A speaking choir of dead people: the making of *Recordare*

Nicola Bernardini

[1]Conservatorio "O. Respighi"
Latina (Rome) – Italy

bernardini@conslatina.it

Abstract. *This paper is a description of the compositional and realization processes behind* Recordare *– Madrigale recitato per suoni elaborati, a 7' minutes-long acousmatic piece that I have created between 1996 and 2001.* Recordare *features some interesting macro-granular techniques which meet specific compositional and aesthetic demands and requirements. Furthermore, although it is an acousmatic work, its fully automated building process allows constant recreation and adaptation to different spaces and venues. Finally,* Recordare *has been developed and written with longevity in mind: it was conceived entirely with Free Software tools in order to guarantee its long-term sustainability and reproducibility.*

1. Introduction

In 1995, my father (a physicist well known for his commitment for peace and nuclear de-commission) presented me with a striking text (*La maledizione del Generale Mladić*) about the serbo-bosniac war that had been raging for some years at that time. In the idea of his author, the text was suitable to be set to music. However, its characteristics (cf.2) made it particularly difficult (at least for me) to conceive some form of lyricization of it. Far from being a generic text despising war and promoting peace, it is a harsh curse against a specific criminal of war (the general Mladić) pointing out the fact that behind every horror there are single human responsibilities that should not be forgotten. The challenges could be summarized as follows:

1. the text deconstruction adopted in contemporary music idioms – a necessary technique to deal with the fragility of contemporary music structures – could not make sense in this context: fragmenting and hiding the words would jeopardize the meaning of the entire piece; however,
2. adopting the text in its entirety would pose problems at the musical level: how expressive could a mere lyricization of the text be, without jeopardizing the message it wanted to convey?

The recent history of contemporary music has shown that dealing with text is always a complex matter – especially when it conveys some form of political or ideological message (cf. for example, the famous Nono-Stockhausen diatribe about Nono's text treating within *Il Canto Sospeso* [1]). Solutions must be seeked almost exclusively outside the current trends, possibly overthrowing and revolutionizing them. The challenge was extremely interesting, the moment was ripe, and it is in this context that *Recordare* saw the light.

2. The text

The form of the text of *Recordare* (*La maledizione del Generale Mladić*) is loosely based on that of Franz Joseph Haydn's *Die sieben letzten Worte unseres Erlösers am Kreuze* (cf.[2]). While Haydn's 1787 instrumental oratorio is organized in 9 separate movements (one introduction, seven movements based on the seven phrases pronounced by the crucified Christ, and a conclusion), *La maledizione del Generale Mladić* is organized in a soloist/choir responsorial fashion as follows:

I	Introduction	Choir
II	*Woman stance I*	Female solo, Choir
III	*Daughter stance*	Female solo
IV	*Man stance I*	Male solo, Choir
V	*Child stance I*	Child solo, Female Choir
VI	*Choral stance*	Child solo
VII	*Child stance II*	Child solo, Choir
VIII	*Man+Woman stance II*	Male solo, Female solo
IX	Conclusion	Choir

Table 1: *La maledizione del Generale Mladić*, formal distribution

Following its form, the content alternates between vivid memories of exemplary victims of generalized massacres and choral condemnation and cursing of the chief responsible for them. However, in contrast with the outrages described, the tone is generally subdued – after all, these are horrible tales recounted by the victims, and they are all dead. This is one of the major strengths of the text: to be able to point the finger at those who are really responsible for atrocities and genocides without giving way to any vindicative form, even linguistically: an aspect that I sought to emphasize in the musical rendition of it.

3. The musical solution

Early in the pre-compositional stage, I decided some aesthetic tenets of the work:

1. the basic sonic material would be constituted by the spoken words themselves, without any other intervening sound (except for a subtle synthetic grindstone sound, accompaning the main material with indifference)
2. the piece would then take the form of a *recitato* madrigal of spoken words without changing in any way the order in the text
3. the voices should represent the voices of a crowd, even when (or even though) they should appear as solo voices

According to these aesthetic requirements, the ideas regarding the base materials of the piece coalesced into:

