Music/Multimedia Technology: Creating Variations from Rhythm Generation and Melody Synthesis Processes of Hybridized Interactive Algorithmic Composition Model

Garba Joshua,Wajiga, Gregory Maksha

ABSTRACT

The Hybridized Interactive Algorithmic Composition (HIAC) Model was developed by Dr. Etemi Joshua Garba under the supervision of Professor Gregory Maksha Wajiga (Garba and Wajiga, 2014a). The HIAC model is an aggregate of various algorithmic composition models in order to minimize the weaknesses experienced when such models are used singlehandedly in music composition.The hybridization of the models (at different stages of composition) capitalized on the advantages of such models. The HIAC model was presented as software framework. In this paper, however, we shall take a review of how variations are created from the rhythm generation and melody synthesis processes of the HIAC model. The variation aspect of the HIAC model is expected to be useful to software designers and application developers in the field of Music and Multimedia Technology.

Keywords: music and multimedia technology; variations; rhythm generation; melody synthesis; hybridized interactive algorithmic composition model.

Classification: H.5.5, H.5.1

Language: English

London Journals Press

LJP Copyright ID: 884367
ISBN 10: 1537631683
ISBN 13: 978-1537631684

London Journal of Research in Computer Science and Technology

Volume 17 | Issue 1 | Compilation 1.0

Music/Multimedia Technology: Creating Variations from Rhythm Generation and Melody Synthesis Processes of Hybridized Interactive Algorithmic Composition Model

Garba, Etemi Joshua α & Wajiga, Gregory Maksha σ

I. ABSTRACT

The Hybridized Interactive Algorithmic Composition (HIAC) Model was developed by Dr. Etemi Joshua Garba under the supervision of Professor Gregory Maksha Wajiga (Garba and Wajiga, 2014a). The HIAC model is an aggregate of various algorithmic composition models in order to minimize the weaknesses experienced when such models are used singlehandedly in music composition. The hybridization of the models (at different stages of composition) capitalized on the advantages of such models. The HIAC model was presented as software framework. In this paper, however, we shall take a review of how variations are created from the rhythm generation and melody synthesis processes of the HIAC model. The variation aspect of the HIAC model is expected to be useful to software designers and application developers in the field of Music and Multimedia Technology.

Keywords: music and multimedia technology; variations; rhythm generation; melody synthesis; hybridized interactive algorithmic composition model.

Author α σ: Department of Computer Science, Federal University of Technology (Modibbo Adama University of Technology) Yola.

II. INTRODUCTION

The development of the HIAC model became necessary since the existing algorithmic composition models do not single-handedly take care of all the stages of music composition. Therefore, the HIAC model harnesses the strength of existing algorithmic composition models in order to overcome their weaknesses – when they are used individually.

The models for algorithmic composition used included: Mathematical (Gilkerson and Owen, 2005), Grammar (Rule-Based) (Towsey, Brown, Wright and Diederich, 2009), stochastic (Microsoft Encarta Encyclopedia, 2009), Knowledge-based (Järveläinen, 2000), and Evolutionary (Genetic Algorithm) (Holland, 1975). The HIAC 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.

III. ALGORITHMIC COMPOSITION

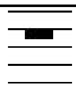
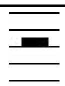
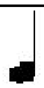
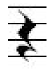
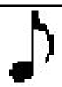
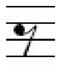
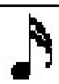
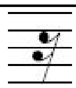
Algorithmic composition is the technique of using algorithms to create music. According to Fernández and Vico (2013), algorithmic composition is the partial or total automation of the process of music composition by using computers programs. There have been many studies on automatic music composition using computer since the conception of the computer, and some automatic music composition models have been proposed (Unehara and Onisawa, 2009), (Todd and Werner, 1999) and (Espi, Ponce de Leon, Perez-Sancho, Rizo, Inesta, Moreno-Seco, and Pertusa, 2009).

Music composition is perceived a complex and challenging activity for those not having musical knowledge or skill (Unehara and Onisawa, 2009). According to Garba (2003), music is the alternation of sound and silence. From the computational (logical) point of view, when sound is produced (or a note/tone is played) the binary value is 1, otherwise it is 0. Therefore, the binary musical concept perceives music as a stream of 1s and 0s. This means that musical ideas and concepts are logical – hence computable.

