media is the notion of all media representing possible containers for other media (see [Bei04] and [Whi04]). Another reccuring approach concerning the categorization of media is their assignment to human senses – primarily to the visual, aural and tactile domains, and, of course, their intersections (see [Hou04]). While music is clearly located in the aural domain, it might on the other hand surprise that it is regarded as a medium of its own by most authors (see [Bei04], [Hou04] and [Mar04], [Whi04]).

Evidently, following the argumentation that music is in fact its own medium, it is though a very common example of a medium contained within a different medium. Beitler [Bei04] remarks that music is often part of mass media such as film, television, or radio. It can, however, also be assigned to the internet (see [Whi04]) or, more recently, telephone. Whitehead [Whi04] points out that this nestedness of media also contributes to the public perception of media, but in many cases the nested media do not require each other. He brings the example of radio and music: If music didn't exist, the public notion of radio would alter, but it would not cease to exist (see [Whi04]).

Regarding the perceptional spheres used for the categorization of media might also contribute to the objective of retrieving possible fields of application for media music. Houlihan [Hou04] uses three circles in her graphical representation of a media taxonomy, referring to the three senses of sight, hearing and touch. It might be obvious that media relying solely on the human sense of sight and/or touch will not be a valuable basis for the application of music. Although music can be described or otherwise represented by media such as writing, painting, photography or braille, it cannot unfurl its functions as described in this chapter. Media depending entirely or partially on the sense of hearing on the other hand are often influenced by music, and are able to profit from musical additions. While it seems obvious that radio or speech are closely related to and influenced by music, the intersection with visual media also has a long tradition: Music theater (e.g. opera, musical etc.), television and cinema have been benefitting from their musical ingredients, partly for centuries. However, especially the intersection of all three senses, visual, oral and tactile ones, offers a lot of novel applications for music, some of which are already widely accepted and exploited by the industry. These applications include cell phone ringtones, music for videogames, web sound tracks as well as music for advertisement (for a detailed explanation of the underlying media taxonomy, see [Hou04]).

Concludingly, a possible way of constructing a taxonomy of media music could start by dividing the potential applications into sensory groups:

- music for audio-only media (radio, speech)
- music for audio-visual media (theater, cinema, television, video art)
- music for audio-tactile media (e.g. telephone)
- music for audio-visual-tactile media (cellular phones, video games, the internet, advertisement, infotainment, edutainment etc.)

2.6.2. Common Media Music Categorization Scheme

As outlined above, several terms for different use cases of media music have emerged in the industry, some of which are shortly described here (see [Sch97], pp. 70ff for a detailed explanation). Mostly these terms refer to intended meanings or functions, or to positions within a composite media product. Certainly, this list can be extended further by adding other target media or groups.

Jingle A short music clip signifying the start (or end) of a certain program or block (e.g. commercial break, station ID, news headlines etc.).

Music Bed Mostly ambient music with considerable rhythmic components, which are used to underlay visual (in case of television) or oral (in case of radio or television) content that does not necessitate direct connection with the music track. Here mostly loops are utilized, which presents a main starting point for revision and refinement, where generative approaches could also contribute to the originality and diversity of such musics.

Signation Usually a short musical sequence of high recognition value, used to announce the beginning of a television magazine, for example.

Theme Music A recurring piece of music of medium length, signifying the start of a television series (sometimes the term is also associated with feature films).

Illustration Music This term refers to film music that is used to accompany the narrative of a movie, although it is also used in many television formats.

Bridge Short sequences that are used to connect a certain scene to a following one (often used in television contexts).

3. Musical Representation of Emotions and Moods

The research conducted in the preceding chapter has identified several functions which media music incorporates, the most popular one probably being the function of delivering or depicting emotional content. That is to say, interrelations between emotions or moods and their musical representations have been subject to many extensive studies since psychology became an accepted discipline in the early 20$^{\text{th}}$ century (see e.g. [BW05], [BW06], [Mey56], [Sch06]). However, as argued in section 2.1 (p. 5), many interactive non-linear media applications still lack an integrated emotional musical design, which could help improve the user experience of the product (see [MK06], p. 37). Therefore, this chapter focuses on the underlying principles concerning how emotional content is expressed by musical structures.

The first question arising here is, whether music indeed contains the potential to evoke emotional reactions, or to represent affective states. While this has been the accepted conviction of many musicians, composers and listeners for centuries (see [Mey56], p. 7), there is also evidence for physiological reactions to musical stimuli, which are caused by emotions (see e.g. [Mey56], pp. 10f, [MK06], [Sch06], p. 14). The representation of moods and emotions by music gestures, on the other hand, can be explained by their similarity to behavioral gestures: both are characterized by energy, tension, direction and continuity (see [Mey56], p. 268).

Even though these two phenomena may accompany each other, it is equally possible that the emotional content illustrated by a music gesture is understood, but does not trigger any affective reaction at all (see [Mey56], p. 268 and [Sch06], p. 26). The focus of this thesis in terms of emotional design of music is laid on the representation of feelings, for use as ambient, illustrative music. Hence, the evocation of emotions is not discussed in this chapter.

Several emotion theories also differentiate between *emotions* (which are considered temporary occurences which follow a certain intensity gradient) and *moods* (more permanent and constant states of affect) (see [Mey56], p. 7 and [Sch90], p. 81). Schneider ([Sch97], p. 32) also mentions the musical representations of these two different concepts: timbre and melody for emotions; accompaniment, rhythmic and harmonic structures for moods. For the purpose of this thesis, it is considered sufficient to emphasize on the mood characteristics of music, as such long-term, slowly evolving affective states are more apt for the expression by a generative algorithm.

In the following, this chapter gives an overview over the existing taxonomies for mood classification of music, and investigates Russell's circumplex model of affects, which is widely accepted and used by many computer musicians (see section 3.1.2, p. 35). Afterwards, the modeling of mood music is examined by assembling and evaluating relevant literature on this topic. Thereafter, the musical parameters derived from this study are mapped onto the above mentioned circumplex model, for the purpose of developing a simplified two-dimensional user interface. Finally, the prototype's prerequisites concerning the graphical user interface, and what musical parameters the application has to be able to process, are specified.

3.1. Mood Classification

The classification of moods that are expressible by music is a central aspect regarding the development of a music-generating application. Especially concerning the emphasis on a simplified user interface for the manipulation of emotional content, it seems vital that the terms and tags used are all understood and accepted by the target group, in this case media producers. It seems obvious that the musical expression of mood and emotions can be broken down into categories, which on the other hand have no clear boundaries and tend to intermingle (see [Bru07], p. 26). The following two sections will give a condensed overview of useful techniques for the categorization of mood music.

3.1.1. Taxonomies

Several attempts have been made to develop a categorization scheme for the mood qualities of music. One rather simple approach – used for the adaptive production of music for game characters – divides the conveyable affects into *positive, neutral* and *negative* as follows ([ENF06], p. 4).

Positive affects

- Amusement
- Interest / Excitement
- Enjoyment / Joy
- Relief
- Satisfaction

Neutral affects

- Confusion
- Surprise / Startle

Negative affects

- Distress / Anguish
- Fear / Terror
- Anger / Rage
- Shame / Humiliation
- Sadness
- Guilt

Even though this scheme may have the advantage of being simplistic and easy to use in a gaming environment, it is strictly one-dimensional and discrete. Thus, it lacks the ability to define hybrid states consisting of two or more adjacent categories, or to continuously adjust the emotional parameters e.g. of a scene.

Another approach originating from the discipline of Music Information Retrieval (MIR) deals with automatic music mood estimation for the emotional categorization of a music library [SMvdP07]. For that purpose, a reduced term set was developed by user surveys, emphasizing on *easiness of use* and *importance* of the used expressions (see [SMvdP07], p. 345). The resulting 12 categories are

- arousing-awakening
- angry-furious-aggressive
- calming-soothing
- carefree-lighthearted-light-playful
- cheerful-festive
- emotional-passionate-touching-moving
- sad
- loving-romantic
- powerful-strong
- restless-jittery-nervous
- peaceful
- tender-soft

This term set, although also one-dimensional, tackles the problem of the high amount of subjectivity connected with mood categorization by involving results from listener surveys. To achieve this, the classes that most subjects seem to agree upon concerning their meaning and importance were identified (see [SMvdP07], p. 345). A similar approach is taken by Hu et al. ([HBD07]): By retrieving the top rated mood tags attached to a standardized set of musical pieces from music information services such as *last.fm*[1], a simplified but practicable term set is constructed (see [HBD07], p. 309). So, those mood descriptions achieve justification by being derived from the pragmatic context of social music networks. The top 19 mood tags were taken and clustered thereafter by comparing the tag occurences in the dataset (see [HBD07], p. 310). This analysis yielded 3 clusters with the top tags

[1] http://www.last.fm/

- aggressive, angry
- mellow, calm
- upbeat, happy

The remaining 13 top-rated tags (*sad, relaxing, sexy, romantic, dark, cool, melancholy, funny, powerful, emotional, soft, energetic, depressing*) were situated somewhere between those three clusters (see [HBD07], p. 310). This interesting piece of research shows that even though emotional music categories display a tendency to agglomerate themselves with other categories to higher-level classes, their boundaries become blurred when investigating them in more detail. Here, the categories are also aligned in a two-dimensional space after clustering, allowing for a better visualization of how they interrelate with each other. Hu et al. argue that this set is probably over-simplified, but since it is grounded in a real-world context, it may be sufficient for retrieval and consumer purposes (see [HBD07], pp. 309f). For production purposes, though, it still lacks clear structure to be useful for the construction of an optimized user interface.

3.1.2. Circumplex Model of Affects

The so-called *Circumplex Model of Affects* was proposed by James A. Russell in 1980 [Rus80]. The central concept behind this model is that every affective state is composed of the interconnection of two separate neurophysiological systems: one connected to *valence* (pleasure/displeasure), and the other to *arousal* or *activation* (see [RPP05], p. 716). Each emotion or mood is thus representable as a linear combination of these two axes (see [RPP05], p. 716). This fact is illustrated by figure 3.1: For example, a sad emotion consists of a medium degree of activation combined with a low degree of valence/pleasure. A feeling of tension, on the other hand, includes a higher level of activation and medium valence. Interestingly, Russell et al. ([RPP05], p. 719) also remark that

> "[...] the 2-D structure is found consistently across a large number of studies."

The other implication that this model brings about, which also coincides with the observations from the preceding section, is that emotions do not possess discrete boundaries that would separate affective states from one another (see [RPP05], p. 719). This is underlined by the fact that subjects seldom report the feeling of a

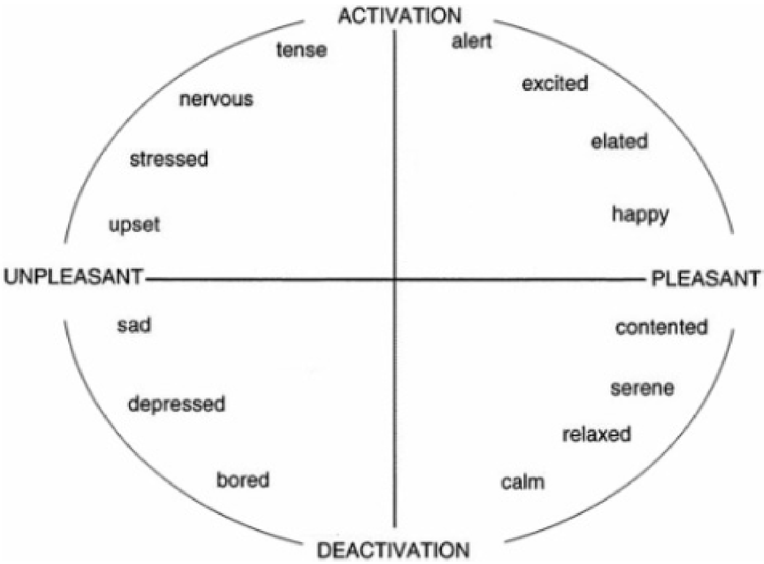

Figure 3.1.: Circumplex Model of Affects ([RPP05], p. 716)

particular positive emotion without also experiencing other emotions of high valence (see [RPP05], p. 719). Moreover, this fact is also consistent with the findings of Hu et al. (see section 3.1.1, p. 34) – there, many songs frequently tagged with one specific emotion were often tagged with another specific emotion as well.

In computer science, the circumplex model has also acquired broad acceptance, since the inherent two dimensions are easily managable in terms of computations (see [KCBM08], p. 583). For instance, Wingstedt et al. use the circumplex model for their studies concerning the correlations of emotions and musical parameters (see [BW05], p. 2); Knox et al. [KCBM08] use it as the basis of a music emotion classification system.

For the purpose of this thesis, the circumplex model seems ideal out of three reasons:

- As indicated above, the boundaries between affective states appear to be blurred. Therefore, to provide seamless musical transitions between two or more moods, an underlying model supporting this prerequisite is needed. The circumplex model directly implements this postulation by providing the two independent dimensions of valence and arousal.

- Furthermore, the immanent two axes allow for a specific mapping of musical parameters onto them. For example, if tempo should prove to be a relevant factor contributing to the representation of activation, the straightforward conclusion would be to map it onto the arousal axis, and define suitable border values for it.
- A central part of the prototype to be designed consists of the construction of an optimized user interface. Since the circumplex model is often illustrated as a circle with some exemplary emotions aligned around it (see figure 3.1), this representation also seems apt for the conceptual design of an interface.

3.2. Modeling of Mood Music

The first step towards a functional prototype which implements a musical representation of moods includes identifying musical parameters which are generally ascribed to certain emotions. Thereafter, those parameters have to be evaluated regarding their aptness to be mapped onto the circumplex model of affects. Finally, the prerequisites (in terms of implemented musical parameters) for the algorithms used within the prototype have to be determined.

Certainly, prior to that some constraints which are connected to the task have to be examined. One problem that has to be addressed are the cultural and social interrelations and dependencies that come along with musical representations of moods and emotions. There seems to be some consensus that concerning the classification of moods, cultural differences do not play a decisive role (see [KCBM08], p. 582). However, regarding the perception and interpretation of musically represented moods, it has been noted that different cultures use different conventionalized musical figures to express certain emotions (see [Mey56], pp. 266f). For reasons of simplicity, in the context of this thesis the assumption is made that the target listeners either belong to the Western culture area, or have been trained by Western media to become accustomed to their standardized musical expressions.

Another issue is related to large-scale temporal structures which cannot be modeled using simple musical parameters, such as tempo, rhythm or harmony. In other words, musical dramaturgic progressions, such as the expected conclusion of a melody or the resolution of dissonances (which certainly would contribute to the overall emotional impact of the music, see [Mey56], pp. 25ff) etc., cannot be predetermined by a basic set of musical variables.

3.2.1. Musical Parameters

Leaving cultural and social aspects aside, there seems to be a certain extent of agreement among Western musicologists concerning the relationships between certain musical parameters and the emotional experience connected to them (see [BW06], p. 66). It also appears to be accepted that several musical parameters together form the representation of a specific mood or emotion (see [Bru07], p. 22 and [ENF06], p. 5). At any rate, the variation of those parameters is believed to cause the alteration of the perceived mood (see [BW06], pp. 65f). In this section, eleven musical parameters are investigated concerning their ability to transport emotional values.

Mode

Probably one of the most popular stereotypes related to musical mood representation are the common interpretations of the *major* and *minor* modes. Even though their connotations appear to be quite straightforward - the major mode being associated with joy, and the minor mode with sadness, their use is not always that unambiguous. Moreover, other modes, for example church modes, pentatonic, whole tone or chromatic scales are often totally neglected regarding their emotional significance.

Apart from happiness, the major mode also appears in the context of serenity or solemnity (see [BW06], p. 66). The minor mode, because of its division in *natural*, *harmonic* and *melodic* scales[2], needs a more sophisticated analysis. Because of their potentially chromatic nature, e.g. composed of a melodic ascending and a natural descending, the minor modes also spread a sense of mystery (see [Mey56], p. 222 and pp. 224f). In some contexts, minor modes are used to create a feeling of tension, dreaminess, disgust and anger (see [BW06], p. 66), but also neutrality and harmony (see [ENF06], p. 5).

Whole tone or chromatic scales are sometimes used to display a gloomy or depressed mood (see [ENF06], p. 5). Church modes, e.g. the Lydian mode, are occasionally employed to convey a feeling of unfamiliarity (see [Sch97], p. 134).

[2] see Minor Modes glossary entry

Instrumentation / Timbre

As outlined in section 2.5.2 (p. 23), timbre is a key sound characteristic regarding its emotional implications, and carries a large potential for a better representation of moods. Beginning in the 19th century, timbre and instrumentation respectively have reached a higher level of acceptance concerning their impact on the affective perception of music (see section 2.5.3, p. 27). For example, mixed timbres of various different instruments can convey a feeling of mystery, as they cannot be dissected anymore (see [Sch97], p. 200). Furthermore, synthetical sounds often express surrealism or time dilatation, and are therefore used in science fiction or horror contexts (see [Sch97], p. 207)

Regarding the spectral envelope, a bright timbre (consisting of a fundamental frequency with a rich overtone spectrum) might indicate a feeling of potency, disgust, anger, terror, activity or astonishment (see [BW06], pp. 66f). On the other hand, a rather dull sound leads to associations with pleasentness, joy, affection, unhappiness or boredom (see [BW06], pp. 66f).

Tempo

Tempo is regarded as a major parameter concerning mood representation (see [BW06], p. 67 and [Bru07], p. 22). Tempo fluctuations are considered a powerful means to express the mood swings of a certain character (see [Sch97], p. 145).

A fast musical tempo leads to the impression of joy, excitement, astonishment, anger or terror (see [BW06], p. 67). Slow tempo is connected to sadness, calmness, solemnity, affection, boredom and disgust (see [BW06], p. 67).

Meter

The most frequently used time signature being 4/4 (see p. 26), it transports symmetry and stability (see [Raf02], pp. 277f and [Sch97], p.143). Other meters, such as 3/4, 5/4, 6/8, 7/4 are used to convey a feeling of activity and liveliness (see [Raf02], p. 278). A special role is attributed to polymeters: because of the superposition of more than one metric framework, there is no fixed first beat anymore, which can lead to a sensation of ecstasy and transpersonality (see [Sch97], pp. 143f).

Eladhari et al. have used different time signatures to represent angry (5/4), annoyed (7/8), neutral (4/4), cheerful (6/8) and exultant (3/4) affective states (see [ENF06], p. 6).

Rhythm

Rhythmic structures are often regarded as a representtaion of energy and arousal (see [ENF06], p. 3). Rhythms can be very simple, or can achieve a high degree of complexity either by the omission of pulses, or by dividing pulses into smaller parts and combining them to a higher-level structure (see [Bir03], p. 102).

Simple rhythms are used in joyful as well as solemn contexts (see [BW05], p. 165, [Bru07], p. 22 and [Sch06], p. 23). Complex rhythmic structures, on the other hand convey e.g. sadness (see [BW05], p. 166 and [Bru07], p. 22).

Accentuation

Accentuation, or accent evenness expresses the loudness variations of a sequence of musical notes (see [BW06], p. 67). Putting an emphasis on the musical pulse can e.g. enhance an expression of hecticness, stress, amusement or fear, whereas a constant accentuation can convey sadness, calmness, or seriousness (see [Raf02], p. 276 and [BW06], p. 67).

Articulation

Musical articulation is characterized by the ratio between the actual duration of a note and the length of a musical pulse. In other words, *legato* articulation means that every note's actual duration is equal to the length of a pulse, thus connecting the notes without temporal gaps between them. This form of articulation is used to communicate a feeling of sadness, affection or solemnity (see [BW06], p. 67).

Staccato, on the other hand means that every note is only played out for a certain percentage of the duration it should possess, according to the pulse. This can result in a representation of high energy, delight, arousal, terror or anger (see [BW06], p. 67).

Volume

By volume, the overall sound level of a piece is expressed (see [BW06], p. 67). Of course, a piece's volume is determined by the individual volumes of the participating instruments. *Crescendo* (increasing volume) and *decrescendo* (decreasing volume) have been discovered as a stylistic device for emotional expression in the early 19th century (see [Sch97], p. 195).

