*TeleSon* -- A Composition for Two “reacTables” and Four Players

Computer music composers are very often also instrument-builders. Thus, composing *TeleSon* was an extension of my previous work in making software instruments for collaborative performance. I was attracted to the reacTable by how it made the experience of playing in a networked musical environment accessible to any person, which were similar to experiences I had with expert users in the network band "The Hub". I also liked that the focus of this interaction was on creating and editing synthesis instruments in real-time, with a fluidity modeled on analog instrument paradigms. I wanted to make a piece that arose from the special capabilities of such an interactivity-enhanced digital instrument, rather than imitating acoustic compositional models.

The reacTable's software design was also a natural fit for the SuperCollider3 (SC3) synthesis environment that I have been composing with for many years, since inter-process communication in its software uses Open Sound Control (OSC), which is also used in SC3 for all communication between the synthesis and control threads. I decided to develop my own SC3 synthesis engine, and by using MTG's software reacTable simulator, I could compose the piece in my studio in California while the new physical instrument was being developed in Barcelona.

I made a set of 28 SC3 Synths that would respond to a control object’s data stream describing its position, proximity, angle-to-center, and speed. I used as much of this data as possible to affect each sound, on the principle that the more parameters used in defining each gesture, the more sensitive the instrument would be. In the end, each player had available the same set of 16 sound generators (square objects), 8 filters or sound processors (circle objects), 4 mixer/modulators, (hexagons), 4 low frequency control generators (rounded squares), and 2 sync generators (half-domes). The generators included: 3 wavetable oscillators and a phase-modulation oscillator, whose frequencies were set by angle and rotation, with timbres set by proximity to the center of the table; ostinato percussive generators, including bell, string, noise, and fm-drum types, whose ostinato frequency was derived from center proximity; and sine, noise and feedback granulators, whose grain frequency and density were related to their center and radial positions. I intentionally omitted samplers, because I wanted the character of the music to arise from the configuration and interaction of sound modules, and pre-recorded samples would distract from this. The filters included two band-pass filters, a linear-frequency shifter, a comb delay, a flanger, reverb, waveshaper, and an FFT bin-shifter. Controls were sine, pulse, noise, and stepped-noise generators, with their depth of modulation controlled by proximity to the object they were modulating. Sync-generators are controls that affect any number of objects within their proximity: one was an impulse generator that produced a sharp timbral change at a low frequency set by the angle and rotation of the object. The other was a pattern sequencer, which sent differently changing control data to affected objects matched to each section of the piece. The player

manually selected the section by its center angle position according to the score, which also changed the background color of the video projection on the table as a cue for the ensemble.

Next, I composed patches using all of these synths in different combinations. These had to be small enough to be played within a quarter of the physical space of the table, and were textural, rather than dependent on other inputs for melodic or rhythmic elaboration. Using the reacTable simulator, I saved graphical screen-shots of each patch, gave each a descriptive name, and these became the components of the score. All 30 are distributed among the four players, an only one is ever played by more than one player. The score is a chart that choreographs the entrances and exits of each player playing a particular patch, resulting in a sequence of solos, duos, trios, and finally a quartet at the end. The score has 28 scenes, divided into 8 sections, matching those selected by the pattern sync generator described above.

*TeleSon* is thus defined by the process of constructing and deconstructing sound generating patches, interleaved and coordinated by cues between the four players. Each player had to learn to build, elaborate, and eventually remove them in a musical way. At any point during this process there is opportunity for improvised interactions, both by adjusting the parameters of the patch and, more radically, by intersecting with other player's patches and creating new hybrid ones. There was less of this latter type in the first two performances than I hoped, rather we remained a bit territorial in our behaviour, as our patches struggled for space on the table. More rehearsal and familiarity with all the elements in the score would hopefully lead to greater exploration of this potential in the instrument and piece.

The idea of composing for two remotely located reacTables arose later in the process, with the opportunity to make a concert linking the ICMC and Ars Electronica. But collaborating over distances was already part of the process, since udp data streaming provided the connections between components of the software, and our team coordinated its development using email, voice, and video-chat. It became natural to conceive of a single instrument consisting of two physical interfaces in very remote locations. Since there is always a distance in computer-based instruments between a gesture, its interpretation by the machine, and the sonic response, there is an attractive aesthetic transparency in highlighting that aspect of the instrument and making it part of the music.