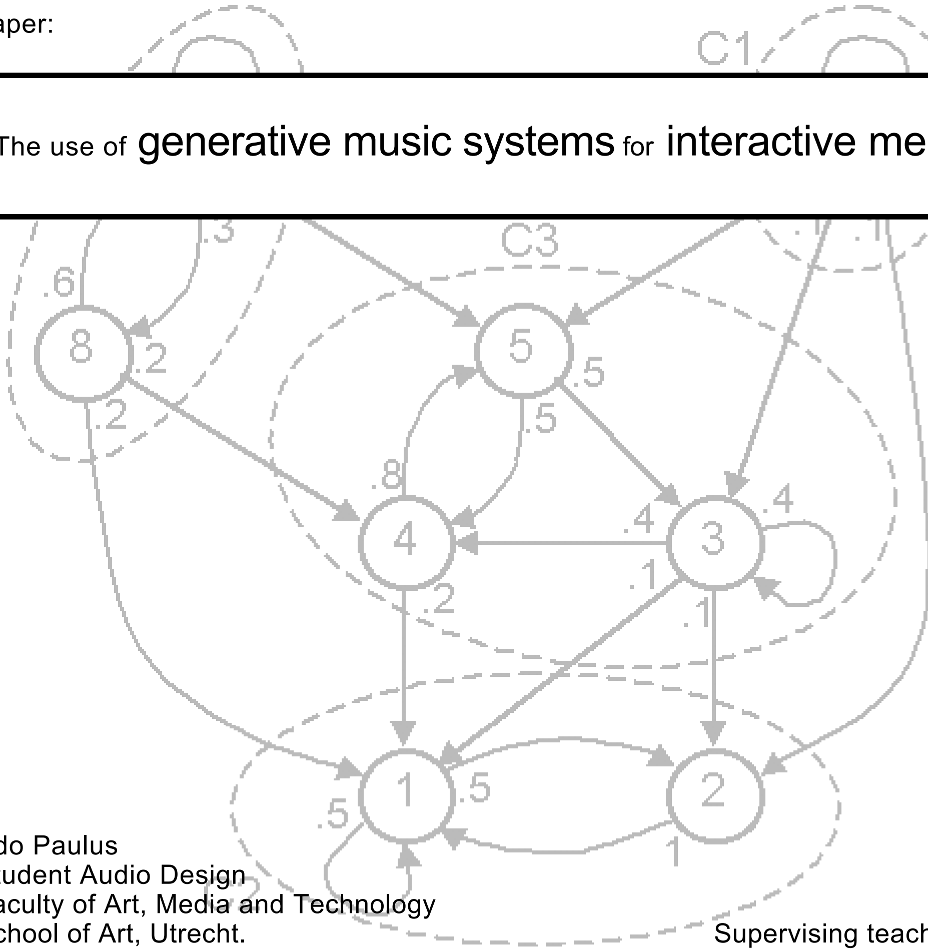
Paper:

The use of generative music systems for interactive media.



Edo Paulus
Student Audio Design
Faculty of Art, Media and Technology
School of Art, Utrecht.
November 2001
-small revisions July 2002

Supervising teacher:
Hans Timmermans

Contents

Introduction

For about three years I'm consciously interested in and practically working on interactive and/or algorithmic music systems. This resulted in various projects: A computergame with interactive music and sound, autonomous computer applications generating non-linear soundscapes, interactive musical systems for live performances with musicians, installations with interactive and/or generative music systems, an internet site with interactive music and a CD-ROM with interactive music. Therefore it seems logical for this paper to be about non-linear music.

Non-linear music, interactive music, generative music, automatic music, adaptive music, algorithmic music. A lot of terms are being used in this field, denoting different types of non-linear music but also very often overlapping or sometimes covering almost identical subfields. This is due to it's broadness, it's recent developments and to the fact that it has influence on very different subcultures: The computer music scene, the computergames scene, the internet scene, the computerart scene and many other scene where electronic music is involved. One thing is obviously similar in all those scenes: The use of computers. The computer, and especially the software, is more and more being recognised as more than a virtual, dare we say 'cheaper', variant of realworld music production tools. The computer has some unique potential. It can render musical result without it being preconceived by the user and it has the possibility of reacting to the user in a complex manner (we all know that!). Making positive use of these potentials is, in my opinion, a more elegant way of using computers.

Then, how can we turn these potentials into concrete, in this case musical, results? This is of course a much to broad subject to fully cover in this paper. Therefore I've narrowed the question to the following:

> What are the advantages, disadvantages, problems and possible solutions of using generative music systems for making music for interactive media?

This is still quite broad, but I will try to give an answer at the best of my capabilities.

In the first chapter of the paper I will make a focus movement starting with the general broad field of systems capable of creating non-linear music (which theoretically covers about all music) gradually zooming in to the specific field I'd like to elaborate about in this paper: generative music systems for interactive media. This to make more clear what this paper is about, and what not, and to get some insight in similarities and differences with neighbouring fields of systems that create non-linear music and last but not least to get to understand some terminology and definitions.

The second chapter will be about the "Interactive media". What types are there and how is the music generally being used? Also I'll investigate the advantages, disadvantages and problems for using interactive and especially generative music for these media.

The last two chapters will be about possible ways to realise generative music systems for interactive media. Trying to exploit the advantages and to consider the disadvantages and problems, these part can be seen as possible 'solutions'.

In the third chapter I will survey the most well known generative methods in use in algorithmic computermusic and judge them on the usability in interactive media.

And in the last part I will offer possible solutions of ways to structure and control the generative techniques that render the final musical output.

A lot of aspects involved in making generative music systems for interactive media are not covered here, though: Musical styles and it's consequences for the design of a generative music system (this paper is at certain points maybe a bit focussed on computer composition- and soundscape styles, but the concepts and techniques mentioned can be applied on many different styles), the purpose and effect of music accompanying vision (for this aspect one can find a lot of information with music for film, there is a nice link at the end of this paper), the technical realisation (on this aspect there are also some links on the final page), the production process (I'd only like to say that it is a <u>lot</u> more work than producing linear music).

About Musical Systems making Non-Linear Music

Linear vs. Non-Linear Music[1]
Only since the invention and widespread application of mechanically reproduced sound man has been able to hear the same music for repeated times in exactly the same and predictable way. This was the first time man actually heard linear music. Fully linear music has all its musical events predeterminedly organised in time whereas with non-linear music possibilities are left open until the relevant moment. Before the possibility of mechanically reproducing sound the score for a musical piece was the closest existing form of linearity in music. But before a person could really hear it, it had to be played by musicians. And every performance, everyone knows, is always a different and unique experience. The music is never played twice the same. One could of course still, and rightfully, argue about differences in reproduction equipment and acoustics, and also a needle might jump out of its groove! So theoretically linear music still isn't, and probably never will be, fully realised.

Non-linearity has always been an important aspect of the experience of music performance. The unpredictability of what was to come set the listener in a heightened state of attention and curiosity. With mechanical reproduction of sound man has always tried to make the music more controlled, more fixed into a linear medium. Only since recently, since the mechanical means are becoming more rigid and very widespread used (like, for instance, the CD-player), man is acknowledging that the non-linearity is actually desirable (that's why the less rigid analogue synthesisers are sometimes preferred over digital synths). Now man is also consciously trying to make the mechanical reproduction more non-linear, more like a performance.