A high overall sound level leads to the impression of joy, intensity, power, tension and anger (see [BW06], p. 67). A low volume conveys a feeling of sadness, affection, solemnity or fear (see [BW06], p. 67).

Register

A note's register, or pitch level also seems to have an immense impact on the representation of affective states (see [BW06], p. 67 and [Sch06], p. 23). A high pitch denotes the expression of joy, grace, dreaminess, activity, astonishment, terror or anger, whereas a low pitch often carries the meaning of sadness, solemnity, boredom or pleasantness (see [BW06], p. 67).

Melodic Range / Ambitus

The melodic range or ambitus of a piece of music is defined by the lowest and highest occuring notes respectively. A wide melodic range is often associated with happiness/joy (see [BW05], p. 165 and [Bru07], p. 22), whereas a narrow ambitus is mostly connected to sadness or depression (see [BW05], p. 166 and [Bru07], p. 22).

Harmony

Harmonic features of a piece of music are mostly expressed by the polarity between consonance and dissonance. Meyer ([Mey56], p. 230) states that

> "[...] consonance and dissonance are not primarily acoustical phenomena, rather they are human mental phenomena and as such they depend for their definition upon the psychological laws governing human perception, upon the context in which the perception arises, and upon the learned response patterns which are part of this context."

One constant of human perception, he argues, is the inclination of the human mind to organize the stimuli it receives in the simplest way (see [Mey56], p. 231). In other words, taking the Gestalt laws into account, consonance presents a stable total entity, whereas dissonance is a less stable Gestalt (see [Mey56], p. 231). Dissonances are regarded as tendencies which exert their affective potential by delaying the approach of a stable entity, the consonance (see [Mey56], p. 232). Moreover, the perception of harmony (or its absence) is largely influenced by cultural and social contexts: While a medieval definition of consonant intervals would only have included octaves, fifths and fourths, it was later expanded by the major and minor thirds and sixths (see [Raf02], p. 99).

Many studies have been conducted regarding the psychoacoustic conditions that lead to a perception of dissonance or consonance (see [Raf02], pp. 99ff). For reasons of simplicity, in this thesis the above-mentioned intervals (octave, fifth, fourth, major and minor thirds and sixths) are regarded as consonant, all the others as dissonant (see also [Sch97], pp. 120ff).

Consonant harmonies are often experienced as happiness, solemnity, or softness (see [Sch06], p. 23, [Bru07], p. 24 and [BW05], p. 165), whereas dissonances are mostly interpreted as sadness or excitement (see [Sch06], p. 23 and [BW05], p. 166).

3.2.2. Parameter Mapping

In order to be apt for the construction of a user interface implementing the circumplex model of affects, the musical parameters investigated above have to be suitable for the mapping onto the valence and arousal axes. This leads to the hypothesis that each linear combination of valence and arousal that can be found in the circumplex model of affect, has a unique set of musical parameters attached to it expressing the mood in question. For the purpose of retrieving the relevant musical parameters for the two dimensions, exemplarily the diametrically opposed moods of sadness/joy (low and high valence) and tension/calmness (low and high arousal, see figure 3.1) are analysed concerning the combination of musical characteristics that lead to their representation.

Sadness / Joy

Undoubtedly, this pair of affects has been the most widely studied in musicologic literature (see e.g. [BW05], pp. 165f, [Bru07], p. 22 and [Sch06], p. 23). Here, the 11 musical parameters evaluated in section 3.2.1 (p. 38ff) are listed, and aligned with proposed values (see table 3.1). For the suggested tempo and articulation values see [BW05], p. 167, even though they were adapted here.

	Sadness	**Joy**
Mode	minor	major
Timbre	dark	dark/medium
Tempo	(rather) slow (approx. 60-100 bpm)	(rather) fast (approx. 80-120 bpm)
Meter	4/4	6/8 or 3/4
Rhythm	complex	simple
Accentuation	constant	slightly accented
Articulation	(rather) legato (80-90% of pulse length)	(rather) staccato (60-70% of pulse length)
Volume	soft	loud
Register	low	high
Melodic Range	narrow	wide
Harmony	dissonant	consonant

Table 3.1.: Parameter Mapping Example 1 – Sadness/Joy

Tension / Calmness

The affective pair of tension and calmness has been less studied in literature. Therefore, the suggestions in table 3.2 are derived mainly from the parameter evaluation in section 3.2.1.

Application of the Valence and Arousal Dimensions

From the exemplary analysis above, some correlations between musical parameters and the valence and arousal axes become apparent:

	Tension	Calmness
Mode	minor	major/minor
Timbre	bright	dark
Tempo	fast (>120 bpm)	slow (<70 bpm)
Meter	5/4 or 7/8	4/4
Rhythm	medium complexity	medium complexity
Accentuation	accented	constant
Articulation	staccato (50-60% of pulse length)	legato (90-100% of pulse length)
Volume	loud	soft
Register	high	low
Melodic Range	medium	medium
Harmony	dissonant	consonant

Table 3.2.: Parameter Mapping Example 2 – Tension / Calmness

1. **Valence**
   The most prominent correlation is again the mode, which appears to be strongly connected to valence. Also rhythmic complexity, melodic range and harmony show a connection with the valence axis. Figure 3.3 displays the interrelations and proposed limiting values.

	Low Valence	High Valence
Mode	minor / whole tone / chromatic	major / minor
Rhythm	complex	simple
Melodic Range	narrow (1 octave)	wide (2-3 octaves)
Harmony	dissonant	consonant

Table 3.3.: Mapping of Musical Parameters on the Valence Axis

2. **Arousal**
   One of the most obvious interrelations between the arousal axis and a musical parameters is related to tempo: slow tempo seems to indicate low arousal whereas high tempo denotes high arousal.

Other correlations can be assumed concerning articulation, timbre and volume. A slight relation can also be suspected between arousal and accentuation. The respective correlations and proposed limiting values are described in table 3.4.

	Low Arousal	**High Arousal**
Tempo	slow (50 bpm)	fast (150 bpm)
Articulation	legato (90-100% of pulse length)	staccato (50-60% of pulse length)
Timbre	dark	bright
Volume	soft (-24...-6 dB)	loud (-18...0 dB)
Accentuation	constant	accented

Table 3.4.: Mapping of Musical Parameters on the Arousal Axis

3. **Cross-Correlations**

Concerning register and meter, the correlation cannot be determined completely, a cross correlation between the valence and arousal axes seems to be the case. Low register is simultaneously associated with low arousal and low valence, and the opposite seems to be true for high register.

As for meter, it appears that a 4/4 time signature is preferred to express low arousal (e.g. calmness), whereas higher levels of activation are represented by other meters. Obviously there is a cross-correlation with valence concerning the time signature: positive affects are expressed by a 3/4 or 6/8 meter (e.g. joy), as opposed to negative affects (e.g. tension), where a more uneven meter such as 5/4 or 7/8 is preferred. As a simplification, table 3.5 shows the valence and arousal quadrants connected with the respective register and meter values.

	Low Valence	**High Valence**
High Arousal	5/4 or 7/8 time signature	high register 3/4 or 6/8 time signature
Low Arousal	low register 4/4 time signature	4/4 time signature

Table 3.5.: Cross Correlations between Valence and Arousal axes

3.2.3. Prototype Prerequisites

Implementable Musical Parameters

In order to outline the prerequisites for the algorithms used for automated composition, it is crucial to evaluate the evaluated musical parameters regarding their scalability and implementability.

Some characteristics have to be considered as global parameters, affecting the performance of the whole application. For instance, for reasons of intelligibility, it appears plausible to define tempo and meter, as well as volume as global parameters, applicable for all instruments that are included. This limitation leads to the omitting of polymeters and other phenomena, but the benefit for the overall clarity of the output seems to justify this restriction. Concerning their implementability, an algorithm for these three parameters should be straightforward to develop.

Another global parameter is given by the key of the piece. It should be noted that to obtain the key, the mode parameter has to be extended by the definition of a key tone. Modulation (i.e. switching between different keys) should only be possible by user interaction, i.e. the user is entitled to alter the key note by a user interface component.

All the other parameters (rhythm, accentuation, articulation, melodic range, harmony, register) can be regarded as local parameters, which can be varied for each individual instrument connected to the application. It also seems feasible to group the rhythm, accentuation and articulation parameters into one module, as they all depend on meter and tempo. Similarly, melodic range, harmony and register – all the parameters depending on the key of the piece – should be merged into a module.

In chapter 4, several algorithms' possibilities concerning the implementation of these musical parameters will be evaluated. However, a problem arises concerning the timbre parameter: Since timbral features are not part of the Musical Instrument Digital Interface (MIDI) protocol, which is going to be used for the communication between the generative composition algorithms and the instruments addressed by them, it has to be omitted here. It should be remarked, though, that for future research on this topic, the timbre parameter seems to be a pivotal one concerning the representation of affective states.

User Interface

As the underlying affective model is a two-dimensional one, it seems legitimate to build a user interface that directly implements these valence and arousal axes. The first device that can be thought of could be a joystick, a trackball or something similar. As a screen-based representation, it seems sufficient to integrate a two-dimensional slider, where the two axes can be affected simultaneously and independently.

4. Algorithmic Composition

4.1. Introduction to Algorithmic Composition and Generative Music

4.1.1. Definitions

Delineating the borders of algorithmic versus traditional composition appears to be an unrewarding task. Any compositional technique following a specific ruleset, e. g. contrapuntal, or according to some blues or jazz scheme could be considered algorithmic. In the modern sense of the word, however, algorithmic compositions mostly involve a computer which is generating music according to a specified algorithm. Together with a random element, this approach usually leads to a pre-determined, but nonetheless unpredictable aesthetic outcome (see e.g. [Mir01] or [Asc08]).

Generative music, on the other hand, is a more universal term. In theory, every structure-generating process (e.g. natural phenomenons and their impacts) can be utilized to spawn aesthetically fascinating musical gestures (see [BW07], p. 1). The interconnection to algorithmic composition is made up by the uniqueness of the artifact being produced. Brian Eno, a pioneer in generative ambient music, concludes ([Eno96]):

> "Generative music is unpredictable, classical music is predicted. Generative unrepeatable, classical repeatable."

What else unifies the two paradigms is the postulation of an unfinished, open piece of art which embraces the listener's participation and encourages him to transform his role from a totally passive to a more active one, contributing to the process of creating music by influencing the generative process behind it (see [Ess91]).

4.1.2. Historical Context

Apart from early algorithmic applications, such as Mozart's musical dice game (see [Mir01], p. 41), such methods have gained a high level of significance and interest in the 20th century, especially within the movements of *serialism* and *aleatoric music*. These developments were mainly inspired by the post-war necessity to create a new musical syntax and dispose of old-fashioned compositional traditions (see [Ess96], p. 11).

Serialism

Emerging from dodecaphonic principles of the Second Viennese School (whose most renowned members were Arnold Schönberg, Alban Berg and Anton Webern), serialism required the composer to submit to a strict ruleset for the creation of music. The basic law postulated by Schönberg instructs the composer to construct a series of 12 tones, of distinct ordering. This ordering has to be kept throughout a piece, and the repetition of a note before all the other notes of the series have been employed is forbidden (see [Mir01], p. 53). To overcome the drawback of replaying the same sequence of notes again and again, some transformations of the series are allowed (*transpose*, *retrograde*, *invert*, and *retrograde inversion*, see [Mir01], pp. 53f).

Later this approach was extended, e. g. by Pierre Boulez, so as to organize other musical parameters (such as pitch, duration, dynamics or timbre) by making use of the root series (see [Mir01], pp. 54f and [Ess96], pp. 13f). The most prominent example of serial music is probably Boulez' *Structures*, a piece for two pianos (see [Asc08], p. 9 and [Mir01], p. 55). The legacy of serialism as regards algorithmic composition clearly lies in the application of formalized and explicit rules for the composition of music (see [Mir01], p. 55).

Aleatoric Music

This countermovement to serialism (*alea* is the Latin word for dice), whose most prominent member was John Cage, deliberately abandoned the deterministic serial approach and included elements of chance for compositional purposes (see [Asc08], p. 10). In a way, aleatoricism reflects the shift in natural sciences that took place in the 20th century – the movement away from strictly deterministic Newton mechanics towards the more indeterministic quantum mechanics (see [Ess96], p. 21). The chaotic and random element introduced by aleatoric music is also perpetuated in most of the basic algorithms used in automated composition approaches, which will be further explained in this chapter.

4.1.3. Computational Prerequisites

As a computer essentially does not possess any understanding of musical theory or even acoustic events, musical parameters have to be converted to input/output parameters that the machine is able to utilize. This process of finding a sophisticated parametric representation of music and the mapping between musical values and their computational counterparts is a pivotal requirement of an automatic composition system (see [Asc08], pp. 23f). It will be showed in this chapter that different algorithms allow for different ways to influence the musical output of the algorithm by altering the input parameters or initial states.

4.2. Algorithms & Input Parameters

4.2.1. Probabilistic Approaches: Stochastic Processes & Markov Chains

Introduction

As outlined in section 4.1.2 (p. 49), the idea of randomness influencing the musical expression of a composition is no novel concept. However, as arbitrary as the outcome of a randomly generated sequence of notes would be, the same would apply to the aesthetic value of such a piece (see [Mir01], p. 61). The great advantage of a computer in this context is the fact that it can be used to evaluate the output of a random number generator, or to restrict the algorithm that this generator is based on itself, so that only those output values coinciding with the composer's aesthetic intentions are permitted (see [Mir01], p. 61).

The principles of probability calculus are mathematically well researched, so that *stochastic generators* (one that creates musical sequences according to a certain probability distribution) can be easily constructed. Moreover, stochastic generators can refer to past events by using *conditional probability*, that is, their outcome is affected by preceding results (see [Mir01], p. 62).

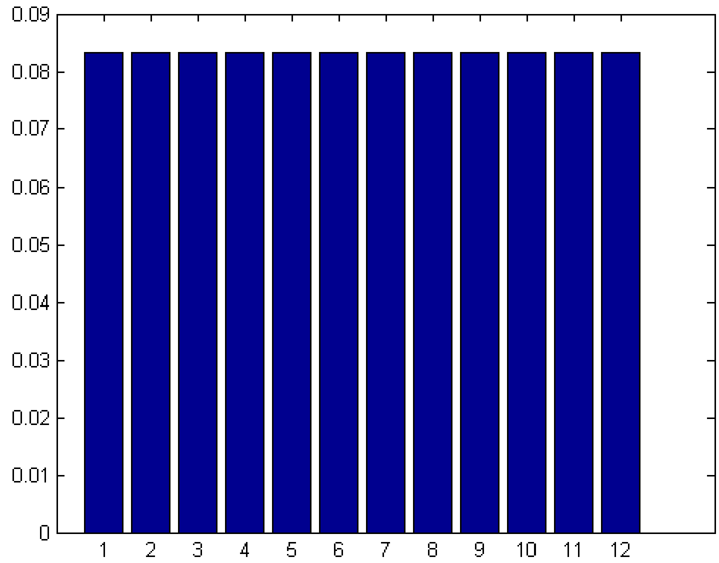

Figure 4.1.: Uniform probability distribution for 12 pitches

Stochastic Processes

Probability distributions can be used to shape the data produced by a random number generator according to a specified rule. The random variables used can be discrete (e.g. pitches) or continuous (e.g. durations) (see [Asc08], pp. 27f); their probability can be manipulated by altering their frequency of occurrence in the sample space. Thus, for example certain notes or intervals can be preferred over others (see [Mir01], p. 62). Exemplarily, four common probability distributions will be discussed in this section. Note that for all distributions it is essential that the sum (or integral, in the case of continuous random variables) of all probabilities has to amount to the value of 1.0.

- **Uniform Distribution**
  In a uniform probability distribution, all events have the same likelihood of occurrence (see [Mir01], pp. 62f). Figure 4.1 shows a discrete uniform distribution, where all pitches in one octave have the same probability.

  To produce such a distribution, merely one random number generator is needed (see listing 4.1).

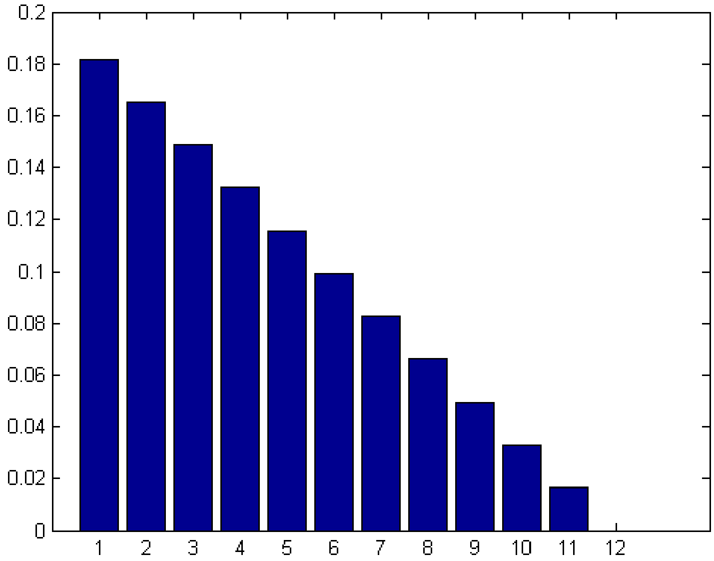

Figure 4.2.: Linear probability distribution for 12 pitches

```
float uniformDistribution()
{
  return random(1.0);
}
```

Listing 4.1: Algorithm for uniformly distributed random values ([Mir01] p. 63)

- **Linear Distribution**

  Figure 4.2 displays a linearly distributed random variable, i.e. values on the left hand side of the abscissa are more likely to occur than those on the right hand side (see [Mir01], pp. 63f).

  A generator algorithm (as displayed in listing 4.2) for such a probability distribution would have to include two random number generators implementing a uniform distribution, compare their output values and always return the larger one.

```
float linearDistribution()
{
  float a = random(1.0);
  float b = random(1.0);
  if(a > b)
    return a;
  else
    return b;
}
```

Listing 4.2: Algorithm for linearly distributed random values ([Mir01] p. 64)

- **Exponential Distribution**

  Exponentially distributed random values occur according to the following equation ([Asc08], p. 31):

  $$f(x) = \lambda\, e^{-\lambda x} \quad \forall x \in \mathbb{R} : x \geq 0 \tag{4.1}$$

  The rate of decrease is determined by the magnitude of λ, an increasing value results in a more rapid decay of probabilities; figure 4.3 depicts such a distribution. Although this function has no upper limit, very high values are very improbable to occur (see [Asc08], pp. 31f). A generator algorithm for exponentially distributed random values is given in listing 4.3.

```
float exponentialDistribution(float lambda)
{
  // provide a random number greater than
  // 0 and less than 1
  float a = random(1.0);

  // divide by lambda
  float b = a / lambda;

  // return the natural logarithm
  return ln(b);
}
```

Listing 4.3: Algorithm for exponentially distributed random values ([Mir01] p. 64)

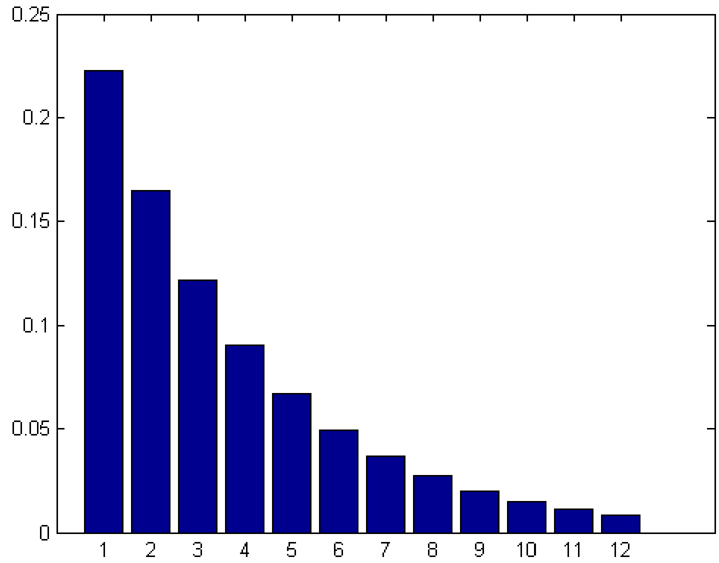

Figure 4.3.: Exponential probability distribution for 12 pitches, $\lambda = 0.3$

- **Gaussian Distribution**
  This function is known to resemble the value distribution of many natural phenomena, and is given by ([Asc08], p. 32):

$$f(x) = \frac{1}{\sqrt{2\pi\sigma}} \exp\left[-\frac{(x-\mu)^2}{2\sigma^2}\right] \tag{4.2}$$

  The function describes a bell curve (see figure 4.4), with μ being the average value around which the random values are centered (and the peak of the curve), and σ representing the standard deviation defining the span of the distribution: 68.26% of all results will be gathered in the area of $\mu \pm \sigma$, and 99.74% of all occurrences lie within $\mu \pm 3\sigma$ (see [Asc08], p.32).