IV. RHYTHM GENERATION PROCESS OF THE HIAC MODEL

Rhythm Generation Process of the HIAC Model is based on both mathematical and grammar models. The Beat Binary code is the stream of bits required to represent a beat. See Table 1.

Table 1: Beat Binary Code

Note Type	Note symbol	Rest symbol	Beat Binary Code		
			Bits	Note Binary Value	Rest Binary Value
Whole			16	1000 0000 0000 0000	0000 0000 0000 0000
Half			8	1000 0000	0000 0000
Quarter			4	1000	0000
Eight			2	10	00
Sixteenth			1	1	0

$$B_{BC}=Random(Abs(2^n-1)) \quad (1)$$

Where: B stands for Beat, BC stands for Binary Code, Abs stands for absolute value and n specifies the type of note/rest assigned to one beat in a meter. The Rhythm Binary Code, which is made up of a measure, is a function of the Beat Binary code.

$$R_{BC}=\{B_{BC1},B_{BC2}, \ldots,B_{BCm}\} \quad (2)$$

Where: R stands for Rhythm, BC stands for Binary Code and m is the numerator that indicates the number of beats in a measure of a given meter. Note: R_{BC} logically represents a musical Rhythmic Pattern (RP). For example, an R_{BC} = {1010, 1000, 1110}, could be musically represented as shown in Figure 1.

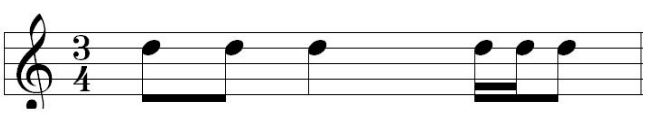


Figure 1: Musical notation of Rhythmic Pattern.

The rhythm generation process of the Hybridized Interactive Algorithmic Composition Model was developed from the concept formulation of the research by Garba(2012). The result of the rhythm generation process is a rhythm pattern devoid of melody. See Figure 2.

