

The Virtues of the Musifier: A Matter of View

Published at www.interamus.com 2004-09-01.

© [Jonas Edlund](#) and [InterAmus Music Systems](#).

Abstract

There are endless ways in which music can support storytelling, be it opera, theatre or film. Interactive narratives like computer games pose specific problems for the musician due to their non-deterministic nature. Using the terminology of the theatre we conclude that a number of requirements must be met in order to generate highly responsive music with the ability to convey continual changes in multiple independent dimensions: The orchestra need a clear view of the stage. We show that for computerized systems this indicates that frequently updated multi-floating-point views are better than the integer-based views in use today. The InterAmus Musifier is presented as a proof of concept for this approach. The term *musification* is discussed in mathematical terms as a function from \mathbf{R}^n into the space of musical expression.

Musical representation of processes

Musification is the musical representation of data. The term is analogous to *visualization*. Data can be *visualized* by means of graphics and *musified* by means of music. Since music always has a duration in time musification works best for data with a time dimension, i.e. it works best for representing processes. In musical terms: Musification is a musical interpretation of a course of events. In technical terms: Musification is the musical representation of *source processes*. In the following we use the words *source process*, *course of events* or *stage* for that which is being represented. That which *musifies*, *represents* or *interprets* is called *musifier*, *interpreter*, *musicians*, *orchestra*, or simply the *music*.

Deterministic and non-deterministic source processes

Source processes and musifiers come in various kinds. The patterns of a CD and the sound system on which it is played is one example, an improvised dance performance accompanied by a group of improvising musicians is another. In the former example, the source process is deterministic and the musification is in one sense trivial. The music sounds the same each time you play the record. In the latter case the source process is non-deterministic and so is, or at least should be, the musical interpretation. The musification of a deterministic process can be made in advance; the work of the film composer is complete before the opening night, whereas a meaningful real-time interpretation of a non-deterministic process must in some sense be created as it is performed.

Quality of interpretation

All music can be judged on its own merits. Does it sound good? But in addition to that, the quality of an alleged musical interpretation of a non-deterministic process can be judged from three specific criteria. They concern responsiveness, continuity and complexity.

1. Responsiveness

The way the orchestra responds to specific changes in the source process is obviously an artistic matter. But if there is a way in which we can control the source process we can detect the overall responsiveness of the musician, i.e. its ability to musically convey the changes in the source process as they occur. Is the music really an interpretation of the source process?

Responsiveness must not be confused with variation. A musical "interpretation" can be of high musical standard and varied in every possible sense and at the same time be totally unresponsive, totally blind to what goes on in the source process.

2. Continuity

A process can contain continual changes. Fast or slowly, a dancer moves gradually further and further away from the audience and not by teleportation. The sun sets slowly into the sea, it doesn't plunge straight into it all in one go.

Music can also contain continual changes, crescendi/diminuendi, accelerandi/ritardandi, gradual changes in timbre or tempo, in register towards higher/lower notes, in the number of parts etc. To what extent can the musician convey continual changes in the source process in continual changes in the music?

3. Complexity

We define, in this context, the complexity of the source process as the number of independent dimensions in which the process can evolve. In the example with the dance performance, the movements of the dancers, changes in the lighting and in the set can all occur totally independently. The performance is very complex.

Components of music can also vary independently. Dynamics (loud/soft), melodic material (big/small intervals), rhythm, register (high/low), timbre and tempo can all be varied independently even within a single note melody. Add more parts and the complexity increases dramatically. Can the musician convey changes in independent dimensions in the source process in independent dimensions in the music?

Information flow: the view of the source process

In order for the orchestra to be able to musically interpret a non-deterministic process responsively the musicians must somehow receive information about what is happening on stage, they need a view of the source process. The information between stage and orchestra can flow in many ways, a stage manager can stand in the wings and shout in the intercom to the musicians, who sit in another room playing into a set of microphones: "Piece 34!" Or when finished with one piece the orchestra can ask the stage manager:

“Which piece?”, or the musicians can simply see the stage with their own eyes and interpret the course of events independently or according to guidelines agreed upon beforehand with the people responsible for the overall experience of the audience, be it the producer, the director or whatever. If we define our model so that *all musical interpretation is the responsibility of the musifier* we can declare with certainty that regardless of how the information flows from the source process to the musifier, the sole object and purpose of that communication is to maintain the musifier's view of the source process. This is true even if the stage manager is shouting in the wings: “Song 14, variant 6!” The musicians in the next room will be thinking: “All right, we understand the state of the stage is now such that the music will work fine with whatever is going on there if we play song 14, variant 6, here we go...” Their view is made up by the two integers, 14 and 6.

Automation requires a standardized view

The realization of a computerized musifier with the ability to musify an arbitrary non-deterministic process calls for a standardized view. In the following it is argued that the properties of the view and the way it is updated restrict the musifier's ability to interpret the source process. The musifier's responsiveness is never better than its view.