There are many other probability distributions that would be worth mentioning here, e.g. Cauchy, Poisson or Beta distributions. [Asc08], pp. 32ff provides a detailed examination.

Markov Chains

While the stochastic processes discussed above are independent of past events and have to be calculated seperately each time they are used, Markov chains incorporate

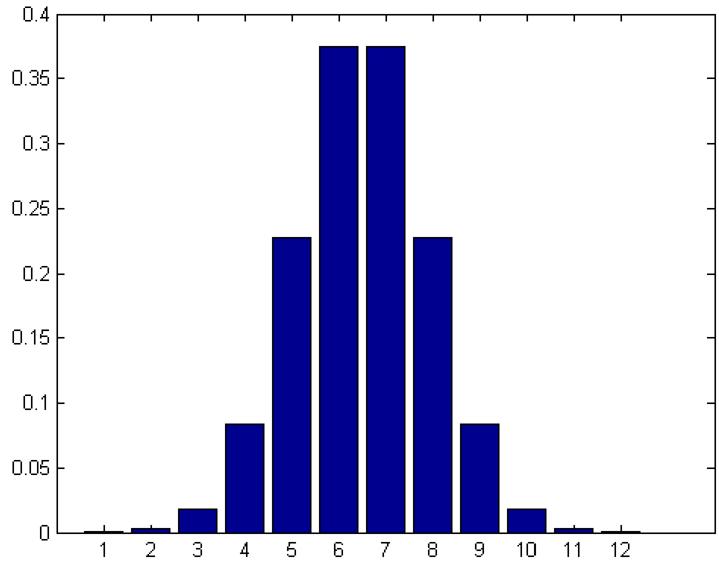

Figure 4.4.: Gaussian probability distribution for 12 pitches, $\sigma = 1.0$

the potential to include the outcome of preceding events. Thus, they are an example of conditional probability systems, where future events can depend on a certain number of past ones. The amount of past events that such a chain is dependent on is called its *order* (see [Mir01], p. 69). Markov chains can be used to model note progressions, note durations and other musical parameters. The probabilities that a chain is made up of can be arranged in a *transition matrix* (see [Asc08], p. 37); table 4.1 displays an example of a transition matrix of a first-order Markov chain modeling the note progression probabilities of an arbitrary twelve-tone scale.

State A is called *reachable* from state B, if it is possible to arrive at state A starting from state B after a finite amount of transitions (see [Mir01], pp. 69f). Two states are said to *communicate*, if they are reachable from each other (see [Mir01], p. 70). This relation plays a crucial role in the dynamics of a Markov chain, because communicating states can be proven to be *reflexive* (meaning that one of the states always communicates with itself), *symmetrical* (two communicating states can do so in either direction) and *transitive* (communicating states can be concatenated to reach other states) (see [Mir01], p. 80).

	C	**C#**	**D**	**D#**	**E**	**F**	**F#**	**G**	**G#**	**A**	**A#**	**B**
C	0.1	0.02	0.1	0.03	0.2	0.1	0.05	0.25	0.0	0.08	0.02	0.05
C#	0.25	0.02	0.08	0.05	0.2	0.07	0.1	0.01	0.02	0.1	0.0	0.1
D	0.1	0.03	0.15	0.02	0.1	0.25	0.01	0.14	0.02	0.1	0.03	0.05
D#	0.0	0.02	0.03	0.1	0.3	0.1	0.02	0.1	0.03	0.1	0.05	0.15
E	0.1	0.0	0.05	0.05	0.02	0.2	0.03	0.25	0.1	0.05	0.03	0.12
F	0.05	0.0	0.2	0.0	0.05	0.0	0.0	0.2	0.0	0.3	0.0	0.2
F#	0.1	0.15	0.0	0.2	0.05	0.0	0.12	0.1	0.18	0.1	0.0	0.0
G	0.05	0.2	0.15	0.0	0.1	0.05	0.0	0.2	0.05	0.1	0.05	0.05
G#	0.03	0.0	0.1	0.1	0.15	0.07	0.05	0.1	0.2	0.0	0.15	0.05
A	0.14	0.05	0.0	0.03	0.0	0.06	0.05	0.27	0.0	0.15	0.2	0.05
A#	0.18	0.25	0.0	0.07	0.1	0.05	0.2	0.1	0.03	0.02	0.0	0.0
B	0.5	0.0	0.27	0.0	0.03	0.0	0.0	0.0	0.18	0.0	0.02	0.0

Table 4.1.: Example of a first-order Markov chain applied to pitch classes

When designing Markov transition matrices, two requirements should be kept in mind: First, the sum of probabilities in each row has to amount to 1.0, as with all probability distributions. Secondly, it is vital to align the probabilities in such a way that no *dead ends* are produced, i.e. that the process of selecting the next state ends up in an infinite loop (see [Asc08], p. 38)

Evaluation

Probabilistic approaches seem appropriate to introduce a touch of randomness to otherwise predictable processes. Stochastic processes, for example, could prove to be valuable to influence note lengths or accentuation of a melody. On the other hand, a well-chosen probability distribution could also determine the overall representation of consonance or dissonance in a piece.

The conditional characteristics of Markov chains could be beneficial in constructing patterns of higher complexity (melodies, harmonies etc.). However, with increasing order, also the difficulty of designing such chains multiplies. A possible approach could be a combination of the two concepts: using certain probability distributions to form the row probabilities of Markovian transition matrices.

4.2.2. Iterative Approaches: Chaotic Systems & Fractals

Introduction

Iterative approaches utilize outcomes of repeated mathematical procedures where the output of such a procedure is used as the input value for the next step of the same procedure (see [Mir01], p. 83). Thus, the process describes a feedback loop, resulting in the outcome of the previous step influencing the next one. Therefore, one of the crucial characteristics of iterative approaches is that they are strictly deterministic, even though their output can range from predictable to chaotic behavior (see [Asc08], p. 41).

Chaotic Systems

Many natural phenomena which exhibit apparently chaotic, random behavior can be modeled using iterative processes. As indicated above, one of the main qualities of chaotic systems is that their output is fully determined (either by one equation or by a system of equations) by the initial parameters, but at the same time it can vary to a large degree (see [Asc08], p. 41). The characterizing features of a chaotic system are its *orbit* and possible *attractors*.

The orbit of an iterative process is represented by the set of its possible output values, (see [Mir01], p. 83). Depending on the initial parameters, three classes of orbits are possible (see [Mir01], p. 84 and [Asc08], p. 42):

1. orbits converging towards a stable value (called a *fixed attractor*)
2. orbits oscillating between a limited set of values (*periodic attractor*)
3. orbits displaying an apparently chaotic behavior (*chaotic / strange attractor*)

Nonetheless, a chaotic orbit is a necessary condition to describe a chaotic system, but it is not a sufficient one. Three principal characteristics are used to define whether a system is behaving chaotically (see [Mir01], p. 84):

1. **High sensitivity to initial conditions:** As has been pointed out above, chaotic systems exhibit a high degree of dependence on starting values. This concept has been illustrated by the metereologist Edward Lorenz using the so-called *butterfly effect*: He showed that using a simplified global weather model,

it was possible to alter the outcoming weather conditions to a great degree by only slightly changing the initial parameters. The picture he used was that of a butterfly whose wings' flapping on one side of the planet could influence the metereological system on the other side of the planet (see [Mir01], pp. 84f).

2. **Period doubling process:** Chaotic systems show a behavior called period doubling, meaning that the number of points in the orbit double successively before they become chaotic. For example, with a certain initial condition a chaotic system can converge towards a fixed attractor, with a slightly different initial condition its attractor begins altering between two values, then four, eight and so on (see [Mir01], p. 85).

3. **Sporadic settlements:** Lastly, chaotic systems possess the ability to establish quasi-stable regions after chaotic motion, which soon arrive at the process of period doubling and chaotic behavior again (see [Mir01], p. 86).

To illustrate these properties, two exemplary chaotic iterative systems are described in the following enumeration.

- **Logistic Equation**
  This iterative function describes a model for population growth using the following equation (see [Mir01], p. 86).

  $$x_{n+1} = r \cdot x_n(1 - x_n) \quad \forall r \in \mathbb{R} : 0 \leq r \leq 4 \tag{4.3}$$

  While the details that form the theoretical background of this model are not relevant in this context, this equation is quite apt to illustrate the above described properties of chaotic systems. Figure 4.5 displays the attractor's bifurcation diagram for different values of r.

  The orbit of this iterative function clearly exhibits a high degree of sensitivity to its initial conditions: As can be seen from the diagram, different values of r produce either fixed attractors ($1 < r < 3$), periodic ones ($3 < r < 3.569$), or chaotic ones ($3.569 < r < 4$). The period doubling property can be observed between $3 < r < 3.569$: first the periodic attractor is formed by only two points which then bifurcate to a four-, eight-, 16- and 32-point attractor. Finally, the attractor exhibits sporadic settlement in some regions between $3.569 < r < 4$: for a short period, it shows an oscillating motion which then becomes chaotic again (see [Asc08], p. 43).

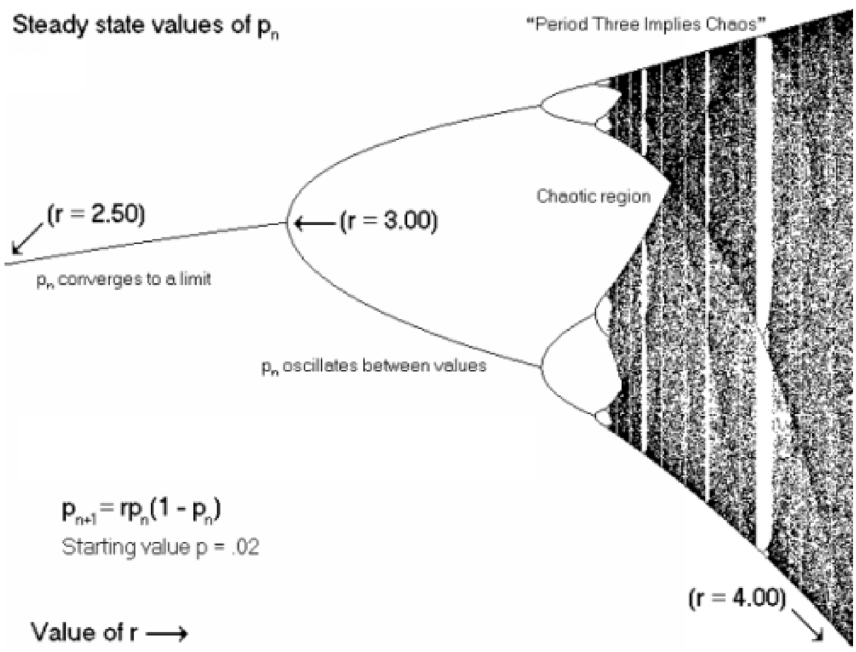

Figure 4.5.: Bifurcation diagram of the logistic equation [Bra08]

- **Hénon Attractor**

  This attractor, also known as the strange attractor, models the trajectory of an object orbiting a gravitational center (see [Mir01], p. 86). The model is composed of the two equations (see [Asc08], p. 47):

$$\begin{aligned} x_{n+1} &= y_n + 1 - ax_n^2 \\ y_{n+1} &= bx_n \\ a, b \in \mathbb{R} &: a, b > 0 \end{aligned} \tag{4.4}$$

  The behavior of the system is again dependent on the constant factors a and b, figure 4.6 shows the two-dimensional attractor map for the values $a = 1.4$ and $b = 0.3$. The nature of the attractor, i.e. whether it is fixed, periodic or chaotic, mainly depends on the value of a (see [Asc08], p. 47).

  Strange attractors are also known to generate self-similar structures, that is, patterns that display a periodic motion which produces similar, but nonetheless different sequences of output values (see [Asc08], p. 45).

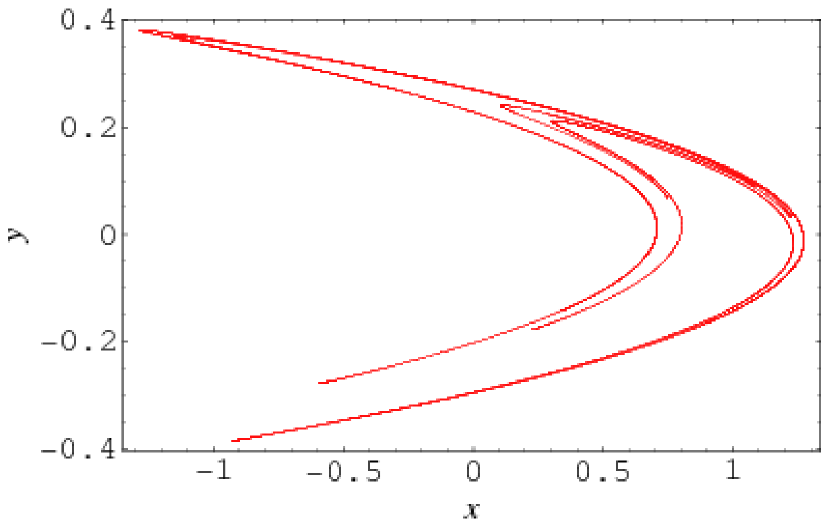

Figure 4.6.: Diagram of the Henon attractor for $a = 1.4$ and $b = 0.3$ [Wei08c]

Fractals

Fractal geometry was proposed by Benoit Mandelbrot in 1982 to overcome the shortages that traditional, Euclidean geometry exhibits when it comes to describing natural objects such as clouds, coastlines, trees and so on (see [Mir01], p. 90). Fractal shapes contain *self-similar* patterns, that is, structures which repeat at different sizes of an object; figure 4.7 shows four well-known examples of such structures. A typical realistic example of fractal geometry is the *coastline paradox*: the measured length of a coastline depends on the precision of the ruler – the shorter the ruler, the longer the total length (see [Wei08b]). A possible method of constructing such patterns is the use of iterative functions (see [Mir01], p. 90).

Self-similarity itself can be divided into three types (see [Mir01], p. 90):

1. **Exact self-similarity:** The same structure is reproduced at every scale of the fractal, as in the Koch snowflake.

2. **Statistical self-similarity:** This means that there is no exact reproduction of the basic shape at different levels, but it is strong enough to be identified. Many natural structures (e.g. plants) exhibit statistical self-similarity.

3. **Generalized self-similarity:** Transformations are applied to the scaled copies of a structure; such a type of self-similarity is more difficult to recognize.

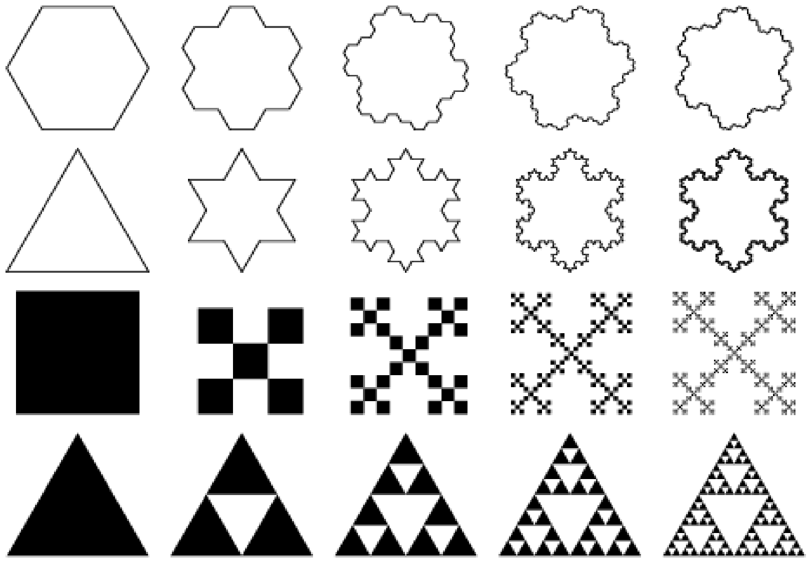

Figure 4.7.: Illustrations of the *Gosper island*, *Koch snowflake*, *box fractal* and *Sierpinski sieve* [Wei08b]

The musical potential of fractals seems to be a controversial issue. Miranda argues that the beauty that some illustrations of fractal patterns show cannot be easily transferred to the field of music, as that beauty emanates from observing the whole picture at the same time. Music, on the other hand, is a time-based art so that a mere transformation from the visual to the acoustic domain would result in unintelligible structures (see [Mir01], p. 95). A more promising approach is applying the iterative basic principles of fractals on a musical motif: By transferring geometrical transformations, such as translation, reflection and rotation, to the area of music, fractal-like patterns can be constructed (see [Mir01], pp. 95f). Thus, main ideas of serialism (see section 4.1.2, p. 49) can be extended and used by a generative system.

Evaluation

The possibilities for algorithmic composition clearly lie within the self-similar qualities of iterative processes. Chaotic as well as fractal systems provide structures that show a certain degree of similarity but never repeat in the exactly identical way – a property that is considered to be an essential ingredient of pleasant music (see [Mir01], p. 88). While fixed attractors will almost certainly result in a rather boring musical experience, periodic (especially those with a large period) and chaotic attractors can of course be regarded as a possibly valuable source of musical structures, such as note successions, durations etc. (see [Mir01], pp. 88f).

A caveat that has to be taken into consideration, however, is the fact that chaotic systems such as an implementation of the logistic equation or the Hénon attractor discussed above, will result in musically meaningless or arbitrary outcomes. The reason for this lies in the fact that as yet there exists no practicable method for utilizing the output of a process that was not originally devised for musical applications for such a system (see [Mir01], p. 89). Another drawback that comes with chaotic systems is grounded in their high degree of sensitivity regarding input values – depending on the iterative process in question, it might be problematic to determine whether an attractor will exhibit fixed, periodic or chaotic behavior before the first few iterations have completed (see [Mir01], p. 89). However, the potential of chaotic systems to produce fascinating patterns could also be used to create variations of an existing musical melody or rhythm (see [Asc08], p. 45).

Concerning fractal shapes, a possible application seems to be the calculation of rhythmic structures by utilizing the self-similar nature of fractals. The superposition of self-similar patterns is considered to generate interesting structures, which could be easily achieved using iterative processes (see [Mir01], pp. 96f). Also, the transformation of existing patterns using fractal algorithms seems a useful approach in this context.

4.2.3. Artificial Intelligence Approaches: Artificial Neural Networks

Introduction

Generally speaking, the probabilistic and iterative approaches discussed in the preceding sections are strictly rule-based. The main limitation of such a technique seems to be obvious: Even though music of all epochs follows certain rule systems, the aesthetic value it contains cannot be solely derived from such a rule set (see [Mir01], p. 103). In this context, neural networks suggest a different approach as compared to the traditional theory of computing including the conventional central processing unit (CPU) vs. memory approach (see [Mir01], p. 103 and [Asc08], p. 52).