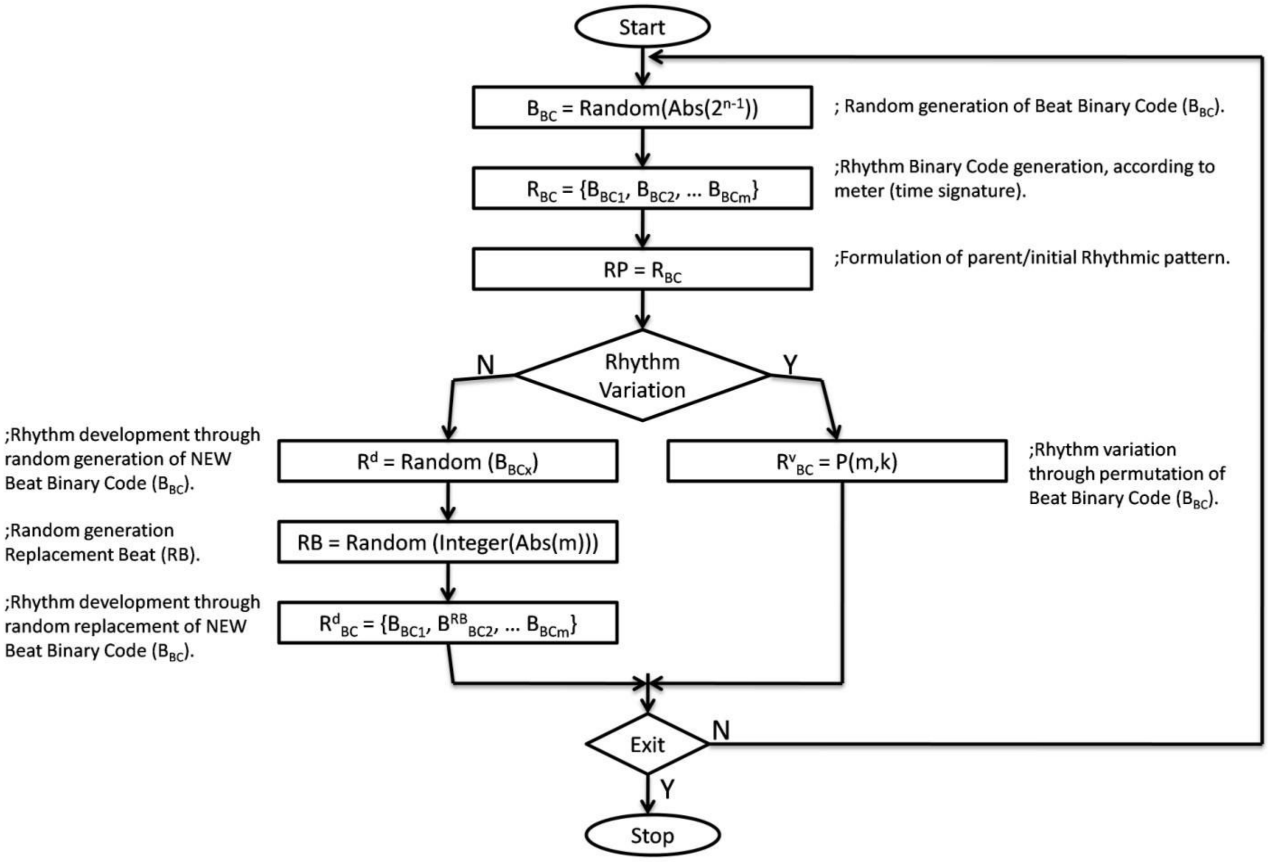


Figure 2: Rhythm Creation Process(Garba, 2012).

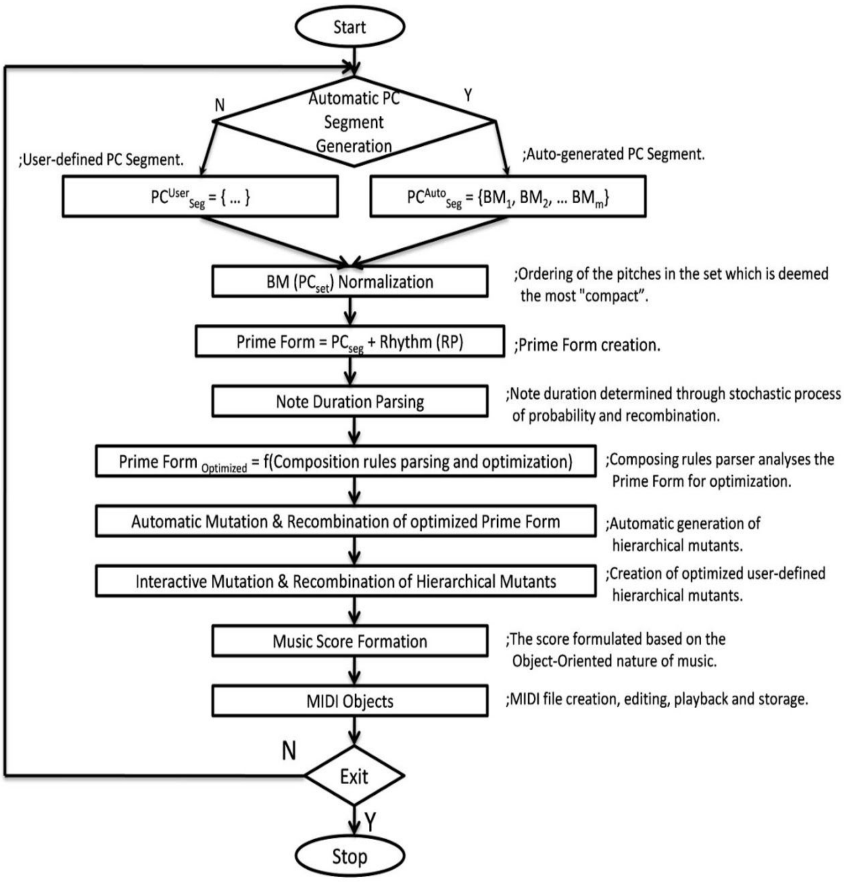


Figure 3: Melody Synthesis Process(Garba, 2012).

