Software tools for automatic music performance

Jonas Masko, Jonathan Fischer Friberg and Anders Friberg

KTH Royal Institute of Technology
School of Computer Science and Communication
Speech, Music and Hearing
Stockholm

Abstract. Two new computer programs are presented for the purpose of facilitating new research in music performance modeling. Director Musices (DM) is a new an implementation of the user interface for the KTH rule system. It includes a new integrated player and several other improvements. The automatic sampler facilitates the sampling of a MIDI-controlled instrument, such as the Disklavier piano from Yamaha. Both programs are open source, cross-platform written in Java and in the case of DM including also different lisp dialects.

1 Introduction

Music performance modeling has been an active research direction for a rather long, in particular since the introduction of the personal computer in the 70's. The field is evolving using available computer tools and methods in artificial intelligence and computer science. In this paper, we will present two different computer programs that can contribute and facilitate further research in music performance.

The work at KTH started with a rule system for music performance run on a minicomputer and using a rule based programming language developed for speech synthesis (Sundberg et al., 1983). This system was later redesigned into a new environment for rule-based performance rules research that was implemented in Lisp (Friberg & Sundberg, 1986). The rule system program Director Musices has been continuously developed and supported and additional rules has been added (see overview in Friberg et al., 2006, new accent rules see Bisesi et al, 2011, Parncutt et al., 2013). While the basic rule system written in standard common lisp was easy to maintain during this long period, the graphical user interface (GUI) had to be reimplemented regularly (e.g. Friberg et al., 2000). Once again there was a need for updating the system in particular regarding cross-platform compatibility which resulted in the new version described below.

The output data of the programs for modeling music performance is often in terms of MIDI, that is, symbolic play instructions. In order to make a convincing performance, a performance program needs to be connected to a model of an acoustical instrument. There has been a considerable improvement in instrument synthesis towards realistic instrument models, in particular for the acoustic piano (e.g. Bank et al.,

Workshop Proceedings of IAS-13
13th Intl. Conf. on Intelligent Autonomous Systems
Padova (Italy) July 15-19, 2014
ISBN 978-88-95872-06-3 pp. 537-544

2010). However, most of the commercial products for music production are focused on contemporary popular music. Therefore, it is still hard to find for example a sampled piano that can express details of articulation used in performance of classical music.

The introduction of the computer-controlled grand piano, such as the Disklavier by Yamaha was an important step forward in making a realistic rendering of performance models. This has been used for example in the RENCON competition for computer-generated piano performance[1] (Katayose et al., 2012, Canazza et al., 2013).

However, the MIDI specification is not specifying explicitly the key velocity of each note with respect to some reference. Therefore it is up to each manufacturer of a MIDI-controlled instrument to implement the mapping between the input MIDI data and the resulting dynamic level. This implies that a performance model that has been developed using one MIDI instrument will not be performed properly on another instrument. This is handled in Director Musices by implementing a mapping that is specific for each synthesizer and then specify dynamics internally in terms of sound level in decibels (Bresin et al., 2002).

In the RENCON competition 2013 in Stockholm this type of calibration data was published in advance for the participants[2]. The data was collected by measuring the response of the particular Disklavier model used in the competition.

An even better solution would be to have a realistic sampling of the particular instrument used in the competition so that each participant can listen to a realistic sampling of the instrument that actually responds also in the same way regarding dynamics. This was the main purpose for developing the automatic sampler described below.

2 Director Musices

This new implementation of Director Musices (DM) is a cross-platform, open-source program containing a several improvements. One addition is the integrated MIDI player which is also more flexible and includes a moving cursor over the music score. Now it is also possible to edit the rule palettes graphically. Also included is a new tentative input format in terms of the ABC notation.

2.1 Implementation

The previous version of Director Musices (DM) was implemented for the Windows platform using the commercial development system Allegro Common Lisp. Although it can be compiled for other operating systems this required in this case a reprogramming of the whole user interface. Therefore, we decided to move to an entirely new environment. This new implementation is based entirely on open-source software. The umbrella environment is Java which makes it cross-platform compatible without any recompiling of the code. It only requires a Java runtime installed in

[1] http://renconmusic.org/

[2] see http://smac2013.renconmusic.org/midi-calibration/

the computer which is more or less standard on all platforms. The core of the rule system is written in common lisp and the original code is now run in armed bear common lisp (ABCL)[3]. The user interface is programmed in Clojure[4] using the Java Swing library, see figure 1. In order to facilitate rule development without the necessity to recompile the whole system, the program can use an external rule library that is loaded interactively.