Musical Systems

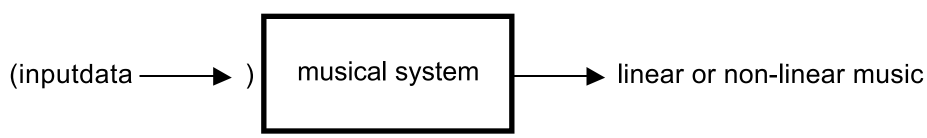


fig. 1

One can imagine music as always coming forth out of a musical system. One can consider a jazz ensemble as a musical system. The system is working and the result of this activity is music. One can imagine a musical instrument, for instance a guitar, also as a musical system. For a guitar to become active and produce some music the guitar clearly has to be influenced by some external force. A musician might do that job. And a third example of a musical system could be a recordplayer. When the recordplayer is playing one hears music, given that the electrical signal coming out of the recordplayer is being converted to airwaves, of course.

Musical Systems Scheme
Now, how can we make a classification of these systems with respect to their creating of more linear or more non-linear music. There are two general aspects to consider here. First, adaptability to input from it's environment, and second, predictability of its output depending to its input. As will be made clear later, one might state that the more adaptive a musical system becomes or the more

[1] Non-linear music should not be mixed up with the term non-linear systems. Where non-linear music is about music not having its musical events predeterminedly organised in time, non-linear systems are systems that do not respond to an input in a proportional way. They can correspond to very simple input in a very complex way. Frequency modulation, waveshaping and chaotic systems (see chapter three) are examples of non-linear systems. For more information on non-linear systems, see Roads (1996, p. 887) or (van Swieten, 1979).
In order to avoid any misunderstanding I will not use the term non-linear system. So, when mentioning non-linearity in this paper, it will only be about not fully predetermined events in time.

unpredictable it becomes, the more non-linear its music output will be. Specifically for this paper, I'd like to add a third aspect to this classification system; the musicality of its input.

Instead of making some discreet categories I'm proposing a three-dimensional scheme (fig. 2). Each musical system can be placed somewhere within this continuum. This is of course a very gross approach, but nevertheless useful in the perspective of this paper.

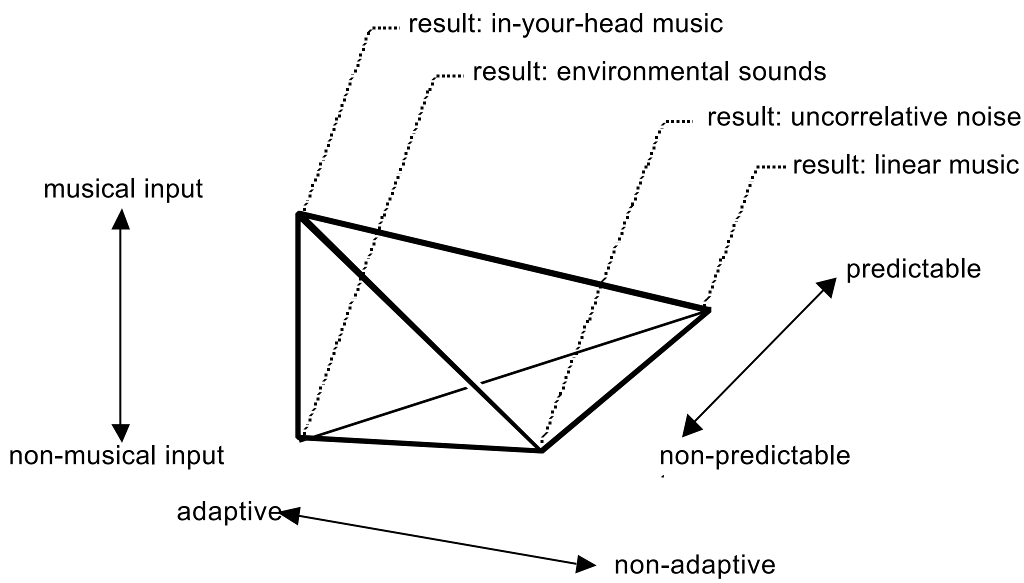


fig. 2

The first dimension distinguishes between adaptive and non-adaptive systems.
If a system works fully of it's own and can not be influenced from the outside then this system is non-adaptive. A CD-player for instance works mostly on it's own, you may only start or stop the machine, so it can be positioned almost fully on the non-adaptive side of this dimension.
The more the output of the musical system is being affected by external controlling the more adaptive it can be considered. The earlier mentioned guitar is quite adaptive but it still imposes a lot of restrictions. It would be quite difficult getting the sound of a symphony orchestra out of the guitar. The guitar as a musical system then would be placed somewhere a bit more to the adaptive side of this dimension.
A totally adaptive music system, which is more of a theoretical situation, would be a system that doesn't limit the input at all. This could be a system that just literally translates the fantasised music in ones head into real sound. In this extreme situation it wouldn't be much of an adapting system anymore, it would become more something like a seamless translator.

Another distinction is between systems rendering musical output in a way more predictable versus more unpredictable.
The CD-player has a predictable way of processing. Add that to the almost non-adaptive aspect of the device and it results in being one of the most linear-music-producing musical systems. By setting the CD-player into shuffle mode the otherwise quite predictable output becomes somewhat non-linear because of the added unpredictability. Or imagine again the jazz ensemble making improvised music. For sure their output is gonna be jazz, but what notes it exactly will be you won't know. Even a well trained orchestra playing a traditional score has some form of unpredictability. The musician will perform different every time its played and the instruments always have some slight uncontrollable variations in their tone.

A totally unpredictable music system would generate only uncorrelative noise. Any form of adaptation seems impossible then.

Finally, we can distinguish between musical and non-musical input.
Musical input can be music itself, a representation of an aspect of music or other input intending to influence the output in a musical manner. Non-musical input is any other type of input.
A musician in the jazz ensemble can on itself also be imagined as a musical system. It listens to the other musicians, the input, and produces additional music influenced by that input. A musical system with non-musical input could be a soundinstallation that's being influenced by weather conditions. One could also fit the musical system used for John Cage's 4,33" (four minutes and thirty-three seconds, 1952) into this scheme. In this piece the performers sit still during the entire length of the piece, while the sounds of the concerthall and outside become the resulting music. The input of this system is non-musical and the system adapts itself completely to the input, so this musical system should be placed in the other extreme corner.

Also notice that the more a musical system becomes adaptive, the less influential the predictable-non-predictable dimension becomes. And the more such a system becomes non-adaptive, the less will be the influence of the musical-non-musical input dimension on the output of the system.

About Music for Interactive Media

Well, where can one place interactive music for interactive media in this non-linear-music-system-continuum? Before we can answer that question we first have to know a bit more about interactive media.

Theoretically interactive media is all media where the user can have influence on the output of the medium. A lot of media can have interactive aspects, like actors in a play playing quit different depending on the audience being completely spellbound or throwing tomatoes at them, but since this paper is about interactive media involving computers media such as CD-ROMs, computergames and internet are a more appropriate mindset.

A Brief History of Interactive Music for Interactive Media

Computergames have always been the most technologically innovative branch in the interactive computermedia in respect to interactive music. This brief history therefor is mostly about the computergames biz.