Artificial Neural Networks

The main concept consists of a quasi-reverse-engineering of the human brain. Other than traditional imperative environments, which use programming techniques such as designing functions and procedures that are then executed in the CPU, *Artificial Neural Network (ANN)* – also known as *Connectionist models* or *parallel distributed processing models* – comprise a system of many interconnected units, resembling human neurons and synapses (see [Asc08], p. 52). Such networks are programmed by presenting them existing examples – they are actually able to learn the implicit rules that are contained within a problem, or rather lead to its solution (see [Mir01], p. 103).

The principal building block of an artificial neural network is constituted by the replication of a human neuron, the so-called *perceptron* (see [Mir01], p. 106). Human neurons are interconnected via *synapses* and may stimulate other neurons by sending an electrical pulse through its *axon*. The receiving neuron may be activated to propagate the signal if the sum of incoming stimuli causes a sufficient deviation of its *equilibrium potential* (see [Mir01], pp. 104ff and [Asc08], p. 53). The perceptron has been modeled in a way that resembles this concept (see figure 4.8 for an illustration): Incoming signals x_j represent the electrical pulses arriving at the neuron which are then multiplied by weighting factors w_j. These factors can possess either positive or negative values, depending on whether the synapse has *excitatory* or *inhibitory* character (i.e. the incoming signals are amplified or attenuated, see [Mir01], p. 107). The weighted signal values are summed up at the *summing junction* and propagated if the value fulfills the *activation function* F (see [Mir01], p. 107):

$$y = F(\sum_{j=1}^{n} w_j x_j) \tag{4.5}$$

This activation function F strongly influences the perceptron's behavior and resembles a biological neuron's sensitivity for incoming electrical depolarizations, also known as the *firing threshold*. It typically takes values between 0 and 1 or -1 and 1 (see [Mir01], pp. 107f). The three basic types of activation functions are *threshold*, *piece-wise linear* and *sigmoid*. A threshold activation function provides a hard threshold value that the sum of incoming weighted signals has to exceed in order to produce an output value of e.g. 1. Piece-wise linear functions possess a range of input values where the input values are linearly weighted and propagated. Sigmoid

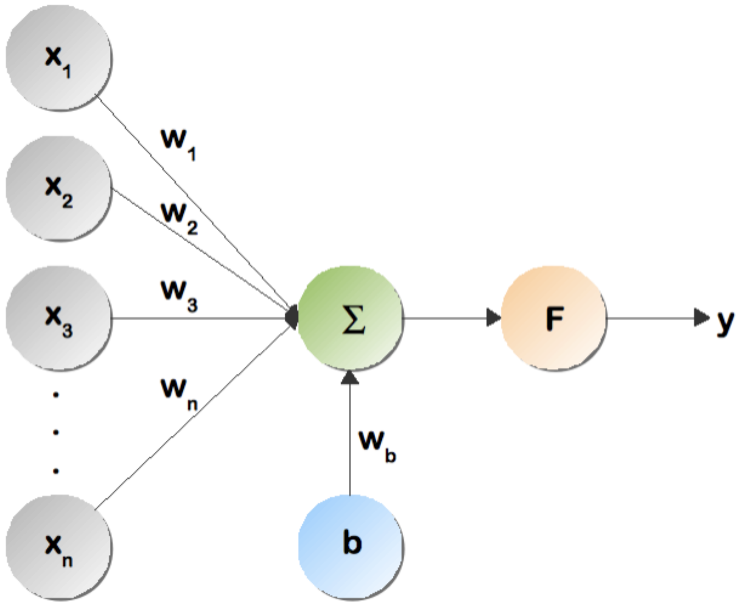

Figure 4.8.: Diagram of a single-layer perceptron

functions present a soft, smooth threshold and are widely used (see [Mir01], pp. 108f for more details).

More sophisticated neural networks are made up of multiple instances of a perceptron which are organized in layers – called *Multi Layer Perceptron (MLP)*, for an example illustration see figure 4.9. Generally, those layers are divided into input and output layers, as well as an arbitrary number of hidden layers in between (see [Mir01], p. 110). A further classification of such networks can be made according to their internal connections (see [Mir01], p. 110):

- **Feed-forward networks** contain no feed-back loops, the information flow happens only in one direction.
- **Feed-back networks** on the other hand can include such feed-back loops; a single node can thus even be connected to itself.
- In **fully connected** (as compared to **partially connected**) networks, all the nodes of one layer are linked with all the nodes of every adjacent layer.

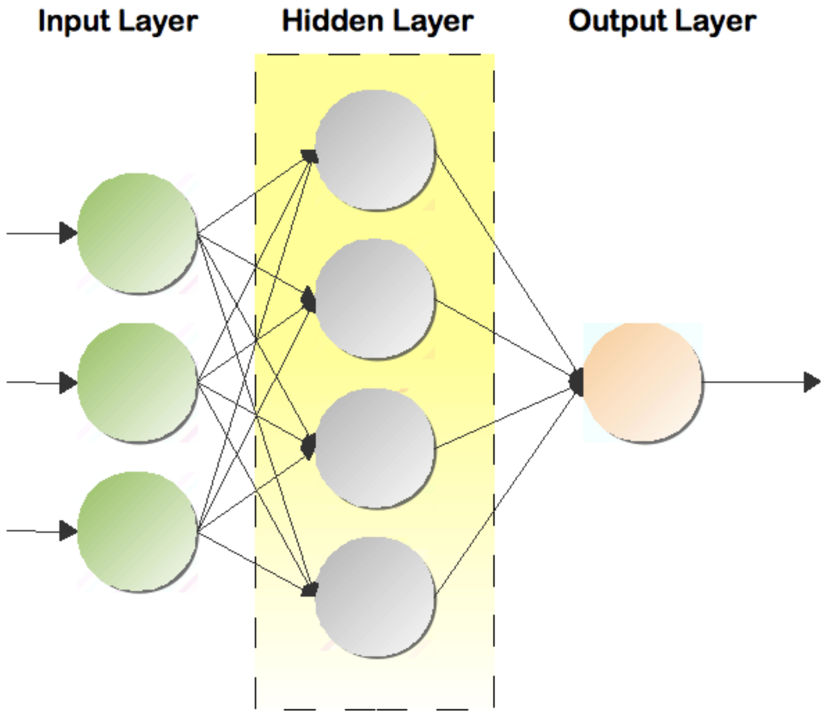

Figure 4.9.: Basic Multi Layer Perceptron

Network Training

As indicated above, an artificial neural network is programmed by training. Although there are several learning techniques, *supervised learning* seems to be the most appropriate for computer music purposes (see [Mir01], p. 112). In this case, the network is presented a set of examples with initial input values and a desired output. The weighting factors of the network are initialized randomly, but have to be changed by the network with each iteration in order to produce the desired output values. This process is called *error correction*, and mostly uses the *back propagation* algorithm (see [Mir01] pp. 112f and [Asc08], p. 55). The starting point of this algorithm is the calculation of an error signal e_k at time n for each neuron k (see [Mir01], p. 113):

$$e_k(n) = d_k(n) - y_k(n) \tag{4.6}$$

In this equation, d_k represents the desired output of the neuron whereas y_k represents its current output. The goal of the algorithm is to minimize the total error denoted by the formula

$$J = E\left[\frac{1}{2}\sum_k e_k^2\right] \tag{4.7}$$

which includes the sum of the squared individual errors e_k and the expected value E (see [Mir01], p. 113). The error correction itself takes place according to the following rule:

$$\Delta w_{kj}(n) = \eta \; e_k(n) \; x_{kj}(n) \tag{4.8}$$

The constant η denotes the rate of learning and has to be used with caution: too small a value will make the learning process smooth, but also time consuming; too large a value will cause the process to diverge and produce an unstable behavior of the network. The parameter x_{kj} stands for the input signals arriving at the neuron (see [Mir01], p. 113).

Musical Use

Regarding the construction of an ANN for musical use, several considerations have to be made, e.g. about the amount of layers, neurons and whether to include feed-back loops etc. (see [Asc08], p. 56). Several approaches have been made to devise musical

ANNs; Miranda describes a general architecture: the output layer of such a network could be used to produces pitches, durations etc., while the input layer could be fed with delayed outputs of the network. Thus, the outcome can be influenced by several preceding notes, depending on the number of delay units (see [Mir01], p. 114). To train such a network in order to produce a desired melody, several examples have to be fed into the network; after each step, the synaptic weights have to be modified in order to produce the next note of the example melody (see [Mir01], pp. 114f).

One advantage of ANNs concerning the generation of music lies in their ability to discover inherent patterns and organizational structures in the training material (see [Asc08], p. 58). Thus, the network is able to construct new motifs from the learned rules. However, this ability is influenced by the amount of hidden neurons in the network's architecture. Too large a number of hidden neurons combined with a small number of examples will result in the network solely memorizing those examples. A better approach seems to be to devise a network with as small a number of hidden neurons as possible, while feeding it a large amount of examples (see [Asc08], p. 58).

Evaluation

Even though ANNs seem to have a large potential regarding the automated composition of music, there are some caveats that have to be taken into consideration. First, designing an ANN always involves a training stage, and there may be applications where either there is no time for such a learning process to take place, or it is simply undesirable out of conceptual reasons. Neural networks are therefore considered not a substitute for, but a beneficial extension to the rule-based approach (see [Mir01], p. 103). Moreover, there are applications where ANNs prove to be useful in connection with artificial life approaches (see section 4.2.4).

It should also be remarked that ANNs offer the possibility to be trained in such a way as to produce *soundalikes* of a certain style, or even of a particular piece of music (see [Mir01], pp. 113f) – a valuable feature for many media music settings.

Finally, an ANN can also be regarded as a classificator, e.g. for the consonance of an interval (see [Mir01], pp. 115ff) – a quality that should also be evaluated regarding the prototype construction.

4.2.4. Artificial Life Approaches: Cellular Automata & Genetic Algorithms

Introduction

Recently, aside from artificial intelligence methods, also *Artificial Life (Alife)* approaches are gaining importance. These techniques include theoretical studies of life and the application of these theoretical backgrounds on experimental lifelike systems (see [BW07], p. 1). Here, processes of interest are the interaction of the system's individual components, the interconnection of such systems and the manifestation of global behavior (see [Mir01], p. 119). The goal of the alife approach is to provide a means to examine living systems using a computer as a platform, which again may serve as a research instrument for diverse fields, such as biology, medicine, social sciences and musicology (see [Mir01], p. 120).

In the computer music context, the probably most interesting cases for research are evolutionary and adaptive systems, that is, systems which are able to respond to their surroundings by adapting and evolving, out of the need to survive (see [BW07], p. 1). In particular, a goal could be to devise a virtual world populated by virtual musicians, composers and listeners incorporating social, ecological or psychological constraints, so as to investigate the system-inherent processes which foster the evolution of music (see [Mir01], p. 120). This section includes an introductory description of two alife paradigms: *cellular automata* and *genetic algorithms*.

Cellular Automata

Cellular Automata (CA) are models of dynamic systems which are used to examine complexity and self-organzation of biological processes or populations (see [Asc08], p. 59). Such automata possess certain limitations: They can be used to model systems where space and time are discrete, and the elements' (called *cells*) conditions are derived from a finite set of discrete states (see [Mir01], p. 121).

A cellular automaton's graphical representation consists of a grid, each field symbolizing a cell in a particular state. The cells' evolution takes place in discrete time steps, wherein all cells' states are updated simultaneously according to a predefined set of transition rules, which can depend on a cell's direct neighborhood as well as its current state (see [Asc08], p. 59 and [Mir01], p. 121). The most notable ex-

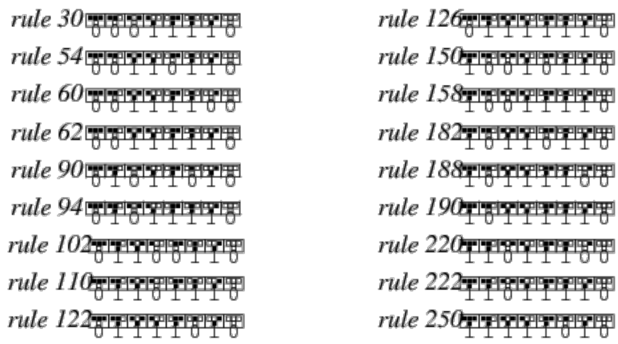

Figure 4.10.: 18 exemplary rulesets for elementary one-dimensional CA [Wei08a]

ample of such a system is John Conway's *Game of Life*, a two-dimensional cellular automaton which has been used e.g. by Miranda for compositional purposes (see [Mir01], pp. 124ff). This section will exemplarily cover a description of elementary one-dimensional CA, and possibilities for their application for the purpose of generating musical patterns.

Elementary one-dimensional CA feature a set of n cells neighboring each other and two possible states (0 and 1) for each cell (see [Asc08], pp. 60f). The transition ruleset of such an automaton possesses a radius $r = 1$, i.e. only a cell's own state as well as the two nearest neighbors' values are taken into account for the calculation of its succeeding state. There are $2^3 = 8$ possible states for the three adjoining cells (000, 001, 010 and so on), therefore $2^8 = 256$ rulesets for elementary one-dimensional CA can be constructed (see [Wei08a]). The rulesets are named by the decimal representation of their binary rule table (e.g. 00011110_2 stands for a rule 30 automaton, see [Wei08a]). A selection of rulesets is shown in figure 4.10, and illustrations of the corresponding CA started with a single living cell in figure 4.11.

Those rulesets are able to produce a variety of structures, including fractal ones (e.g. rule 90 or 150 automata) as well as chaotic ones (e.g. rule 30). Moreover, in some cases the same structure can be spawned by different rulesets (e.g. rule 18, 90, 146 and 218) (see [Asc08], pp. 61f and [Wei08a]). Rule 54 and rule 110 also seem worth considering for compositional purposes, as they produce a stable structure on the one hand while also including significant variations on the other hand (see [Asc08], p. 62).

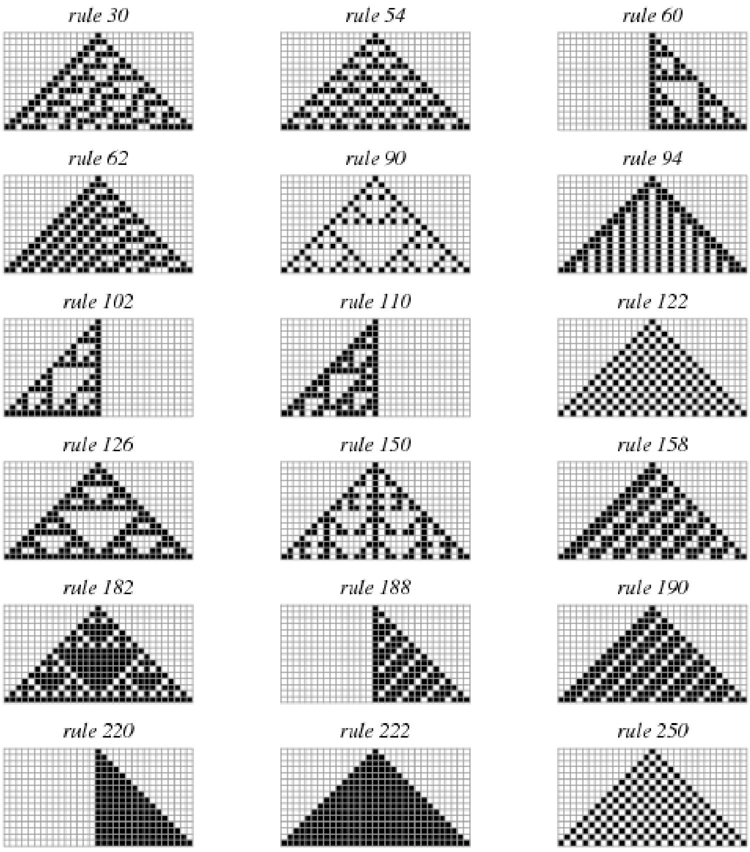

Figure 4.11.: 18 exemplary illustrations of elementary one-dimensional CA [Wei08a]

The transformation of a cellular automaton's output to meaningful musical values can take place in multiple ways: Living cells' coordinates can be converted to pitches or intervals, as in Miranda's CAMUS (see [Mir01], pp. 127ff). Furthermore, CA can also be used for sound synthesis, by simply applying the automaton's grid onto a wave table stored in the computer's memory – a possibility to generate a large variety of interesting sounds (see [Asc08], p. 67).

Genetic Algorithms

Another type of evolutionary computing method is represented by *Genetic Algorithms (GA)*. Evolutionary biological processes such as *reproduction*, *selection* or *mutation* are modeled in order to stimulate the creation of an optimal (fittest) solution to a given problem (see [Mir01], pp. 129ff). In general, GA can be regarded as search algorithms which are suitable for exploring a large search domain (see [Asc08], p. 68). They use a community of many individual possible solutions, each of which contains a codified version of the individual's properties (mostly a binary representation). In analogy to biological nomenclature, this representation is called *genotype*, comprising several *genes* (see [Asc08], p. 68). After the population has been initialized randomly, it iterates over a reproduction cycle, until an optimized solution has been found according to predefined fitness criteria (see [Mir01], pp. 131f and [Asc08], pp. 68ff):

1. **Fitness Evaluation**
   The current population's fitness is rated according to the mentioned fitness criteria. As the initial population will probably fail to provide an optimal solution, the process continues by ranking the individuals according to their fitness and then proceeding to the selection phase. If, however, the fitness criteria are met, this step is also the exit point of the loop.

2. **Selection**
   An arbitrary number of entities are chosen from the population (in the simplest case using stochastic methods which employ the computed ranking from the preceding step) which form the mating pool to breed a new generation of individuals in the next phase.

3. **Reproduction**
   Pairs of individuals are aligned at random which mate by exchanging parts of their codified genotype and thus forming the offspring.

4. **Mutation**
   The simplest way of implementing a mutation on binary codified genotypes is altering the value of a single bit according to a predefined mutation probability. It has to be considered that too high a probability will produce children that bear only little similarity with their parents, thus distorting the selection process.

After the last step has completed, the offspring is added to the population, and the reproduction cycle starts over with a new generation of individuals. Those entities that were not selected for reproduction in the preceding cycle iteration die in most implementations of the algorithm, although there may be applications where they survive (see [Mir01], p. 132).

Fitness Evaluation Techniques One of the crucial aspects in the process is the evaluation of fitness, which can take place automatically (involving a computer routine calculating the fitness function) or interactively (including an expert human user's knowledge to select the most appropriate individuals) (see [Asc08], p. 71).

Automatic fitness assessment methods include methods of deterministic fitness evaluation (by applying a mathematical function to obtain a fitness measure), formalistic fitness evaluation (by comparing individuals to a ruleset that may e.g. resemble a certain musical style), or the measuring of fitness via ANN (which offers the benefit that no precise rules have to be formulated as to what constitutes good music) (see [Asc08], pp. 71ff).

Interactive Genetic Algorithms, on the other hand, make use of a human user's (or a group of human users) knowledge to judge the individual solutions for their musical content. Thus, the process gains subjectivity as compared to automatic fitness evaluation methods, but it also involves the drawback that a manual evaluation may slow down the evaluation procedure considerably. This issue can be at least minimized by involving an automatic assessment method first, and afterwards presenting a reduced set of candidates to the evaluating user (see [Asc08], pp. 73f).

Musical Application The process of composing can be viewed as a search problem – to find the best (or most suitable) composition out of many possibilities. Therefore, GA can be used as a powerful means to assist in filtering the optimal musical structure (see [Asc08], p. 68). To accomplish this task, some preprocessing steps have to be taken before applying a genetic algorithm.

First, the search domain has to be specified. This can be made up of single notes, whole melodies, or it can be organized hierarchically, traversing many levels of musical structure (see [Bir03], p. 101). Generally, a preselection of apt individuals can also take place, e.g. by allowing only a selected set of pitches (see [Asc08], p. 70).

Moreover, in order to generate musically meaningful structures, a codification of musical values has to be introduced. Usually this is done by coding different musical attributes such as pitch, dynamics and so on into the genotype in a binary string form. A basic principle that should be followed is to use the smallest alphabet possible to encode the individuals' characteristics (see [Mir01], p. 133).

