The jChing: a Java-Based Algorithmic Composition Tool

Robert Hamilton
rob@roberthamilton.org

ABSTRACT

The chance-based compositional techniques utilized by composer John Cage in such works as "String Quartet in Four Parts" and "Music of Changes" made use of a compositional framework of gamuts and gamut squares that serves as the object-model for a compositional software application capable of transforming musical data cells using both chance-based and probability-driven functions. Written in Java, the jChing makes use of the MusicXML data format to output transformed musical data in a format compatible with a number of commonly used musical notation applications. This article outlines the functional model and technical specifications for the application and provides basic examples of the jChing workflow.

1. INTRODUCTION

While many composers find the application of algorithmic processes and models to more traditional forms of musical composition to be an effective and evocative method of work, the inherent calculation and mapping of computational data to musical form can be an extremely time-consuming and unnecessarily complex task. And while the use of computer systems has made the application of otherwise prohibitively complex algorithmic processes to musical data possible, most computer-based compositional softwares have had issues of usability, access, or sustainability that should be addressed. Software-based compositional systems such as Koenig's *Project One* and *Project Two* [6] and Xenakis' *Stochastic Music Program (SMP)* [11] were computationally powerful yet left composers the undesirable task of transcribing massive amounts of data into comparatively simple standard musical notations. Issues of software and hardware obsolescence have left software by composers such as James Tenney [7] difficult if not impossible for composers to access. More recently, conversations with French composer Luc Ferrari revealed that his 1998-9 work *Jeu Du Hasard Et De La Détermination* for piano, percussion and recorded sounds [5] was realized with the use of similar software specifically crafted by the GRM (Groupe de Recherches Musicales) but not readily available to the public. For compositional software tools to be truly useful to a larger community of composers, there exists a need to automate both compositional and score-rendering processes, creating a workflow where complex transformations effected upon musical materials can be quickly and easily replicated and transcribed.

The jChing project was intended to provide a compositional tool capable of processing and formatting user-generated musical data using a data-structure compatible with existing musical notation softwares and creating a streamlined workflow without generating a large attentional "footprint" during the compositional process. The tool was designed for use by composers as a non-intrusive step in the compositional process able to produce dynamic reinterpretations of composed materials. Using a standardized data format, the results generated by the jChing could be viewed immediately in standard musical notation with the use of common music notation software.

By establishing a rapid workflow timeline encompassing the transformation of musical data from input through its realization as a standard musical score, such an approach makes possible additional compositional uses of the transformed results. A unique instance of an algorithmically generated composition could itself be generated before any given performance or in real-time during a performance. The MusicXML [9] data format provided a robust and widely accepted data output solution whereby the functionalities of existing and prevalent music notation softwares such as Finale and Sibelius could be leveraged for the rapid presentation and further editing of output scores.

Using a class framework written in Java, it was relatively easy to create a basic object model of Cage's Gamut structures and simple coin-flipping selection processes. While additional functionality added to the project would eventually distance the software from Cage's wholly chance-based techniques, a model based on his concept of cell-sets or gamuts of musical data proved a stable initial goal for the project..

Once the basic system architecture was in place, it became apparent that the jChing software could be augmented to make use of a number of stochastic and probabilistic algorithms, further extending the range of compositional transformations available to a composer/user. The introduction of a basic system of weighting allowed for the possibility that algorithmically generated probability curves could be applied to aspects of the piece, including overall note density, note duration and note dynamics.

2. CHART/GAMUT-BASED COMPOSITION

With his 1949-50 *String Quartet in Four Parts* [2] as well as 1951's *Music of Changes* [3], John Cage developed and made use of a system of charts and gamut cell-structures which enabled the chance-based selection of musical data cells with the use of coin-flips. By representing note data, dynamic data, and duration data in separate charts, and through random selection of data from each chart, Cage sought to escape the inherent determinism in traditional compositional practices and to create an overreaching compositional method which could break free from not only traditional harmonic and rhythmic constraints but from also his own compositional biases and preferences [8].

Cage's system was simple in design yet tedious in implementation. While the methods the composer employed to select pitch and performance data were

chance-based, the resultant composition remained fixed; the static notation of the piece removed any further element of chance from future performances. To create a chance-based compositional form from pre-defined musical phrases, Cage used a set of matrices of musical data cells mapped to cells of the I-Ching. These matrices can be referred to as 'Gamuts' and the individual cells of each Gamut as 'Gamut Squares'. Separate Gamuts are used to hold note-phrase data (groupings of musical pitches and durations), dynamic level markings, and performance technique articulations.