The beginning of interactive music for interactive computermedia started with videogames in the 70's. In those times the only means of sound reproduction were simple tonegenerators. Although the audio was usually done by a programmer in a few spare hours and although the soundquality was very low, there was some interactivity. The games Space Invaders and Asteroids both employed a simple device to aurally communicate urgency to the player: They speeded up the music. Incoming space invaders where represented by four bass notes going down.

By the mid 80's developments in the electronic musical instrument technology made more "real"-sounding music possible. Higher-quality synthesis, digital soundfile instruments played by MIDI scores and especially complete digital audio recordings (first in arcade games, later also in the computergames at home) made an impact on the audience and the industry. The goal was to make it as attractive as the music you'd hear on the radio. It was more about good songs as opposed to a good soundtrack. This sort of music was a big step away from interactivity.

Also In the beginnings of the 90's MIDI, which has some potential to be used for interactive music, had to make way for the complete digital audio recordings. MIDI was limited to the quality of the soundcard in the personal computers and, even worse, it sounded different on every computer. Soundquality was the main issue. The lack of interactivity was often compensated by using different songs as digital audiofiles and to switch between them to respond to the gamestate.

The game Myst, released by Cyan in 1993, was an important step forward for music in games. Instead of songs the music was more like a soundtrack. The game is of the adventure genre and is a sort of slideshow of pictures of an imaginary world mixed with short linear animations. The music is in an ambient style and is subtly blended together with the many environmental sounds. The combination of this musical style with the slideshow interface made it fairly easy the make the music adequately adaptive to the gamestate. When a new picture of the world popped on the screen the appropriate high-quality audiofile could be crossfaded with the previous one. Also some out of sync vertical layering of audiofiles was used to mask the inevitable looping of the audiofiles a little bit. Myst proved that the music and audio could, by adapting to the gamestate, increase the immersiveness of the game-experience while maintaining a high soundquality.

Since then this adaptability of audiofiles has been taken quite a bit further in some games. Fragments of music are being rearranged to make the music instantly adapting to the game. A programmed set of rules decides on how to react and how to combine the fragments in a musical way that it stays rhythmically and harmonically coherent.

Since the late 90's computers are becoming fast enough to do real-time softwarebased soundsynthesis and sample playback. This makes the use of MIDI attractive again. Also the interactive and variation possibilities are larger. Re-rendering the music at the moment of listening instead of using pre-recorded soundfiles results in much smaller filesizes. With the limited bandwidth of the internet this technique also has great potentials for this relatively new medium. The first initiatives in this field are emerging.

These techniques for interactive music nowadays will be worked out later. First let's get to the arguments for using interactive music clear.

Advantages, Disadvantages and Problems with Interactive Music for Interactive Media
The use of non-linear and interactive music for interactive media can have two main advantages opposed to linear music:
Most important is the ability of the music to adapt itself to the changing situation of the interactive media. Instead of a general background tune creating a general mood, atmosphere or vibe, the music can change according to the changes in the interactive media. This way the music can heighten the immersiveness of the total media experience. The goal of this interactive music is actually to be as coherent with the interactive media as a moviescore is with a movie.

The second main advantage is that non-linear music is able to constantly vary itself. It may never sound exactly the same as the previous time the interactive media was experienced. With linear music the same audiofile will be played again when the interactive media is experienced again or when the audiofile has ended. This repeated listening to exactly the same music may often start to annoy a user. Users playing a computergame almost always turn off the music if its a looped audiofile and if its possible to turn it off.

A disadvantage of non-linear music opposed to linear music is that with linear music the total track can be produced to be as much as possible a satisfactory musical experience. "The big problem is that most of the things that one can do to a track to make it interactive are things that take it a step away from being a well arranged song" (Harland, 2000). This disadvantage is not really a problem, in music for films music also has to add to the total experience and cannot be subordinate only to its own laws of beauty and meaning. This is merely an aspect to be aware of.
The main problem is that the possible amount of possibilities how the interactive media is going to behave is practically unlimited. And the interactive music has to change, vary and adapt to every situation in a musically 'nice' way. This is really a different situation compared to film where the specific music is made to fit to the specific scene.

Now, let's place the music system producing this interactive music in the musical systems scheme. By doing this it becomes more clear what aspects to think about when realising such a system.

Classification of Interactive Music Systems for Interactive Media in the Musical Systems Scheme
The scheme in figure 3 roughly represents the situation of an interactive music system for interactive media.

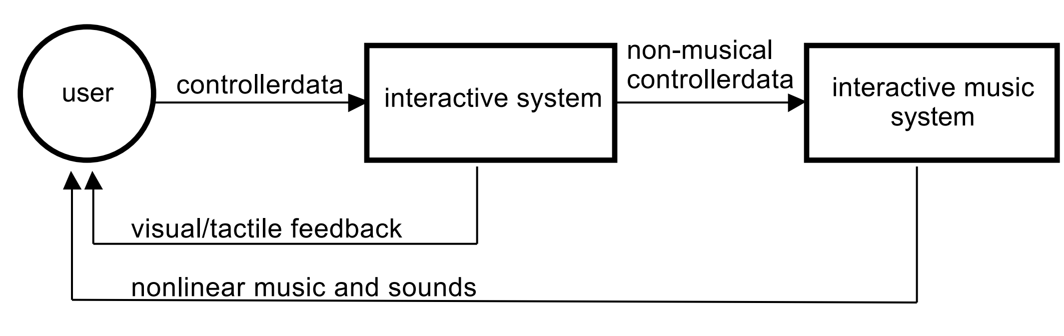


fig. 3

The user is primarily focussed on the interactive system (like a computergame or a website for instance). Through all sorts of physical interfaces the user can control the interactive system. The system in turn gives feedback to the user through different sorts of media. These media can be vision (which usually is the most prominent one), tactile feedback (which usually not used at all) and sound (which is sometimes used). The user in turn can react to that feedback and so on. This continuous reacting between user and system makes this an interactive situation. The sound feedback here is represented by a separate loop. The general system sends controllerdata to the interactive music system. This data is a representation of the situation of the interactive system itself and is generally not coming directly from the user. The interactive music system produces music and other sounds depending on its input.

Now, where can one place this music system in the non-linear-music-system-continuum mentioned in the previous chapter?
The amount of interactivity depends largely on the type of interactive media and on the amount of data being send to the musical system. A website might only send information about the current page being in front of the user, while a computergame might send a lot of information, like the position of the player in a virtual space, the score, the health of the player, and so on. One can say the less information is coming to the musical system, the more the musical system has to have internal unpredictable decision-making to keep producing music with the same amount of non-linearity.
Also the musical system cannot be to unpredictable, because interactive media and the music have to be in some way related. If the music becomes to unpredictable it will have a identity of its own which has nothing to do with the interactive media anymore.
Since the user is not functioning as a musical performer and also cannot directly control the music and sounds, the controllerdata being send to the interactive music system is not intended to influence the output of that system in a musical way.
Thus, one can place interactive music systems for interactive media somewhere in the region marked grey in figure 4.