Evaluation

The rich amount of applications of GA (e.g. [BK08], [Bir03], [BL06], or [Mir01], p. 199) displays their great potential for generative music purposes. The issues concerning the implementation of such algorithms lie in the complexity of finding a suitable fitness evaluation method (see [Asc08], p. 76). If the evaluation function is designed in too simplistic a way, it will possibly yield diverging results, as the evolution of the population is insufficiently directional. However, a deterministic or formalistic representation of musical value or usefulness is tricky to formulate, as it involves a high degree of subjectivity and many composers and musicologists still argue over this topic. The ANN approach for evaluating fitness seems promising, but it also has to be trained (again by a human), which again includes the mentioned subjectivity.

A different possible approach is the development of distributed, interacting agents which are evolved by evolutionary techniques and create musical content together, by listening and judging the other agents' output (see [Asc08], p. 75).

The usefulness for the prototype to be developed depends on the restriction of the search domain on the one hand (e.g. only consonant notes of a scale etc.) and on the possibility to define an appropriate fitness evaluation function, which seems to be the harder task.

One-dimensional CA, however, seem to be an interesting technique for the generation of rhythmic structures, as they are able to produce self-similar output of fractal form. By changing the ruleset or the initial conditions, rhythmic complexity could be manipulated.

5. Prototypical Implementation

5.1. Related Work

As has already been pointed out, a large variety of applications and projects related to the topic of this thesis exist. Some of them, covering aesthetical, technical and practical aspects are presented and evaluated in this section.

CAMUS In this generative music software, two similar two-dimensional cellular automata are used for music composition: the *Game of Life* and *Demon Cyclic Space* (see [Mir02], p. 173).

The first, Game of Life, simulates the evolution of an array of virtual organisms whose survival and birth rates are determined by their eight nearest neighbors on a two-dimensional grid (see [Mir02], p. 173). In CAMUS, it is used to generate note triples by determining the horizontal and vertical coordinates of living cells and transforming them to musical intervals. The timing of the resulting note progressions on the other hand is determined by the cell's neighbors' states (see [Mir02], p. 174).

In the Demon Cycling Space CA, more than two possible cell states exist, which are labeled by numbers from 0 to $n-1$. The evolution rules again define how the cells change their states; however, in this case those rules are not set up by the neighboring cells' states. Here, cells with state k will dominate cells with state $k-1$, that is, change their state to k in the next evolutionary step. Since the automaton is cyclic – state $n-1$ is not the end of the chain, but it is dominated by state 0 – the cells' evolution is an infinite process (see [Mir01], p. 126). In CAMUS, this specific automaton is used to determine the instrumentation of the generated music by assigning different instruments to different states (see [Mir01], p. 128).

Apparently, cellular automata show similar qualities in organisational structures as compared to musical patterns and sounds. Nevertheless, it seems that such automata are more apt to model micro-scale than macro-scale structures (see [Mir02], p. 176).

Real Time Composition Library This library (RTC-lib[1]) was initiated and has been further developed by Karlheinz Essl since 1992, and can be used within the Max/MSP programming environment. It is intended to fill the gap between the strictly MIDI-control-based Max system and additional compositional requirements (see [Ess96], p. 36).

It includes random and list operations as well as harmony, rhythm and envelope generators (see [Ess96], pp. 40f). The great advantage of this library is that it is integrated in the well-known and widely used environment of Max, which makes it ideal for rapid prototyping and bottom-up approaches to algorithmic composition. Moreover, it is well documented, with existing tutorials and example pieces by the main developer. The library thrives on its author's experience in the programming and design of structure generators and generative pieces, making it an ideal choice for many (but not all) algorithmic applications.

AMEE The Algorithmic Music Evolution Engine (AMEE) is a system developed by the Department of Computer Science of the University of Western Ontario (see [HDK07]). It was devised with the purpose to dynamically compose music for interactive applications, e.g. computer games, or as the basis for novel composition tools (see [HDK07], p. 52).

The system's architecture includes a modeling of real-life musical entities, such as Musician, Instrument, Performer etc., which play together in a collaborative manner. Furthermore, pattern libraries that serve as motif repositories of existing or precomposed pieces are part of the system, enabling the reuse of certain compositional elements. The framework also contains an emotion mapper which is used to affect music generation in real-time (see [HDK07], p. 54).

[1] http://www.essl.at/works/rtc.html

Although there is no detailed explanation of the underlying algorithms employed, the value in this seminal development clearly lies in the flexible and extensible design of the system, which allows for the implementation of extension components (see [HDK07], p. 56), as well as the somewhat pioneering application of emotional parameters on generative music.

Continuous-Time Recurrent Neural Networks Oliver Bown and Sebastian Lexer [BL06] researched the development of computer software agents exhibiting musicality, and developed a generic behavioral tool using *Continuous-Time Recurrent Neural Networks (CTRNN)* (see [BL06], pp. 1f). CTRNN are a special type of neural network whose internal state is constantly updated using a differential equation, and enables recurrency, in particular the connection of any node to itself (see [BL06], p. 2).

Networks of different behavioral types were investigated, one of the most interesting ones being networks that produce different output states according to the trajectory that the input states pass (e.g. input state A followed by input state B leads to a different output than when followed by input state C, see [BL06], p. 4).

The very valuable aspect of their approach, however, lies in leaving the evolution of desirable behaviors to the users of such an application themselves, for which they employed a genetic algorithm (see [BL06], p. 6). Moreover, another important insight seems to be the fact that the authors envision a system that is capable of involving the user's as well as the network's evolution and lead to a collaborative, coexistential approach (see [BL06], p. 12).

5.2. Requirements & Limitations

In order to construct a software prototype for the automated generation of media music, several prerequisites and restrictions have to be considered. Technical requirements that have to be met were described in section 3.2.3. It seems advisable to devise a modularized structure for the prototype, wherein different core functionalities can be integrated. As a programming platform, Max/MSP[2] is chosen, because it facilitates rapid prototyping by providing a graphical development environment

[2]http://www.cycling74.com/products/max5

for interactive media applications. Furthermore, with the mxj Java wrapper class, a decent possibility to include external code on a multi-platform basis exists.

To demonstrate how the principles of algorithmic composition can be used to yield meaningful musical expressions, a classical electronic music setup using four instrumental voices (drumkit, bass, pad, lead sound) is chosen. The prototype is intended to create a stream of MIDI note events for each voice which can easily be integrated into any audio sequencer program, or directly connected to a sound-producing software synthesizer or sampler. The musical output is intended to be useful as a music bed contributing to the overall mood of a product presented in any medium involving audio components.

Some issues have to be expected concerning the application of volume values to the instruments. As MIDI note streams contain loudness (*velocity*) information, the transmission of relative volume values is generally possible. However, any software or hardware sound-producing device, and every audio sequencer in particular is equipped with a volume-control interface, so that the relative volume values can be easily overruled by the user. While it is on the one hand possible to affect the overall relative velocity values of the respective instruments, it is impossible to guarantee a certain mix of instrument volumes solely by using the MIDI protocol's inherent possibilities. Although the mix of instrumental voices could be an effective means to influence the emotional perception of the produced music, this quality of the resulting music track is mostly determined by the arranger, not by the composer. In a standalone system accompanying e.g. a computer game or an interactive installation, this feature should of course be taken into consideration.

5.3. User Interface

The user interface was developed on the basis of Russell's circumplex model, containing a two-dimensional area where the valence axis is mapped on the horizontal one, and the arousal axis on the vertical one. Thus, the user can freely decide on the degree of valence and arousal by clicking or dragging his mouse in this area, and does not have to interfere with or understand the system logic used to generate note sequences.

Figure 5.1.: Prototype user interface

Aside from the ability to turn each instrument on or off using a toggle switch, it is also possible to influence the meter and tonic keynote parameters, as well as the lead voice's register (see next section). The patch is started and stopped by a click in the respective start/stop toggle box in the top area of the interface, or by simply hitting the computer's space bar. Finally, in the right top corner of the prototype patch there is a dropdown menu allowing the user to assign the MIDI output device that he wants to use. It has to be remarked, though, that the prototype is capable of producing music in a reasonable way without any user interaction. Figure 5.1 shows a screenshot of the prototype's user interface.

5.4. Modules

This section will cover the overall architecture of the prototype, which has been split into six modules, each encapsulating a certain functionality. Complete Java code listings can be found in appendix B beginning on page 114. All range and register differences mentioned in this section refer to a root MIDI note value of 60, which corresponds to the pitch of C4.

5.4.1. Global Parameters

This module provides several initial values and weighting factors which are needed for the computation of note events by the instrument modules. These parameters include

- the *key note* (represented by its MIDI value)
  The default value for this parameter is 60 (which corresponds to C4), and can be altered by the user using a keyboard user interface component (see following module).
- the *mode* (represented by an integer: 0 for major, 1 for minor)
  According to the position on the valence axis, the instruments are required to produce either major or minor melodies and chords. The threshold is set to the arbitrary value of 20, resulting in a slight predominance of the minor mode on the valence axis.
- the *time difference between pulses* (measured in milliseconds)
  This value is retrieved by mapping the position on the arousal axis onto a scale of 50 to 150 bpm, which is afterwards converted to the time difference between 1/8 notes in milliseconds.
- the *peak volume* to be achieved by all instruments (provided as a MIDI velocity value)
  To determine the overall volume of the produced music, the arousal axis is mapped on a dB range of -6 to 0, which is then converted to an amplitude value, and afterwards returned as a MIDI velocity value.

5.4.2. User-Definable Parameters

The following parameters are definable by the user:

- *Meter:* Because of the cross-correlations observed in chapter 3, it seemed advisable to leave the decision, which meter to employ to the user. Possible values are 3/4 to 7/4 meters.
- *Tonic keynote:* Even though the patch is initialized with a tonic of C, it is possible to overrule this pre-definition.

- *Lead register:* Regarding the register parameter, it seemed apparent from chapter 3 that a distinct correlation cannot be determined. Furthermore, several instruments possess no, or a rather fixed register, as do the drumkit or the bassline. Since the pad instrument utilizes chords with wide range, the register parameter doesn't apply to it as well, leaving the lead voice's register to the user's taste. Possible range values include -1 to +2 octaves.

5.4.3. Bass

While the use of several instruments may be highly dependent on the style or genre of a piece of music, basslines seem to be omnipresent in contemporary music culture. Because of the necessity for a sonic fundament, and because of the simplicity that many basslines often display – they are mostly monophonic, because low-pitched chords, if they can be disassembled by the listener at all, sound rather dissonant (see [Raf02], p. 100) – this module was the first to be implemented during the prototype development phase; the inherent parts will be explained here.

Figure 5.2 shows a flowchart of the bass module's note event calculation scheme. It relies on a Hénon map algorithm for the generation of pitches, triggered by a musical pulse. The HenonMap object is initialized with values $a = 1.4$ and $b = 0.3$ (similar as in [Asc08] pp. 81f); listing 5.1 provides an overview of the used Java code. Its output between -1.3 and 1.3 is mapped on a scale between -3 and 12, thus generating an interval value in semitones which is added to a key note afterwards. The resulting note's dissonance factor is classified by the mode (minor or major); afterwards the pitch is adjusted, depending on the amount of dissonant notes allowed to pass the classifier. The percentage of dissonant notes that are not affected by the classifier is determined by the position on the valence axis, resulting in a value of 80% to 20%. The melodic range occupied by the bassline is also defined by the valence value, ranging from 1 octave for rather negative moods to 2 octaves for positive moods. To obtain this effect, a randomized multiple of 12 semitones (either 0 or 12) is added to the generated note when the user places the knob in the right half of the interface.

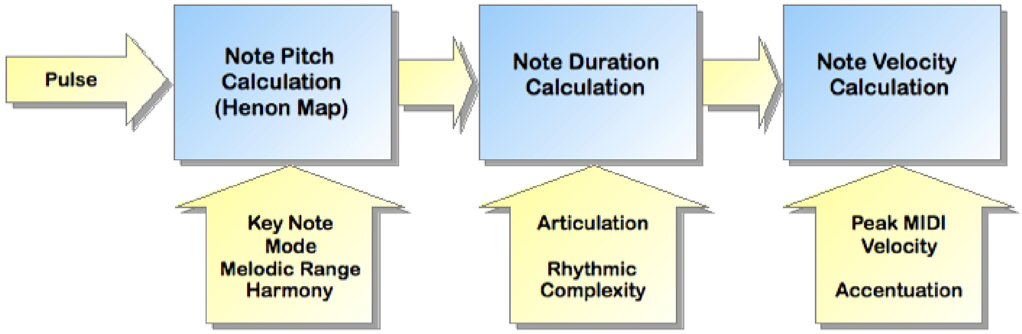

Figure 5.2.: Flowchart of the bass module

```
public class HenonMap extends MaxObject {

  private float a;
  private float b;
  private float x;
  private float y;

  public HenonMap(Atom[] args) {
    /* code omitted */
  }

  /**
  * produce next iteration of Henon map
  */
  private void generate() {
    // apply henon formulas
    float lastX = x;
    float lastY = y;

    x = lastY + 1 - a * (float) Math.pow(lastX, 2);
    y = b * lastX;
  }

  public void inlet(float f) {
    /* code omitted */
  }

  public void bang() {
```

```
    generate();
    outlet(0, x);
    outlet(1, y);
  }
}
```

Listing 5.1: Java code of the HenonMap class

The second step, the computation of the note's duration, takes into account rhythmic complexity, determined by the position on the valence axis, and the note's articulation, according to the position on the arousal axis. To achieve a representation of rhythmic complexity, the valence value is mapped on a scale between 4 and 1, which feeds the standard deviation value of a Gaussian distributed random variable with mean 0. Its absolute value is taken and rounded, so as to determine the amount of pulses that this note is going to occupy. The current note's articulation is represented by the percentage of the absolute duration, and is derived from the arousal value. Low arousal values will produce an average relative note length of 85%, whereas high arousal values result in an average length of 55% – representing legato and staccato articulation. This mean value is again assigned as the mean value of a Gaussian distribution with standard deviation 0.3, so as to randomize the relative note lengths. The output is then multiplied with the absolute note length to obtain the final duration value.

Finally, the note event's volume is composed of the peak MIDI velocity defined in the global parameters module, and a factor between 0 and 1 describing the note's accentuation. This value is again extracted from the current arousal value and fed into a Gaussian distribution object with mean values between 0.85 and 0.7, and a standard deviation between 0.2 and 0.05. The resulting factor denotes whether the note progression displays high or low accentuation, representing high and low arousal.

The resulting pitch, duration and volume values are combined to a MIDI note event and sent to the MIDI output device of Max/MSP.

5.4.4. Pad

A pad instrument is often used in electronic music to provide a background sound sphere to the music, often composed of chords which are changing only slowly and

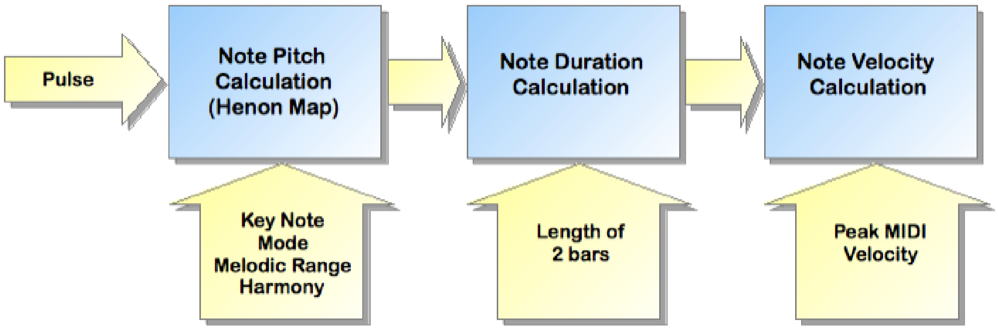

Figure 5.3.: Flowchart of the pad module

almost imperceptibly (see [Raf02], pp. 242f). Therefore, the pad module is set up in a similar way compared to the bass module, with three major differences: Because of the slowly evolving structures, the module abandons the concepts of articulation, rhythmic complexity and accentuation.

Figure 5.3 displays the main functionality of the pad module as a flowchart. Other than in the bass module, two HenonMap objects, one initialized with $a = 1.064$ and $b = 0.3$, the other with $a = 1.4$ and $b = 0.3$, are used to produce four notes to form a chord. The melodic range, according to the valence axis, varies between 1 and 3 octaves, while the harmony factor determines the amount of dissonant notes that are allowed to pass the mode classifier (as described above in the bass module).

The chord's duration has been fixed to a value resembling two bars (or the respective number of pulses according to the specified meter), while the volume matches the peak MIDI velocity. Using these values, a MIDI note event is assembled and sent to the MIDI output device.

5.4.5. Drumkit

For the creation of drum patterns an elementary one-dimensional cellular automaton was chosen, because of its potential to produce fractal, self-similar structures (see section 4.2.4). This particular quality is considered to be of high relevance for a generally rhythm-dominated voice, such as electronic percussion. The computation algorithm involves the initialization of 10 adjacent cells, and a default ruleset. The ruleset can be altered by sending a list of 8 binary integer values (0 or 1) to the inlet of the mxj object. The automaton evolves by applying the current ruleset on the cell population each time a bang is sent to the object's inlet (see listing 5.2). The resulting cell states are then mapped onto MIDI percussion instruments so as to produce a rhythmic pattern.

In order to vary the rhythms produced by the algorithm, the ruleset used to calculate evolving generations of cell states is altered periodically, depending on the rhythmic complexity according to the valence value. The accentuation of the drumkit's generated notes is affected by a Gaussian distributed random variable, similar to the bass module.

```
public class CA_1D extends MaxObject {

  private int[] cells;
  private int[] ruleset;
  private int columns;

  Random r = new Random();

  public CA_1D(Atom[] args) {
    /* code omitted: initialization of default cells and ruleset */
  }

  /**
  * apply incoming list as new ruleset
  */
  public void list(Atom[] args) {
    // fill ruleset with list arguments
    for(int i = 0; i < args.length; i++) {
      ruleset[i] = args[i].getInt();
    }

    // re-initialize original cell state for convenience
    /* code omitted */
  }

  /**
  * evolve next generation of cells
```

```java
  */
  private void generate() {
    int[] nextgen = new int[columns];

    for(int i = 1; i < columns-1; i++) {
      int leftNeighbor  = cells[i-1];
      int thisCell           = cells[i];
      int rightNeighbor = cells[i+1];
      nextgen[i] = applyRules(leftNeighbor, thisCell, rightNeighbor);
    }

    cells = (int[]) nextgen.clone();
  }

  private int applyRules(int left, int cell, int right) {
    if (left == 1 && cell == 1 && right == 1) return ruleset[0];
    if (left == 1 && cell == 1 && right == 0) return ruleset[1];
    if (left == 1 && cell == 0 && right == 1) return ruleset[2];
    if (left == 1 && cell == 0 && right == 0) return ruleset[3];
    if (left == 0 && cell == 1 && right == 1) return ruleset[4];
    if (left == 0 && cell == 1 && right == 0) return ruleset[5];
    if (left == 0 && cell == 0 && right == 1) return ruleset[6];
    if (left == 0 && cell == 0 && right == 0) return ruleset[7];
    return 0;
  }

  public void bang() {
    outlet(0, cells);
    generate();
  }
}
```

Listing 5.2: Java code of the CA_1D class (adapted from [Pro04])

5.4.6. Lead

For the lead voice, a simple genetic algorithm was implemented, so as to explore the potential of an evolutionary approach to produce meaningful melodic patterns. To achieve this, the population is initialized as 1000 (an arbitrarily chosen amount) four-note sequences (see listing B.3 in the appendix), each including the respective note pitches, velocities and relative durations. These parameters are filled with random values; however, the initialization of pitch values takes into account the specified register as well as the melodic range in octaves (again according to the current valence value). Thus, the initial search domain of the algorithm is limited to a predefined set of note pitches.

