**Software Engineering**
2014; 2(2): 15-21
Published online January 08, 2015 (http://www.sciencepublishinggroup.com/j/se)
doi: 10.11648/j.se.20140202.11

# Music/multimedia technology: Modeling and simulation of the hybridized interactive algorithmic composition model

**E. J. Garba*, G. M. Wajiga**

Department of Computer Science, Federal University of Technology (ModibboAdama University of Technology), P.M.B. 2076 Yola, Adamawa State, Nigeria

**Email address:**
e.j.garba@mautech.edu.ng (E. J. Garba)

**To cite this article:**
E. J. Garba, G. M. Wajiga. Music/Multimedia Technology: Modeling and Simulation of the Hybridized Interactive Algorithmic Composition Model. *Software Engineering.* Vol. 2, No. 2, 2014, pp. 15-21. doi: 10.11648/j.se.20140202.11

**Abstract:** The hybridization of models of algorithmic composition models gave birth to Hybridized Interactive Algorithmic Composition (H.I.A.C.) model. This hybridized model was derived from the existing algorithmic composition models – where their strengths were harnessed and their weaknesses transparently subdued. The hybridized model was then analyzed using Unified Modeling Language (UML) as an Object-Oriented Analysis and Design tool. The analysis gave rise to the development of an Object-Oriented software framework for the hybridized model, presented as Java objects with generated JavaDoc hypertext documentation for further developmental references. This research has succeeded in providing music/multimedia programmers with necessary building blocks for implementing and realizing applications which require musical features.This article presents some of the key aspects of the simulation and modeling of the Hybridized Interactive Algorithmic Composition model.

**Keywords:** Hybridized Interactive Algorithmic Composition Model, Algorithmic Composition, Simulation, Modeling

## 1. Introduction

Music is the skillful arrangement and alternation of sound and silence within a given period of time with appreciable aural perception. When musicians play instruments, sound is produced (as the sound-producing parts of the instruments vibrate/oscillate). These vibrations cause air molecules to displace one another in a systematic continuous flow, which moves away from its source. In the end, our ears perceive these air pressure fluctuations as sounds that are translated by our brains as music [1].

## 2. Computability of Music

According to [1], music is the alternation of sound and silence. It is therefore obvious that the concept of music conforms to binary theory; where only the values of 0 and 1 are considered. The binary value of "0" indicates certainly that an event will not occur; while "1" indicates certainly the event will occur.

Logically put, when sound is produced (or a note/tone is played) the binary value is 1, otherwise it is 0. That is:

- 1 ➔ sound is produced (or a note/tone is played)
- 0➔ silence (no note is played)

Therefore, the binary musical concept perceives music as a stream of 1s and 0s. This therefore, means that musical ideas and concepts are logical – hence computable.

## 3. Algorithmic Composition

Algorithmic composition is the technique of using algorithms to create music. Music composition is usually a complex and difficult task for the people not having musical knowledge or skill, and composer's expertise plays an important role. Nevertheless, there have been many studies on automatic music composition using computer since the conception of the computer, and some automatic music composition models have been proposed [2].

The formal technique of generating music was adopted again in the 20th century. The *tone-row technique* was introduced by Arnold Schönberg at the beginning of the century and further developed into *serialism* by Anton Webern and his successors [3]. In the serial technique music was factorized into parameters such as pitch, duration, and timbre that were controlled separately. A permutation was chosen from the possible values of each parameter and arranged into a row. The parameter values changed according to the row or its inversions or retrogrades [4].

According to [3], mentions some common techniques of algorithmic composition; which include state machines, rule-based, grammars, stochastic processes, and genetic algorithms. Therefore, Systems for the algorithmic composition of music can be conveniently categorized into three types; rule-based systems, systems which learn by example – knowledge-based and genetic algorithms [5] and [6].

# 4. An Overview of Hybridized Interactive Algorithmic Composition Model

A hybridized interactive algorithmic composition (H.I.A.C.) model was developed from the existing algorithmic composition models in order to minimize the weaknesses experienced when such models are used singlehandedly in music composition[7]. The hybridization of the models (at different stages of composition) capitalized on the advantages of such models. The models for algorithmic composition used included: Mathematical[8], Grammar (Rule-Based) [9], stochastic[10], Knowledge-based [3], and Evolutionary (Genetic Algorithm) [11].