V. MELODY SYNTHESIS PROCESS OF THE HIAC MODEL

In the HIAC Model, the melody synthesis process involves several steps until the Prime Form of the melody is created (Garba and Wajiga, 2014b). See Figure 3 for full overview of the melody synthesis process. Note that the Prime Form is further analyzed by the Composition Rules Parser based on the basic prevalent contemporary music composition rules. Afterwards, the Prime Form is refined by the Optimizer. The optimization process could be done either manually by the composer or automatically. The final result of this stage is known as Optimized Prime Form. See Figure 4 for an example of an Optimized Prime Form.

Figure 4: Optimized Prime Form.

VI. CREATING VARIATIONS IN HIAC MODEL

The repetition of musical patterns helps listeners to comprehend music without having because that leads to the experience of similarity and variations within the listening process. Musicians do use altered repeated musical patterns to enhance the communication and experience of their musical ideas ((Muller and Kurth, 2007), (Deliège, 2007) and (Meyer, 2000)).

Music variation is strongly supported in HIAC Model through hierarchical mutation and recombination of the Optimized Prime Form that gives birth to what is known as Hierarchical Mutant (which is a variant of the original Melodic Pattern). Hierarchical mutants could be produced from the Optimized Prime Form through these processes: retrogradation, transposition and inversion. See Figure 5.

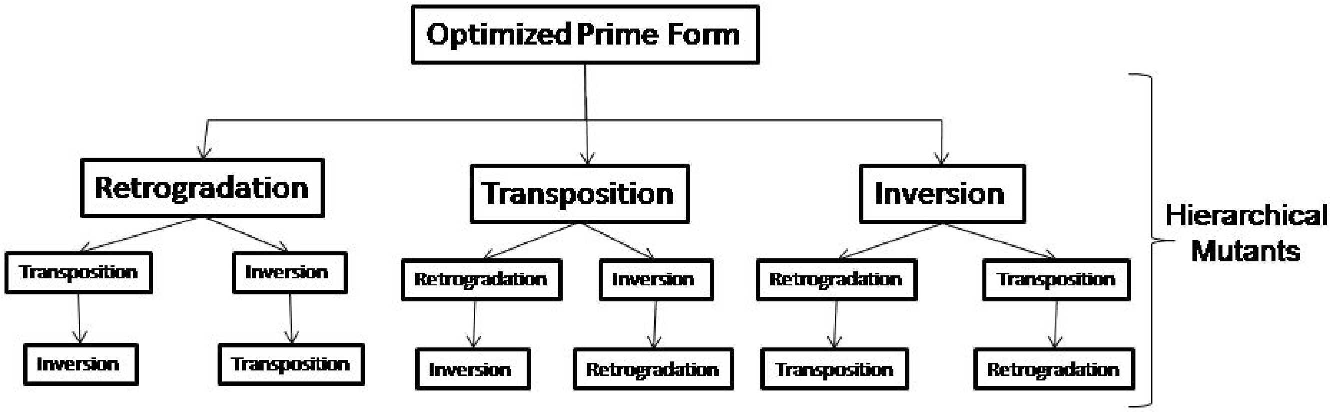


Figure 5: Hierarchical Mutation and Recombination of Optimized Prime Forms.

Transposition

Transposition mathematically captures the restatement of a melody at higher and lower pitch levels in a way that preserves intervals. Transpositional equivalence has also been in place in tonal theory. Any sonority (collection of pcs, or "pc set") can be transposed to another pitch level and retain its character (though the *function* may change in tonal music). For instance, Figure 6 shows a transposition of the Optimized Prime Form in Figure 4. Note that transposition neither changes the rhythm nor the intervallic structure of the melody.

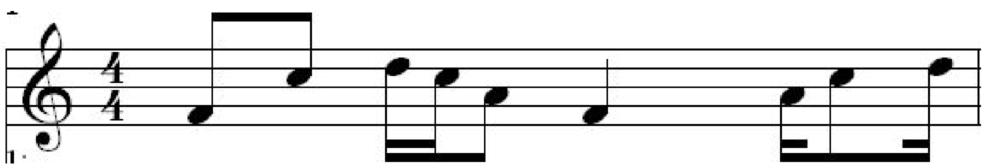


Figure 6: Transposition of the Optimized Prime Form (after transposing the musical piece by +5 cents, the tonic note is "F").