The crucial point in the design of the lead voice was the determination of a suitable fitness evaluation function, which is necessary for the selection of the fittest entities for the mating pool. It showed that such a function would have to take into consideration rhythmic, tonality, harmony, articulation and accentuation factors. In a crude estimation, those factors were multiplied with each other to obtain a measurement for fitness according to the user-specified values. The respective values of those parameters were determined as follows:

- **Rhythm Factor**
  To obtain a measurement for rhythmic complexity, the pauses (resembled by a note value of -1) in the pattern are counted. Furthermore, the pauses' relative positions are also considered to be relevant concerning rhythmic structure, so the on beat pauses are counted to be included in the calculation afterwards, too. Equation 5.1 displays the computation of the rhythm factor f_{Rhythm}.

$$
\begin{aligned}
f_{Rhythm} &= 1 - (f_{Rhythm,User} - f_{Rhythm,Pattern})^2 \\
f_{Rhythm,Pattern} &= \frac{n_{P,P} \cdot (n_{N,P} - n_{P,P})}{n_{N,P}^2} + \frac{n_{P,O}}{n_{N,P}}
\end{aligned}
\tag{5.1}
$$

  To reflect the distance of the user-defined rhythmic complexity value (which is a decimal number between 0 and 1) to the pattern's value in the computation, the difference of the two values is taken and subtracted from 1, so that high values of f_{Rhythm} resemble high values of rhythmic complexity. The pattern's complexity is composed of the addition of two quotients: one involving the total number of pauses $n_{P,P}$ and comparing it to the note amount $n_{N,P}$ in such a way, that a pause frequency of half the pattern's length denotes a local maximum. The second quotient includes the onbeat pauses $n_{P,O}$ which are divided by $n_{N,P}$ and added to the pattern's overall rhythmic complexity factor.

- **Tonality Factor**
  The tonality factor provides a weighting factor describing how many notes of the pattern are in the desired range (see equation 5.2). Whether a note is in

the required range is determined by the user-defined melodic range, keynote and register.

$$f_{Tonality} = \frac{n_{N,Range}}{n_{N,P}} \tag{5.2}$$

- **Harmony Factor**
  The harmonic aspect is covered by the amount of notes that belong to the given scale $n_{N,Scale}$, as well as the harmonic ratio determined by the current valence value (between 0.2 and 0.8), analogously to the rhythm factor.

$$\begin{aligned} f_{Harmony} &= 1 - (f_{Harmony,User} - f_{Harmony,Pattern})^2 \\ f_{Harmony,Pattern} &= \frac{n_{N,Scale}}{n_{N,P}} \end{aligned} \tag{5.3}$$

- **Articulation Factor**
  As the note lengths are stored in a relative way, their average value $f_{Articulation,Pattern}$, as well as the user-defined value $f_{Articulation,User}$ are used for the calculation.

$$f_{Articulation} = 1 - (f_{Articulation,User} - f_{Articulation,Pattern})^2 \tag{5.4}$$

- **Accentuation Factor**
  This factor includes the specified peak MIDI velocity v_{peak} as well as the average value $\bar{v}_{Pattern}$ of the velocities contained in the pattern and their standard deviation $\sigma_{v,Pattern}$. Those values are compared with the user-defined mean $\bar{v}_{User}$ and standard deviation $\sigma_{v,User}$ as follows:

$$\begin{aligned} f_{Accentuation} &= f_{Accentuation,Mean} \cdot f_{Accentuation,Stddev} \\ f_{Accentuation,Mean} &= 1 - \left(\frac{\bar{v}_{Pattern} - v_{peak} \cdot \bar{v}_{User}}{127}\right)^2 \\ f_{Accentuation,Stddev} &= 1 - \left(\frac{\sigma_{v,Pattern}}{127} - \sigma_{v,User}\right)^2 \end{aligned} \tag{5.5}$$

It seems evident that the fitness assessment method employed here is only a primitive approximation of the actual value of the musical content. However, it also shows the difficulty in designing a suitable technique for evaluating the fitness or appropriateness of musical structures. The complications encountered here hint at the possibility of employing other fitness classifiers which are relying on artificial intelligence methods (e.g. ANN).

After the fitness evaluation phase, the fitter half of the population is selected for reproduction, by exchanging halves of their genotype notes, velocities and durations. The offspring is inserted into the population after a mutation phase which involves the altering of one random bit of the note pitches and velocities of every tenth pattern. In a first step, 20 generations of such a population are evolved, and afterwards the four fittest sequences are selected for output.

All in all, the genetic algorithm developed here turned out to be a valuable experiment, even though it contains a vast amount of options and parameters that have to be taken into consideration. For example, the percentage of mutated entities had a significant impact on the outcome, as well as the amount of generations, the total size of the population and the proportions of the entity sets selected for reproduction. It seems apparent that such algorithms are a valuable approach for the generative composition of music, especially if one reviews related research on this topic, as has already been pointed out.

5.5. Evaluation

5.5.1. Subjective Critique

Musical Content

The resulting music shows defined rhythmic and melodic structures, while at the same time exhibiting enough diversity so as not to be considered monotonous or repetitive[3]. While highly and lowly aroused moods seem to be sufficiently discernible, the weighting of the valence axis seemingly has to take more features into account than the ones proposed in this thesis. Especially timbre, or instrumentation, as well as the mix and arrangement of the instruments seem to considerably affect the differentiation between negative and positive mood representations. In a standalone music generation library embedded e.g. in gaming or other interactive environments, this constraint should be addressed rigorously.

Another question that arises here is whether certain musical styles or genres require the usage of a certain selection of instrumental voices or timbres, and vice versa. Since this prototype was constructed without considering matters of musical

[3]In fact, the author experienced no listening fatigue even after a period of 20 minutes.

genre, or stylistic concerns, it is impossible to predict the effect that the resulting music will have on how it is perceived when played back with different sets of instruments. In other words, whether the music produced by the prototype fits the melodic capabilities of the instruments that are used for playback, or the stylistic implications associated with them, is not part of the prototype's architecture.

A special stylistic issue that has to be dealt with is the necessity of a lead voice. It can be argued that in many possible environments, such as image videos or interactive product presentations, a lead voice incorporating the melodic conduct of a piece could be considered disturbing, or at least distracting. In other, more immersive environments, such as computer games or auditory user interfaces, a lead voice cannot be done away with, since it embodies the recognizability and uniqueness of a product, or serves the guiding functions the product is intended to exhibit.

Implemented Algorithms

Of the implemented algorithms, especially the sonified Hénon map proved to be of great value concerning the composition of pitches or intervals (see also [Asc08], pp. 81f). The self similar, yet continuously slightly varying note or chord progressions produced by it seem to be highly suitable for background components that do not require distinct musical gestures, such as bassline or pad.

In the context of this prototype, cellular automata were used to produce rhythmic structures to be fed into a drumkit. As suggested in section 4.2.4, the fractal-like cell evolution exhibited by some specific rulesets is suitable to produce interesting patterns that display self-similarity and variance at the same time. However, the spectrum of patterns that CA are able to produce seems to be limited. For example, they are not able to produce short gestures, such as drum or cymbal fills, in a satisfactory way. Furthermore, most automata create patterns that have an irregular time base, e.g. 6 or 14 pulses, so that in order to obtain different meters it is inevitable to interfere with the algorithm's flow, e.g. by restarting it continuously.

For the variation of note articulation and accentuation, a simple Gaussian probability distribution process turned out to be useful. By simply shaping the expectation value and the standard deviation of a Gaussian distributed random number generator, variance of both loudness and note duration can be influenced in a natural sounding and sufficiently predictable way.

Although the genetic algorithm implemented in the prototype is only a simplification of more sophisticated and powerful ones, it clearly shows that using alife approaches, the borders of computer-based creativity can be further extended. Using ANNs as fitness evaluators could prove to be a valuable option, as well as initializing the GA's population not with random notes, but with precomposed patterns. With suitably trained agents (e.g. neural networks) as fitness classifiers, it even appears to be possible to generate high-level structures using genetic algorithms.

5.5.2. Expert Assessment

By and large, the performed expert interviews confirmed the subjective impression that the valence axis was underrepresented in the prototype as compared to the arousal axis. The intended representations of tension and calmness were identified with a larger degree of certainty in contrast to those of happiness and sadness.

There also seems to be some consensus that stylistic differences and associations contribute to the representation of moods, and that an interrelation between a certain genre and the musical elements used therein exists. In this context, it has also been mentioned that genre-specific representation of affective states, or rather their recognition, depends on the audience's listening experience, or cultural differences and encodings to a large extent.

Even more agreement can be found regarding the relation between timbre or instrumentation and emotional representation. Timbre, and micro-temporal fluctuations seem to be understood and interpreted by the listener in an intuitive manner, and also seem to be closely interconnected with certain genres. It was regarded as evident that timbres of natural, mechanical instruments (as compared to electronic ones) entail a longer tradition and feature certain rhythmic, melodic and harmonic stereotypes that lead to certain interpretations by the audience. Indeed, it would be a valuable addition to this experiment to exchange musical timbres and analyse the changes on mood representation.

One expert reported the frequent occurrence of dissonant intervals, which also hints at the question whether the lead voice could be too distracting in a media context.

At any rate, it signifies that melodies should perhaps be precomposed or at least generated by a more sophisticated random generator.

Generally, cost and time factors were mentioned as a primary advantage of generative approaches toward music composition by all experts. The field of algorithmic composition was seen as a promising, yet not enough explored area of music creation, which could prove to be valuable in interactive media contexts. On the other hand, all experts reported the musical arbitrariness of outcomes and several restrictions concerning the employed instruments and stylistics, as well as issues regarding the musical training of the audience. It seems as though these restrictions should be addressed firmly when refining the prototype.

6. Conclusion

6.1. Discussion of Results

The foremost ambition of this thesis was to discuss the discrepancy between contemporary practice regarding the use of music in media products, and the functions and impacts that this type of music is intended to embody. While tools for the dynamic and automatic generation of music in real time exist and are experimented on in many research and artist communities, numerous media products, especially those equipped with a small budget, still rely on repetitive patterns composed of stereotype music loops.

However, recently there has been a shift in public attention towards generative or reactive music applications, which expresses the necessity for research in this field so as to provide the fundament for the development of intuitive, mentally satisfying music-generating tools.

After all, algorithmic approaches display many features that seemingly make them an excellent choice to create an interactive tool for automatic and adaptive music production. Such an appliance could address the conflict between high quality standards and low production costs that many media producers are facing. At the moment, music tracks are often composed of loops taken from royalty-free music libraries; a fact that many media products are suffering from, as it affects both quality and recognizability. Another drawback of off-the-shelf music loops lies in the inability to apply dramaturgic concepts to them or create high-level structures from them.

The methodology used in this thesis was mainly based on literature research on the topics of media music composition, relevant perceptional principles as well as the representation of moods by different musical parameters. Furthermore, several common algorithms which are used for automatic music generation were investigated and evaluated as to their appropriateness for the design of a prototypical application.

Subsequently, the prototype was developed in the Max/MSP graphical programming environment, incorporating iterative as well as artificial life algorithms. The purpose of this implementation was to show that it is possible to devise an application which is usable by users with little or no musical background. In addition, a graphical user interface enabling the intuitive control of mood representation was created. Finally, using expert interviews and a subjective critique by the author, the produced content's aesthetic value as well as the representation of the specified mood characteristic was evaluated.

The results found during the research conducted in this thesis can be summarized as follows. In the second chapter, principles of the perception of media music, as well as its functions and impacts were discussed. Furthermore, design patterns were investigated along with potential applications and taxonomies that can be used for the categorization of media music. Media music seems to incorporate several functions, the most relevant ones being emotive, informative and guiding functions. It showed that both design patterns and perceptional concepts are closely linked to the fact that sound in general is a time-based medium, allowing for a practicable classification in correlation with the time scale.

The third chapter included an examination of mood representation of music, and the identification of a set of parameters that can be utilized to generate emotionally biased music. Current studies on musical mood classification were cited, and concluded by the introduction of Russell's circumplex model of affects. This emotional model seems to be apt for a sufficiently minute differentiation of affective states, as well as for a computational representation, as it is made up of two independent variables - the valence and arousal axes. Using state-of-the-art research on the modeling of mood music, a set of eleven parameters contributing to the emotional representation of a piece of music was identified (consisting of: mode, timbre, tempo, meter, rhythm, accentuation, accentuation, articulation, volume, register, melodic range, and harmony). In another step, it was attempted to map these parameters on the two independent axes mentioned above, so as to provide the basis for a prototype for the automatic production of mood music. A mapping of mode, rhythm, melodic range and harmony on the valence axis, as well as tempo, articulation, timbre, volume and accentuation on the arousal axis seemed feasible.

In addition, possible generative algorithms were studied as well as their possibilities concerning the mapping of the mentioned musical parameters. It showed that the most common generative approaches are derived from different areas, such as probabilistic or iterative processes, artificial intelligence and artificial life. There are indeterministic techniques, such as stochastic ones, as well as strictly deterministic, but nonetheless unpredictable ones, such as iterative processes, followed by artificial intelligence and evolutionary methods. Due to the diversity of those approches, the mapping of input parameters has to be addressed in different ways: probabilistic and chaotic outputs have to be shaped or filtered, fractal structures and cellular automata depend on their initial parameters, while artificial networks have to be trained to produce the desired output, and genetic algorithms need to be approached with a suitable definition of musical fitness.

In the next step, a software prototype was constructed, based on the findings from the preceding chapters. The objective was to design a user interface derived from the circumplex model of affects, that would facilitate the use of a music-generating tool for people with little or no musical knowledge. This prototype was developed in Max/MSP and enables the seamless, real-time control of the musical output's mood representation. Implemented algorithms include stochastic probability distributions as well as chaotic systems, cellular automata and genetic algorithms.

Finally, the produced musical content was evaluated concerning the correct mapping of the specified input parameters, as well as its aesthetical value. Moreover, it was also analyzed whether the music composed by the prototype indeed offers a representation of the intended moods. In general, the produced music showed perceivable rhythmic and melodic patterns, while possessing a sufficient degree of variation. It can be remarked in this context, that the goal of constructing a compositional automaton that evolves self-similar, yet not repetitive structures has been sufficiently reached.

Regarding the represenation of mood, the results show a tendency toward a better representation of the arousal domain as compared to the valence domain. Despite the common preconception that major and minor scales represent positive and negative affects, apparently timbre as well as the mix or arrangement of instruments have a higher degree of influence than is often suspected. Concludingly it can be noted that although the research conducted in this thesis indicates several tendencies toward mood representation in automatic music composition systems, music is still a time-based art whose impact is dependent on a multitude of parameters at the same time. Therefore it seems necessary to identify fields of future research in this context.

6.2. Future Research

Some classes of functions of media music were identified in this thesis; however, another wide area of possible future research lies in the semantic aspects of such music. Depending on the type of usage (commercial, informational, educational etc.), music is able to transport or encode meaning. It would be a worthwile task to study these features and identify parameters in the musical content that are responsible for the inclusion of semantic attributes.

Furthermore, the taxonomy of media-based music developed in this thesis can only be viewed as a starting point for a minute examination of this area. Such a fundamental taxonomy would present an essential component to facilitate access to the field of media music, and a source for many further studies regarding functions and impacts.

An important result of the conducted research is the fact that positive and negative affects seemingly need further studies concerning their musical representation. A valuable approach could be the examination of timbral aspects, as well as instrumentation or arranging techniques. Also, high-level musical structures can be suspected to contribute to this representation of valence.

As it was shown that genetic algorithms, or artificial life approaches in general, can be regarded as a promising branch of algorithmic composition, special effort should be laid on the conception of more appropriate fitness criteria for mood music. In general, low-level characteristics of music, such as were employed in this thesis should be taken into consideration as well as high-level structures. As dramaturgy presents a central factor in the composition of music, it would be of high value if these aspects could be integrated in such an approach.

Finally, the expert assessment that was carried out during the evaluation of the prototype should be augmented by extensive listener surveys, so as to gain better insight into the correlations of the mentioned low-level musical parameters with mood representation.

List of Figures

List of Tables

List of Listings

Index

Glossary

Auditory Icon A sonic element of an auditory user interface, which resembles a realistic everyday sound (sometimes enhanced or stylized) and thus needs no further learning or interpretation by the user (see [Bro07] p. 88). 5

Earcon A sonic element of an auditory user interface, which is used to transport information – similarly to visual icons – in an abstract way (e.g. by using a musical interval), and thus needs to be learned and interpreted by the user (see [Bro07] p. 88). 5

Formant Denotes a frequency range that emerges in the frequency spectrum of an acoustic signal according to the characteristics of the source's resonating body (see [Raf02], p. 52). 24

Granular Synthesis A sound synthesis technique comprising the division of an original sample into *grains* – segments of only a few milliseconds – and their resynthesis by reassembling them in an arbitrary sequence (see [Raf02], p. 239). 25

Leitmotif A musical motif that is attached to a certain character or mood. 19

Minor Modes Diatonic scales that exhibit a minor third interval. Commonly divided into

- natural minor (half step between second and third, and fifth and sixth degree)
- harmonic minor (seventh degree raised a semitone)
- melodic minor (sixth and seventh degrees raised a semitone)

. 40

Physical Modeling A sound synthesis technique which uses computer simulations of oscillating bodies (e.g. strings, membranes etc.) in order to generate certain timbres and temporal structures of a sound (see [Raf02], p. 240). 25

Synchresis Chion defines synchresis the following way:

> "The forging of an immediate and necessary relationship between something one sees and something one hears at the same time (from synchronism and synthesis). The psychological phenomenon of synchresis is what makes dubbing and much other postproduction sound mixing possible."

([Chi94], p. 224) . 9

Acronyms

Alife Artificial Life. 71

ANN Artificial Neural Network. 66

CA Cellular Automata. 72

CTRNN Continuous-Time Recurrent Neural Networks. 81

GA Genetic Algorithms. 75

MIDI Musical Instrument Digital Interface. 48

MLP Multi Layer Perceptron. 68

Bibliography

[Asc08] Daniel Aschauer. Algorithmic composition. Master's thesis, Technische Universität Wien, 2008.

[Bea96] James Beauchamp. Music 4c introduction. URL, http://ems.music.uiuc.edu/beaucham/software/m4c/m4c_intro_html/M4C_intro.html, 1996. accessed December 20, 2008.

[Bei04] Lauren Beitler. Taxonomizing media: The mediazoo. URL, http://csmt.uchicago.edu/taxonomy/taxonomybeitler.htm, 2004. accessed January 19, 2009.

[BH07] Axel Berndt and Knut Hartmann. Strategies for narrative and adaptive game scoring. In *Audio Mostly - 2nd Conference on Interaction with Sound - Conference Proceedings*, pages 141–147, Ilmenau, Germany, September 2007. Fraunhofer Institute for Digital Media Technology.

[Bir03] David Birchfield. Generative model for the creation of musical emotion, meaning, and form. In *ETP '03: Proceedings of the 2003 ACM SIGMM workshop on Experiential telepresence*, pages 99–104, New York, NY, USA, 2003. ACM.

[BK08] Scott Beveridge and Don Knox. Control of sound environment using genetic algorithms. In *Proceedings of the Audio Mostly Conference - A Conference on Interaction with Sound*, pages 50–55, Piteå, Sweden, October 2008. Interactive Institute, Sonic Studio Piteå.

[BL06] Oliver Bown and Sebastian Lexer. Continuous-time recurrent neural networks for generative and interactive musical performance. In *EvoMusArt Workshop*, Budapest, 2006. EuroGP.

[Bla01] Alan Blackwell. Human computer interaction notes. URL, http://www.cl.cam.ac.uk/teaching/2000/AGraphHCI/HCI/hcinotes.html, 2001. accessed January 19, 2009.

[Bra08] Larry Bradley. Chaos & fractals. URL, http://www.pha.jhu.edu/~ldb/seminar/logdiffeqn.html, 2008. accessed January 1, 2009.

[Bro07] Kai Bronner. *Audio-Branding: Entwicklung, Anwendung, Wirkung akustischer Identitäten in Werbung, Medien und Gesellschaft*, chapter Schöner die Marken nie klingen ... Jingle all the Way? Grundlagen des Audio-Branding, pages 82–96. Verlag Reinhard Fischer, München, 2007.

