

A Visual Language Framework For Music Rendering Using L-System

LIM CHEN KIM, ABDULLAH ZAWAWI TALIB

School of Computer Sciences

Universiti Sains Malaysia

11800 USM Pulau Pinang

MALAYSIA

kim86_lavender@hotmail.com, azht@cs.usm.my

Abstract: - Although the existing Lindenmayer Systems (L-Systems) have been traditionally used to model plant development graphically, it is also possible to interpret L-Systems for music rendering. However, many of the existing L-System modelling applications, especially for musical rendering do not cater a larger circle of people who may not be well-versed in music and neither for those with programming skill nor prior knowledge in L-Systems. In this paper, a visual programming framework is proposed for L-System music rendering in order to cater a larger circle of people including non-experts in L-System and music. A visual language framework for L-System has been developed and the usefulness and the efficiency of visual language in L-System music rendering have also been studied. The visual language framework includes the L-System attributes and standard musical grammar rules in the form of simplified icon-based visual language. Furthermore, the music rendering has also been further fine-tuned by using stochastic and context-sensitive L-Systems. Based on the evaluation of the proposed visual language framework, it has been shown to be effective in the sense that anyone who does not have prior knowledge in L-System music may also fully utilize the easily generated musical sound based on the original L-System concepts.

Key-Words: - Attributes, L-System, Plant Modelling, Visual Language Framework, Music Rendering

1 Introduction

Visual representation has been employed to model or serve as a means of expressing with respect to some particular purposes. As time evolves, the use of pictorial information in visual programming languages (VPLs) has been applied in various aspects. VPL is an application development environment that utilises symbols, charts and diagrams explicitly or specifies them through other forms of representations in order to increase the visualization of the system.

According to Menzies, many researchers claim numerous benefits of visual framework for its friendly computing environment through the representations of pictures that are used to interact with a program in many different circumstances [1]. The visual language framework proposed by Siew and Talib cover the L-System attributes and grammar in the form of icon-based visual language for plant modelling [2]. On the other hand, OpenMusic allows composers to create formal computing models of musical structures and processes [3]. However, the execution can only be done provided that the composer keys in the L-System grammar strictly. Another recent tool which is related to this area is the OpenAlea by Pradal et al. which is a visual programming and component-based software platform specially tailored for plant modelling but without involving the L-System [4].

The Lindenmayer System or better known as L-System is actually a mathematical modelling system which is mainly used for multi-cellular organism and

development and modelling of plants [5]. In L-Systems, grammar productions are applied in parallel and simultaneously replacing all the letters in a given string. Subsequently, the interpretation of L-System was proposed by Worth and Stepney with a view of turning plant modelling into music rendering [6]. Hence, a versatile tool for music rendering is needed to emphasize the relation between plants and music. In order to interpret the L-System as music, the existing LMUSE [7] maps the standard turtle's 3-D movement, and the orientation directions (forward, up, and left) into musical pitches. In L-System music, there are three distinct rendering techniques namely parallel, sequential and Schenkerian rendering for modelling of leaf, distorted leaf, bush and twig.

The recent research on L-System is geared towards music rendering in creating many interesting musical scores. Lourenco et al. proposed an L-System using the genetic operators and strings that are generated in order to increase the variability [8]. Thus, it is essential to further extend the L-System in the visual world to music rendering since the focus of the L-System nowadays is converging to the world of music.

2 Related Works

OpenMusic [3] allows the users to work with sound related data and provides a control over the low-level musical processes that are traditionally delegated to predefined tools. Although extra features of the

OpenMusic have been provided to allow users to work with sound related data, the cross connections between those data can be upgraded to provide an overall control over a higher level musical process and the control must be an easy and user-friendly one for the users. OpenAlea [4] provides a visual and interactive interface to the inner structure of a Functional-Structural Plant Modelling (FSPM) application. However, the visual programming environment has been designed for model integration and connection rather than for modelling feedback and retroaction between models.

Siew and Talib have clearly compared the advantages and disadvantages of their modelling tools [2]. Their visual language framework is a good basis that allows users who want to draw a tree even though they have no prior knowledge in programming and L-System. Menzies has tested the numerous benefits of visual frameworks and defined the criteria that satisfy the definition of visual programming [1]. Furthermore, there are still many arguments on the advantages and disadvantages of visual programming because it depends on the scenarios.

