Director Musices: The KTH Performance Rules System

Roberto Bresin, Anders Friberg, Johan Sundberg

Department of Speech, Music and Hearing
Royal Institute of Technology - KTH, Stockholm
email: {roberto, andersf, pjohan}@speech.kth.se

Abstract

Director Musices is a program that transforms notated scores into musical performances. It implements the performance rules emerging from research projects at the Royal Institute of Technology (KTH). Rules in the program model performance aspects such as phrasing, articulation, and intonation, and they operate on performance variables such as tone, inter-onset duration, amplitude, and pitch. By manipulating rule parameters, the user can act as a metaperformer controlling different feature of the performance, leaving the technical execution to the computer. Different interpretations of the same piece can easily be obtained. Features of Director Musices include MIDI file input and output, rule palettes, graphical display of all performance variables (along with the notation), and user-defined performance rules. The program is implemented in Common Lisp and is available free as a stand-alone application both for Macintosh and Windows platforms. Further information, including music examples, publications, and the program itself, is located online at http://www.speech.kth.se/music/performance.

This paper is a revised and updated version of a previous paper published in the Computer Music Journal in year 2000 that was mainly written by Anders Friberg (Friberg, Colombo, Frydén and Sundberg, 2000).

1 Performance Rules

Performance rules previously presented in several articles (e.g. Sundberg, 1988; Friberg, 1991; Friberg, Frydén, Bodin and Sundberg, 1991; Sundberg, 1993; Friberg, 1995; Friberg, 1995; Friberg, Bresin, Frydén and Sundberg, 1998; Friberg and Sundberg, 1999; Bresin and Friberg, 2000; Bresin, 2001) constitute the core of Director Musices. They are used to modify the nominal values of various performance variables, such as duration and amplitude, as shown in Figure 1. Most of the rules have a global quantity parameter *k* (whose default value is 1) regulating the magnitude of all modifications caused by that rule. Further adjustment of rule effects can be attained by additional rule parameters. The selection of rules, k-values and rule parameter values can drastically change the performance and many different but still musically acceptable performances can be obtained. An overview of the current rule system is given in Table 1. Previous implementations of a subset of the rules are the Windows program MELODIA (Bresin, 1993), and Japer (Bresin and Friberg, 1997), a Java program available on the World Wide Web.

One of the goals of our performance research has been to find rules that are independent of musical style and which corresponds to basic performance principles used by musicians. The rules can be divided in three categories according to their apparent communicative purpose (Sundberg, 1999): (1) *grouping rules* that mark boundaries between smaller and larger tone groups (e.g. *Punctuation* rule and *Phrase arch* rule), (2) *differentiation rules* that increase differences between categories (e.g. *Duration contrast* rule and *High loud* rule), (3) *ensemble rules* for the interaction between musicians in an ensemble (e.g. *Ensemble swing* rule and *Melodic sync* rule). Thus, the rules are mainly related to basic aspects of performance such as simply marking the structure. Yet, by rule selection and by adjusting rule parameters the rules can create performances that differ in emotional quality, e.g, “happy” or “sad” (Bresin and Friberg, 2000). Another recent development is the GERM model (Juslin, Friberg and Bresin, In press) combining four different performance rule types: Generative (described above), Emotional, Random variations and associated Motion.

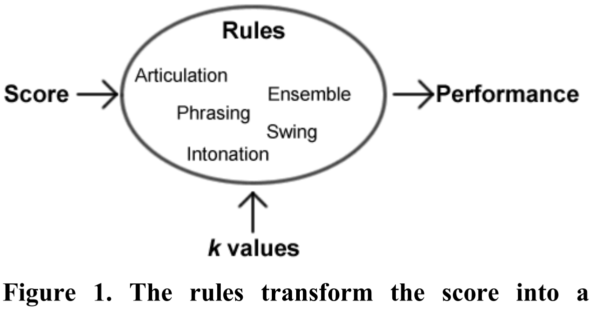


Figure 1. The rules transform the score into a performance according to the rule parameters (*k* values).

Table 1. Most of the rules in Director Musices, showing the the affected performance variables (sl = sound level, dr= interonset duration, dro= offset to onset duration, va= vibrato amplitude, dc= cent deviation from equal temperament in cents)

Marking Pitch Context

Rule Name	*Performance Variables*	*Short Description*
High-loud	sl	The higher the pitch, the louder
Melodic-charge	sl dr va	Emphasis on notes remote from current chord
Harmonic-charge	sl dr	Emphasis on chords remote from current key
Chromatic-charge	dr sl	Emphasis on notes closer in pitch; primarily used for atonal music
Faster-uphill	dr	Decrease duration for notes in uphill motion
Leap-tone-duration	dr	Shorten first note of an up-leap and lengthen first note of a down-leap
Leap-articulation-dro	dro	Micropauses in leaps
Repetition-articulation-dro	dro	Micropauses in tone repetitions