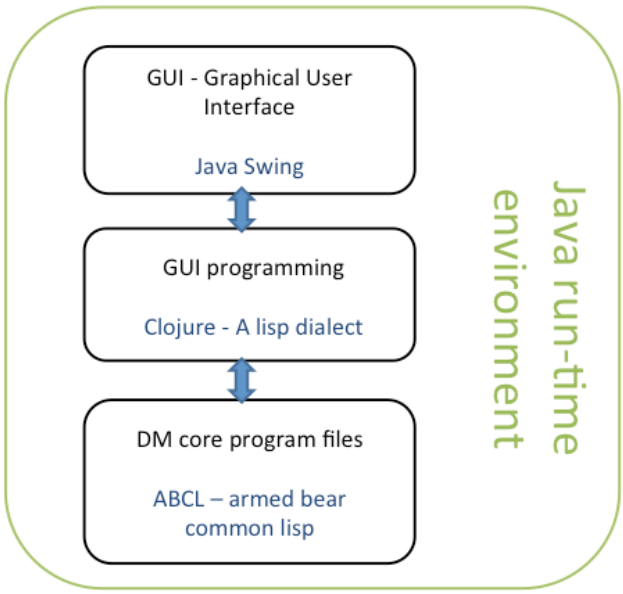

Figure 1. The different software components in the Director Musices implementation.

2.2 User interface

The user interface consists of one main window divided in different vertical panes, see Figure 2. The different pane are:

The MIDI player (top pane). It includes selection of MIDI output (otherwise missing in the Windows system environment).

The track (second pane). It sets basic properties of each track, such as MIDI channel, overall volume and panning. It is also possible to chose the dynamics mapping for different MIDI instruments as discussed above.

The score (third pane) includes music notation (shown here is the Mozart A major Sonata), optional graphical displays of performance parameters (in this case the rela-

[3] abcl.org
[4] clojure.org

tive IOI variation) for each track, display of phrase analysis in different colors above the score, and manual editing of any note parameter (not shown in figure).

The rule palette (bottom pane) is where the different performance rules can be applied. Several palettes can be loaded at the same time for comparison with different settings. It is also fully editable.

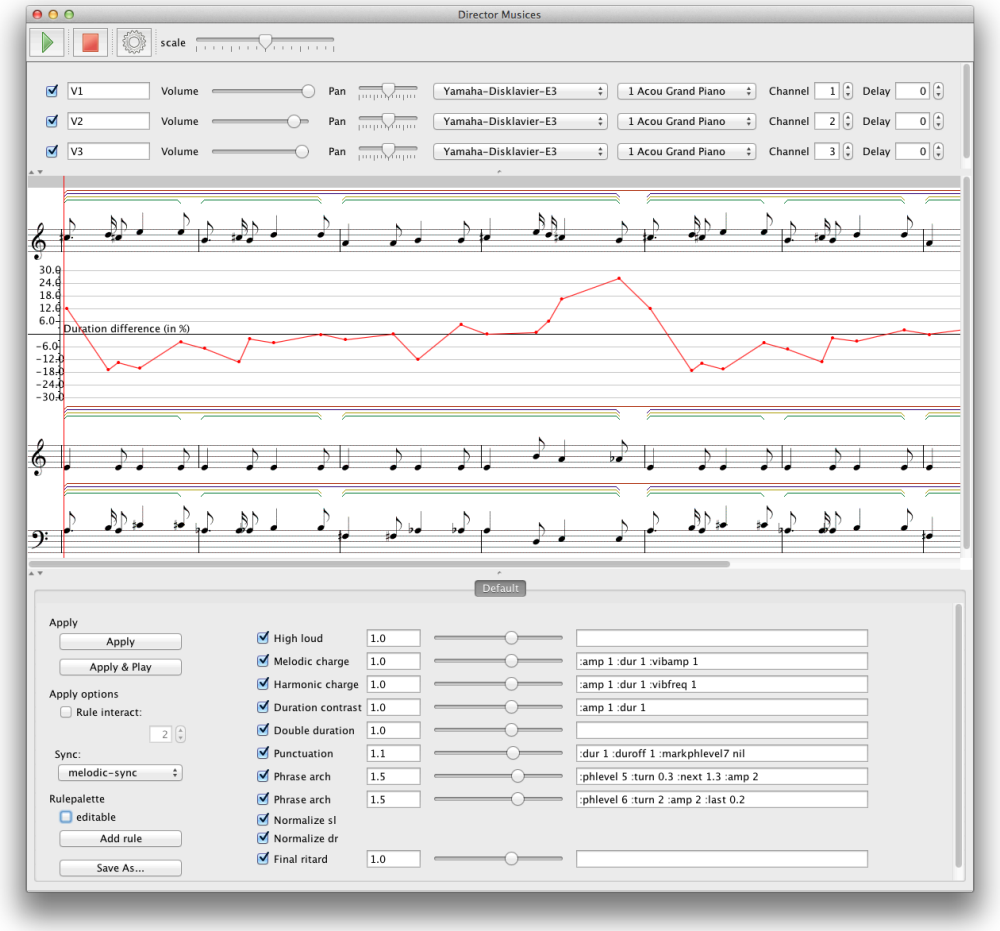

Figure 2. A screen shot of the main window of Director Musices.