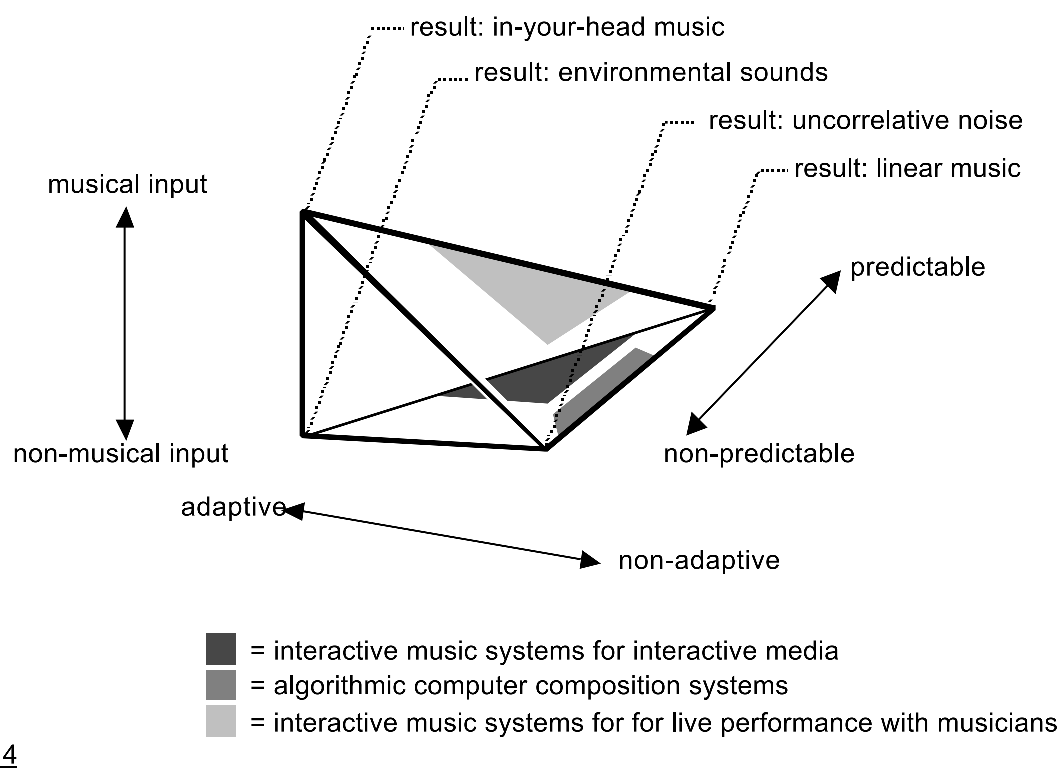


fig. 4

There are other regions in this scheme which have some similarities and some differences with interactive music system for interactive media.

One is the field of algorithmic computer composition systems. This region is marked with horizontal lines in the figure 4. These systems are primarily focussed on getting the computer to generate music, or musical material in an automatic way. There is usually not much interaction involved in this generative process. The system itself is modified to suit the needs and wishes of the composer, or the output of such systems is used as raw musical material and organised and modified further to get to a satisfying composition. Since the composer often uses this method to generate music which he/she would never have created using his/hers own imagination, these systems are often made to be quit unpredictable.
Another region is the field of interactive musical systems for live performances with musicians. In figure 4 this region is marked with vertical lines. These systems are controlled by a musician in a performance situation where the input can be real-time manipulation of the rules inside the system that generate the music or, as more often is being the case, controllerdata of musical gestures or actual audiosignals. In these situations the input is always intended to be musical.

For making interactive music systems for interactive media both these other fields can offer solutions in how to utilise the previously mentioned advantages of interactive music systems for interactive media and how to get around the disadvantages and problems.

'Solutions' for Interactive Music with Interactive Media
The interactive music system has to be some sort of improviser, reacting instantly to a situation, composing the music using of a collection of rules and predetermined musical material and performing it at the same time. This is a complex situation. How can one get a clearer view of the inside organisation of an interactive music system performing such a complex task?

"The processing chain of interactive computer music systems can be conceptualised in three stages" (Rowe 1992, p. 9). The interactive music systems Mr. Rowe talks about in his book are meant to be used in live performance situations with musicians. I have adopted some of his concepts and modified them to suit the situation of the non-musical interactive media. Figure 5 visualises the concept of the three stages of the processing chain.

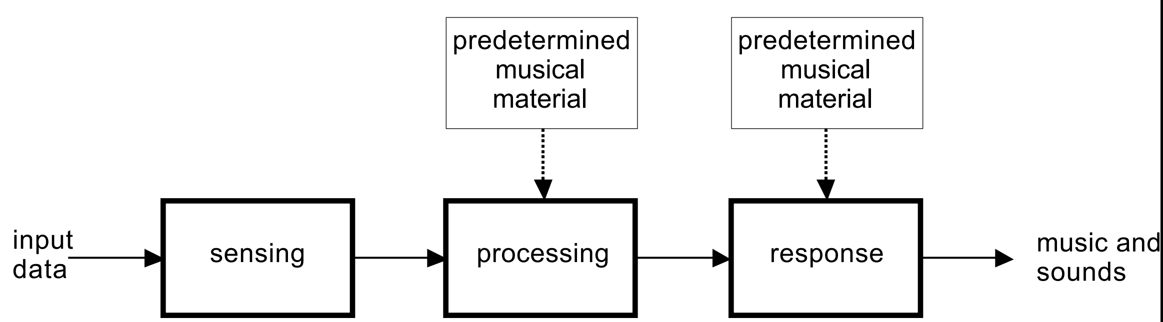


fig. 5

The first is the sensing stage. This stage deals with managing the datainput coming from the interactive media. The most important aspects here are what kind of data is coming in and how this is going to be mapped to the processing part.
The second stage, the processing stage is where the composing happens. Here, in the core of the system, are the decisions being made on how to produce the music. These decisions are made depending on the inputdata, on the musical situation the system is in, and also on a set of rules, so called algorithms. The result of these musical choices are being send to the last stage.
In the last stage, the response stage, the audiosignal is being rendered.

As briefly mentioned before one can distinguish two different types of methods of creating the non-linear music coming out of this system: Sequenced or generative.

Sequenced techniques use pre-recorded music fragments (both MIDIfiles or complete audiofiles). These fragments can be rearranged both vertical (textual re-orchestration) and horizontal (re-ordering of fragments in time). The processing stage decides which fragments to play at what time. These choices go to the response stage where the relevant fragments are being selected from the database of predetermined musical material. The fragments are being played and mixed thus resulting in the final musical output.
With sequenced techniques the processing stage is relatively small. Most of the musicality of the system is already fixed inside the prefab music fragments.
In the case of audiofiles this method can use up a lot of the datastorage capacity of a computer. On the other hand, the use of CPU-power is usually less than with the other, generative method.

A generative method has a much more elaborate processing stage. Instead of deciding on fragments this stage decides on what sorts and values of soundparameters to send to the response stage. These soundparameters can vary from pitch, loudness and timbre of a sound to all sorts of soundsynthesis parameters. Since there are a lot more choices to be made the amount of algorithms is larger and the algorithms are more complex. This method is also referred to as algorithmic music.
Apart from the large amount of algorithms there is usually also a collection of predetermined musical material. This collection consists of elementary source material such as pitchscales, sets of durations, possible chord progressions or sets of synthesisparameter settings.
The musicality of a generative system is largely being defined by the kind of algorithms used and how they are implemented. Also the elementary source material has a large influence on the character of the music.
The response stage uses one or more types of software based synthesis methods to render the final output. These synthesis methods can, of course, be any. Most used are samples, but physical modelling would actually be a more or even most generative method. The response stage may also use some predetermined musical material, like for instance presets of soundsynthesis parameters, basic waveforms (samples) or envelope curves.
Generative methods use a lot less datastorage capacity than the sequenced method but will probably use more CPU-power.