The focal point of this H.I.A.C. model is based on Interactive Algorithmic Composition approach where human creativity in music improvisation and composition is complemented by the leverage of speed and accuracy of the computer.

The development of the H.I.A.C. model is necessitated upon the fact that the existing algorithmic composition models do not singlehandedly take care of all the stages of music composition. In essence, the hybridized model harnesses the strength of existing models in order to overcome their weaknesses (when used individually).

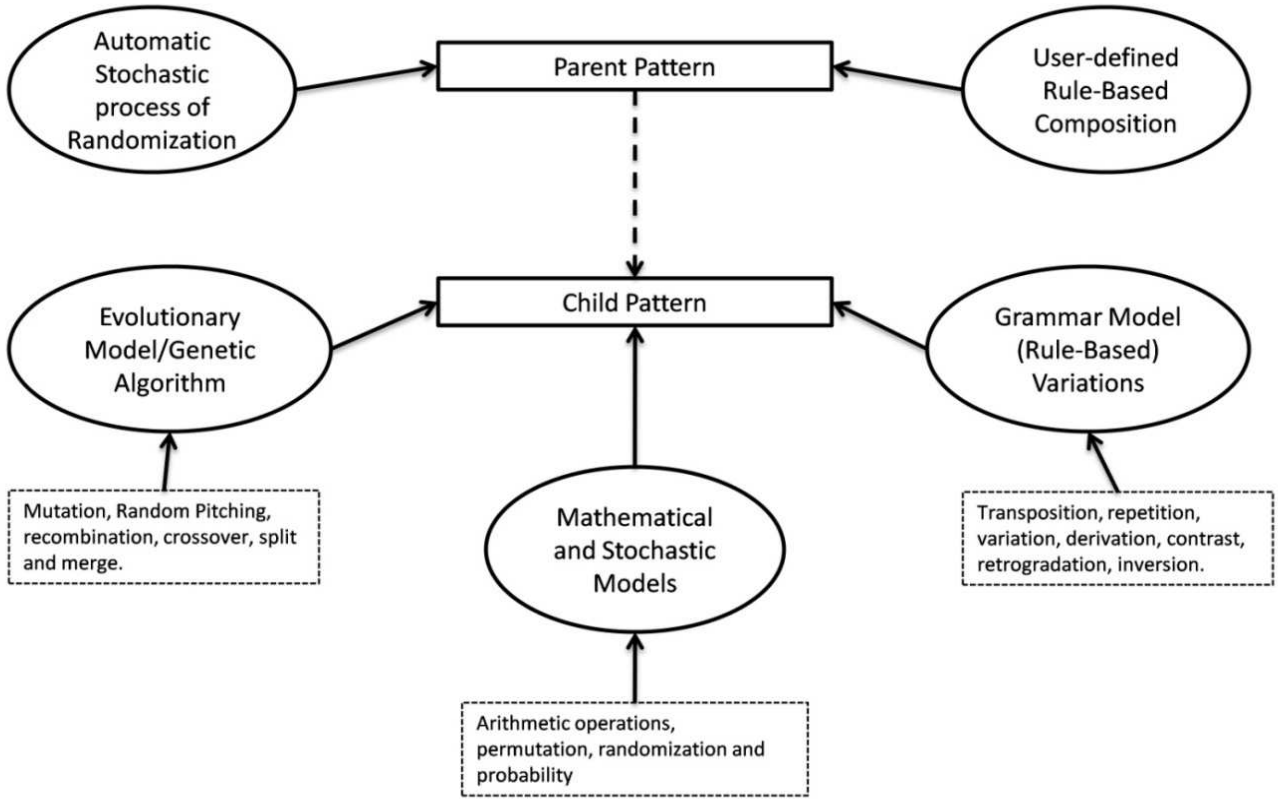


***Figure 1.*** *Overview of the Hybridized Interactive Algorithmic Composition Model[7]*

### *4.1. Rhythm and Melody Creation Processes of the H.I.A.C. Model*

In view of the fact that the existing algorithmic composition models do not clearly distinguish between the process of creation of both rhythm and melody (which makes music composition non-definitive as regards rhythm or melody creation); in the H.I.A.C. model, however, both rhythm and melody creation, generation and synthesis are made explicit. This way, music composition is made both interactive and flexible – in the sense that a generated rhythm (for instance) could be applied to several melodic patterns to produce rhythm-based variations; on the other hand, a melodic line could be applied to a number of rhythmic patterns to produce melody-based variations [12],[13]. Figures 2 and 3 describe both rhythm and melody creation processes of the H.I.A.C. model.