Lourenco et al. presented the concept of Genetic L-System where the set of productions rules can be changed between successive iterations and the changes are made by two genetic operations such as crossover and mutation and generates the MIDI files [8]. It has some limitations that must be improved. Worth and Stepney presented two musical renderings that produce pleasant sounds from classic 'plant' L-Systems [6]. In their work, Stochastic L-Systems are used to enforce some kind of structure on the score and context-sensitive L-Systems in order to offer the best potential for creating longer pieces of music. The work on producing sounds from the classical plant works well with the two musical renderings method but the attempt to get simultaneously pleasing graphics starting from a musical grammar has been less successful.

The scores generated by Prusinkiewicz are quite interesting [9]. They are relatively complex but they also have a legible internal structure. The score generation is very attractive but it can be further improved by having other important parameters in music such as by adding volume and tempo into the pitch and duration to speed up the performance, and fine tuning the musical string symbols generated by an L-System.

The L-System has started off with the idea of making improvement to best mimic the development of biological plant in reality. This area has been further extended from plant modelling to musical rendering. The L-System is capable of traversing plant to music and vice versa. Within the L-System, context sensitive and stochastic L-System are the two most outstanding musical rendering method which helps in generating more "pleasing" music melodically.

Majority all these available tools need the user to have the experience and exposure on L-Systems as well as music and programming. Therefore, this research aims to provide a visual language framework for music rendering that will not only benefits experts in music and L-System, but also cater those who have no prior knowledge in these areas. In order to resolve the deficiency of L-System as well as visual language in musical environment, this research also aims to provide a visual a set of grammar models that allows the interpretation of "pleasing" melody, and improve further the pieces of music by introducing stochastic and context-sensitive grammars in visual music rendering.

3 System Methodology

3.1 L-System Music Study

As shown in Figure 1, the overall methodology of this research starts with L-System music study before other steps. The L-System music study is required in order to ensure each attribute work, and to look into the restrictions and limitations of the L-System music classes in this research. L-System music study is concerned with spatial, sequential and Schenkerian rendering technique. Besides, analysis on the stochastic and context-sensitive grammar and how it is beneficial to this research is done before the implementation stage. There are three types of musical rendering techniques such as the parallel, sequential and Schenkerian rendering [6].

In parallel rendering, the beginning and end of branches is used to push and pop the time. F adds to the time, to make the shorter branches play first followed by longer ones after a period of time proportional to their difference in length, and makes the music not so pleasing. There is a period of silence in parallel rendering before this start and no branches have length 0. The reason to this is that the time at which a note is played is dependent on the length of the branch it is associated with. However, this silence can be removed by using sequential rendering. Thus, the rendered music is melodically more pleasing but, there is still no rhythm as the length of branches is being ignored.

Although the sequential rendering has the advantage over the parallel rendering but the interpretation of 'F' remains. Schenkerian put these two ideas together by having the number of F's in a branch determines the length of a note, and notes are played sequentially. This is done so that within the brackets of the musical note, the duration of the entire note is proportional to the number of F's. The result of this combination renders a wonderful melody in spite of the unfit into a 4/4 framework. Even with a metronome

beating standard 4/4 tempo behind it, the melody is still very musically projected. The stochastic rules are vital to read the grammar differently to meliorate the music rendering. Most of the time, every occurrence of the predecessor is simply replaced by the successor that is bounded by a set of production rules represent by the derivation symbol \rightarrow . The derivation of successor is obtained through a set of predecessor which maps a set of probabilities. The production is done by replacing the successor which has higher probability based on the frequency a note appears in a set of input notes, starting from an axiom. The combination of the probability of each musical note is sum to 1. In order to experience variety of pitches of melody, this implementation contrives to allow the expansion of context-sensitive rules that concerns with the other symbols surrounding the symbol to be replaced. The context-sensitivity of a rule is portrays like $AC=DEF$ which express that the B will be replaced by DEF provide A is on the left of B and C is on the right of B. The rule can also be one-sided at times.