As before with the classification of musical systems the sequential and generative methods are also not two discrete categories. All sorts of hybrid structures are possible. For instance, with a generative method, the collection of predetermined musical material of the procedure stage could contain a set of musical fragments in a MIDI kind of representation. Then these fragments could be manipulated in different sort of ways in the processing stage, like transposing, different loudness, variable speeds, playing backwards and so on. This would be a method using both generative and sequenced techniques.

The next part of this paper will be only about the generative method. In chapter three different algorithmic techniques will be investigated to be used in the processing stage. Chapter Four will digress on the sensing stage: The mapping of the incoming data from the interactive media to the different algorithmic procedures in the processing stage. The response stage is not further elaborated on since this has already been done in a lot of other literature. The soundsynthesis techniques are explained in a lot of literature about electronic music in general. The mapping of the output of algorithmic procedures in the processing stage to these soundsynthesis techniques is an important aspect of algorithmic composition in general and has been investigated already many times.

Different Algorithmic Methods for Music Composition

This Chapter will give a description of most well known algorithmic computer composition methods and in addition it will judge them on their usability in interactive media.
The most important criterion is whether the algorithmic method can react appropriately to changes in the situation of the interactive media. A lot of computer composition methods are focussed on generating a compositional form. The total time of the resultant linear composition is being divided and structured according to algorithmic procedures. These methods are of course not suitable for interactive music. In the situation explored in this paper, the interactive music system needs to be able to respond at any moment, during the performance of the music, to the input from the interactive media. It's, so to say, working in *real-time*. The algorithmic methods more appropriate are the ones making the musical decisions on the same time as they are being heard, or scheduled not to far away into the future.

Deterministic vs. Stochastic
There are two contrasting ways for computers to make choices:
deterministic versus stochastic. "A deterministic process generates output by carrying out a fixed but possibly complicated compositional task that does not involve random selection. (…) Stochastic procedures, on the other hand, integrate random choice into the decision-making process." (Roads 1996, p. 834-6)
Both techniques are usable for music with interactive media, but for different reasons.
Some of the variables supplied to a deterministic procedure may come from the interactive media. In that way the output of such a process will be different when the input is different and thus result in variation of the music, but more important; the musical result can have a clear relation with the interactive media.
Stochastic processes can be useful for adding variation and unpredictability. This will make the musical output more enjoyable for longer periods of time or when listened to repeatedly. Also "Stochastic techniques may be used to produce *fuzzy edges* and to 'humanise' computer-generated sounds." (Jones 1981). Keep in mind though that large changes in the musical output due to stochastic decision-making may result in the music being a less coherent part of the complete multi-media experience.
An other, more technical, benefit of stochastic processes is that it may offer useful means of datareduction. When using audiofiles, they can claim a lot of the computer's datastorage capacity. By using stochastic means for variation one needs lees audiodata for the same amount of diversity.

One should add, though, that stochastic processes aren't always more unpredictable than deterministic processes. "Stochastic generative schemes may produce results that sound far more ordered than what might be produced by a supposedly deterministic system" (Jones, 1981). Deterministic systems such as cellular automata and chaotic systems, both mentioned later, can have much more unpredictable results than for instance brown noise, also to be mentioned later.

Aleatoric Processes
The most simple stochastic techniques are aleatoric processes. An aleatoric process generates random numbers between a maximum and minimum value.
By changing the boundaries over time one can create so called *tendency masks* (fig. 6) (Roads 1996, p.842).

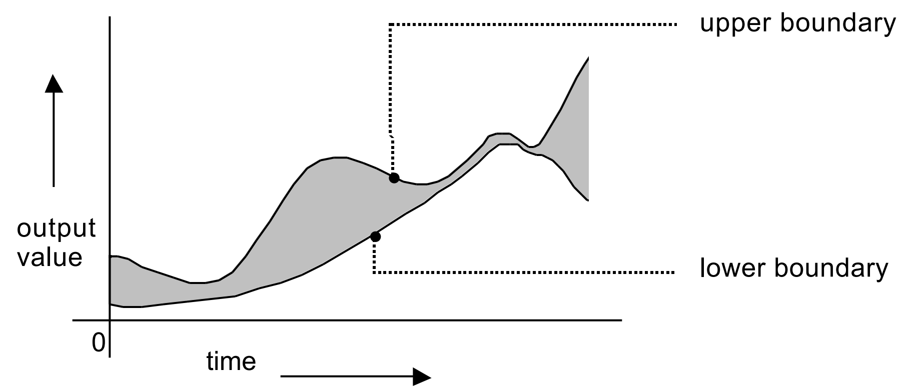


fig. 6

This method for instance has been widely used for granular synthesis techniques.
It can also be used very well in generative music systems for interactive media. One can use tendency masks to control the ambitus of a melody or the notes of a harmonic progression for instance, or the dynamic intensity of percussive sounds.

Probability Distribution
Instead of aleatoric techniques, where every value between the maximum and minimum value has an equal chance of occurring, one often uses probability distribution. A lookup table, used for this technique, shows the likelihood of occurrence of all possible output values. Figure 7 gives an example of a probability table lookup.

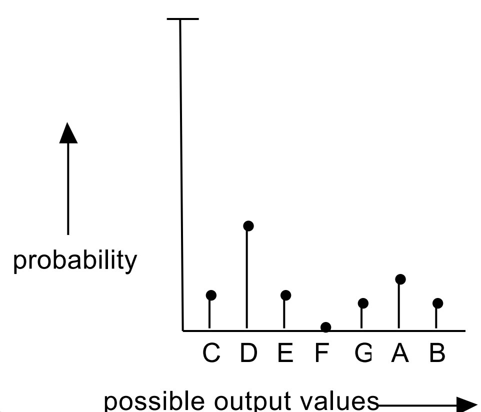


fig. 7

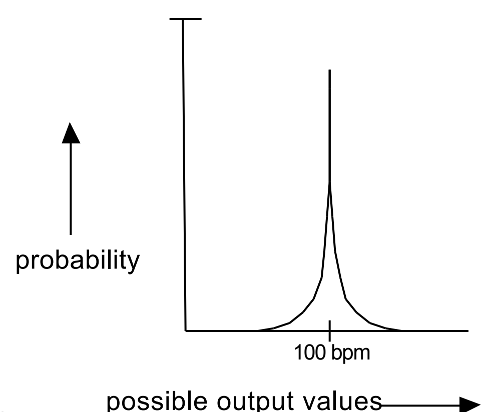


fig. 8

With probability distribution one can make certain output values occur more often than others. One can for instance suggest a fundamental tone. Or one could make slight variations in timing of some music by mapping the output of the lookup table in figure 8 to the metrum speed.
A logical way to extend the possibilities of this technique would be to change the probabilities over time (Roads 1996, p. 874).For instance to interpolate from one probability lookup table to an other. This approach also makes it very useful in an interactive environment, where the probabilities are being influenced by controllerdata coming from the interactive media.

Brown Noise
A typical from of random is brown noise, also sometimes referred to as random walk. Brown noise differs from the previous mentioned noisetypes in the way that it's output is dependent on its previous output. Normally, the next chosen output value is equal, one higher or one lower than the previous output value. This result in a gradually changing value.
By controlling the stepsize (the allowed range to which the next value can increase or decrease) and the probability of going up or down this parameter can react very well to the interactive media. Such a parameter can for instance control the loudness of a wind-like soundeffect.