Retrogradation

The retrograde form of the Optimized Prime Form is created by writing the notes in the original version in reverse order. The retrograde, or backward form of the Optimized Prime Form, is the possible manipulation that can be used in serial composition. Since it is simply the reverse of the prime form, it is most conveniently found in a matrix by reading the desired musical piece backwards (from right to left). Figure 7 shows the retrograde of the Optimized Prime Form in Figure 4. After retrogradation, though the rhythmic structure of the melody changes, the intervallic structure of melody remains unaltered.

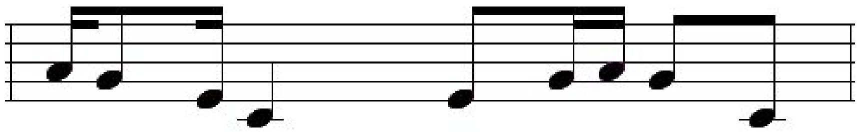


Figure 7: Retrogradation of the Optimized Prime Form (after retrogradation, "A" is now the tonic note of the musical piece).

Inversion

Inversion is another way to create the musical variation while preserving the intervallic sound of a melody (although it does not preserve the exact intervals). Inversion is the operation of turning the melody contour upside down across a horizontal line of symmetry. Figure 8 shows the graphical representation of the inversion operation, in which the Pitch Class sets (PC sets) are "flipped" around the 0 axis ((Nelson, 2007), (Huron, 1994), (Pendergrass, 2013), (Lewin, 2001), (Mead, 1988) and (Mead, 1989)).

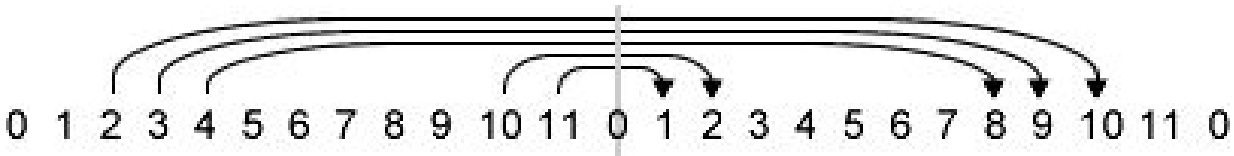


Figure 8: Horizontal line of symmetry used in the inversion operation of a Pitch Class sets.

Table 2 shows how an Inverted Prime Form is created from an original Prime Form using the horizontal line of symmetry in Figure 8.

Table 2: Creation of Inverted Prime Form

Prime Form	c	_	g	_	a	g	e	_	c	_	_	_	e	g	_	a
PC Number	0	_	7	_	9	7	4	_	0	_	_	_	4	7	_	9
Inverted PC Number	0	_	5	_	3	5	8	_	0	_	_	_	8	5	_	3
Inverted Prime Form	c	_	f	_	d#	f	g#	_	c	_	_	_	g#	f	_	d#

Melody Variation

Melody variation is realized by exchanging notes positions. That is, the permutations of the existing pitched notes, without altering the Rhythmic Pattern. This implies that a given Rhythmic Pattern could produce several Melodic Patterns through variation. Figure 8 shows the creation of a melody variation of the Optimized Prime Form in Figure 4.

Figure 9: Melody Variation of the Optimized Prime Form.

Melody Repetition requires using the same Melodic Pattern over and over again without making changes to both the Melodic and Rhythmic Patterns. Melody derivation is achieved when the Melodic Pattern is kept constant, while the Rhythmic Pattern is changed. Melody contrast is created when both the Melodic and Rhythmic patterns practically remain unchanged, but long notes are replaced with shorter ones or vice versa.

VII. CONCLUSION

The composer can interactively re edit/ recompose parts of the hierarchical mutants to create more customized user-defined Melodic Patterns. This task is realizable through variation, repetition, derivation and contrast. At this stage the hierarchical mutants are analyzed and optimized by the Composition Rules Parser and Optimizer of the HIAC Model.