2.1. Using the jChing – Basic Workflow

Short musical phrases of pitch values and duration values are composed and entered into the jChing as individual Gamut Squares. The entire set of Gamut Squares makes up the larger unit of the Gamut. Composers wishing to apply probabilistic processes to their materials can enable weighting values for Gamut Squares. Standard dynamic markings such as ***pp*** *(pianissimo)* and ***f*** *(forte)* are already defined in the system and can be given a weighting percentage to set the probability that those dynamics will be attached to any given note. Other system settings allow the composer to randomly or algorithmically assign various pitch transpositions and rhythmic expansions or diminutions to the cells. Once all desired parameters have been set, running the jChing program will generate a data file in the MusicXML format which can then be viewed and manipulated further using compatible notation software.

3. OBJECT DATA STRUCTURES

The highest hierarchically ordered object within the jChing object model is the Piece object. The Piece object acts primarily as a container for both the Gamut object, the principle data-storage container for input data, and the Staff object, the principle data-storage container for output data. Piece objects provide logical data storage for Piece-level data elements such as "Composer Name" and "Piece Title". Both the Gamut object and the Staff object store Gamut Squares with the Gamut object reading Gamut squares note-by-note from the input data file and the Staff object ordering, transforming, and outputting Gamut Squares into a MusicXML formatted output file.

3.1. Data Input Structures

Contained within the Piece object, the Gamut and Gamut Square objects serve as key grouping data structures for the jChing data input process. Each Piece has one Gamut object, which in turn is made up of any number of Gamut Squares - cells of Note objects upon which transformations can be later performed.

As data is read from the input data file, individual Gamut Square objects are created for each successive Gamut Square. Within each Gamut Square, individual Note objects are created containing pitch and duration data for each note found in the input file. By the time the data input file has been fully processed, the Gamut object is populated with a set of Gamut Square objects, each respectively populated with a series of one or more Note objects. A musical rest is considered a Note in this context and is stored as such.

3.2. Data Output Structures

During the algorithmic selection of Gamut Squares and their subsequent transformation, Staff objects are used both as containers for processed Gamut Square objects as well as representative staves within a multi-voice musical piece. When calculating the sequence of Gamut Squares for output, Gamut Squares are selected from the Gamut object based on the desired chance-based or probabilistic selection criteria and placed into a new order within the Staff object. In a multi-staved musical piece, all Gamut Squares for one Staff object are selected sequentially. Only after all squares for one staff have been selected will squares for the next staff be selected.

4. GAMUT SQUARE TRANSFORMATIONS

The jChing is designed to allow users to effect transformations, either chance-based or probabilistic in nature, upon three core attributes of note sets within a piece: the ordering of Gamut Squares found in the piece, the relative amplitudes or dynamics for each note of the piece, and the absolute temporal durations of each note in the piece.

4.1. Gamut Square Ordering

The most basic form of transformation of musical cells in a Gamut-based system is the relative order in time that each cell will be performed - essentially a virtual shuffling of Gamut Squares. By selecting cells at random a timeline of cells can be created to form a chance-based compositional structure. By applying simple weighting values to each data cell, specifying the relative likelihood of each cell's selection within the Gamut, a simple stochastic system can be created. Generating the weights of relative cells based on probabilistic functions such as Gaussian distributions can impose a more ordered stochastic form upon the cells, giving the composer even more control over what many would still consider a relatively "random" compositional form. Similarly both independent probabilities and conditional probabilities [4] can be used during selections of gamut squares to reinforce weighted compositional tendencies.

Selected squares are placed in their new order into a Staff object. If the piece being generated contains multiple voices or parts, multiple Staff objects are populated with Gamut Squares until a pre-set limit of either Gamut Squares per Staff or a total length of beats is reached, at which time the next Staff object is populated.

4.2. Dynamic Weightings

Just as a Gamut structure can be used to facilitate algorithmic selections of note cells, so too can a simple table structure be used to apply occurrence-weightings to different musical dynamics. Composers can set

weighting values for each desired level of dynamic from ***pppp*** to ***ffff***. A table of values in the input data file matches dynamic markings to percentages (from a combined total of 100 for the entire piece). For instance, if the ***p*** dynamic is to appear more frequently than the ***f*** dynamic, the composer can set the ***p*** to have a weight of 20, while the ***f*** could be set to 1; therefore the probability of selecting ***p*** is twenty times greater than ***f*** and will most likely appear significantly more frequently in the piece

On a note-by-note basis, dynamics can be selected from this table and applied to the Note object currently being processed by jChing. Functionally, the table with added indices, more accurately realized as a multi-dimensional array, acts in a manner similar to a Gamut.