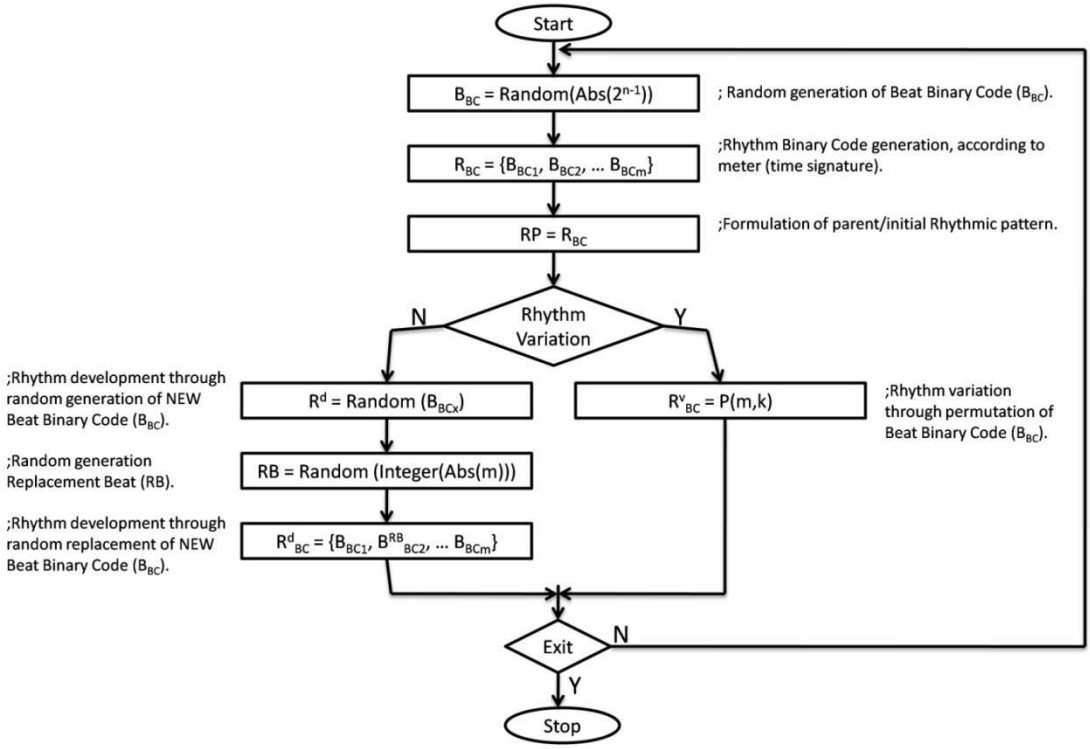


***Figure 2.*** *Rhythm Creation Process[7]*

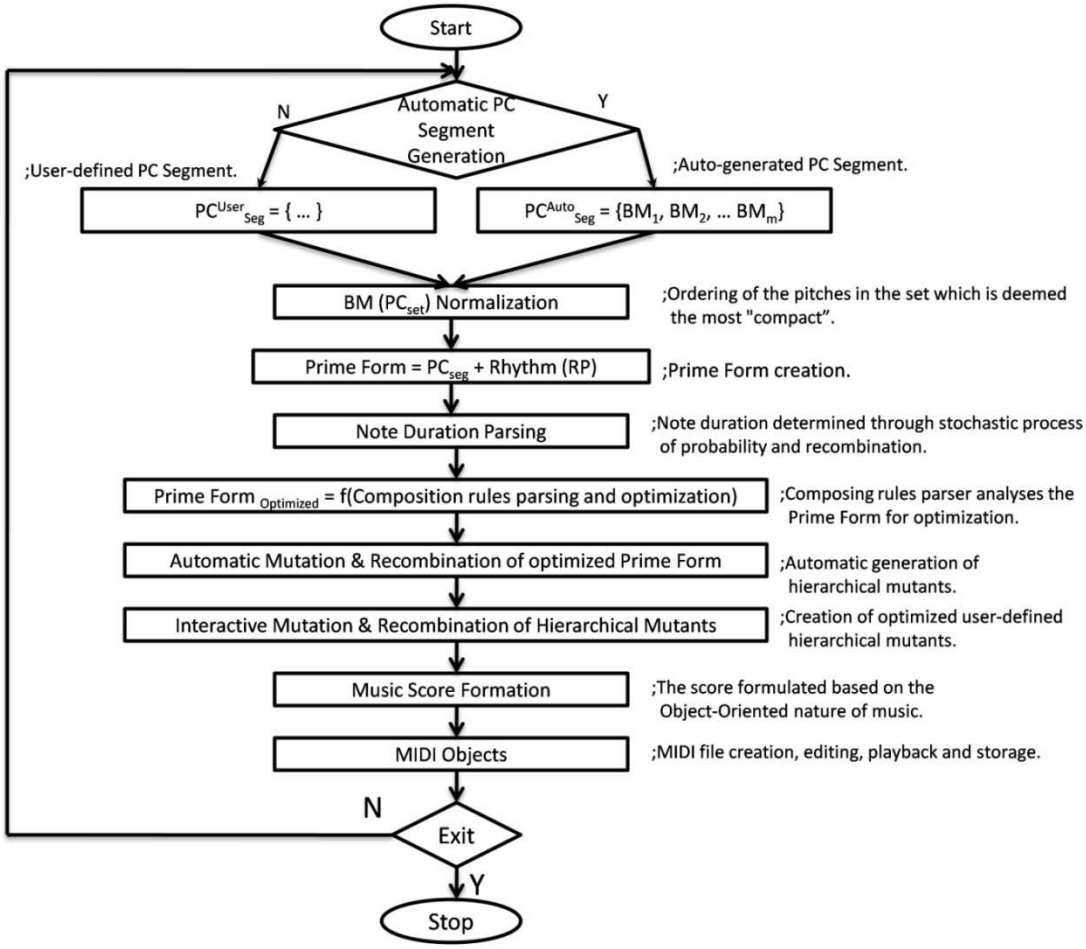


***Figure 3.*** *Melody Creation Process[7]*

## 5. Object Oriented Modeling

Unified Modeling Language (UML) is a visual language for capturing software designs and patterns. Dig a little deeper, though, and you'll find that UML can be applied to quite a few different areas and can capture and communicate everything from company organization to business processes to distributed enterprise software. It is intended to be a common way of capturing and expressing relationships, behaviors, and high-level ideas in a notation that's easy to learn and efficient to write. UML is visual; just about everything in it has a graphical representation [14].