View and interpretation

In computers there are computer words each consisting of a number of bits, 'zeros and ones'. These words are interpreted as values of various types, integers, floating point numbers, characters etc. The view that the computerized musifier has of its source process will at this level consist of a number of values of one or more types. The responsibility of the musifier is to convey those values in music, as the values change and as time goes by. The number of elements in the view and the types of those elements restrict that interpretation.

Ex. 1

View: One integer.

1.1

One solution is to map the integer to pitch and generate a single note melody that directly monitors the changing values. The responsiveness is obviously limited, the orchestra cannot do much with the information, but it's ok, they just play a single note melody anyway.

1.2

A possibly more interesting solution would be for the musifier to react as a jukebox, changing song with each new value. It would then be possible to give the musifier a set of numbered audio clips beforehand. Regardless of the quality of the music in those clips, however, this does not improve the responsiveness. Continuity may become a problem if the order of the clips is not known beforehand and if clips are changed to often we might prefer the single note melody.

A more sophisticated jukebox musifier would generate a transition from the old song to

the new, but if the value of the integer changes frequently this is not an attractive solution. Complexity is limited to one dimension; changes in the source process are monitored in one dimension only.

1.3

Yet another solution would be to let the musifier choose between different continuations of the music at certain points *in the music*. This would be as if the orchestra, when arriving to the end of a section in the music, would ask the stage manager "what's going on on stage?", and depending on the answer they would choose different continuations. This is a more music centred approach and it solves problems with continuity but does not deal with the lack of responsiveness. And all the possible "ways through the music" must be carefully crafted beforehand. Complexity is still limited to one dimension.

Ex.2

View: Two integers.

2.1

One solution is for the first integer to represent song or section and the other a specific variant of that song. Each song could then be crafted in a number of variants and the transition between variants could be easier, if not totally painless, compared to transitions between songs. The variants could be used for changing the music gradually along with the changing "variant variable" in one or more musical dimensions. This would improve responsiveness. But note that the dimensions in which the music changes are not independent as long as we have only one variant variable. The more variants, i.e. the more different values of the variant variable that we have a new variant for, the smoother the transitions we can generate, but obviously, the more work for supplying the musifier with all those variants. In this case also we can consider the possibility of generating transitions. Transitions between songs probably remain a problem if the order of the songs is not known beforehand.

In the more music centred solution in ex. 1.3 the extra integer could denote which variant of the next section that should be played.

Ex.3

View: Multiple integers.

3.1

To expand on the first solution of ex.2: The first integer could be interpreted as "song variable" the rest as "variant variables". This means that we beforehand can supply the musifier with variants in as many dimensions as we have variant variables. If we have n variant variables and each variant variable can assume m different values we could create nm (n times m) variants of each song and then, during generation, maybe with some extra computations, maybe get the musifier to transit between those variants, and thus really be able to handle n independent dimensions in the music and independent smooth transitions in all of them, if m is big enough. Maybe. But with certainty, it would take a lot of fiddling with variants. And then it would be great if we could handle transits between songs in a way that provides for continuity between arbitrary songs.

3.2

Another possible interpretation of a view with many integers would be an interface in which you can trigger multiple audio files in a controlled manner or play multi-track songs with specified tracks muted, including control of levels and so forth.

Conclusions

We can conclude, not surprisingly, that the more information in the view, the greater the possibilities for the musifier. To be specific, we can draw three conclusions.

1. Responsiveness

The musifier's ability to respond to changes in the state of the source process is dependent on the frequency of the updating of the view.

2. Continuity

The ability to interpret continual changes in the source process into gradual changes in the music is dependent on the size of the domains of the variables of the view, i.e. the number of values that the variant variables in the examples above is allowed to assume. The more values, the smoother transitions. The unpleasant consequence, the fiddling with variants, may be the price we have to pay as artists.

3. Complexity

We cannot musically interpret, independently, more dimensions in the source process than there are components in the view. Mathematically it is very simple, a u -dimensional space cannot, without loss of information, be projected into a v -dimensional space, if v is smaller than u . 2D is not 3D.

A floating-point view of arbitrary cardinality?

The examples 1 – 3 are all based on some kind of song-variant-transition paradigm and they all have integer views. Systems available to the computer game industry are all, to our knowledge, based on this paradigm.

This paradigm obviously gets strained as rising demands for responsiveness, continuity and the ability to handle independent dimensions force the number of variables in the view and the size of their domains, as well as the number of precrafted variants of the music, to increase. Are there other views? It seems obvious that for representing gradual changes floating point numbers are more adequate than integers.

With the terminology of mathematics, we can define the musification, the musical interpretation of a view in \mathbf{R}^n , i.e. consisting of n real numbers, as a function from \mathbf{R}^n into the space of musical expression.

Can we construct a computerized musifier that interprets an arbitrary sized floating-point view, updated with the frequency of our choice? Ready to obey our musical instructions without us having to create all those variants beforehand? With high responsiveness, and with the ability to handle continuity, dimensional complexity and transitions between arbitrary songs?

Yes, we can. It exists as a prototype, it's called the [Musifier](#), it can sound like [this](#) and it works according to a principle called [musical morphing](#).