4.3. Gamut Square/Note Scalings

Composers wishing to effect duration changes to Gamut Squares or Notes can enable an option whereby a percentage value entered in the input file will determine the probability that any given Gamut Square or Note object will have its durational value multiplied by a multiplier value. Multipliers are entered as a range of numbers and a divisor; as cells are selected for scaling, a value derived from the range (in increments of the divisor) will be used to multiply the cell's duration. For logistical reasons, the cell can be either a Gamut Square (whereby all durations of notes contained within will be scaled) or an individual Note.

5. INPUT DATA FORMATTING

In the early stages of development, it became clear that input data needed to be organized in a manner that was both comprehensible to the human user as well as formatted in a robust and logical way to facilitate easy data parsing and processing. After considering a number of possible data formatting solutions, a modified version of the SCORE (L.Smith, 1987) music data format was chosen for its simplicity and comprehensive coverage of musical expressions and data types. The SCORE 4.0 Music Data Entry Reference Manual [10] stands as the standard formatting model for all musical expressions of data input. At this time support for additional input formats including MusicXML input is under development.

5.1. Header Data Declarations

Data representing elements such as the Piece-Name and Composer-Name, as well as structural characteristics such as the number of staff-systems, respective system clefs, and the overall size of the excerpt are entered into the Header Declaration of the .gam input file. Data from the header declaration will be used to populate Piece-level data in the jCching object model. From the following header declaration, we can see that each data-value is prefaced by a data-tag in capital letters. Multi-part data such as the four values for SYSTEMCLEFS are separated using "/" marks.

```
; This is a comment line
PIECENAME Dianae Sumus In Fide
COMPOSER Robert Hamilton
POET Catullus
RIGHTS Copyright 2003, CDS. Publishing
SOFTWARE jChing
ENCODINGDATE February 14, 2004
SYSTEMS 4
SYSTEMCLEFS T/T/A/B/
PARTS Soprano/Alto/Tenor/Bass/
SIZE 18
```

Figure 1. Piece-level header data

5.2. Dynamic Weighting Declaration

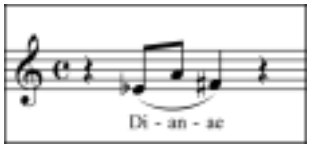
Following the header data the user can define relative weightings for the range of dynamic values to be used in the piece. In the following partial definition of dynamic levels, the "NULL" weighting is used to define a cell with no specific dynamic marking.

```
DYNAMICS
NULL    45
PPPP     0
PPP      0
PP      10
…
```

Figure 2. Dynamics weighting table

5.3. Gamut Square Declarations

Figure 3 depicts a composer-defined musical cell in both standard musical notation as well as in the jChing input data format. Pitches are declared as a note name and accidental followed by an octave number.




```
START
NUMBER 1
METER /3 4/
CLEF TR
KEYSIGNATURE K1S
DURATIONS Q/H/Q/H/
NOTES Eb4/R/CN2/G#3/
WEIGHTING 1
END
```

Figure 3. Individual Data Cell

The .gam data file lists each Gamut Square cell used in a piece, complete with individual numbers, meters, clefs, key signatures, note pitches, and note durations. Each cell is prefaced by a START data-tag and ended with an END data-tag. For Gamut Square 1, we can see that there is a Treble-clef in a key with no flats or sharps and four notes with respective durations of one quarter-note, one-eighth note, one-eighth note, one quarter-note and one quarter-note. Both Note durations and Note pitch values are entered in "/"-delineated lists, with the order of values in each list corresponding to the order of notes found in the particular Gamut Square.

The METER data field shows that the cell can be interpreted to have a time signature of 4/4. Additionally, the cell is given a WEIGHTING value of "1", meaning it is given no additional weight in probabilistic computations.

6. DATA OUTPUT FORMATTING

Scores output by jChing make use of the MusicXML data format as defined in the MusicXML 1.0 Tutorial [9]. MusicXML score data can be grouped either in a "partwise" manner, where each score part or instrument is hierarchically superior to the measures contained within, or in a "timewise" manner, where each measure is considered one by one, grouping each individual part as belonging to the particular measure. Scores created using jChing are currently formatted using the partwise definition.