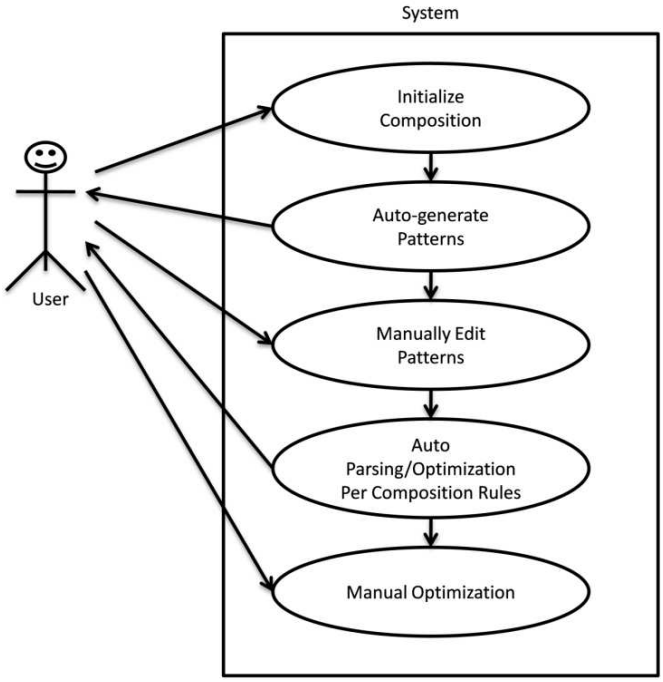


***Figure 4.*** *Use Case Model for Human-Machine Interactivity of the Hybridized Interactive Algorithmic Composition Model*

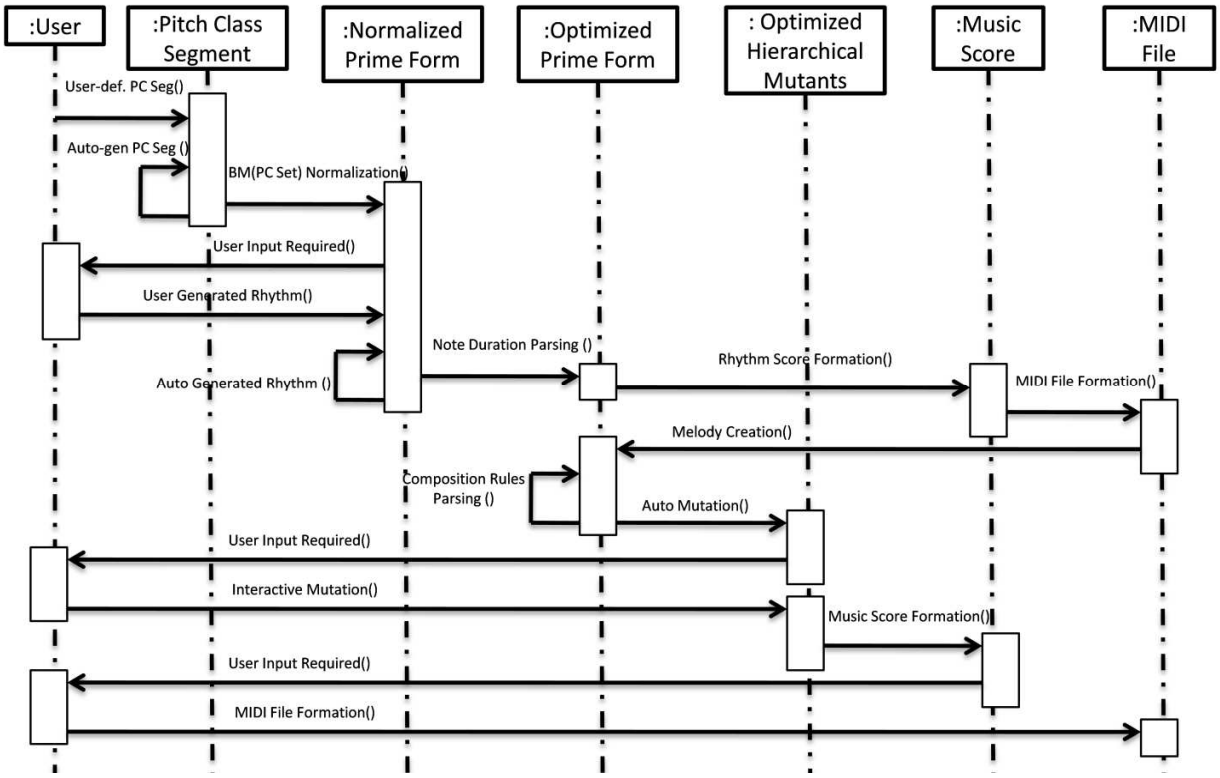


***Figure 5.*** *Sequence Diagram for Interactive Music Composition of the Hybridized Interactive Algorithmic Composition Model*