1. the text would be recorded integrally several times, with all the vocal gender characteristics required (adult female, adult male, child, young, old, etc.)
2. no dramatization and no expressivity would be sought in recordings (this proved to be in fact the most difficult objective to reach)
3. the recordings should then be spliced and interleaved at dyphone/syllable level and then concatenated together to reconstitute the original text
4. the fragments of dyphone/syllabic materials would constitute a database which would be statistically searched and picked up to generate approximate "statistical gender characters" close to the characteristics expressed in 1
5. the choral parts would be realized out of the loosely synchronized superimposition of several "voices" realized as described in 3
6. the whole piece whould be accompanied by a subliminal and wavy but constant grindstone sound – a sort of "basso continuo" that would serve as expressive bass line accompaniment
7. finally, the result should be spatialized to promote the perception of a choir facing the audience, walking slowly towards it

With such requirements in mind, it was clear that the piece could only exist in an acousmatic form. Trying to outweigh the shortcomings of acousmatic pieces, another further idea which was explored in *Recordare* was to build a creation environment that would be fully sustainable for any foreseable future, allowing to recreate the piece at any time, for any configuration of loudspeakers and any audience room geometry.

4. The realization

Thus, the creation of *Recordare* implied the following processes:

1. create a consistent number of recordings from a variety of voices
2. segment all recordings tracking the beginning and the end of each syllable
3. create a searchable database of all the segments obtained in 2
4. reconstruct the text combining fragments of different voices
5. layering the voices according to some formal requirements
6. creating a *mise en espace* for them

These passages are briefly described in the sections that follow.

4.1. Vocal material selection and recording

The vocal material was picked up from members of my extended family. The members were selected on the basis of voice qualities. 16 different recordings of the full text were made with 10 different native italian speakers, with several multiple takes to catch different voice qualities out of the same speaker. No editing was performed on the recordings, and no recordings were left out from the final result.

In the table below the first column indicates the number of recordings, the second one the voice typologies, and the third one the number of different persons and recording takes that were used to create the work.

N. recs	Typologies	People involved and n. of takes
3	children	2 children, 1+2 takes
4	aged males	3 men, 2 takes (slow/fast) of 1 man
2	aged females	2 women
4	adult females	2 women, 1+3 takes (high/low/whisper)
3	adult males	1 man, 3 takes (high/low/whisper)

Table 2: List of recorded voices

4.2. Fragmentation process

Each recording was fragmented in syllables using a Free Software application called `transcriber` [cf.[3]]. This software allowed quick and easy trascription of audio text into its millisecond-precise segmentation. It produced a set of `xml` formatted files (one for each recording) with all the relevant information concerning the start and end points of each syllable in the text. Since the text present in the recordings was well known, some pre-conditioning of the transcription could be performed by creating an xml file with approximate positions of the syllables generated automatically adapting a general template to the specific readings. The work was thus reduced to set the precise cue points, considerably speeding up the whole process.

4.3. Database creation

Once the segmentation was completed for all recordings, the resulting files were organized in a rudimentary database built out of `awk` (cf. [4]) and `perl` scripts[1](cf. [5]) in which each fragment contained all the information concerning itself, including a) the start and stop position of the fragment, and b) its average rms amplitude envelope. The database was conceived to be completely non-destructive in respect of the original voice files, so that small modifications and adjustments could be performed without having to recreate it anew each time.

4.4. Voice reconstruction

Once the database described in 4.3 was in place the `perl` library was extended to be able to interpret a metadata text file whose task was to reposition every fragment of every voice somewhere along the general timeline according to some general rules. A sample line of this metadata file will clarify its algorithmic role (cf. Fig.1).

[1] these being the only two fairly powerful scripting languages at the time that *Recordare* was conceived.

score.data

```
63 ##
64 ## section 2+3+4 (csound section 2)
65 ##
66 section|63.704|Donna - Coro II (3+1) - Donna
67 voice|00.000|29-66|clcbassa|clcbassa:61.8,silvia:23.6,catbassa:9,clcalta:5.6|2:0|0,2.8:0,2.8|0.201
```

Figure 1: `score.data` **metadata file, lines 63–67**

Fig.1 represent lines 63-67 of the `score.data` metafile. These lines provide an exhaustive sample of the "small language" used to create the sectioning and layering of the vocal parts. What follows is a synthetic description of each line:

