(M01) Composing with Data Flow Programming
Laura Meskanen-Kundu, New Media
Student num. 588247
27.11.2016

Report:
Max at Seventeen by M. Puckette

"Max at Seventeen" introduce the development, history, drawbacks, strengths, relation to competitors, concepts or thought patterns, including theories, research methods, postulates, and standards of the Max music software. It offers an audience with an illustration of similar occurrences and not meant to lead into the same conclusion as the development of Max, but shows a personal accountant of its advancement. It gives the philosophical and theoretical framework for Max within which theories, laws, and generalizations and the experiments performed in support of it where formulated.

Firstly, it talks about the pre-designed building blocks that make up the configurations useful for real-time computer music performance and explains it to be a graphical representation and editor for patches. To bring about this graphical interface an opinion is given that computer music software most arises because of interactions between artists and software writers where artists are around to remind of their needs of that software. Next, the background and influences are made known during the time between 1980-1990's when a wide variety of influences acted on it. There was Max Mathews's RTSKED program, Curtis Abbott's earlier 4CED program, Mathews's earlier GROOVE program, Csound and MUSIC N languages all that shaped the Max we have today. Interestingly the idea of a graphical patch language was not new and had appeared by 1987 when starting to write the Max "patching" GUI.

Next the development of Max, Jmax, and Pd is shown as these are the competitors in the market. We begin with the first version of Max, Music500, that had no graphical front end and ran on machines where the number crunching couldn't be realized in a high-level language, but this version never made it to the point of making sound in real time. Next came a version where "objects could have only one inlet and one outlet apiece, connections could be specified simply by naming each object and listing its destination objects, by name, in its creation arguments." Then Max became a MIDI program and was commercialized through the efforts of David Zicarelli. From 1991 onwards a project called Max/FTS originating from older max was extended at IRCAM, who now distribute it under the name Jmax. Meanwhile in 1994, to make improvements to the original Max, a new program named Pure Data, or Pd was created.

The design issues in Max lay in that sometimes for reasons of practicality and sometimes of style, it breaks many of the rules of computer science orthodoxy. The software is more oriented toward processes than data, where a patch is a collection of boxes interconnected by lines and the contents of the boxes themselves are usually hidden from the user. Because each type of box is free to store data in its own way, there is no uniformity across Max in how data is stored. Communication across boxes is by Max messages that are linear lists and can be numbers or strings. The main inlet, always the leftmost one, is used to trigger outputs, while the other inlets are used to set internal

state in the box. Also, if a “previously saved Max patch is reopened, all the values which had been sent to boxes’ inlets are forgotten and replaced anew by values supplied by the creation arguments (or failing that, to their default values)”.

When it comes to scheduling objects are activated by passing messages to them from other objects. For a calculation to work, the leftmost inlet is triggered after all relevant messages have been sent to the other inlets and since the first one is the ‘hot’ one it should be the last to receive messages. Control computations are order-dependent whereas the audio sample clock is predictable.

Programming in Max tries to make the environment resemble what it is, a system for scheduling real-time tasks and managing intercommunication between them. It tries to avoid pushing a stylistic bias on the musician’s output and shouldn’t restrict the musician’s doings by purifying it of cultural biases and built-in assumptions.

While Max and its predecessors are perfect for real-time music production and their development has been long, it is important to note that the graphical programming language scares strong believers in traditional software architecture. The language has in built logic for its objects that are hidden behind the boxes and this makes it hard for a first timer to understanding what exactly is happening when run. As an instrument, when the restrictions, flow and concepts behind making music with Max are understood is when the musician has a way of producing music from this software. It never the less, holds a high learning curve and a pre-requisite in music theory, history of the program, philosophy and understanding of its structure to get started.