Markov Chains
An other stochastic technique which output is also dependent on its previous output is the Markov chain. "A Markov chain is a probability system in which the likelihood of a future event is determined by the state of one or more events in the immediate past." (Roads 1996, p. 878). In a *state-transition matrix* all probabilities of the Markov chain are laid out (fig. 9).

curren t state	next state 1	2	3	4	5	6	7	8
1	0	1	0	0	0	0	0	0
2	.5	.5	0	0	0	0	0	0
3	.1	.1	.4	.4	0	0	0	0
4	0	.2	0	0	.8	0	0	0
5	0	0	.5	.5	0	0	0	0
6	.1	0	.1	0	.1	.7	0	0
7	0	0	0	0	.4	0	.3	.3
8	0	.2	0	.2	0	0	.6	0

fig. 9

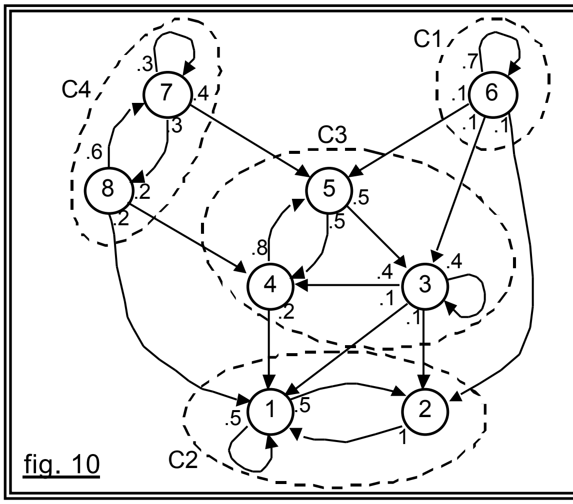


fig. 10

(Fig. 9 & 10 are examples used from Jones 1981)

The row of probabilities corresponding to the current state is used to derive the next state. The row corresponding to this new state is used to derive again a following state, and so on: hence the chain aspect of this kind of stochastic algorithm.

The example in figure 8 is a *first-order* Markov chain; it looks back only one state to choose the next state. A *second-order* Markov chain looks back two states and consequently has a three-dimensional state-transition matrix. Higher-order Markov chain are also possible resulting also in even more dimensional matrices. The higher the order the more coherent the produced patterns will be. Taken to far however such systems can become very predictable or very time consuming in order to work intentionally.

In a musical algorithmic system a state can defined as all sorts of musical parameters. A state can be the pitch of a note, thus generating a melody. Or a state can be a duration which makes the Markov chain generate rhythmic patterns. Also a state can stand for a combination of values for these parameters (for instance a set of a pitch-, duration-, amplitude-, soundcolor- and following-silence value). One can also think about chords, resulting in chordprogressions, or phrases, bars, complete audiofiles. A state can represent a set of soundparameters of any soundsynthesis method between which the system can interpolate.

Markov chains are very well usable in a generative musical system for interactive media. By changing the probabilities in the state-transition matrix the controllerdata coming from the interactive media can have far-reaching influence on the patterns coming out of the process. This way the output can vary from a completely predictable periodic pattern, to a periodic pattern with small variations, to brown-noise-like patterns, to quite random progressions with certain preferred behaviours, to completely aleatoric results. In fact all previous mentioned stochastic processes can also be achieved by using Markov chains which makes it a general all-round technique when using stochastic generative processes.

Classifications of States in a Markov Chain

"When one defines a Markov chain for a specific musical task, a number of properties of states can be very useful for describing relationships and classifications"(Jones 1981) and thus to get some insight in the behaviour of that system. Figure 10 is a *state-relation diagram* corresponding to state-transition matrix in figure 8. As clearly seen, the states can be grouped into different classes (C1 to C4 in this example). The states inside one class may all be reached, directly or indirectly, from any state in that same class. Further, different types of classes can be defined. Once a process has entered a class and can never move out of it, this class is said to be recurrent. In figure 10, class 2 is such a recurrent class. On the other hand, if it is possible for a process to leave a class, then this class is said to be transient. One can also conclude that if a process leaves a transient class, it is impossible return to it. Class 1, 3, and 4 in figure 10 are transient classes.

The distinction between transient- and recurrent classes is very usable for use with interactive media. By changing the probabilities of a Markov chain the situation of the process may change from being in a recurrent class to being in a transient class, this way the transient class is being used for making a coherent transition from one type of behaviour of the Markov chain to an other.
One would usually want only one recurrent class at a time in a Markov chain, because that makes the type of behaviour more predictable a thus more intended for that specific situation in the interactive media. With two recurrent classes two types of behaviour of the Markov chain could be possible with the same information coming from the interactive media! One should consider this possibility to be desirable.

Musical Coherence of Markov Chain Processes

Once a process reaches a recurrent class though, the output can after a while exhibit a kind of aimless meandering. This is a general property of Markov chains. There are several ways to make the output on a global scale more musically coherent.
As previously mentioned one can increase the order of the Markov chain. Also mentioned before is defining states as larger elements such as phrases, (multiple) bars or prefab audiofiles. But this is actually only slowing the pace of the Markov chain and lengthening the time before the aimless meandering becomes notable, it is not really making the Markov chain work more musically coherent.
A third way would be to use hierarchical Markov chains. One chain would generate high-level structures which would decide on changing the probabilities of a chain generating low-level patterns. More levels result in a longer musical coherent progressions, but also, of course, make it more laborious. And still, the topmost chain won't be any more coherent, it would only be employing a longer durations between succeeding states..
How many hierarchical levels one needs depends mostly on the speed and type of progression of the interactive media and thus finally also on the user.

Automata

"An automaton is a procedure whose output depends on its internal state and its input." (Roads 1996, p. 857) An example is an automaton that harmonises an input note alternately by a perfect fourth or fifth interval. It maintains an internal state (the previous interval) and its output notevalue depends on its input value. Automata can use stochastic processes but are mostly deterministic. Automata can be used in different places in a interactive music system together with other algorithmic procedures to influence the behaviour of specific musical parameters, such as pitch, loudness, soundcolor, etc.

Automatons can also be linked with one another in a network. At each step of a global clock all new states and outputs off all automata are calculated depending on the input coming from other linked automata. Once such a system is started it can run autonomously. When mapping the output of such a system to musical parameters one can use the output of only one automaton, but also of several or maybe all.

The *weight* of the links between the automata can be de- or increased and/or inverted in order to change the influence of one automaton on the behaviour of an other automaton. This is a way to make such a system controllable by the interactive media.

One speaks about *cellular automata* when all the automata, or *cells*, in a network are identical and when all interconnections between the automata don't change over time and are ordered in a regular symmetrical structure. Such a structure can, for instance, be an array of cells, or a two-dimensional field of cells.
The interconnections usually work both ways. A famous two-dimensional cellular automaton is The Game Of Life.
For musical application cellular automata have been used to generate tone parameters (pitch, loudness, etc.), soundsynthesis parameters (amplitude envelopes of additive synthesis, for example) and have emulated physical objects in their generation of sound (a type of physical modelling).