- lines beginning with a '`#`' are comments ignored by the interpreter
- lines beginning with a double '`##`' are comments carried out in the resulting `csound` score
- other lines are pipe ('`|`') separated fields which begin with the type of record
- there are two types of record, `section` and `voice`
- the `section` record has three fields:
    1. the `section` tag
    2. the start time of the section, in seconds
    3. a descriptive text of the section (which is out in the `csound` score)
- the `voice` record has eight fields:
    1. the `voice` tag
    2. the start time of the voice, in seconds; this start time is relative to the start time of the current section
    3. the range of words which are going to be reconstructed, in the format `<start word number>-<end word number`
    4. the reference voice (the basic timing of the utterance is built the time of a reference voice which may change for each utterance)
    5. voice appearance probabilities: a comma-separated list of voice tags and their selection probabilities in the following format `<voice tag>:<percentage probability>`
    6. the dynamic gain in `dB` to be applied to the result. This may be a comma separated list of `<dB>:<time>` tuples where the time is in seconds relative to the beginning of the voice; multiple dynamic indications interpolate the gain value between them
    7. a colon separated list of bidimensional positions of the voice; these positions feature x, y positions at the beginning and at the end of the utterance
    8. the average segment duration for this voice

The `score.data` metascore contains nine `section` lines and fifty-one `voice` lines. From them, the interpreter is able to construct the `csound` score of the voice parts from scratch.

A similarly interpreted metafile is used to create the "basso continuo" gravel/grindstone sound and its whole evolution during the piece (cf. 4.5).

4.5. Synthesis and layering

All the work described in sections 4.1, 4.3 and 4.4 were aimed at producing several scores in the `csound` score language (cf. [6]). There were essentially two synthesis systems, one for the voice and one for the continuo. They are both briefly described in the subsections below (cf. 4.5.1 and 4.5.2).

4.5.1. Voice resynthesis

The synthesis system for the voice was a sort-of well-informed *concatenative synthesis* aimed at reconstructing integrally the text out of a statistical picking of each syllable from different voices, following the rules set up in the voice metadata score (cf. 4.4). Essentially, it comprised a soundfile reader with some mechanism to perform a smooth energy crossover with an appropriate windowing mechanism (see below) and to normalize the amplitude among fragments (see below). The only element that has been kept untouched is the phase of each fragment (except for the borders of each fragment, where windowing squashed the waveform) – because sudden phase changes give that expressive touch of broken voice which was sought in this context.

The window selected to perform an equal-power windowing among fragment was some sort of "*quasi-boxcar*" window with constant inverse cosinusoidal half-sides and a variable-length unity gain middle section, to be able to maintain constant gain during transitions while adjusting for the specific duration of each fragment. This window was not available to the `csound` version of the time, so it had to be approximated with `csound`'s `GEN 0 6` table generator (segments of cubic splines) (cf. 2 for an example of such window).

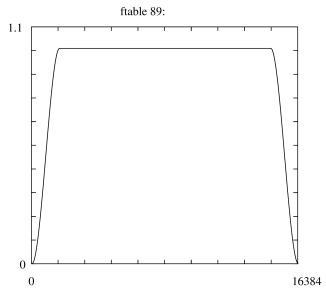

Figure 2: Example of fragment windowing function

Nowadays, `GEN` routine plugins can be designed anew, so such a generator could indeed be designed to obtain inverse cosinusoidal sides. The metadata interpreter software (cf. 4.4) was designed to re-calculate a different window for each fragment in order to adjust to different durations. Given the considerable memory requirements connected to the generation of several hundred tables for each piece section (which was a concern at the time of composition), the tables were created and deleted independently for every section.

Another issue that was tackled in voice resynthesis was that of normalization. Different amplitudes among different fragments needed to be smoothed out in the synthesis phase, but "usual normalization" procedures would not work in this case: the prosodic and expressive elements of a vocal utterance (although minimal) needed to be preserved. The algorithm devised was the following:

1. each section, or each voice of choral section, had to have a "reference voice" which was selected manually keeping in mind the statistical process of fragment-picking (cf. 1)
2. all fragment amplitudes were made to match the amplitudes of the respective fragments in the "reference voice".

This mechanism was quite effective in maintaining the prosodic and expressive elements in the reconstructed utterance.

4.5.2. Continuo synthesis

The other element of the piece could be called the "continuo" part, as it is a countinuously accompaning element which evolves slowly throughout the piece. I had in mind the shivering sound of a grindstone, something that could provoke, at a subliminal level, a perceptual discomfort in the listener. This result was obtained with a granulator which was reduced to obtain slightly-randomized single pulses which was then passed to a bank of butterworth bandpass filters slowly moving towards resonating frequencies throughout the piece. These filters were tuned to frequencies derived from a distributive algorithm encountered in [7, p.193] which was supposed to recreate the partial frequencies of tubular bells.