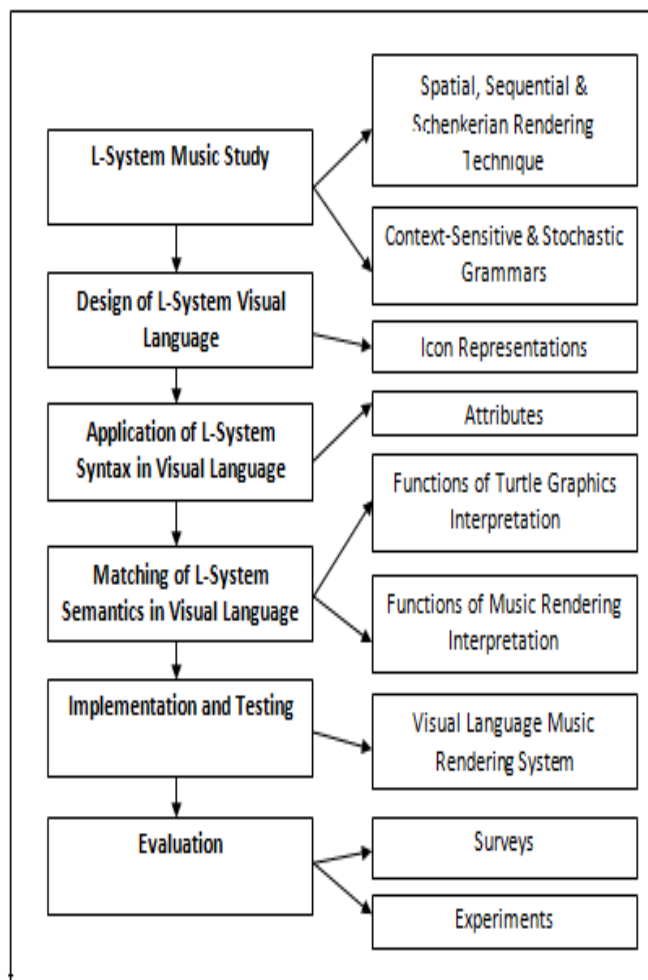


Figure 1. Stages of Music Rendering in Visual Language Environment

3.2 Design of L-System Visual Language

The design of L-System Visual Language is the second stage and the most interesting stage. Here, the visual language framework is to be developed in order to produce more user-friendly interface for users with little or without any knowledge of L-System so that they can still produce the music using L-System. The visual language consists of elementary object icons which are the step size and the angle, composite object icons which are the production rules and the process icons are 'play', 'transpose up', 'transpose down', 'push' and 'pop' that are the attributes from the Bracketed OL-System class.

3.3 Application of L-System Syntax in Visual Language

The third stage of this research is the application of L-System syntax in Visual Language. The syntax part of the visual language framework is the orderly arrangement of the icons in visual language. In other words, syntax is a set of rules for forming admissible string from a set of production rules in a grammatical arrangement. Basically, the framework is designed to follow the L-System syntax and a line of visual language sentence in the framework is interpreted from left to right. The semantic part of this framework is the meaning behind a group of composite icons. The meaning running at the background of a set of composite icons is the semantic portion of this particular framework.

The syntax of the framework and the functions of each icon also contribute to the semantic area. The attributes used namely 'F', 'Angle', '+', '-', '[', and ']' are adopted through the Deterministic Context-free L-System (DOL-System), the graphics turtle interpretation of strings and the Bracketed OL-System. The attribute 'F' distinguishes string of the initial state, the attribute of 'Angle' provides the increment value of the pitch or octave, the attribute '+' moves up one note in the scale of the chosen key, the attributes '-' moves down note in the scale of the chosen key, the attribute '[' pushes everything in the current of the turtle into a stack and the attribute ']' retrieves everything in the current state but not the event time.

3.4 Matching of L-System Semantics in Visual Language

The fourth stage is matching of L-System semantics in Visual Language. Each of these attributes has their own functionalities that can be converted into visual language representations. The semantic can as well simply generate a single function or it may bring a lot of complexity to the complex musical score rendering and generation. In general, the framework semantic is

dependent on the L-System music semantic. As a result, when different icons are being selected and put together, different output of L-System musical scores are generated.

3.5 Implementation and Testing

The implementation is the fifth stage in the methodology. The visual language framework is implemented in a music modelling L-System application. The L-System code is taken from a open source website which is Java L-System application. In the visual programming domain, the emphasis of research is dependence on the application of visual formalism because from the view of the programming thus, it is regarded to be more effective than the textual formalism.

The system called VL Music Rendering aims to be an application which its development environment is designed based on the dataflow programming model. The usage of dataflow model has a few advantages which results in easier debugging process because it allows an immediate access to the program state. Besides, dataflow languages also support automatic parallelization which is vital for the music rendering and score generation process. As the processes encountered in the implementation of VL Music Rendering are concurrent processing scenarios, hence the dataflow model is well suited in this case.

3.6 Evaluation

Finally, in the evaluation stage which is the final stage, a survey together with a set of experiments are carried out to determine the acceptability and the quality of the techniques and methods derived in this research. The main objective is to gather the information and comments of the Visual Language music modelling system as opposed to LMUSE. In the survey, questions and comments regarding the quality of the output will be prompted to the respondents through a set of experiments. The respondents comprise experts and non-experts in music fields from the School of Art, USM. A rating system is used for the survey. The rating scale is from 1 to 5, 1 being the worst and 5 being the best rating. At the end of the survey, users will need to comments on visual language framework compared to the existing music rendering tool LMUSE so that future work for this research can be identified.