[Bru07] Herbert Bruhn. *Audio-Branding: Entwicklung, Anwendung, Wirkung akustischer Identitäten in Werbung, Medien und Gesellschaft*, chapter Musik als Repräsentation von vorgestellten Handlungen - Ausdrucksmodelle als Erklärung für die Wirkungen von Musik, pages 20–31. Verlag Reinhard Fischer, München, 2007.

[Bur98] Heinz-Wilfried Burow. *Musik, Medien, Technik: ein Handbuch*. Laaber Verlag, 1998.

[BW05] Jan Berg and Johnny Wingstedt. Relations between selected musical parameters and expressed emotions - extending the potential of computer entertainment. In S. Zhou, Z. Ying, and L.S. Ping, editors, *ACM SIGCHI International Conference on Advances in Computer Entertainment Technology ACE*, pages 164–171, Valencia, Spain, 15-17 June 2005. New York: ACM.

[BW06] Jan Berg and Johnny Wingstedt. Perceived properties of parameterised music for interactive applications. *Journal of Systemics, Cybernetics and Informatics*, 4(2):65–71, 2006.

[BW07] Oliver Bown and Geraint A. Wiggins. On the meaning of life (in artificial life approaches to music). In *Computational Creativity Workshop*. Goldsmiths, University of London, 2007.

[CGH06] Stuart Cunningham, Vic Grout, and Richard Hebblewhite. Computer game audio: The unappreciated scholar of the half-life generation. In *Proceedings of the Audio Mostly Conference - a Conference on Sound in Games*, pages 9–14, Piteå, October 2006. Interactive Institute, Sonic Studio Piteå.

[Chi94] Michel Chion. *Audio-Vision - Sound on Screen.* Columbia University Press, 1994.

[ENF06] Mirjam Eladhari, Rik Nieuwdorp, and Mikael Fridenfalk. The soundtrack of your mind: mind music - adaptive audio for game characters. In *ACE '06: Proceedings of the 2006 ACM SIGCHI international conference on Advances in computer entertainment technology*, page 54, New York, NY, USA, 2006. ACM.

[Eno96] Brian Eno. Generative music. URL, http://www.inmotionmagazine.com/eno1.html, July 1996. accessed July 30, 2008.

[Ess91] Karlheinz Essl. Computer aided composition. URL, http://www.essl.at/bibliogr/cac.html, 1991. accessed October 22, 2008.

[Ess96] Karlheinz Essl. *Beiträge zur Elektronischen Musik, Band 5*, chapter Strukturgeneratoren - Algorithmische Komposition in Echtzeit. Institut für Elektronische Musik (IEM) an der Hochschule für Musik und darstellende Kunst in Graz, 1996.

[Far07] Andy Farnell. Synthetic game audio with puredata. In *Audio Mostly - 2^{nd} Conference on Interaction with Sound - Conference Proceedings*, pages 76–83, Ilmenau, Germany, September 2007. Fraunhofer Institute for Digital Media Technology.

[Flü01] Barbara Flückiger. *Sound Design - Die virtuelle Klangwelt des Films.* Schüren Verlag, 2001.

[HBD07] Xiao Hu, Mert Bay, and J. Stephen Downie. Creating a simplified music mood classification ground-truth set. In *Proceedings of the 8^{th} International Conference on Music Information Retrieval*, pages 309–310, Vienna, Austria, September 2007. Oesterreichische Computer Gesellschaft.

[HDK07] Maia Hoeberechts, Ryan J. Demopoulos, and Michael Katchabaw. A flexible music composition engine. In *Audio Mostly - 2^{nd} Conference on Interaction with Sound - Conference Proceedings*, pages 52–57, Ilmenau, Germany, September 2007. Fraunhofer Institute for Digital Media Technology.

[Hou04] Kasia Houlihan. Interlocked circles of the senses: A new taxonomy of media. URL, http://csmt.uchicago.edu/taxonomy/taxonomyhoulihan1.htm, 2004. accessed January 19, 2009.

[Jør06] Kristine Jørgensen. On the functional aspects of computer game audio. In *Proceedings of the Audio Mostly Conference - a Conference on Sound in Games*, pages 48–52, Piteå, October 2006. Interactive Institute, Sonic Studio Piteå.

[KCBM08] Don Knox, Gianna Cassidy, Scott Beveridge, and Raymond MacDonald. Music emotion classification by audio signal analysis: Analysis of self-selected music during game play. In *Proceedings of the 10th International Conference on Music Perception and Cognition*, pages 581–587, Sapporo, Japan, August 2008. The Japanese Society for Music Perception and Cognition.

[Mar04] Starr Marcello. Media taxonomy. URL, http://csmt.uchicago.edu/taxonomy/taxonomymarcello.htm, 2004. accessed January 19, 2009.

[Mey56] Leonard B. Meyer. *Emotion and Meaning in Music.* The University of Chicago Press, London, 1956.

[Mir01] Eduardo Reck Miranda. *Composing Music with Computers.* Elsevier Ltd, Oxford, 2001.

[Mir02] Eduardo Reck Miranda. Sounds of artificial life. In *C&C '02: Proceedings of the 4th conference on Creativity & cognition*, pages 173–177, New York, NY, USA, 2002. ACM.

[MK06] David C. Moffat and Katharina Kiegler. Investigating the effects of music on emotions in games. In *Proceedings of the Audio Mostly Conference - a Conference on Sound in Games*, pages 37–41, Piteå, October 2006. Interactive Institute, Sonic Studio Piteå.

[Mur05a] Walter Murch. Dense clarity - clear density. URL, http://www.transom.org/guests/review/200504.review.murch2.html, 2005. accessed January 9, 2009.

[Mur05b] Walter Murch. Womb tone. URL, http://www.transom.org/guests/review/200504.review.murch.html, 2005. accessed November 7, 2008.

[PC00] F. Pachet and D. Cazaly. A taxonomy of musical genre. In Centre des Hautes Etudes Internationales d'Informatique Documentaire C.I.D, editor, *Proceedings of Content-Based Multimedia Information Access (RIAO) Conference*, volume 2, pages 1238–1246, College de France, Paris, 2000.

[Pro04] Processing.org. Wolframs's 1d ca. URL, http://www.processing.org/learning/topics/wolfram.html, 2004. accessed January 2, 2009.

[Raf02] Hannes Raffaseder. *Audiodesign.* Fachbuchverlag Leipzig, 2002.

[Raf07] Hannes Raffaseder. *Audio-Branding: Entwicklung, Anwendung, Wirkung akustischer Identitäten in Werbung, Medien und Gesellschaft*, chapter Klangmarken und Markenkl"ange: die Bedeutung der Klangfarbe im Audio-Branding, pages 102–117. Verlag Reinhard Fischer, München, 2007.

[RPP05] James A. Russell, Jonathan Posner, and Bradley S. Peterson. The circumplex model of affect: An integrative approach to affective neuroscience, cognitive development, and psychopathology. *Development and Psychopathology*, 17(3):715–734, 2005.

[Rus80] James A. Russell. A circumplex model of affect. *Journal of Personality and Social Psychology*, 39:1161–1178, 1980.

[Sch67] Pierre Schaeffer. *Traité des objects musicaux.* Rev. ed. Paris: Seuil, 1967.

[Sch90] Norbert Jürgen Schneider. *Handbuch Filmmusik I - Musikdramaturgie im Neuen Deutschen Film.* UVK, 1990.

[Sch97] Norbert Jürgen Schneider. *Komponieren für Film und Fernsehen.* MDS, 1997.

[Sch06] Jörg Schönberger. *Musik und Emotionen.* VDM Verlag Dr. Müller, 2006.

[SMvdP07] Janto Skowronek, Martin McKinney, and Steven van de Par. A demonstrator for automatic music mood estimation. In *Proceedings of the 8th International Conference on Music Information Retrieval*, pages 345–346, Vienna, Austria, September 2007. Oesterreichische Computer Gesellschaft.

[Tan06] Atau Tanaka. *Consuming Music Together - Social and Collaborative Aspects of Music Consumption Technologies*, chapter Interaction, Experience and the Future of Music, pages 271–292. Springer Verlag, 2006.

[Wei08a] Eric W. Weisstein. Elementary cellular automaton. URL, http://mathworld.wolfram.com/ElementaryCellularAutomaton.html, 2008. accessed January 1, 2009.

[Wei08b] Eric W. Weisstein. Fractal. URL, http://mathworld.wolfram.com/Fractal.html, 2008. accessed January 1, 2009.

[Wei08c] Eric W. Weisstein. Hénon map. URL, http://mathworld.wolfram.com/HenonMap.html, 2008. accessed January 1, 2009.

[Whi04] David Whitehead. Media taxonomy. URL, http://csmt.uchicago.edu/taxonomy/taxonomywhitehead.htm, 2004. accessed January 19, 2009.

[Win98] Todd Winkler. *Composing Interactive Music - Techniques and Ideas Using Max.* MIT Press, 1998.

[Win04] Johnny Wingstedt. Narrative functions of film music in a relational perspective. In *Proceedings of ISME - Sound Worlds to Discover*, Santa Cruz, Teneriffe, Spain, 2004. International Society for Music Education.

A. CD-ROM Contents

- Rubisch_-_Generative_Music_for_Media_Applications.pdf (this thesis in electronic form)

Prototype

- CA_1D.class
- CA_1D.java
- GeMMA_prototype_v0.5.maxpat
- HenonMap.class
- HenonMap.java
- major_note_classifier.maxpat
- minor_note_classifier.maxpat
- Sequence.class
- SimpleGA.class
- SimpleGA.java

Questionnaire

- qu1.mp3
- qu2.mp3
- qu3.mp3
- qu4.mp3
- Questionnaire_Hannes_Raffaseder.doc

- Questionnaire_Martin_Parker.rtf
- Questionnaire_Matthias_Husinsky.doc
- Questionnaire_Michael_Jaksche.doc

References

- **Literature**
  - Aschauer_-_Algorithmic_Composition.pdf [Asc08]
  - AudioMostly_2006.pdf [CGH06] [MK06] [Jør06]
  - AudioMostly_2007.pdf [BH07] [HDK07] [Far07]
  - AudioMostly_2008.pdf [BK08]
  - Beveridge_-_Music_emotion_classification_by_audio_signal_analysis.pdf [KCBM08]
  - Birchfield_-_Generative_Model_for_the_creation_of_musical_emotion.pdf [Bir03]
  - Bown_Lexer_-_CTRNN.pdf [BL06]
  - Bown_Wiggins_-_On_the_Meaning_of_Life.pdf [BW07]
  - Eladhari_-_The_Soundtrack_of_your_Mind.pdf [ENF06]
  - Essl_-_Strukturgeneratoren.pdf [Ess96]
  - Hu_-_Creating_a_Simplified_Music_Mood_Classification_Ground-Truth_Set.pdf [HBD07]
  - Miranda_-_Sounds_of_Artificial_Life.pdf [Mir02]
  - Pachet_-_A_Taxonomy_of_Musical_Genre.pdf [PC00]
  - Russell_-_circumplex_model_of_affects.pdf [RPP05]
  - Skowronek_-_A_Demonstrator_for_Automatic_Music_Mood_Estimation.pdf [SMvdP07]
  - Tanaka_-_Interaction,Experience_and_the_Future_of_Music.pdf [Tan06]
  - Wingstedt_-_Narrative_functions_of_film_music.pdf [Win04]
  - Wingstedt_-_Perceived_Properties_of_Parameterized_Music.pdf [BW06]
  - Wingstedt_-_Relations_between_Selected_Musical_Parameters.pdf [BW05]
- **Websites**

- Bea96.html [Bea96]
- Bei04.html [Bei04]
- Bla01.html [Bla01]
- Bra08.html [Bra08]
- Eno96.html [Eno96]
- Ess91.html [Ess91]
- Hou04.html [Hou04]
- Mar04.html [Mar04]
- Mur05a.html [Mur05a]
- Mur05b.html [Mur05b]
- Pro04.html [Pro04]
- Wei08a.html [Wei08a]
- Wei08b.html [Wei08b]
- Wei08c.html [Wei08c]
- Whi04.html [Whi04]

B. Code

```
import com.cycling74.max.*;

public class HenonMap extends MaxObject {

  private float a;
  private float b;
  private float x;
  private float y;


  public HenonMap(Atom[] args) {


    declareAttribute("a");
    declareAttribute("b");
    declareAttribute("x");
    declareAttribute("y");

    declareTypedIO("bffff","ff");

    setInletAssist(new String[] {
            "bang to output next x and y values",
            "float to set value of a",
            "float to set value of b",
            "float to set value of x",
            "float to set value of y"});

    setOutletAssist(new String[] {
            "x value (float)",
            "y value (float)",
            "info outlet"});
  }

  /**
```

```java
    * produce next iteration of Henon map
    */
  private void generate() {

    // apply henon formulas
    float lastX = x;
    float lastY = y;

    x = lastY + 1 - a * (float) Math.pow(lastX, 2);
    y = b * lastX;
  }

  public void inlet(float f) {
    int inletNo = getInlet();

    switch(inletNo) {
      case 1: a = f; break;
      case 2: b = f; break;
      case 3: x = f; break;
      case 4: y = f; break;
    }
  }

  public void bang() {
    generate();
    outlet(0, x);
    outlet(1, y);
  }
}
```

Listing B.1: Full Java code of the HenonMap class

```
import com.cycling74.max.*;
import java.util.Random;

public class CA_1D extends MaxObject {

  private int[] cells;
  private int[] ruleset;
  private int columns;

  Random r = new Random();

  public CA_1D(Atom[] args) {
    declareAttribute("cells");
    declareAttribute("ruleset");
    declareAttribute("columns");

    columns = 10;

    // first and last cells are left blank and will not be dealt with
        by ruleset

    // initialize default starting cell order: 0 1 0 0 0 1/0 0 0 0 0
    cells = new int[columns];
    cells[0] = 0;
    cells[1] = 1;
    cells[2] = 0;
    cells[3] = 0;
    cells[4] = 0;
    cells[5] = r.nextInt(2);
    cells[6] = 0;
    cells[7] = 0;
    cells[8] = 0;
    cells[9] = 0;

    // initialize default ruleset: 90
    ruleset = new int[8];
    ruleset[0] = 0;
    ruleset[1] = 1;
    ruleset[2] = 0;
    ruleset[3] = 1;
    ruleset[4] = 1;
    ruleset[5] = 0;
    ruleset[6] = 1;
```

```java
    ruleset[7] = 0;
  }

  public void list(Atom[] args) {
    // fill ruleset with list arguments
    for(int i = 0; i < args.length; i++) {
      ruleset[i] = args[i].getInt();
    }

    // re-initialize original cell state for convenience
    cells[0] = 0;
    cells[1] = r.nextInt(2);
    cells[2] = 0;
    cells[3] = 0;
    cells[4] = 0;
    cells[5] = 1;
    cells[6] = 0;
    cells[7] = 0;
    cells[8] = 0;
    cells[9] = 0;
  }

  private void generate() {
    int[] nextgen = new int[columns];

    for(int i = 1; i < columns-1; i++) {
      int leftNeighbor  = cells[i-1];
      int thisCell          = cells[i];
      int rightNeighbor = cells[i+1];
      nextgen[i] = applyRules(leftNeighbor, thisCell, rightNeighbor);
    }

    cells = (int[]) nextgen.clone();
  }

  private int applyRules(int left, int cell, int right) {
    if (left == 1 && cell == 1 && right == 1) return ruleset[0];
    if (left == 1 && cell == 1 && right == 0) return ruleset[1];
    if (left == 1 && cell == 0 && right == 1) return ruleset[2];
    if (left == 1 && cell == 0 && right == 0) return ruleset[3];
    if (left == 0 && cell == 1 && right == 1) return ruleset[4];
    if (left == 0 && cell == 1 && right == 0) return ruleset[5];
    if (left == 0 && cell == 0 && right == 1) return ruleset[6];
```

```java
    if (left == 0 && cell == 0 && right == 0) return ruleset[7];
    return 0;
  }

  public void bang() {
    outlet(0, cells);
    generate();
  }
}
```

Listing B.2: Full Java code of the CA_1D class (adapted from [Pro04])

```java
class Sequence implements Comparable {

  private byte[] notes;
  private byte[] velocities;
  private float[] durations;
  private float fitness;

  private int length = 4;

  private Random r = new Random();

  public Sequence() {

  }

  public Sequence(int keynote, int melodicRange, int register, int
      meter) {

    this.length = meter;

    notes = new byte[length];
    velocities = new byte[length];
    durations = new float[length];

    for(int i = 0; i < length; i++) {
      int note = r.nextInt(13);
      note -= 1;

      if(note >= 0) {

```

```
        float rangeFactor = (float) r.nextInt(melodicRange) - melodicRange/2.0f;

        note += register * 12;

        note += rangeFactor * 12;

        note += keynote;
      }

      notes[i] = (byte) note;

      int velocity = r.nextInt(128);

      velocities[i] = (byte) velocity;

      float duration = r.nextFloat();

      durations[i] = duration;
    }
  }

  public int compareTo(Object o) throws NullPointerException {
    if(o == null) throw new NullPointerException();

    Sequence s = (Sequence) o;

    if(this.fitness < s.getFitness()) return -1;

    if(this.fitness == s.getFitness()) return 0;

    if(this.fitness > s.getFitness()) return 1;

    return 0;
  }

  public float getFitness() {
    return fitness;
  }

  public byte[] getNotes() {
    return notes;
  }
```

```

  public byte[] getVelocities() {
    return velocities;
  }

  public float[] getDurations() {
    return durations;
  }

  public void setNotes(byte[] notes) {
    this.notes = notes;
  }

  public void setVelocities(byte[] velocities) {
    this.velocities = velocities;
  }

  public void setDurations(float[] durations) {
    this.durations = durations;
  }

  public void calculateFitness(float rhythmicComplexity, float
      articulationMean, float accentuationMean,
    float accentuationStddev, int mode, int keynote, int melodicRange,
        int register, float harmony, int peakVelocity) {

    int pauses = 0;
    int pausesOnbeat = 0;
    int notesInRange = 0;

    int notesInScale = 0;

    float sumLengths = 0.0f;
    float averageLength = 0.0f;

    int sumVelocities = 0;
    int averageVelocity = 0;

    for(int i = 0; i < length; i++) {
      // count pauses
      if(notes[i] == -1) {
        pauses++;
        if(i \% 2 == 0) {
```

```
            pausesOnbeat++;
          }
        }

        // evaluate if notes are in melodic range
        if((notes[i] <= keynote + 12*register + melodicRange*12/2
          && notes[i] >= keynote + 12*register - melodicRange*12/2)
          || notes[i] == -1) {
          notesInRange++;
        }

        // evaluate if notes are in given scale
        if(mode == 0) {                              // major
          int rootNote = (notes[i] - (keynote - 60)) \% 12;

          switch (rootNote) {
            case 0: notesInScale++; break;
            case 2: notesInScale++; break;
            case 4: notesInScale++; break;
            case 5: notesInScale++; break;
            case 7: notesInScale++; break;
            case 9: notesInScale++; break;
            case 11: notesInScale++; break;
          }
        }

        if(mode == 1) {                              // minor
          int rootNote = (notes[i] - (keynote - 60)) \% 12;

          switch (rootNote) {
            case 0: notesInScale++; break;
            case 2: notesInScale++; break;
            case 3: notesInScale++; break;
            case 5: notesInScale++; break;
            case 7: notesInScale++; break;
            case 8: notesInScale++; break;
            case 10: notesInScale++; break;
          }

        }

        // sum note length
        sumLengths += durations[i];
```

```

      // sum note velocities
      sumVelocities += velocities[i];
    }

    averageVelocity = sumVelocities / length;

    float sumSquareVelocityDifferences = 0.0f;
    for(int i = 0; i < length; i++) {
      sumSquareVelocityDifferences += (float) Math.pow((velocities[i] -
          averageVelocity), 2);
    }
    float velocitiesStddev = (float) Math.sqrt(
        sumSquareVelocityDifferences/(length-1));

    // calculate rhythm factor
    float rhythmFactor = (float) (pauses * (this.length - pauses))/(
        float) Math.pow(this.length, 2);

    // pause positions: onbeat pauses count more
    rhythmFactor += pausesOnbeat/this.length;

    rhythmFactor = 1 - (float) Math.pow((rhythmicComplexity-
        rhythmFactor), 2);

    // calculate tonality factor
    float tonalityFactor = (float) notesInRange / (float) length;

    // calculate harmony factor
    float harmonyFactor = (float) notesInScale / (float) length;

    harmonyFactor = 1 - (float) Math.pow((harmony - harmonyFactor), 2);

    // calculate articulation factor
    averageLength = sumLengths / (float) length;

    float articulationFactor = 1 - (float) Math.pow((averageLength -
        articulationMean), 2);

    // calculate accentuation factor
    float accentuationFactor = 1 - (float) Math.pow((averageVelocity -
        peakVelocity * accentuationMean)/127, 2);