3 Automatic MIDI Instrument Sampler

This is an application for automatically collect samples from MIDI-controlled instruments. The samples are saved into audio files intended to be used in the Kontakt software sampler from Native Instruments or similar software. The application is written in Java and is currently in an alpha state.

The entire thought behind this application is to automatize as much as possible the otherwise time-consuming task of sampling an acoustic instrument. Any instrument which can handle MIDI-input can be computer controlled. Usually keyboards/pianos are thought of when it comes to MIDI instruments – but MIDI support for other instruments are both available and under development/improvement.

The application can in its current state automatically collect samples from a MIDI controlled instrument. It collects all samples within a given note range, at specified velocity/velocities, and saves this into wave audio file/files.

It can automatically detect silence, allowing to record no more than what is needed. This saves a great amount of time and memory when collecting many samples. Also, maximum/minimum record time for each sample as well the total record time can be specified.

After setting the parameters and checking that the recording works the whole procedure is fully automatic. Thus, the system can be left unattended. For a fine resolution of dynamic levels the whole recording procedure will typically take several hours.

3.1 Requirements

- Java runtime installed on the machine, minimum version is not verified, but it's verified to work in 1.7.x.
- MIDI out (software MIDI will also work)
- Audio input from sound card or internally
- A microphone (software recording will also work)
- Sampler software for playing the recorded samples. It is tested using Kontakt by Native instruments. Any other sampler will work but may require more time for transferring the samples and setting up the player.

3.2 Limitations

This application is in an alpha state, and is specifically designed to play notes on and record whatever sound a MIDI-controlled instrument produces. Technically it will work on any MIDI controlled instrument, but a piano has very few variables when it comes to input; namely what note and at what velocity. Should a more complex instrument be sampled, say a MIDI-controlled violin, variables such as modulation (for the hand movement) must also be accounted for, which drastically increase the amount of combinations which can be recorded.

The application is also specifically designed to generate recordings that can easily be imported in the Kontakt sampler from Native Instruments. This means that one continuous audio file is generated per velocity level which covers the entire note range. Kontakt is designed to automatically split the notes and map these files to a virtual keyboard.

Workshop Proceedings of IAS-13
13th Intl. Conf. on Intelligent Autonomous Systems
Padova (Italy) July 15-19, 2014
ISBN 978-88-95872-06-3 pp. 537-544

3.3 Technical description

The application structure consists of a midi interface, an audio processor/recorder and a graphical user interface (see Figure 3). As of the alpha version, these three parts are hard coded together and cannot be separated (Figure 4) – in future releases we will strive to make this application into more of a library than a standalone application.

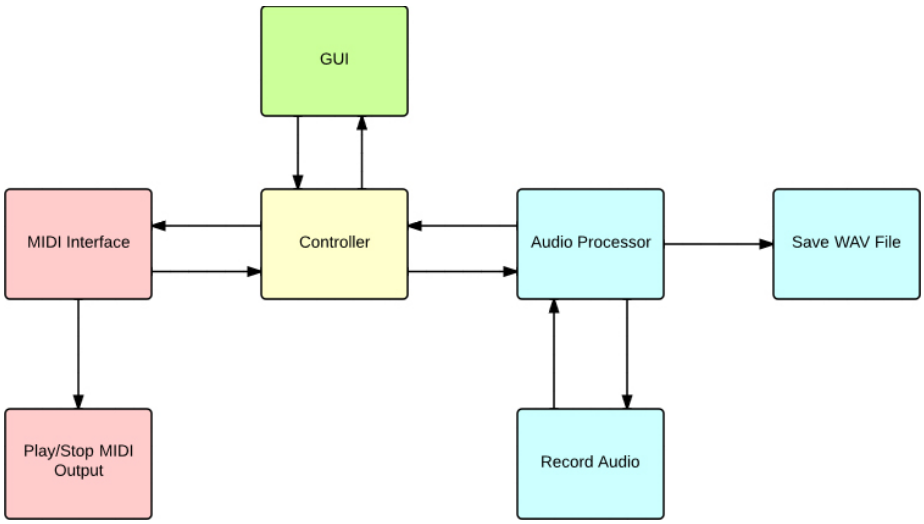

Figure 3. The software structure of the automatic sampler.