In the pauses in between sections, the same impulse generation system was passed through a different set of filters replicating the standard formants of female voice vowels whose data was recovered from the italian literature of the time (cf. [8]). The sequence of vowels selected to match the sequence of pauses was: U – E – E – E – E – I – O – A – I, with a progressive movement of formants towards higher frequencies.

4.6. *Mise en espace*

Finally, the materials synthesized as described in sections 4.5.1 and 4.5.2 were all passed, separately, to a spatialization system. The musical idea was to have soloists and choir slowly advance frontally towards the audience throughout the whole piece. In order to provide a vivid sense of movement and tri-dimensionality of the voices, each of them was treated separately with small deviation movements from their central location.

To achieve this result, I have used a "brute force" spatialization system which I had devised in previous years along with my colleague Alvise Vidolin to provide easily configurable robust mechanisms for any sound spatialization need and any room and loudspeaker geometries (cf.[9]). This system simply calculates, given an x, y position of a sound source in a virtual space which is outside an "inner room" marked by the speakers, delays and amplitudes for direct and first-order reflections for every speaker – as indicated by Chowning and Moore (cf. [10], [11]). After the first reflections all higher-order reflections and reverberation tail are synthesized by the `babo` opcode (cf.[12]).

In 2024, I have added – as a separate software project – a `csound` "room instrument designer" software written in the `ruby` language to automate the re-writing of that specific instrument according to the geometry of inner and outer rooms (cf.[13]).

5. Replicability and sustainability

The inevitable acousmatic nature of *Recordare* posed a further problem: I wanted the work not to be fixed once and for all but to adapt to very different spaces and venues (hearphone listening, small rooms, big halls, etc.); not so much to *equalize* the space but to be *enhanced* by each space, along with technological advances of various sort (such as computers having wider and wider words and computational power, etc.). Furthermore, from previous interpretative and compositional work I was well-aware of the huge problem of sustainability and replicability of contemporary electroacoustic works (cf.[14]) – so I was looking into ways of securing the longest possible life to this work. In short, I wanted to be able to resynthesize the whole work at any time in a very simple way – so as to provide a very simple set of instructions to myself and to anybody willing to re-build the piece.

This apparently simple problem hides in fact many complicated aspects. Generally, acousmatic music is considered saved once and for all on a given medium, and a live performance interpretation of the piece reduces itself to an appropriate *mise en espace* of the fixed media. It is often unfeasible or even unthinkable to "rebuild" the work anew. Famous cases of this impossible task are provided by pieces like Karlheinz Stockhausen's *Studie I* and *Studie II*, or Franco Evangelisti's *Incontri di Fasce Sonore*, which are all well described and possess complete scores, instructions, etc. These works cannot be rebuilt anew – they can only be copied from one medium to the other in order to be preserved.

As I have written, I was seeking exactly the opposite: I wanted the piece to be infinitely re-buildable, possibly with "interpretative modifications" and changes – following the same logic of Free Software or Creative Commons artworks (cf. [15], [16]). This was easier said

than done, and I had to resort to several tools which had been available for a long time to software developers (but which are generally unknown in the world of electroacoustic music). A summary outline of these tools includes:

- a SCM (Source Code Management) system – *Recordare* started under *local CVS*, then *networked CVS* (cf. [17]), was then migrated to *Subversion* (cf. [18]), and then was moved to a public *Git* platform (cf. [19]) where it resides to date – many migrations without a single change to the repository tree;
- a set of small programs and libraries all written in long–existing Free Software interpreters such as `awk` (1979) and `perl` (1994) and by specialized Free Software like `csound` (1984) (for sound synthesis) and `pic` (1986) (cf.[20]) or `gnuplot` (1986) (cf.[21]) for drawing/scoring/plotting;
- a collection of nested `Makefile` files which could be read and executed by the program `make` (1979) (cf.[22])

In this way, ideally the piece could *always* be reconstructed by issuing a single `make` command from a console positioned in the root directory of the repository. A change in the geometry configuration would require an addition into the geometry configuration file and then the `make` command to rebuild.