REFERENCES

1. Deliège, I. (2007). Similarity relations in listening to music: How do they come into play? Musicae Scientiae, Discussion Forum 4A, 9-37. Retrieved from https://jyx.jyu.fi/dspace/bitstream/handle/123456789/19315/Deliege_MS2007D4A.pdf?sequence=1
2. 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 Sistemas Informaticos University of Alicante, Spain. Retrieved from http://193.145.231.49/repositori/grfia/pubs/186/ wijcai07.pdf.
3. Fernández, J. D. and Vico, F. (2013). AI Methods in Algorithmic Composition: A Comprehensive Survey. *Journal of Artificial Intelligence Research*, 48 513-582.
4. Garba, E. J. (2003). *Computer Music – Rhythm Programming, Processing and Mastering*. Bloomington, Canada: Trafford Publishing Canada.
5. 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).
6. Garba, E. J. and Wajiga, G. M. (2014a). Music/Multimedia Technology: Modeling and Simulation of the Hybridized Interactive Algorithmic Composition Model. *Software Engineering*. 2(2), 15-21. doi: 10.11648/j.se.20140202.11.
7. Garba, E. J., Wajiga, G. M. (2014b). Music/ Multimedia Technology: Melody Synthesis and Rhythm Creation Processes of the Hybridized Interactive Algorithmic Composition Model. *American Journal of Software Engineering and Applications*. 3(6), 106-111. doi: 10.11648/j.ajsea.20140306.17
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. Holland, J.H. (1975). *Adaptation in Natural and Artificial Systems*. USA: Ann Arbor, University of Michigan Press (Second edition: MIT Press, 1992).
10. Huron, D. (1994). Interval-Class Content in Equally Tempered Pitch-Class Sets: Common Scales Exhibit Optimum Tonal Consonance. *Music Perception: An Interdisciplinary Journal*, 11(3) 289-305. University of California Press. Retrieved from http://www.jstor.org/stable/40285624. Accessed: 08/04/2016 15:28
11. 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.
12. Lewin, D. (2001). Special Cases of the Interval Function Between Pitch-Class Sets X and Y. *Journal of Music Theory*, 45(1), 1–30.

13. Mead, A. (1988). Some Implications of the Pitch-Class/Order-Number Isomorphism Inherent in the Twelve-Tone System: Part One. *Perspectives of New Music*, 26(2), 96–163.
14. Mead, A. (1989). Some Implications of the Pitch-Class/Order-Number Isomorphism Inherent in the Twelve-Tone System: Part Two. *Perspectives of New Music*, 27(1), 180–233.
15. Meyer, L.B. (2000). *The spheres of music*, Chicago, IL, USA: University of Chicago Press.
16. Microsoft Encarta Encyclopedia. (2009). *Stochastic*. Microsoft Encarta 2009 [DVD]. Redmond, WA, USA: Microsoft Corporation.
17. Muller, M. and Kurth, F. (2007). Towards structural analysis of audio recordings in the presence of musical variations. *EURASIP Journal on Applied Signal Processing*, 2007(1), 163–163.
18. Nelson, P. (2007). *Pitch Class Sets*. Retrieved from http://composertools.com/Theory/PC-Sets/PCSets1.htm#_Toc72662199. Accessed: 08/04/2016 13:46
19. Pendergrass, M. (2013). *Two Musical Orderings*. Department of Mathematics and Computer Science, Hampden-Sydney College, Hampden- Sydney, Virginia, May 24, 2013. Retrieved from http://www.hsc.edu/Documents/academics/MathCS/Pendergrass/twoMusicalOrders_rev1_arxiv.pdf. Accessed: 08/04/2016 16:45.
20. 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. Retrieved from http://citeseerx.ist.psu.edu/viewdoc/summary? doi=10.1.1.71.943
21. 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.
22. 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. Retrieved from http://www.Idemployee.id.tue.nl/g.w.m.rauterberg/conferences/CD_doNotOpen/ADC/final_paper/549.pdf.