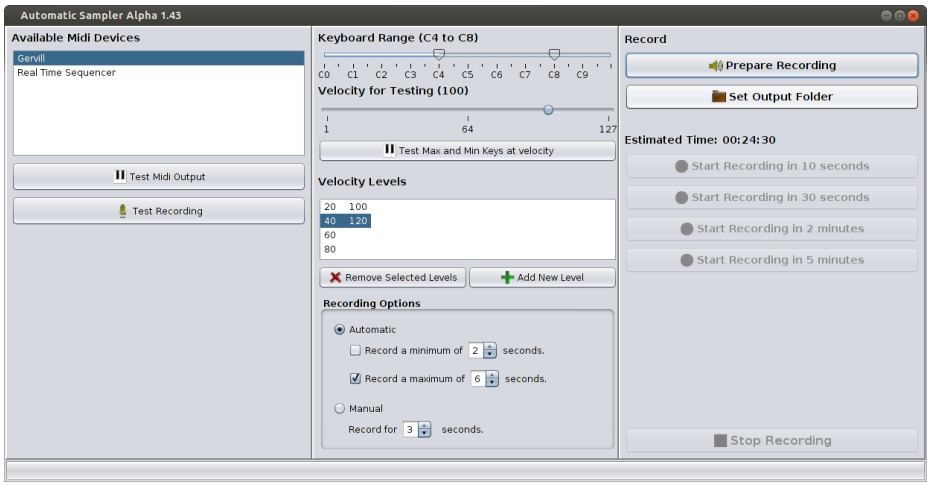

Figure 4. The graphical user interface of the automatic sampler (alpha release).

3.4 Future development

The application records every single note within the given range. It is time consuming to record large note spans, especially when it's sufficient to record only a few notes. Adding support to sample instruments which use modulation and other midi variables is a very interesting feature. This will not have any effect on the current target – the grand piano – but will allow complex recording of other MIDI-controlled instruments. Control on how to save the samples can also be implemented, so that it would be easy to also use the recordings in other applications than the Kontakt sampler currently targeted.

4 Conclusions

While doing research in music performance models a great deal of time is devoted to technical aspects such as inputting the score and integrate the performance models with existing instrument models. We present here two different programs that are purposely designed to simplify this process. There are several improvements possible for future development, as outlined above. However, the presented software can in its present state be used in research about music performance. The new Director Musices program has already been used in recent developments of the KTH rule system (e.g. Parncutt et al., 2013) including participation in the Rencon 2013 competition. The automatic sampler will potentially be useful in particular for the Rencon participants. It would then be possible to sample the piano used in the competition so that the development of the performance systems can be tested using a realistic model of the actual piano used. Both program are or will be released as open source. Director Musices can be downloaded at http://www.speech.kth.se/music/performance/download/. The automatic sampler will be available at the same address.

References

Bank, B., Zambon, S. & Fontana, F. (2010). A Modal-Based Real-Time Piano Synthesizer. *IEEE Trans. on Audio, Speech and Language Processing*, 18(4), 809-821.

Bisesi, E., Parncutt, R., & Friberg, A. (2011). An accent-based approach to performance rendering: Music theory meets music psychology. In International Symposium on Performance Science (ISPS 2011) (pp. 27-32).

Bresin, R., Friberg, A., & Sundberg, J. (2002). Director musices: The KTH performance rules system. In Proceedings of SIGMUS-46 (pp. 43-48). Kyoto: Information Processing Society of Japan,.

Canazza, S., De Poli, G., & Rodà A. (2013). How do people assess computer generated expressive performances? Proc. Sound and Music Computing Conference (SMC13), 353-359

Friberg, A., Bresin, R., & Sundberg, J. (2006). Overview of the KTH rule system for musical performance. Advances in Cognitive Psychology, Special Issue on Music Performance, 2(2-3), 145-161.

Friberg, A., Colombo, V., Frydén, L., & Sundberg, J. (2000). Generating Musical Performances with Director Musices. Computer Music Journal, 24(3), 23-29.

Friberg, A., & Sundberg, J. (1986). A Lisp Environment for Creating and Applying Rules for Musical Performance. In Proceedings of the International Computer Music Conference 1986 (pp. 1-3).

Katayose, H., Hashida, M., De Poli, G., & Hirata, K. (2012). On Evaluating Systems for Generating Expressive Music Performance: the Rencon Experience. *Journal of New Music Research*, *41*(4), 299-310.

Parncutt, R., Bisesi, E., & Friberg, A. (2013). A Preliminary Computational Model of Immanent Accent Salience in Tonal Music. In Proceedings of the Sound and Music Computing Conference 2013, SMC 2013, Stockholm, Sweden (pp. 335-340).

Sundberg, J., Askenfelt, A. & Frydén, L. (1983). Musical performance: A synthesis-by-rule approach. Computer Music Journal, 7, 37-43.

Workshop Proceedings of IAS-13
13th Intl. Conf. on Intelligent Autonomous Systems
Padova (Italy) July 15-19, 2014
ISBN 978-88-95872-06-3 pp. 537-544