

nogo0.1

live-composition for a computer and a concert

Georg Holzmann

Feb. 2005, Graz

nogo0.1 is a composition for the end of a concert. During the whole concert 50-80 random samples of all the other compositions and pauses are recorded in a regular interval. In nogo0.1 these recorded sounds are now recycled, condensed and removed from their linear temporal context.

1 concept

1.1 simultaneity

In nogo0.1 a lot of the previous compositions of the concert are processed at the same time. They get condensed.

With that I try to dissolve the linearity in narrative structure. Instead of the succession the simultaneity is the most important idea.

Furthermore there exist many possible focus for the listener because of that compression. You can pay your attention to many parts of the dense sound, so it's possible to hear your "own" composition out of it.

Also you may recognize parts from the compositions which were played before.

1.2 repetition

With the simultaneous repetition of already known sounds I also try to avoid the narrative form and question the linear progress or development of time in music.

This principle is now used on two different layers:

- with the repetition of the compositions from the concert the idea of progress in the concert (in the whole music ?) is questioned
- with the repetition and layering of the recorded sounds the progress inside of nogo0.1 is questioned

1.3 state

nogo0.1 consists of 6 independent states.

In every state different parts of the concert with different condensations are playing together. So not-narrating because of too much or too few happenings, because of the simultaneity of temporal separated parts.

2 performance, technical realization

2.1 microphones

The microphones are positioned inbetween the musicians (or the loudspeakers, if there are other electroacoustic compositions) - see figure 1.

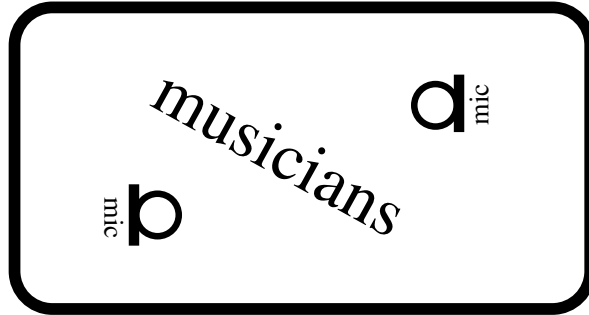


Figure 1: example of the microphone positions

Thereby the recording is focused, e.g. some instruments are louder, more quiet, different timbres, etc.

It is of course also possible to use more than 2 microphones.

2.2 recording of the sounds

During the concert, on 25-50 points in time, samples, each 20 seconds, are recorded in a regular interval (so with 2 microphones you'll get 50-80 samples) ¹.

If there is a break in the concert also the record should be paused (because between the single compositions are enough pauses).

2.3 implementation

The composition is implemented under Linux in *PD* ², a graphical, open-source programming language for audio/video processing in the tradition of MAX/MSP (see figure 2 for the interface).

With the PD-patch the samples of the other compositions are recorded and modified electronically a little bit. During the actual performance the different, recorded parts of the concert are combined with the help of structure-generating algorithms and specific probability distributions.

It is often very complicated to generate more complex musical algorithms in patcher-languages like PD or MAX/MSP. Because of this I developed the external-library *PDContainer* ³ in C++, which allows the use of different datastructures and the communication between them, so that the implementation of algorithmic compositions should be easier.

¹my first realization: 2 samples every 4 minutes, altogether: 2x30 samples

²Pure Data: <http://www.puredata.org>

³PDContainer: <http://grh.mur.at/software/pdcontainer.html>

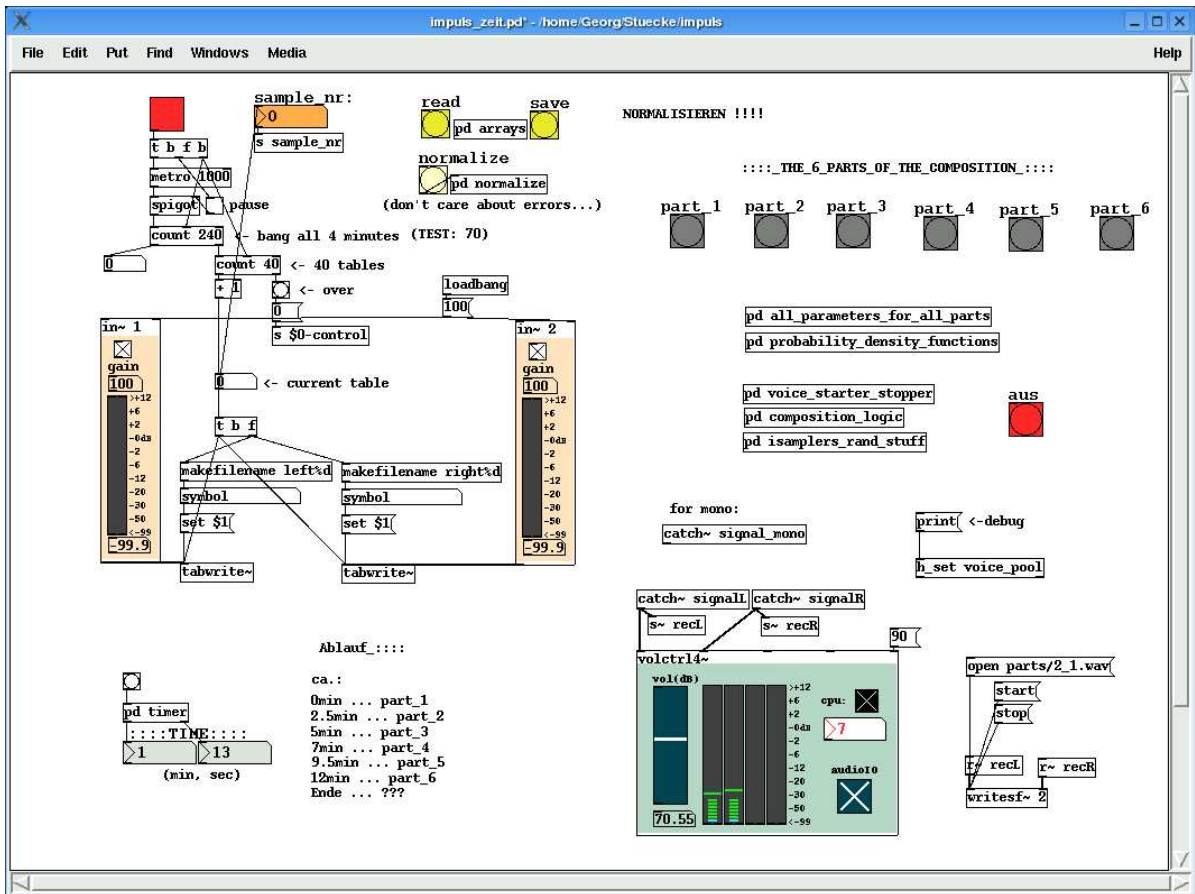


Figure 2: screenshot of the PD-patch

The PD-patch (something like the "score") of nogo0.1 is downloadable for free ⁴ and can be reproduced under Windows, IRIX, Linux or MacOS X (because PD is open-source and multi-platform).

2.4 miscellaneous

duration of the performance: ca. 15 (+/- 2.5) minutes

more infos, a recording and the PD-patch: <http://grh.mur.at/projects/nogo01.html>

⁴PD-patch nogo0.1: <http://grh.mur.at/projects/nogo01.html>