A primary characteristic of cellular automata is the appearance of *complex emergent behaviour* (Mc Alpine, Miranda and Hoggar, 1999). The rules inside a cell are usually very simple and the complex results are a consequence of the large amounts of cells used. Complex emergent behaviour is very useful when specifically working on a musical composition: It generates a lot of not-intended but internally coherent musical information which later can be used or discarded by the composer to make a composition. When designed for use with interactive media on the other hand, an interactive musical system exhibiting complex emergent behaviour is usually much less appropriate because the resulting music becomes an unpredictable identity of its own instead of being coherent with the other media. This makes cellular automata a much less usable method for generating music with interactive media.

This complex emergent behaviour can also be the case when using different automata with varying connections, especially when using feedbackloops in such a system. A way to prevent this or to keep it controllable would be to use less automata and instead make the processes internally inside the automata more elaborate and more designed for a specific musical task.

Other Algorithmic Methods
Of course there are many more types of processes being used in algorithmic composition. Here I will give a brief summary of some of them and judge them on their usability with interactive media.

Chaotic Systems are very useful deterministic processes for generating a lot of internally coherent musical data because, like with cellular automata, they exhibit a lot of complex emergent behaviour. This is also the reason why they are not very useful for being used with interactive media. For more information about Chaotic Systems see Bidlack (1992) and Roads (1996, p.886).

Fractals (or *1/f noise*) are, like many other processes mentioned before, another class of probabilistic systems that make decisions depending on previous output. These processes haven't been used a lot in a real-time, interactive situations, though, but mainly for the generation of coherent musical data to be used for autonomous compositions. Nonetheless, there might be some usable ways to regulate the amount and type of change in the output of such systems. I leave this open for further investigation. For more information on Fractals see Roads (1996, p.880).

Genetic techniques are also being used for in algorithmic composition. These processes use evolutionary means, selection through evaluation of fitness and mating and mutation of these fittest, to get the most appropriate musical data for certain conditions. Although the evaluation of fitness might be a good, but possibly difficult, way to select appropriate music for certain environmental conditions (the situation of the interactive media), These processes are not very usable because they are not suitable for a situation where real-time music generation is required. For more information on Genetic techniques see Burton & Vladimirova (1999).

Summary
The most usable algorithmic methods to be used for interactive music with interactive media are both deterministic and stochastic processes, different sorts of noises (possibly also fractal noises), automata, and Markov chains. And important positive aspect of the better processes is their choice-making being dependent on previous choices. The algorithms exhibiting complex emergent behaviour

are less usable in co-operation with interactive media since it makes them less predictable and controllable.

Connecting Algorithmic Music Systems to Interactive Media

Now that there seem to be some suitable algorithmic methods, how can they be made to react in a coherent way to the interactive system? In other words, what kind of data does the interactive musical system need from the interactive media and how can one map this input to the musical parameters? Of course there are innumerable ways to do this. This chapter will mention aspects to keep in mind and a few suggestions of possible solutions.

Organisation of Algorithmic Procedures

The previous chapter presented some usable core algorithms to generate musical data. An important aspect of many of these processes is that its output is dependent on its previous output. This aspect gives the musical output coherence in time. But apart from musical events following each other in time, different musical events are also happening at the same time next to each other. An important aspect, and an important cause for the appreciation of music, is its multi-layeredness. Layers behaving differently, but being coherent. How can this coherence between multiple layers be achieved? Multiple layers sound coherent when they use a shared metrum or when the pitches of different tones all fit in the same scale or the same chord, for instance.

Coherence between the behaviour of different algorithmic processes can be achieved in many different ways. A familiar way to do this is to organise the algorithms hierarchically in multiple levels of abstraction (fig. 11).

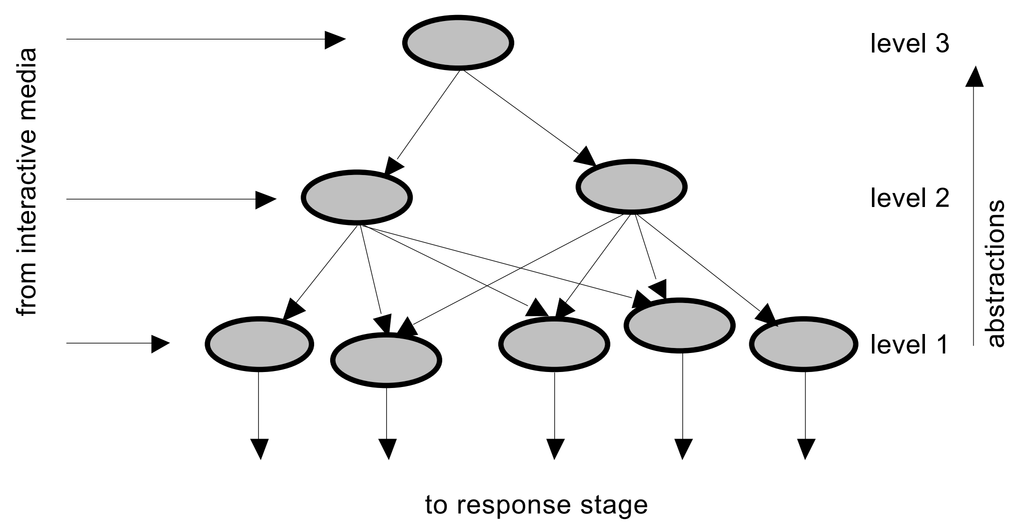


fig. 11

The algorithms with higher levels of abstraction control the behaviour of the lower level algorithms.
A higher level algorithm controls a collection of lower level algorithms, thus creating more coherence between the different lower level algorithms.
The behaviour of the higher level algorithms also changes at a lower pace. This creates more coherence in time

The example in figure 11 uses three levels of abstraction. The amount of levels is very much dependent on the amount of time for which music has to be produced. A larger time asks for more levels of abstraction. In an interactive situation this can almost never be predicted precisely, but some indication must be made. This aspect was already mentioned in the previous chapter on the aspect of Markov chain coherence.

The algorithms in level 1 generate the actual data that is send to the response stage. MIDI-like Information like notes with pitch, loudness and duration; program changes; continuous controllers. One higher level In level 2 the algorithms generation of abstractions like chords, note density, rhythms. In level 3 the generation of general abstractions like scale, amount of layers, consonance of pitches or tightness of timing takes place.
For example a third level algorithm might select a scale to play in and an other algorithm in that level might decide on the amount of consonance of the pitches, then a second level algorithm could generate a chordprogression within that scale also depending on the overall consonance of the music. Another second level algorithm would determine the note density while yet another would generate an ambitus range. A first level process then would finally decide how to play the individual notes depending on chord, note density, ambitus range and possible other parameters coming from second level algorithms.

In this hierarchical structure also connections between different algorithms in different layers can be changed, thus changing the behaviour of the lower level algorithm, or algorithms can be turned on and off or be replaced by another process. For instance a melodic line might first be generated by a Markov chain, while at another moment a predetermined sequence would decide on the succeeding pitches.

Mapping the Input coming from the interactive Media
How can the input coming from the interactive media be mapped to variables in such a structure? This of course depends greatly on the type and amount of data coming in which is different in every situation, but certain general properties can be distinguished.

