A Computer Music System for
Hierarchical Sound Construction
David P. Chapman,
The Media Technology Research Centre, School of Mathematical Sciences
University of Bath, Bath, BA2 7AY, United Kingdom
E-mail: dpc@maths.bath.ac.uk

ABSTRACT: This paper outlines the techniques behind a computer music system in which compositions are
constructed hierarchically from more simple elements. The construction mechanism described views sound
generating objects as occupying a point in a space of parameters governing its properties. The construction
process is therefore one of linking these objects in a parameter-space, and the transformation of such structures
is achieved via motion. The paper further outlines the user interface paradigm used to view this process.

Introduction:
The fundamental problem of digital sound synthesis arises through the complete generality of the digital represen-
tation of sound. The quantity of data required to achieve this generality is large enough to render its manual
specification impractical. As a result, pressure functions are specified using various synthesis techniques, and
languages, requiring a smaller amount of data to drive them. This does by necessity, however, lead to a loss of
generality (Smith). The MusicV program, and its descendant Csound, have used the technique of building an
orchestra of instruments from unit generators, and specifying a score as a list of parameters to be supplied to
these instruments at a given time (Pope). These systems, therefore, fail victims to a problem inherent in all
score representations. Namely that all score representations are, in some sense, too general, or not general
eough (Dacken). If we are to synthesize a composition which consists of parts of varying styles for which
no one score representation is ideal, we can only proceed by providing a technique in which these parts can be
generated using the most appropriate representation, and then linked to form the desired piece. Thus a
composition is production via the hierarchical assembly of smaller components (Punch).

If we now suppose that we have constructed a sound from many levels of smaller parts, we have the prob-
lem of how we are to manipulate this composite. We would like to be able to change certain aspects of it, such
as pitch, duration, amplitude, timbre etc, but without affecting the smaller parts from which it is constructed.
To achieve this we consider that a sound generating object, whose sound depends on a number of parameters,
occupies a point in a parameter-space defined by these parameters. Hierarchical construction can be achieved by
linking the parts, to a single point in the parameter-space, by defining the relative distance in this space, to
this point for each of the parts. Thus when the composite, i.e. the single point, is moved, say in the dimension
of frequency, the smaller parts are also moved due to their links to the composite. We will now consider what
we mean by a parameter-space, and in particular the nature of each dimension.

Dimensions: We define a dimension in terms of sets $\Gamma$ and $\Delta$. Then $\forall \gamma \in \Gamma, \gamma$ is an absolute measure in the dimension and
$\forall \delta \in \Delta, \delta$ is a relative measure in the dimension. Finally we have a function $\Phi: \Gamma \times \Delta \rightarrow \Gamma$. Hence a dimension is defined as a triple $\Omega = (\Gamma, \Delta, \Phi)$. Put simply $\Gamma$ is a set of points. $\Delta$ is a set of routes between the points. The function $\Phi$ is a method for going along a route from one point to another. As an example we might let $\Gamma = \mathbb{R}^n$ and $\Delta = \mathbb{R}$. We have the interpretation that $\gamma \in \Gamma$ is a frequency in Hz and
$\delta \in \Delta \Rightarrow \delta$ is a frequency difference in Hz. We therefore have the very simple function $\Phi: (\gamma, \delta) \rightarrow \gamma + \delta$.
$\gamma, \delta \in \mathbb{R}$, $\forall \delta \in \mathbb{R}$ where $\gamma + \delta > 0$. This gives the frequency reached by the displacement $\delta$ from $\gamma$. Hence $\Omega = (\Gamma, \Delta, \Phi)$ defines a dimension of frequency in Hz.

Parameter Functions: Suppose that we have a tree of nodes as in figure 1. Each node has a number, possibly zero, of sound generating
objects, associated with it. Further, it will have associated with it a time, which is to be considered as the local
time at that node. Each of these generators will have slot values governing their output, taken from the points of
various dimensions. In particular Node0, found in figure 1, has associated with it an object $\lambda$, with a parameter
$p$ from dimension D. In total figure 1 shows a tree of three nodes connected by two arcs. Each arc in this
example has associated with it, two functions. Arc1 has the functions $f_1 : T \rightarrow T$ and $f_2 : D \times T \rightarrow D$. Arc2 has the functions $f_3 : T \rightarrow T$ and $f_4 : D \times T \rightarrow D$. Where $G = (T, \Delta, \Phi)$ is a dimension of time and $\Delta_t = (D, \Delta_t, \Phi_t)$ is a dimension of another parameter e.g. amplitude.

Support that given a tree of nodes of this kind, that all objects associated with it take parameter values from $n \in N$ dimensions. Then we have dimensions $G \equiv (T, \Delta_t, \Phi_t) \forall i = 1 \ldots n$. This is not including $\Delta_t$ which is a dimension of time. Each arc, therefore, will contain functions of the form $f_i : \Gamma_1 \times T \rightarrow \Gamma_2 \forall \Gamma_2$ and a time function of the form $f_i : T \rightarrow T$. When the structure is rendered, to produce sound, the time at the nodes and the parameters of objects associated with them are determined by the function of the arc directly above the node. Before a formal definition of this process is given we will make the following definitions. $u$ is a function of the form $u : N o d e s \rightarrow N o d e s$ such that $u(n)$ gives the node directly above $n$ in the tree. In figure 1, for example $u(N o d e 0) = N o d e 2$, $v$ is a function of the form $v : N o d e s \rightarrow A r c s$ such that $v(n)$ gives the arc directly above $n$. In figure 1 for example $v(N o d e 0) = A r c 2$. Finally $f_i$ denotes the function for dimension $\Delta_t$ of arc $j$. The time at the node at the top of the tree we will call $t_0$. When the sound is rendered $t_0$ increases uniformly. The time at node $n$ is given by the function $e_i : N o d e s \rightarrow T$ thus:

$$e_i(n) = \begin{cases} t_0 & \text{if } n \text{ is the root} \\ f_i(e_i(u(n))) & \text{otherwise} \end{cases}$$

Hence in figure 1 we have $e_i(N o d e 2) = t_1 = f_1(t_0)$ and $e_i(N o d e 0) = t_2 = f_2(t_2) = f_2(f_1(t_0))$. Now if $n$ is a node let $n_0$ denote the time in that room. Suppose that an object, at a node $n$, has a parameter $p \in \Gamma_i$ for some $i$. The value during rendering is given by the function $e_i : \Gamma_i \times N o d e s \rightarrow \Gamma_i$ thus:

$$e_i(p, n) = \begin{cases} p & \text{if } n \text{ is the root} \\ e_i(f_i(p, (u(n)))) & \text{otherwise} \end{cases}$$

Hence in figure 1 the value of parameter $p$ of object $A$ is $e_i(p, N o d e 3) = f_1(f_1(p, t_1), t_2)$. Therefore if the object $A$ has a pressure function $g : \Gamma_4 \times T \rightarrow \text{pressure value}$, the result during rendering will be $g(e_i(p, Room 3), t_3)$. The pressure value of the composite at time $t_0$ is thus given by the summed pressure values of all the objects in the tree.

User Interface:
The user interface to the system represents the nodes in the hierarchy as rooms, two-dimensional surfaces, upon which sound generators associated with the node tools such as score editors, etc can be found. These rooms are viewed through scrollable windows, across which the icons representing objects may be moved, and applied to each other. Arcs are represented as doors between rooms, to which the functions produced by score editors can be applied. The user can view, and experiment with, any part of this hierarchy at any time, and render the sound generated by a particular part of the structure.

References: