INTERACTIVE LEARNING OBJECT FOR MUSIC TUTORING BASED IN SEQUENCE ALIGNING ALGORITHM

Daniel de Souza Carvalho[1], Sandra Maria Dotto Stump[2]

Abstract *— Music tutoring, piano in particular, is an one-to-one relationship between the student and professor, in most cases. In order to verify if it is possible to provide a machine-student tutoring for music, a prototype instructional system was developed as an Interactive Learning Object. The system shows to the student a musical peace on screen, in form of music sheet. And compares the student played song by sequence aligning algorithm, calculating the distance between the requested and played songs. The system verifies the student performance and them provides a new musical peace with appropriate level, in a continuous evaluation feedback. The student data can be sent to a learning management system or web server, for a tutor (professor) evaluation.*

Index Terms — Educational Technologies, MIDI, Learning Objects.

INTRODUCTION AND CONCEPTS

Music learning requires a lot of practical training with the instrument. Usually for piano it is a one-to-one relationship between student and professor. In order to check if it is possible to provide tutoring by software. A prototype instructional system named "jMusicTutor" was developed. This instructional system uses artificial intelligence (IA) algorithm to check student performance. The "exam" end "evaluation" is made by the instructor (professor), but the system provides details about user behavior and can guide the student according its performance in different levels of musical pieces (exercises).

According Silver, Strong and Perini [8], "Playing the piano requires at least three intelligences: musical, for following the music and keeping time; bodily-kinesthetic, for manipulating the fingers and feet appropriately; and special, for determining the relationship between keys and the sounds they produce. In additional other intelligences, such as interpersonal for responding to facial and bodily cues from bandmates during a performance or intrapersonal for creating personally relevant, impassioned musical expressions may be used as well."

It is a challenge subject to test IA for human tutoring because playing the piano is a sophisticated human activity that requires distinct abilities [8] such as Multiple Intelligences (MI) [4] and Learning Styles (LS) [3, 8].

INTERACTIVE LEARNING OBJECT

There are some distinct definitions for Learning Objects (LO) [5, 6], as follows:

- "any entity digital or non-digital that may be used for learning, education or training" [7];
- "any digital resource that can be reused to support learning" [6];
- "A new way of thinking about learning content. Traditionally, content comes in a several hour chunk. Learning Objects are much smaller units of learning, typically ranging from 2 minutes to 15 minutes" [6, 5].

An Interactive Learning Objects (IAO) is a sub set of LO [1, 2], they are active LO where students can interact with. ILO is an interactive small application [1, 2] (software) to assist education.

ILO FOR MUSIC TUTORING

The music tutor "jMusicTutor" is an ILO, it has an user interface that shows an music sheet (Figure 1), and a MIDI controller (musical keyboard, digital piano) should be connected to the computer by MIDI USB interface, in order to student to play the musical peace. The system record the user performance and compare with the proposed peace, if the student succeed the system propose a more difficult peace (promote student to an upper level), if the student fail (have a big distance from the peace proposed) the system presents an easier song (recommends an lower level to the student), it can keep at the same level according and average performance (equivalent level). The parameters that defines the error levels can be set at the configuration file; the professor can decide what are the appropriate error accepted at the three levels, "upper", "lower" and "equivalent", other levels can be create providing new parameters.

TOOLS AND SYSTEM DEVELOPMENT

The tools used to build up the instructional system were:

- Programming language: Java SE 1.6;
- Integrated Development Environment: NetBeans 6.5 and Eclipse 3.4.2;

1 M.Sc. Daniel de Souza Carvalho, Universidade Presbiteriana Mackenzie, Electrical Engineering Program, Rua da Consolação, 896 – 8 andar – São Paulo, Brazil , danielscarvalho@gmail.com

2 PhD. Sandra Maria Dotto Stump, Dean of Research and Graduation Degrees for Universidade Presbiteriana Mackenzie, Rua da Consolação, 896 – 8 andar – São Paulo, Brazil, sstump@mackenzie.br

- Local data base: XML files at the file system to store the student information, configuration file and songs;
- MIDI controller: M-Audio Prokeys 88sx – 88 keys;
- Interface: MIDI USB (usb cable);
- TrueType font: MusiQwik and MusiQwikB with musical notation characters;
- Operating System: MS Windows or Linux.