Some connections between the interactive media and the music system ask for a clear link between an event in the interactive media and a musical event. For instance the emerging of a character in a game can be accompanied by a melodic motif. On the other hand, more global changes in the interactive media ask for more global changes in the music. The gamescore or health of a player are examples of such more global information. To realise this one would probably connect more global and long-term information of the interactive media to the higher level algorithms in the music system. One-on-one links between interactive media events and music events should be handled by the lower level algorithms.

Another general property about the connection is that the more inputdata is coming in, the more the generative music system will constantly adapt and, thus, change. In this situation the system can work quite predictable and still, because of the constant change, vary a lot. On the other hand, a situation with very few input, like for instance a website only giving a signal as the user navigates to a new location (page), could use more unpredictable behaviour in the music system in order to generate enough variation in the musical output to stay pleasing for the user.

Also, there should be some sort of balance between the coherence between the music system and the interactive media on the one side versus the internal variation of the musical system on the other side. Too large changes in the music due only to variations in the music system without a clear cause in the interactive media, will lessen the coherence between the music and the interactive media. A way to prevent this is to make the connection between the interactive media and the higher level algorithms more tight and predictable than with algorithms in lower levels. The lower level processes may use more unpredictable variations and because they are controlled by the more higher level algorithms, they will still be adaptive to the interactive media.

Translating types of controllerdata
The sensing stage, as mentioned in the second chapter, what does it do in a generative music system for interactive media? Apart from the mapping as mentioned above it can also rescale and convert different types of controllerdata to suit more the needs of the processing stage.

There are two types of controllerdata: continuous signals and separate events. The rescaling involves the offsetting and multiplying of incoming continuous signals to adapt them to the valueranges used in the processing stage.
The two types can also in various ways be converted into each other. Continuous signals can be converted into separate events by using tendency masks, as mentioned in chapter three, or thresholds. For example, if the health of a player in a game drops below 10% then the music will change to dissonant sounds.
Separate events can be converted into continuous signals by measuring time intervals between succeeding events and interpolating between these values. Or an algorithm could meter the density of incoming events. For instance an algorithm that calculates the density of cars hitting other cars in a game could influence the amount of disharmonic, metal-like sounds in the music.

Conclusion

Well, with this paper one can get some insight into the possibilities, problems and aspects to think about when using generative music for interactive media. Chapter two clearly states the advantages of interactive music, being adaptivness and variation. The problems, from the perspective used in this paper, seem to be solvable and the two other fields of interactive music systems for performance with musicians and computer composition offer a lot of methods to work with and aspects to think about. What could be investigated further are a bit more precise details about the organisation of algorithms as mentioned in chapter four. Also more investigation on recent implementations of generative methods, both in consumer products and authoring software could be done.

But nevertheless, from the conceptual and methodical approach from this paper it seems possible for generative music to become in the near future an important element of widely used interactive media. Other perspectives not elaborated on in this paper may point in a different direction, though. In my opinion the production process may be the largest obstacle. Making a generative music system is quite different for a composer opposed to making linear audio recordings and also for the process of a multidisciplinar team creating a multimedia product it’s quite a different ballgame. At first this would require a lot of willingness and extra time to get it to a successful result.

There is beauty and elegance in the effectiveness and subtleness with which generative processes use computer resources opposed to the ever more, faster and larger demanding developments like DVD, broadband internet, etc, etc…

Reference

Interactive Music Systems - Robert Rowe (Cambridge, MA: MIT Press, 1992)

The Computer Music Tutorial - Curtis Roads (Cambridge, MA: MIT Press, 1996)

Electric Sound: The Past and Promise of Electronic Music – Joel Chadabe (Upper Saddle River, NJ: Prentice Hall, 1996)

Composing Interactive Music, Techniques and Ideas Using Max - Todd Winkler (Cambridge, MA: MIT Press, 1998)

"Compositional Applications of Stochastic Processes" – Kevin Jones (Computer Music Journal, 5:2, 1981. Republished in Curtis Roads, ed., The Music Machine, Cambridge, MA: MIT Press, 1989)

"Interactive Composing" – Joel Chadabe (Computer Music Journal, 8:1, 1984. Republished in Curtis Roads, ed., The Music Machine, Cambridge, MA: MIT Press, 1989)

"Chaotic Systems as Simple (but complex) Compositional Algorithms" - Rick Bidlack (Computer Music Journal, 16:3, 1992)

"Making Music with Algorithms: A Case Study System" - Kenneth Mc Alpine, Eduardo Miranda and Stuart Hoggar (Computer Music Journal, 23:2, 1999)

"Generation of Musical Sequences with Genetic Techniques" – Anthony R. Burton and Tanya Vladimirova (Computer Music Journal, 23:4, 1999)

"Composing for Interactive Music" – Kurt Harland (Gamasutra, february 17, 2000. http://www.gamasutra.com/features/20000217/harland_01.htm)

"Adaptive Music" – Andrew Clark (Gamasutra, may 15, 2001. http://www.gamasutra.com/resource_guide/20010515/clark_01.htm)

"Interactive Music…er, Audio" – Rob Ross (Gamasutra, may 15, 2001. http://www.gamasutra.com/resource_guide/20010515/ross_01.htm)

"The Difference Between Music for Films and Music for Games" – David Javelosa (Multi-Media, august 1998. http://www.filmmusicmag.com/mmdindex.html)

"SoundScapeSavers" – Edo Paulus (1999. http://www2.hku.nl/~edo/Adem_Wanden/adem_wanden.html soon new website at: http://www.bumpersound.com/sss/)

"Systeem- en regeltechniek" – Aad C.M. van Swieten (Leiden, VSSG, 1979)

"Geschiedenis van de Westerse Muziek" – Donald J. Grout and Claude V. Palisca (Amsterdam, Uitgeverij Contact, 1994. Original english title: "A History of Western Music")

Longman Dictionary of Contemporary English (England: Longman, 1985)

Further literature of interest

This literature is outside the direct scope of this paper but offers interesting perspectives which have made an influence on my motivation for working on generative music systems.

“Gossip is Philosophy” – Kevin Kelly interviewing Brain Eno (Wired 3.05, may 1995. http://www.wired.com/wired/archive/3.05/eno_pr.html)

“Generative Music” – Brain Eno (lecture in San Francisco, June 8 1996. http://www.inmotionmagazine.com/eno1.html)

Silence – John Cage (Wesleyan University Press, 1973)

Articles on David Tudor by John D.S. Adams a.o. (http://www.emf.org/tudor/Articles/)

http://www.filmsound.org/
This link gives a lot of information about aspects of sound and music for film. A lot of these aspects are also very useful for interactive music with interactive media.

Software of interest

Although implementation of the discussed concepts and methods is not a part of this paper, here is still a list of applications in which this implementation, to varying degrees, is possible.

Algorithmic composition:

Csound	http://www.hku.nl/~jorrit/software/csound.html

Interactive performance:

Max/MSP	http://www.cycling74.com
M	http://www.cycling74.com
Jamfactory	
Super Collider	http://www.audiosynth.com
CPS	http://www.bonneville.nl/cps

Internet audio:

Director (shockwaves)	http://www.macromedia.com/software/director/
Flash	http://www.macromedia.com/software/flash/
Beatnik	http://www.headspace.com/
Koan	http://www.sseyo.com
MPEG4 structured audio	http:///www.saol.net

Gameaudio engines:

Microsoft’s Directmusic	http://msdn.microsoft.com/library (search for directmusic)
Staccato’s Synthcore SDK	http://www.staccatosys.com