While MusicXML addresses the current needs of the jChing software, the lack of one definitive industry standard in music notation data formats makes it possible that future versions of the software will use a different output data format. The separation of composition and output processes in the current software makes it relatively easy to add support for additional data varieties such as GuidoXML or SDML for use with other notation softwares and approaches such as ScoreSVG or Lilypond. Current projects such as Kevin Baird's *No Clergy* real-time interactive system are even making use of output formats such as PNG (Portable Network Graphics) in conjunction with MusicXML to address similar concerns [1].

6.1. Staff-based output ordering

Just as the MusicXML partwise data-definition sets up a macro-structure whereby individual measures are grouped together within parts, jChing orders data for output by selecting and grouping individual Gamut Square objects within Staff objects. Each Staff object acts as a container for all Gamut Squares that will make up a specific voice or part within the score; all notes from these Gamut Squares will be placed on the same staff on the written score. When transformations upon Gamut Squares are performed, the Staff objects monitor overall length and pitch ranges for the entire part and if need be can apply constraints to prevent the staff from growing too long or the range of the individual notes from exceeding the range possible for the staff's chosen instrument.

After outputting header information, the jChing processing steps through individual Staff objects and renders each Staff object as a separate part. Parts consist of a number of measures, each defined individually with both a set of measure-wide attributes and a set of note values.

6.2. Measure Calculation

One of the most important aspects of properly presenting scaled Gamut Squares on a written staff is the calculation of individual measure durations within each staff. When Gamut Squares are ordered in partwise fashion, each staff is ordered without consideration of events happening on other staves at the same time. Since the modification of time signatures for individual measures can have subtle differences in meaning for performers, the clearest solution of this issue is to effect a rule-based system of note splitting and tying. This system will be able to fit groupings of notes into a locked measure time signature and create subdivisions of tied notes (e.g. two-tied quarter notes instead of one half-note) to span measure lines. Such a system is currently under development.

7. CONCLUSIONS

While a significant amount of work has been carried out to-date in the field of algorithmic composition software, there still exists the need for composer-oriented easy-to-use software solutions capable of processing and realizing algorithmically driven compositional forms. In this light, the primary goals of the jChing project are to incorporate chance-based and probabilistic compositional processes into a streamlined score-generating compositional tool. Additional development for the project is ongoing to incorporate more features and to streamline existing processes. The scope of enhancements will include not only additional stochastic and deterministic algorithms but also greater support for data input and output with MusicXML as well as other music data-formats, a graphical-user interface for more user-friendly operation, and compatibility with the Max/MSP 4.5 implementation of Java for real-time use.

8. REFERENCES

[1] Baird, K. "No Clergy:Real-Time Generation and Modification of Music Notation". *Proceedings of the 2005 SPARK Festival*, Minneapolis, USA, 2005.

[2] Cage, J. *String Quartet in Four Parts*. (Score). New York: Henmar Press, C.F. Peters, 1960.

[3] Cage, J. *Music of Changes*. (Score) New York: Henmar Press, C.F. Peters, 1960.

[4] Dodge, C. and Jerse, T. *Computer Music: synthesis, composition and performance*. New York: Schirmer Books, 1997.

[5] Ferrari, L. *Jeu Du Hasard Et De La Détermination.* France: Harmonia Mundi:, 1999.

[6] Koenig, G. *Aesthetische Praxis/Texte zur Music*, Band 3, 1968-1991, PFAU Verlag, 1993; online: http://www.earlabs.org/text/historic/earkoenig.html.

[7] Polansky, L. Liner Notes, *James Tenney: Selected Works, 1961-1969*, New World Records, 2003.

[8] Pritchett, J. *The Music of John Cage*. Cambridge University Press, 1993 Pritchett, J. *The Music of John Cage*. Cambridge University Press, 1993.

[9] Recordare, "MusicXML 1.0 Tutorial", Recordare LLC, www.recordare.com, 2004.

[10] Sapp, Craig S. "SCORE 4.0 Music Data Entry Reference Manual", ccrma.stanford.edu/~craig/score/input, 2003.

[11] Xenakis, I. *Formalized Music: Thought and Mathematics in Composition.* (Revised Edition). Stuyvesant, NY: Pendragon Press, 1992.