# 6. Algorithmic and Software Framework

***Listing 1.*** *The Java code shows how a rhythmic Pattern used for creating the beats binary code.*

```
public java.util.ListbeatBinaryCode(int m, int n, String rhythmicPattern){
        java.util.List<Object>beatList = new ArrayList<Object>( );
        //break down the rhythmic pattern into Beat Binary Code...
        for(int i=0;i<m;i++){//m=6/n=8, 11,11,11
                //String beatBinaryCode =
rhythmicPattern.substring(i*n,(i*n+m)+1);
                String beatBinaryCode =
rhythmicPattern.substring(i*(16/n),(i*(16/n)+16/n));
                beatList.add(beatBinaryCode);
        }//for i...
return beatList;
}//beatBinaryCode method...
```

This employs algorithmic techniques, such as flowchart to define the implementation nuggets of the framework. Consequently, the nuggets are presented as JAVA objects in form of variable, methods, classes, interfaces and code fragments/listings. Here, the framework for both Rhythm Generation and Melody Synthesis processes are described and presented.

The formulation of the H.I.A.C. Model provides the basis for developing the object oriented software framework for Interactive Music Composition. An object oriented framework, presented as Java objects (with JavaDoc hypertext documentation), is expected to be useful to music/multimedia programmers who develop music application/tools that would facilitate music creation and interaction (which reflects the needs of the user/musician in light of the conceptual context).

### *6.1. JavaDoc Framework Documentation*

Javadoc is a tool that parses the declarations and documentation comments in a set of source files and produces a set of HTML pages describing the classes, interfaces, constructors, methods, and fields. Javadoc facilitates "code reuse" by allowing other developers to use this framework during software development. See Figure6 for a sample ofJavaDoc documentation of framework for rhythm generation and melody synthesis.

| Method Summary | |
|---|---|
| java.util.List | beatBinaryCode(int m, int n, java.lang.String rhythmicPattern)<br>beatBinaryCode - this method generates serial beats binary codes for any given rhythmic patterns. |

***Figure 6.*** *The Javadocdocuments Java code as software framework.*

### *6.2. Simulation of the H.I.A.C. Model*

This section demonstrates the adoptability and realization of the developed software framework for the Hybridized Interactive Algorithmic Composition Model. This software framework (in form of Java objects) provides the basic building blocks for developers who wish to integrate audio, sound and multimedia into their applications. Thus, a prototype application has been developed to demonstrate the implementation of the framework.

The rhythm creation process through the process of randomization generates a rhythm binary code. Variations of the rhythmic patterns could be generated or created through randomization or permutation of the beat binary code. The result of the rhythm creation process of the Hybridized Interactive Algorithmic Composition Model is a rhythmic pattern devoid of melody.

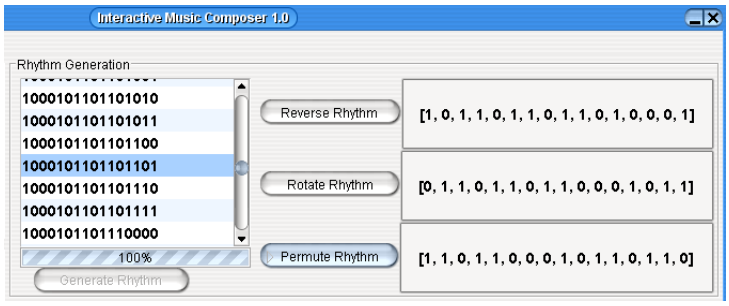


***Figure 7.*** *Demonstration of Rhythm Creation Process*

The rhythm creation process starts with the beat binary code; which could be generated through the process of randomization, makes up a rhythm binary code (which in essence is a rhythmic pattern based on the given meter/time signature).

Variations of the rhythmic patterns could be generated or created through randomization or permutation of the beat binary code. The result of the rhythm creation process of the Hybridized Interactive Algorithmic Composition Model is a rhythmic pattern devoid of melody.

The melody synthesis process creates pitch class segment, which is made up of beat motifs and could either be defined

by the user/composer or auto-generated through randomization process. The resulting pitch class segment passes through the process of normalization of the various pitch sets. The normalization process involves the ordering of the pitches in the set to obtain the most compact form.