The IA algorithm that the system uses is "Levenshtein Distance" [11, 12, 13]. It does a sequence aligning [9, 10] between the notes of two musical pieces, if the distance is zero, the sequences are equal, as more the resulting value is, bigger the error is. The Levenshtein Distance algorithm uses insertion, deletion or substitution operations in the two sequences in order to find the minimum number of edits to transform the sequences in equivalents; it is possible to check according the operations the distance between the two sequences. An example for the musical notation is:

```
Required        E3 A3 E3 A3 A3
Value (Points)  0  1  0  0  0
Played          E3 B3 E3 A3 A3
```

At the second note the student has pressed B3 instead of A3, it add one the value, that is the distance between the two sequences.

```
Required       E3 -  A3 E3 A3 A3
Value (Points) 0  1  0  0  0  0
Played         E3 B3 A3 E3 A3 A3
```

At this second example, the student has played on additional note B3, the align algorithm add one space at the required sequence and calculate and difference of one between the two sequences.
This algorithm is used to check the two sequences and evaluate the student performance. The algorithm can be adjusted/adapted to set the value of points for each operation and give different values for insertion, deletion or substitution.
The time of each note is evaluated according its average, for example:

```
Required       1 1 1 1   4
Value (Points) 0 0 0 0.5 0
Played         1 1 1 0.5 4
```

The difference between the time required and played is 0.5 seconds. Because of MIDI clock precision and the human feeling about time, it is possible to set a percentage of tolerance for appropriate evaluation of notes timing.

The user interface (Figure 1) shows the following details:

- One music sheet (musical peace) that should be executed by the student. There are two grand staffs, with the bass staff and treble staff;
- Name of the musical peace or exercise;
- Author name of the musical peace, if know;
- Tempo of the musical peace or exercise;
- Keyboard image with 88 keys, it highlights the key pressed at the MIDI controller;
- The MIDI input and output device in use;
- The latency of output MIDI device (in seconds);
- Buttons bar and menu;
- Record indicator, it shows a red bullet when the system is recording by the student at the MIDI controller.
- The functional velocity, volume of a note or chord;
- Student's name;
- Record button: it start recording the student execution;
- Check button: it verifies (compare) the musical peace executed by the student with the one proposed, and show results on screen; it is a post processing.
- Next button: the system select the next exercise according the least student performance analysis.
- Replicate button: it download and upload the XML files with an WEB server (or LMS).

The system keeps the users files and songs files from the application root subdirectory "users" and "songs" respectively, all files extensions are ".xml", and the file name is the name of user or song/exercise. The configuration file "jMusicTutorConfig.xml" is kept at the application root directory.
The user files record information about: user name, current level, score of all executed exercises, date and time of the exercises execution. The song file contain: song level (score), title, author, tempo, notes sequence, etc.
The TrueType fonts "MusiQwik" and "MusiQwikB" should be previously installed at the system in order to correctly show the musical notation character on screen.
To run the system the Java Runtime Environment (JRE) is required, and a MIDI controller too.
To avoid latency problems, the system can select the MIDI output device that can be the MIDI controller or an external MIDI module. A MIDI output made by software can carouse latency, 0.6 seconds for example, and it is inappropriate for education and real time application. There are operating systems parameter to minimize this problem (for MS Windows or Linux), and there are real time operating systems versions that can minimize this issue too. The system works in asynchronous mode to minimize the latency, and the audio output of MIDI controller can be used directly.