4 Implementation

4.1 Processes of VL Music Rendering

Based on L-System, the plant modelling then can be interpreted for music rendering. On the front end of the

implementation, the system graphical user interface (GUI) is designed in a music environment. The visual language framework is implemented with musical icons that must be able to drag and drop and is written in Java using Netbeans IDE in order to integrate with the context-sensitive and stochastic concept that is taken partially from LMUSE system. A snapshot from the front end is portrayed in Figure. 2.

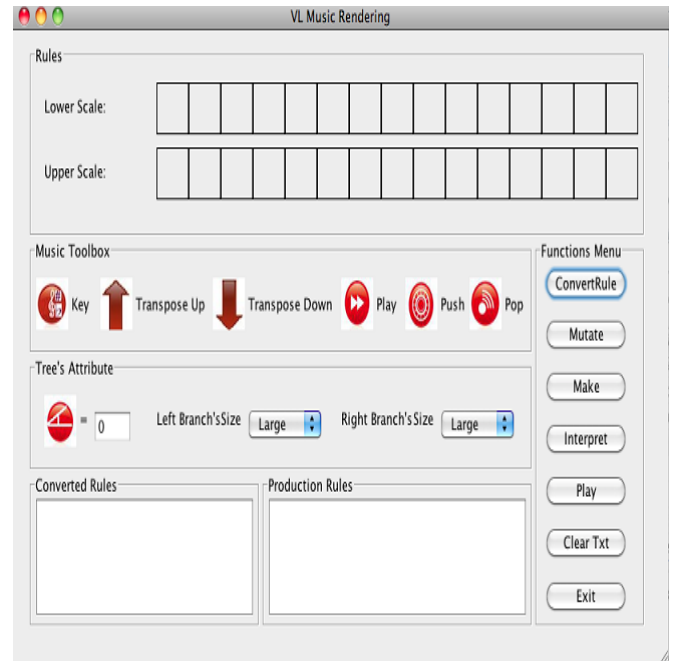


Figure 2. A Snapshot of VL Music Rendering Interface

As for the backend, the engine runs the L-System theory that matches the grammar of plant to music. The lower scale and upper scale need the input from the icons before it can be stored in an array. The array is then passed for conversation into L-System grammar by rule matching in order to enable plant modelling. In this case, data flow model will be used to aid the stages of the interpretation of the grammar to plant modelling and finally the music sound is rendered through the Musical Instruments Digital Interface (MIDI) library.

4.2 Icon for the Interface

After studying the L-System classes, it is found that there is a necessity to adopt the attributes used through the Deterministic Context-free L-System (DOL-System), the graphics turtle interpretation of strings and the Bracketed OL-System. Each of these attributes has their own functionalities that can be converted into visual language representations. The visual language consists of elementary object icons which are the step size and the angle, composite object icons which are the production rules and the process icons namely 'F', '+', '-', '[', and ']' are the attributes from the Bracketed OL-System class.

5 Evaluation

In order to evaluate the usefulness and the efficiency of the proposed framework for music rendering, a set of experiments and questionnaires has been designed to obtain feedback from various respondents.

In the first experiment, a user is given two test cases. A user needs to type in the complete grammar until the music is rendered on LMUSE while on the VL Music Rendering system, a user is required to simply drag and drop to render the musical sound. With this experiment, a user is able to rate the user-friendliness of the both systems.

In the second experiment, a user needs to predict and understand the attributes of the L-System when a rule is interpreted on LMUSE and when an icon function name is selected on VL Music rendering system. Then, users are able to answer the question on which system helps them to understand the theory of L-System at the basic level better.

In the third experiment, a user needs to listen to the musical sound rendered by LMUSE as well as VL Music Rendering system and then make comparison between the both sounds. With these sound, the user would have to evaluate and consider that which systems rendered a more harmonious music. The harmony of the musical sound rendered is defined as the complication of note played per tab on the instrument.

The results of the questionnaires are compiled into the categories of VL Framework, L-System and VL Music Rendering respectively using bar graph as shown in Figure 3.