I stressed the birthdate of all these software tools because, unlike the sick computer world narrative "newer is better", in this case precisely the opposite is true ("older is better") because we can easily assess that if the tools mentioned above have been flawlessly working for 35+ years, they are more likely to work for any foreseeable future.

5.1. On the importance of using Free Software

It is also worthwile noting that rigorously *all* the software used to create (and re-create) *Recordare* is licensed as Free Software (cf. [15]). This is not (solely) an ideological choice: it is a choice mandated by the sustainability and continuous re-creation requirements mentioned in Sec.5. The (re-)creation of an artwork cannot be dependent on the whims of some proprietary code software market: several show-stopping problems (such as version incompatibilities, software/hardware incompatibilities, etc.) prove to be fatal to artworks which do not adhere to specific large market targets – and sometimes they are fatal even to those that do. Furthermore, the complete control of the means of production which Free Software allows are paramount to true freedom of expression and creativity.

6. Conclusions

This paper is an account of the making of *Recordare*, an acousmatic piece I composed from 1996 to 2001. *Recordare* posed several conceptual and technical challenges whose solutions were summarized in the preceding sections. The complete details of all the work carried out can be found in the repository of its sources which is publicly available at https://gitlab.com/nicb/Recordare.git.

References

[1] Karlheinz Stockhausen. Musik und sprache. *Darmstädter Beitrage*, I, 1958.

[2] Franz Joseph Haydn. Die sieben letzten Worte unseres Erlösers am Kreuze, 1787. Hob.XX:1.

[3] Claude Barras, Edouard Geoffrois, Mark Lieberman, and Zhibiao Wu. `transcriber`. `https://trans.sourceforge.net/en/presentation.php`, 1997–2002.

[4] Alfred V Aho, Brian W Kernighan, and Peter J Weinberger. `awk` – a pattern scanning and processing language. *Software: Practice and Experience*, 9(4):267–279, 1979.

[5] Larry Wall et al. The `perl` programming language, 1994.

[6] Barry Vercoe et al. `csound`. `https://csound.com/index.html`, 1984–present.

[7] John Robinson Pierce. *The Science of Musical Sound*. Scientific American library. Scientific American Library, 1983.

[8] Pietro Cosi, Franco Ferrero, and Kyriaki Vagges. Rappresentazioni acustiche e uditive delle vocali italiane. In *Atti del 23°Convegno Nazionale AIA*, pages 151–156, Bologna, 1995.

[9] Nicola Bernardini and Alvise Vidolin. Sound motion and space parameters on a stereo cd. *Journal of New Music Research*, 31:171 – 175, 2002.

[10] John Chowning. The simulation of moving sound sources. *Journal of the Audio Engineering Society*, 19(1):2–6, 1971.

[11] F. Richard Moore. Spatialization of sounds over loudspeakers. In *Current directions in computer music research*. MIT Press, 1989.

[12] Davide Rocchesso. The Ball within the Box: a sound-processing metaphor. *Computer Music Journal*, 19(4):47–57, 1995.

[13] Nicola Bernardini. `crb`. `https://gitlab.com/nicb/crb.git`, 2024.

[14] Nicola Bernardini and Alvise Vidolin. Sustainable live electro-acoustic music. In *Proceedings of the Sound and Music Computing Conference – SMC05*, 2005.

[15] Richard M. Stallman. *Free Software, Free Society: Selected Essays of Richard M. Stallman*. GNU Press, Boston, MA, 2002. Available online at `https://www.gnu.org/philosophy/free-software-for-freedom.html`.

[16] Lawrence Lessig. *Free Culture: How Big Media Uses Technology and the Law to Lock Down Culture and Control Creativity*. Penguin Press, New York, NY, 2004. Available online at `http://www.free-culture.cc/`.

[17] The CVS Team. Concurrent versions system, 1990. Accessed: 2025-06-01.

[18] The Apache Software Foundation. Apache subversion, 2000. Accessed: 2025-06-01.

[19] Junio C Hamano and the Git Community. Git, 2005. Accessed: 2025-06-01.

[20] Brian W. Kernighan. The `pic` language, 1986. Available online at `https://www.cs.princeton.edu/~bwk/pic.html`.

[21] Thomas Williams and Colin Kelley. `GNUplot`, 1986. Available online at `http://www.gnuplot.info/`.

[22] Stuart Feldman. `make`: A program for maintaining computer programs, 1979. Available online at `https://www.cs.princeton.edu/~bwk/btl/make.pdf`.