Next, is the development of prime form – which is the result of the synthesis of pitch class segment and rhythmic pattern (rhythm binary code). The resultant prime form is analyzed through the note duration parser to determine note lengths using stochastic process of probability and randomization.

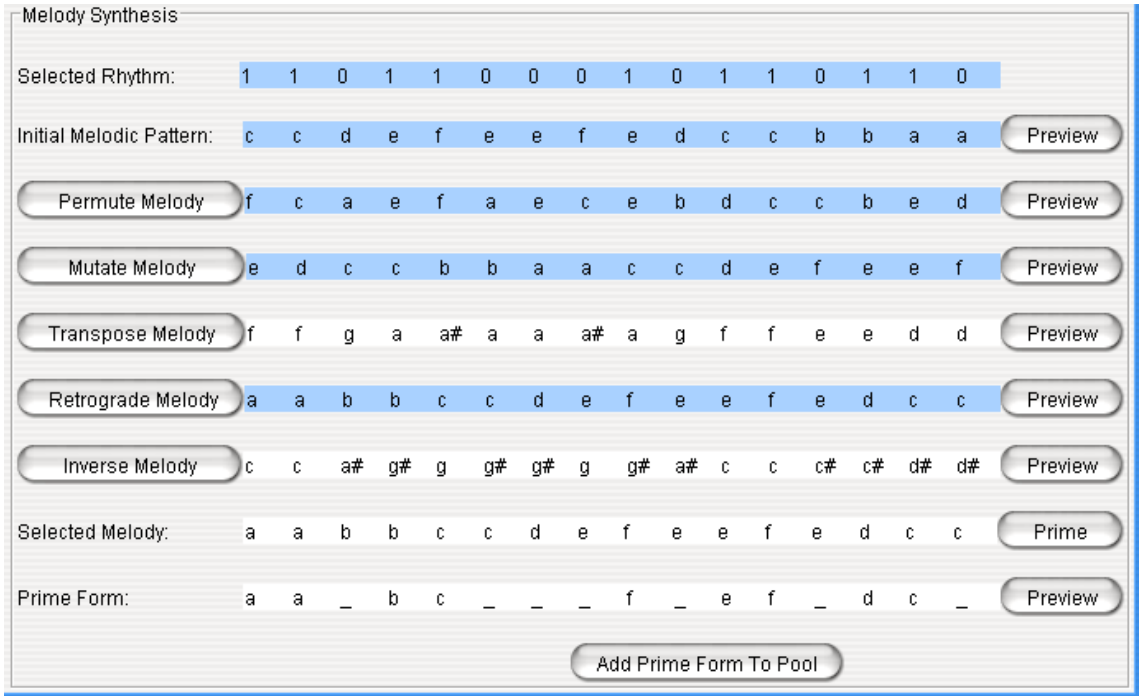


***Figure 8.*** *Demonstration of Melody Synthesis Process*

## 7. Conclusion

The use of algorithms for music composition guarantees speed and accuracy while relieving the user/composer of the herculean tasks and calculations, such as probability, permutations and randomization. Furthermore, the developed Hybridized Interactive Algorithmic Composition model specifically emphasized interactivity as a critical feature in algorithmic composition – where both the user/composer and the music application (without predefined constraints) participate in the process of music creation.

The Objected Oriented Modeling and Simulation of the H.I.A.C. model led to the development of Java objects (which represents the object oriented software framework) ensures code reusability in a robust and multiplatform systems and environments. This is a critical prerequisite of best practices in contemporary computer programming.

## References

[1] Garba, E. J. (2003). Computer Music – Rhythm Programming, Processing and Mastering. Trafford Publishing, Canada.

[2] Unehara, M. and T. Onisawa. (2009). Construction of Music Composition System with Interactive Genetic Algorithm. University of Tsukuba, 1-1-1 Tennodai, Tsukuba, Ibaraki, 305-8573, JAPAN. http://www.idemployee.id.tue.nl/g.w.m.rauterberg/conferences/CD_doNotOpen/ADC/final_paper/549.pdf.