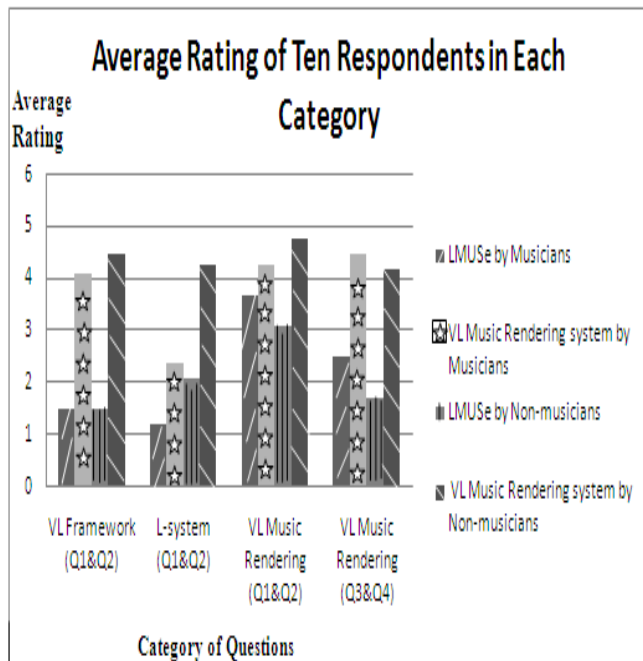


Figure 3. A Bar Graph which Indicates the Average Rating of Ten Respondents in Each Category

6 Discussion

From the collection of results, this research has hit the ultimate contributions to produce a Visual Language Framework for Music Rendering using L-System. First and foremost, this research has designed and produced a simplified icon-based visual language framework for L-System music rendering that improves an expressive music environment of such tool. Secondly, this research has completed the objective to construct visual language grammar models for L-System music rendering that is to be fed into the system in order to be more effective for the proficiencies of plant modelling to music rendering transformation. Thirdly, VL Music Rendering system has enhanced the method of visual generation of the musical notes by using stochastic and context-sensitive L-Systems. These L-Systems can melodize and fine tune the music rendering significantly by playing a few musical notes at one time.

7 Conclusion and Future Work

Introduction of music rendering in visual environment is rather successful so far. However, there are several potential extensions for further improvements of the work. There can be a room for musical score generation with better MIDI support as VL Music Rendering system has much potential to be a tool for basic music learner. The generation of musical score can aid the learners to read musical notes while listening to the musical sound being rendered simultaneously. In terms of the harmony of musical sound, researchers can also perform a more sophisticated mutation process in an effort to address the level of harmony in melody.

References:

- [1] T. Menzies, Evaluation issues for Visual Programming Language: *Handbook of Software Engineering and Knowledge Engineering*, Vol. 2, 2000, pp. 93-101.
- [2] B. H. Siew and A.Z. Talib, Visual Language Framework for Plant Modelling Using L-System, *Lecture Notes in Computer Science, Springer Berlin/ Heidelberg*, Vol. 5857, 2009, pp. 696-707.
- [3] J. Bresson and C. Agon, Sound Writing and Representation in a Visual Programming Framework, *DMNR-06 Doctoral Research Conference, Digital Music Research Network, Goldsmiths College, University of London, United Kingdom*, 2006.
- [4] C. Pradal, S. Dufour-Kowalski, F. Boudon, C. Fournier, C. Godin, OpenAlea: A visual Programming and Component-Based Software

- Platform for Plant Modelling, *Functional Plant Biology*, Vol. 35, 2008, No. 10, pp. 751-760.
- [5] P. Prusinkiewicz, and A. Lindenmayer, The Algorithmic Beauty of Plants, *Springer-Verlag New York Inc.*, 1990.
 - [6] P. Worth, and S. Stepney, Growing Music: Musical Interpretations of L-Systems, *Lecture Notes in Computer Science, Springer Berlin/ Heidelberg*, Vol. 3449, 2005, pp. 545-550.
 - [7] D. Sharp, LMUSE [Online]. [Accessed 28th January 2010]. Available from World Wide Web: <http://www.geocities.com/Athens/Academy/8764/lmuse/lmuse.html>, 2001.
 - [8] B. F. Lourenco, J. C. L. Ralha, M. C. P. Brandao, L-Systems, Scores, and Evolutionary Techniques, *Proceedings of 6th Sound and Music Computing Conference, Portugal*, 2009, pp. 113-118.
 - [9] P. Prusinkiewicz, Score generation with L-systems, *Proceedings of the International Computer Music Conference, Den Haag, Netherlands*, 1986, pp. 455-457.
 - [10] S. Teresi, Scott Teresi's website [Online]. [Accessed 25th January 2010]. Available from World Wide Web: <http://teresi.us/html/main/programming.html>, 2003.