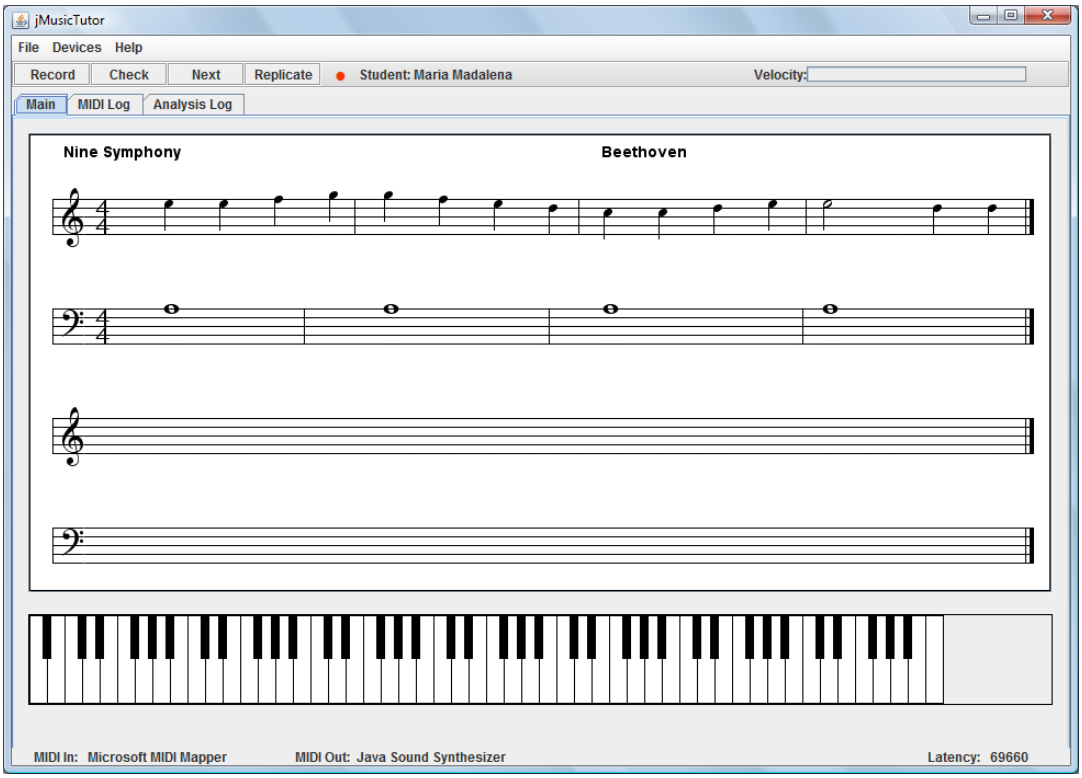

FIGURE. 1
INSTRUCTIONAL SYSTEM JMUSICTUTOR USER INTERFACE.

FINAL THOUGHTS

AI algorithms can be used to check students' performance and provide tutoring by software, but it is important to keep the evaluation to a professor responsibility.
More research is need for music tutoring by software, because the musical notation is very sophisticated and it requires a lot of effort for a fully functional support.
Other algorithms can be tested and compared to provide musical evaluation such as neural networks.
The jMusicTutor needs to be tested with a set of students to check it usage in practice and real music students.

REFERENCES

[1] Carvalho, D, S, Stump, M, D, Mustaro, P, N, "Interactive Learning Objects to Aid Education for The Digital Generation", World Congress on Communication and Arts *, pp. 60 ,* 2007.

[2] Carvalho, D, S, "Objetos de aprendizagem interativos : ferramenta de apoio a disciplinas teóricas", Master's thesis, Universidade Presbiteriana Mackenzie, São Paulo, Brazil, 2009.

[3] Felder, R, M, Silverman, L, K, "Learning and Teaching Styles in Engineering Education", 2002.

[4] Gardner, H, "Multiple Intelligences: New Horizons in Theory and Practice", Basic Books, 2006.

[5] Northrup, P, T, "Learning Objects for Instruction Design and Evaluation". INFOSCI, 2007.

[6] Wiley, D, A, "Connecting Learning Objects to Instructional Design Theory: A Definition, a Metaphor, and a Taxonomy", The Instructional Use of Learning Objects, Bloomington, IN: Agency for Instructional Technology, 2002.

[7] IEEE, IEEE 1484.12.1-2002, "Draft Standard for Learning Object Metadata", IEEE Learning Technology Standards Committee (LTSC), 2002.

[8] Silver, H, F, Strong, R, W, Perini, M, J, "So Each May Learn Integrating Learning Styles and Multiple Intelligenes", ASCD, 2000.

[9] Hodas, Nathan, O, Alberts, Daniel, P, "Efficient computation of optimal oligo-RNA binding", Nucleic Acids Res. Oxford University Press, 2004.

[10] Gunewardena, Sumedha, "Information Categorization in Biological Sequence Alignments", Technical Report #RPG-RR-01-24, Computing Laboratory, University of Oxford, 2004.

[11] Levenshtein, "The Levenshtein-Algorithm", {HYPERLINK: http://www.levenshtein.net/}, 2010.

[12] Gilleland, Michael, "Levenshtein Distance, in Three Flavors", {HYPERLINK: http://www.merriampark.com/ld.htm}, 2010.

[13] Emerick, Chas, "Levenshtein Distance Algorithm: Java Implementation", {HYPERLINK: http://www.merriampark.com/ldjava.htm}, 2010.