[3] Järveläinen, H. (2000). *Algorithmic Musical Composition*. Helsinki University of Technology, Telecommunications software and multimedia laboratory. Retrieved December 5, 2007, from www.tml.tkk.fi/Studies/Tik-111.080/2000/papers/hanna/alco.pdf.

[4] Grout, D. J. (1996). History of Western Music. W.W. Norton & Company, 5th edition.

[5] Todd, P. M. and G. M. Werner. (1999). Frankensteinian Methods for Evolutionary Music Composition. In Griffith and Todd, P. M. (Eds.) *Musical networks: Parallel perception and performance*, 313-339.

[6] Espi, D., P. J. Ponce de Leon, C. Perez-Sancho, D. Rizo, J. M. Inesta, F. Moreno-Seco, and A. Pertusa. (2009). A Cooperative Approach to Style-Oriented Music Composition. Departamento de Lenguajes y SistemasInformaticos University of Alicante, Spain. http://193.145.231.49/repositori/grfia/pubs/186/wijcai07.pdf.

[7] Garba, E. J. (2012). *Multimedia Technology: A Software Framework for Interactive Music Composition*. (PhD Thesis in Music/Multimedia Technology, School of Pure and Applied Sciences Federal University of Technology, Yola, Nigeria).

[8] Gilkerson, J., Li, W. and Owen, D. (2005). *An Introduction to Random Number Generators and Monte Carlo Methods*. Retrieved June 1, 2007, from http://www.mgnet.org/~douglas/Classes/cs521/rng-mc/RandomMonteCarlo2005.ppt.

[9] Towsey, M., Brown, A., Wright, S. and Diederich, J. (2009). *Towards Melodic Extension Using Genetic Algorithms*. Queensland University of Technology Kelvin Grove, QLD 4059, Australia. Retrieved September 23, 2009, from http://eprints.qut.edu.au/169/1/towsey.pdf.

[10] Microsoft Encarta Encyclopedia. (2009). *Stochastic*. Microsoft Encarta 2009 [DVD]. Redmond, WA, USA: Microsoft Corporation.

[11] Holland, J.H. (1975). *Adaptation in Natural and Artificial Systems*. USA: Ann Arbor, University of Michigan Press (Second edition: MIT Press, 1992).

[12] Garba, E. J. and Wajiga, G. M. (2011). A Review of Multimedia and Music Technology: Mathematical Foundations of Rhythm Generations in Algorithmic Composition, *Bagale Journal of Pure and Applied Sciences*, Volume 8, Number 1, December 2011, pp. 64-74. http://www.mautech.edu.ng/

[13] Garba, E. J., Wajiga, G. M. and Oye, N. D. (2011). Multimedia and Music Technology: Mathematical Foundations of Melody Synthesis in Algorithmic Composition, *International Journal of Computer Vision & Applications*, Volume 1, Number 1, April 2011, pp. 9-14. http://www.3kbioxml.com/3k/index.php/IJCVA

[14] Pilone, D. and N. Pitman. (2005). *UML 2.0 in a Nutshell*. Sebastopol, USA: O'Reilly Media, Inc. 1005 Gravenstein Highway North.

# Author Biographies

**Etemi Joshua Garba** – is a Music/Multimedia Consultant with Ethereal Multimedia Technology. He is also a Senior Lecturer (Assistant Professor) in the department of Computer Science at the Federal University of Technology (ModibboAdama University of Technology), Yola, Adamawa State, Nigeria. He is a graduate of B.Sc. Computer Science and has Masters Degree in Software Engineering in Computer Science from the St. Petersburg State University of Information Technology, Mechanics & Optics. St. Petersburg, Russia.

**Gregory Maksha Wajiga** – is a Professor of Computer Science in the department of Mathematics and Computer Science at the Federal University of Technology, Yola, Adamawa State, Nigeria. He is a graduate of B.Sc. Mathematics from the Ahmadu Bello University, Zaria, Nigeria, M.Sc. Industrial Mathematics, Aston University, Birmingham, U.K. and has a PhD degree in Computer Science from the AbubakarTafawaBalewa University, Bauchi, Nigeria.