```

```
    accentuationFactor *= 1 - (float) Math.pow(velocitiesStddev/127 - accentuationStddev, 2);

    this.fitness = rhythmFactor * tonalityFactor * harmonyFactor * articulationFactor * accentuationFactor;
  }


  public void mutate() {

    float f = r.nextFloat();
    if(f > 0.9f) {

      int notePos = r.nextInt(length);

      int pos = r.nextInt(8);

      int[] noteArray = byteToIntArray(notes[notePos]);

      if(noteArray[pos] == 0) {
        noteArray[pos] = 1;
      } else {
        noteArray[pos] = 0;
      }

      notes[notePos] = intArrayToByte(noteArray);

      int[] velArray = byteToIntArray(velocities[notePos]);

      if(velArray[pos] == 0) {
        velArray[pos] = 1;
      } else {
        velArray[pos] = 0;
      }

      velocities[notePos] = intArrayToByte(velArray);
    }

  }


  private int[] byteToIntArray (byte byteIn) {

```

```
    int[] byteOut = new int[8];

    if(byteIn >= 0) {

      int temp = byteIn;

      for(int i = 0; i < 7; i++) {

        byteOut[i] = temp % 2;
        temp /= 2;

      }
    } else {

      byteOut[7] = 1;
    }

    return byteOut;
  }

  private byte intArrayToByte (int[] intArray) {

    if(intArray[7] == 1) {

      return -1;

    } else {

      byte b = 0;

      for(int i = 0; i < 7; i++) {

        b += Math.pow(2, i) * intArray[i];

      }

      return b;
    }
  }
}
```

Listing B.3: Full Java code of the Sequence class used in SimpleGA

```
import com.cycling74.max.*;
import java.util.Random;
import java.util.Arrays;

public class SimpleGA extends MaxObject {

  private float harmony = 0.5f;
  private int mode = 1;
  private int keynote = 60;
  private int register = 0;
  private int melodicRange = 2;

  private int meter = 4;
  private float rhythmicComplexity = 0.5f;
  private float articulationMean = 0.7f; // mean note length, stddev
      0.3 gaussian

  private int peakVelocity = 127;
  private float accentuationMean = 0.77f;
  private float accentuationStddev = 0.13f;

  private static final int POPULATION_SIZE = 1000;

  private Random r = new Random();

  private Sequence[] population = new Sequence[POPULATION_SIZE];

  public SimpleGA(Atom[] args) {

    declareAttribute("harmony");
    declareAttribute("mode");
    declareAttribute("keynote");
    declareAttribute("register");
    declareAttribute("melodicRange");
    declareAttribute("meter");
    declareAttribute("rhythmicComplexity");
    declareAttribute("articulationMean");
    declareAttribute("peakVelocity");
    declareAttribute("accentuationMean");
    declareAttribute("accentuationStddev");


    declareTypedIO("bfiiiiiffiff","l");
```

```java

    setInletAssist(new String[] { "bang to output next pattern",
            "float to set harmony value",
            "int to set mode value",
            "int to set keynote value",
            "int to set register value",
            "int to set melodic range value",
            "int to set meter value",
            "float to set rhythmic complexity value (0.0-1.0)",
            "float to set mean articulation note length value",
            "int to set peak MIDI velocity value",
            "float to set mean accentuation",
            "float to set accentuation std dev"});

    setOutletAssist(new String[] { "pattern of 8*meter notes",
            "info outlet"});

  }

  private void evolve() {

    // evaluate population fitness
    for(int i = 0; i < POPULATION_SIZE; i++) {
      population[i].calculateFitness(rhythmicComplexity, articulationMean, accentuationMean, accentuationStddev, mode, keynote, melodicRange, register, harmony, peakVelocity);
    }


    // select POPULATION_SIZE/2 fittest entities
    Arrays.sort(population);

    Sequence[] selectedForReproduction = new Sequence[POPULATION_SIZE/2];

    for(int i = 0; i < POPULATION_SIZE/2; i++) {
      selectedForReproduction[i] = population[(POPULATION_SIZE-1)-i];
    }

    // reproduce 50 fittest entities
    Sequence[] offspring = new Sequence[POPULATION_SIZE/2];
    for(int i = 0; i < POPULATION_SIZE/4; i++) {
      Sequence[] parents = new Sequence[2];
```

```
      parents[0] = selectedForReproduction[i];
      parents[1] = selectedForReproduction[(POPULATION_SIZE/2-1)-i];

      Sequence[] children = reproduce(parents);

      // mutate
      children[0].mutate();
      children[1].mutate();

      offspring[i*2] = children[0];
      offspring[i*2+1] = children[1];
    }

    // merge with population
    for(int i = 0; i < POPULATION_SIZE/2; i++) {
      population[i] = selectedForReproduction[i];
    }
    for(int i = POPULATION_SIZE/2; i < POPULATION_SIZE; i++) {
      population[i] = offspring[i-POPULATION_SIZE/2];
    }

  }



  private Sequence[] reproduce(Sequence[] parents) {

    Sequence[] children = new Sequence[2];

    byte[] notes1 = new byte[meter];
    byte[] notes2 = new byte[meter];

    byte[] velocities1 = new byte[meter];
    byte[] velocities2 = new byte[meter];

    float[] durations1 = new float[meter];
    float[] durations2 = new float[meter];

    for(int i = 0; i < meter; i++) {
      notes1[i] = parents[0].getNotes()[i];
      notes2[i] = parents[1].getNotes()[i];

      velocities1[i] = parents[0].getVelocities()[i];
```

```
      velocities2[i] = parents[1].getVelocities()[i];

      durations1[i] = parents[0].getDurations()[i];
      durations2[i] = parents[1].getDurations()[i];
    }

    byte temp = notes1[0];
    notes1[0] = notes2[0];
    notes2[0] = temp;

    temp = notes1[1];
    notes1[1] = notes2[1];
    notes2[1] = temp;

    temp = velocities1[0];
    velocities1[0] = velocities2[0];
    velocities2[0] = temp;

    temp = velocities1[1];
    velocities1[1] = velocities2[1];
    velocities2[1] = temp;

    float temp2 = durations1[0];
    durations1[0] = durations2[0];
    durations2[0] = temp2;

    temp2 = durations1[1];
    durations1[1] = durations2[1];
    durations2[1] = temp2;


    children[0] = new Sequence();
    children[0].setNotes(notes1);
    children[0].setVelocities(velocities1);
    children[0].setDurations(durations1);

    children[1] = new Sequence();
    children[1].setNotes(notes2);
    children[1].setVelocities(velocities2);
    children[1].setDurations(durations2);

    return children;
  }
```

```

private void resetPopulation() {

  for(int i = 0; i < POPULATION_SIZE; i++) {
    Sequence s = new Sequence(keynote, melodicRange, register, meter)
        ;
    population[i] = s;
  }

}

public void inlet(int i) {
  int inletNo = getInlet();

  switch(inletNo) {

    case 2: mode = i; break;
    case 3: keynote = i; break;
    case 4: register = i; break;
    case 5: melodicRange = i; break;
    case 6: meter = i; break;
    case 9: peakVelocity = i; break;
  }
}

public void inlet(float f) {
  int inletNo = getInlet();

  switch(inletNo) {
    case 1: harmony = f; break;
    case 7: rhythmicComplexity = f; break;
    case 8: articulationMean = f; break;
    case 10: accentuationMean = f; break;
    case 11: accentuationStddev = f; break;

  }
}

public void bang() {

  resetPopulation();

  for(int i = 0; i < 10; i++) {
```

```java
            evolve();
        }

        // evaluate population fitness
        for(int i = 0; i < POPULATION_SIZE; i++) {
            population[i].calculateFitness(rhythmicComplexity,
                articulationMean, accentuationMean, accentuationStddev, mode,
                keynote, melodicRange, register, harmony, peakVelocity);
        }

        // select 4 fittest entities
        Arrays.sort(population);

        String outputString = new String();

        int counter = 1;

        for (int i = POPULATION_SIZE-1; i > ((POPULATION_SIZE-1)-4); i--) {
            for (int j = 0; j < meter; j++) {

                outlet(0, new Atom[]{Atom.newAtom(counter), Atom.newAtom(
                    population[i].getNotes()[j]),
                  Atom.newAtom(population[i].getVelocities()[j]), Atom.newAtom(
                    population[i].getDurations()[j]) });

                counter++;
            }
        }

    }
}
```

Listing B.4: Full Java code of the SimpleGA class

C. Expert Interviews

Five interviews were conducted, two of them in English, and three of them in German; all of them are printed in their original language. The experts were asked to evaluate four exemplary sound files named qu1 to qu4 regarding their emotional content. The test files were produced using the prototype application with the settings *calm, happy, sad* and *tense* respectively.

C.1. DI (FH) Matthias Husinsky

Lecturer, University of Applied Sciences, St. Pölten, Austria
(interview conducted in German)

1. **Sind die vorgelegten Musikstücke Ihrer Meinung nach geeignet, Emotionen oder Gefühlszustände auszudrücken? Wenn ja, welche? Wenn nein, welche musikalischen Elemente fehlen?**
   Meinem Geschmack nach transportieren eigentlich alle Musikstücke eine ähnliche Emotion, die ich am ehesten als "Verwirrtheit" oder "Orientierungslosigkeit" beschreiben würde. Verursacht wird diese Gefühlsregung durch das Vorkommen vieler dissonanter Intervalle. Ein zusätzliches Element – "Spannung" oder "Aufgeregtheit" - kommt durch unterschiedliche Tempi hinzu, wobei eine höhere Geschwindigkeit bei diesen Stücken mit höherer Spannung korreliert.

2. **In welchem Ausmaß hat die Wahl des musikalischen Stils oder Genres Einfluss auf die repräsentierten Gefühlszustände?**
   Die Stücke haben einen sehr ähnlichen Stil, variiert wird vor allem das Tempo (und eventuell der Crash Einsatz des Schlagwerks). Der Stil ist hier bei diesen Stücken nicht entscheidend, da die oben erwähnten Dissonanzen zuviel Gewicht haben und andere Parameter überdecken.

3. **In welchem Ausmaß hat die Wahl der Instrumentierung bzw. der eingesetzten Klangfarbe Einfluss auf die repräsentierten Gefühlszustände?**

Leider schwer zu sagen, da die Instrumentenwahl in allen Stücken gleich ist. Hier gilt dasselbe wie oben – die Dissonanzen wiegen mehr als andere Elemente, also auch die Instrumentierung. Würde das Stück musikalisch "gewohntere" Formen aufweisen, würden die andere Parameter mehr in den Vordergrund rücken.

4. **Worin sehen Sie die Vor- und Nachteile des Einsatzes von generativen Ansätzen zur automatischen, dynamischen Komposition von Medienmusik (zum Beispiel für Videoclips, interaktive Installationen usw.)?**
   Vorteil: Große Arbeits (= Zeit + Geld) Ersparnis.
   Nachteil: Auch hier muss ein gewisses musikalisches Verständnis vorhanden sein, um zu brauchbaren Ergebnissen zu kommen. Ich sehe eine solche Applikation momentan zwischen den beiden Polen:
   1. strenge Musikalische Vorgabe mit Einflussmöglichkeiten auf eher primitive Parameter (Tempo, Tonart, Intensität einzelner Instrumente)
   2. große Freiheit in der musikalischen Gestaltung, aber damit auch mehr "Fehlerquellen" für untrainierte Anwender.

   Ausrichtung des Systems auf ersten Punkt führt zu "funktionierenden" Stücken, die aber wenig Unterschiede zueinander aufweisen und daher schnell uninteressant werden. Ausrichtung auf zweiteren Punkt setzt wieder ein musikalisches Vorwissen (bei der Steuerung der Parameter) voraus, das eben vielen Menschen fehlt, weshalb das System nicht sinnvoll einzusetzen ist.

   Hier einen brauchbaren Kompromiss zu finden ist die Herausforderung.

C.2. Mag. Michael Jaksche, MA

Lecturer, University of Applied Sciences, St. Pölten, Austria
(interview conducted in German)

1. **Sind die vorgelegten Musikstücke Ihrer Meinung nach geeignet, Emotionen oder Gefühlszustände auszudrücken? Wenn ja, welche? Wenn nein, welche musikalischen Elemente fehlen?**
   qu1: Trott, Freudlosigkeit, Mühsal
   qu2: Aufbruch, Stimmungsumschwung nach einer guten Nachricht, Aufgekratztheit
   qu3: Melancholie, Nachdenklichkeit
   qu4: etwas kündigt sich an – Erwartung, Ungewissheit

   Anmerkung: Die perkussive Stimme hemmt bei allen vier Beispielen die Wirkung eher (subjektiver Eindruck); liegt vermutl. 1. an den perkussiven Sounds (Klangfarbe), 2. an den Lautstärkeverhältnissen (Perkussion zu präsent?), 3. an der statischen Rhythmik

2. **In welchem Ausmaß hat die Wahl des musikalischen Stils oder Genres Einfluss auf die repräsentierten Gefühlszustände?**
   Der Einfluss hängt vermutl. am stärksten mit den musikalischen Hörgewohnheiten des Rezipienten/der Rezipientin zusammen – die Gefühlszustände, die "erkannt" werden, stehen zu diesen in Relation

3. **In welchem Ausmaß hat die Wahl der Instrumentierung bzw. der eingesetzten Klangfarbe Einfluss auf die repräsentierten Gefühlszustände?**
   Starker Einfluss! Ein möglicher Grund: Viele Klangfarben werden mit unterschiedlichen mechanischen Instrumenten assoziiert, die allerdings unterschiedliche Spielweisen verlangen; insofern sind Klangfarbe und andere musikalische "Parameter" wie Rhythmik, Melodik und Harmonik eng verwoben und müssen aufeinander abgestimmt werden. Klänge, die keine solchen Assoziationen auslösen können im Sinne wirkungsvoller automatischer Komposition vermutlich freier gehandhabt werden.

4. **Worin sehen Sie die Vor- und Nachteile des Einsatzes von generativen Ansätzen zur automatischen, dynamischen Komposition von**

Medienmusik (zum Beispiel für Videoclips, interaktive Installationen usw.)?
Vorteile: Kostenersparnis (inkl. Zeitersparnis in der Produktion), RezipientIn kommt eine aktivere Rolle ("Prosumer"), dynamische Komposition bedeutet auch, dass keine zwei Durchgänge exakt gleich sind (kann auch ein Nachteil sein)
Nachteile: Vermutung, dass generative Musik (die nicht in einem zweiten Durchgang nachbearbeitet wird) puncto Melodik Schwächen (Beliebigkeit u.ä.) aufweist

P.S.: Beim nochmaligen Durchhören in anderer Reihenfolge hat sich wieder bestätigt, dass der Kontext eine enorme Rolle hinsichtlich der Wirkung spielt.

C.3. Dr Martin Parker

Programme Director of the MSc Sound Design course, University of Edinburgh, United Kingdom
(interview conducted in English)

1. **Are the provided pieces of music, in your opinion, capable of expressing emotions or moods? If yes, which ones? If not, which musical elements are missing?**
   No, not capable of expressing the emotions or moods but certainly capable of underscoring them. 3 and 4 are particularly effective, 1 is much less so. They "resemble" music and in a context where the viewer is not just listening by looking too, these could be stretched out to work well.

2. **To what extent does the choice of musical style or genre have influence on represented moods?**
   This is important as these things are "culturally encoded" in the listener. In general, most listeners know how to interpret a classical orchestra, a rock band and drum and bass. These "styles" say more about the emotional intention than the content perhaps.

3. **To what extent does the choice of instrumentation or timbre have influence on represented moods?**
   Huge. Leading on from answer to question 2, the instrumentation is massively important as is the way each instrument is "played". Micro shadings on the attack time of keyboards, amplitude of drum patterns, the genre and attitude of the genre are the first things to be read and understood by the viewer. Be good to hear an example with some NI battery instruments on the drums, perhaps sending the keyboards thrrough a VST guitar plugin and using "real" drumkit samples would make a huge difference, especially on track 2.

4. **What advantages and disadvantages do you see in the use of generative approaches for the automated, dynamic composition of media music (e.g. for short video clips, interactive installations etc.)?**
   Advantages
   Shorter time span between demand and result.
   Generation of sequence data that can then be shaped into something more interesting by a composer speeds the compositional process : Define the type

of area, generate data, shape data by hand into desired result
Data can be mapped to any sounds
Thematic links between different voices are generated naturally
An interesting line of investigation that has potential to be very fruitful.

Disadvantages
Seeds for the system need to be interesteing in their own right
Many other parameters need to be controlled to create a musical experience
Musical timing is perhaps too accurate, requires some "jitter". Mapping of data to instruments needs to be done carefully, can't be done by someone who has not got experience.
Tricky to emulate genres but genre is the most important "signpost" to understanding emotional signifiers between music and image.
It is very hard for an audience to interpret electronic sounds with image, they are acousmatic and uncomfortable. However, the sounds of instruments are easier for the audience to bear and understand. (Michel Chion).

C.4. FH-Prof. DI Hannes Raffaseder

Programme Director of the Telecommunications & Media Masters course and Head of Institute of Media Production, University of Applied Sciences, St. Pölten, Austria (interview conducted in German)

1. **Sind die vorgelegten Musikstücke Ihrer Meinung nach geeignet, Emotionen oder Gefühlszustände auszudrücken? Wenn ja, welche? Wenn nein, welche musikalischen Elemente fehlen?**
   Ja, zumindest in Ansätzen.
   qu1: langweilig, traurig,...
   qu2: freudig, ausgelassen, aufgeregt
   qu3: ängstlich,...
   qu4: stressig,hektisch,...

2. **In welchem Ausmaß hat die Wahl des musikalischen Stils oder Genres Einfluss auf die repräsentierten Gefühlszustände?**
   Tempo, Taktart, Dichte der musikalischen Elemente, Klanggestaltung, Melodieführung etc. haben sicher einen großen Einfluss auf die rep. Gefühlszustände. Zum Teil bestimmten diese Elemente auch die stitlistische Zuschreibung. Es sollte daher also ein gewisser Zusammenhang bestehen. Es gibt aber durch vielfältige Variation der Parameter auch innerhalb einer Stilistik ein große Bandbreite um Stimmungen darzustellen.

3. **In welchem Ausmaß hat die Wahl der Instrumentierung bzw. der eingesetzten Klangfarbe Einfluss auf die repräsentierten Gefühlszustände?**
   Meiner Meinung nach eine sehr große. Die Beispiele zeigen aber, dass selbst bei weitgehend gleicher Klanggestaltung durch Variation anderer Elemente Stimmungen dargestellt werden können.

4. **Worin sehen Sie die Vor- und Nachteile des Einsatzes von generativen Ansätzen zur automatischen, dynamischen Komposition von Medienmusik (zum Beispiel für Videoclips, interaktive Installationen usw.)?**
   Vorteile: rapid Prototyping, automatische Anpassung an User-Eingaben bei interaktiven Anwendungen, Möglichkeit für "neue" musikalische Ansätze jenseits gängiger, oft verbrauchter Filmmusikklischees

Nachteil: Musikalische Qualität, Einschränkung der musikalischen Parameter, Einschränkung bei den Klangfarben, Einschränkung auf elektronische Sounds bzw. Samples