Marking Duration and Meter Context

Rule Name	*Performance Variables*	*Short Description*
Duration-contrast	dr sl	The longer the note, the longer and louder; and the shorter the note, the shorter and softer
Duration-contrast-art	dro	The shorter the note, the longer the micropause
Score-legato-art	dro	Notes marked legato in scores are played with duration overlapping with interonset duration of next note; resulting onset to offset duration is dr+dro
Score-staccato-art	dro	Notes marked staccato in scores are played with micropause; resulting onset to offset duration is dr-dro
Double-duration	dr	Decrease duration contrast for two notes with duration relation 2:1
Social-duration-care	dr	Increase duration for extremely short notes
Inegales	dr	Long-short patterns of consecutive eighth notes; also called swing eighth notes
Ensemble-swing	dr	Model different timing and swing ratios in an ensemble proportional to tempo
Offbeat-sl	sl	Increase sound level at offbeats

Intonation

Rule Name	*Performance Variables*	*Short Description*
High-sharp	dc	The higher the pitch, the sharper
Mixed-intonation	dc	Ensemble intonation combining both melodic and harmonic intonation
Harmonic-intonation	dc	Beat-free intonation of chords relative to root
Melodic-intonation	dc	Close to Pythagorean tuning, e.g., with sharp leading tones

Phrasing

Rule Name	*Performance Variables*	*Short Description*
Punctuation	dr dro	Automatically locates small tone groups and marks them with lengthening of last note and a following micropause
Phrase-articulation	dro dr	Micropauses after phrase and subphrase boundaries, and lengthening of last note in phrases
Phrase-arch	dr sl	Each phrase performed with arch-like tempo curve: starting slow, faster in middle, and ritardando towards end; sound level is coupled so that slow tempo corresponds to low sound level
Final-ritard	dr	Ritardando at end of piece, modeled from stopping runners

Synchronization

Rule Name	*Performance Variables*	*Short Description*
Melodic-sync	dr	Generates new track consisting of all tone onsets in all tracks; at simultaneous onsets, note with maximum melodic charge is selected; all rules applied on this sync track, and resulting durations are transferred back to original tracks
Bar-sync	dr	Synchronize tracks on each bar line

2 Input and Output

Director Musices supports three music formats: (1) *scores*, a simple text-based custom format, (2) *performances*, similar to the score format but with added performance variables; and (3) MIDI files. Normally a new score is entered by an external score editor and then transferred to Director Musices as a MIDI file. The MIDI file reader converts any MIDI file to an internal score object, keeping note durations and assigning a note value to each note for the music notation. The assigned note value has no influence on the performance since the rules operate on the real durations. This means that the rules can be applied also to MIDI performances. As each track is basically assumed to contain one voice only, simultaneous notes in the same track are truncated at a new note onset, thus creating a track suitable for rule application. Key velocities are currently disregarded. Harmonic and phrase analysis, needed by some rules, as well as other score variables can be inserted directly in Director Musices.

Decibel to MIDI velocity conversion

In Director Musices, deviations of intensity level for each note are calculated in decibels (dB). The mapping between dB values and MIDI velocity and MIDI volume is not a linear relation and varies with synthesizers. In particular the relation between dB and MIDI velocity is polynomial of the 3rd degree. For these reasons conversion functions are needed for each synthesizer used in the reproduction of performances produced by Director Musices. In Figure 2 conversion functions from dB to MIDI velocity for five sample-based musical instruments and two sound card synthesizers are presented. All curves are normalized so that 0 dB corresponds to MIDI velocity 64. The behaviour of all instruments is almost the for MIDI velocity in a range between 64 and 90. For MIDI velocity values lower than 64 and higher than 90, synthesizers can perform significantly different. For instance a value of –15 dB can corresponds to a MIDI velocity between 18 and 35. Therefore, in order to have a more correct reproduction of performances, for each track of the music score, users must choose which synthesizer to use from the pull down menu *Synth* (see Figure 4).

3 Score representation

The representation of the score in Director Musices is straightforward, similar to that of a MIDI file. A score object contains a list of track objects which in turn contains a list of segments. Each track corresponds to one melodic part and a segment generally corresponds to one note or one chord (a chord is any number of simultaneous notes sharing the same performance variables). The segment object contains all score and performance variables. The performance variables (except durations) can vary over time by assigning a *time-shape* object, typically in the form of a break-point and an interpolation function. The time-shape can be dynamically coupled to a note or phrase chunk. Thus, when the duration of a note is changed, the time-shape of this note is scaled accordingly.

The performance variables are expressed in physical measures such as duration in milliseconds and sound level in decibels. The translation to MIDI variables is made in a synthesizer object, one for each track, making the rule effects independent of the synthesizer used.

Although, the performance has mostly been realized in terms of MIDI, other output representations such as Csound can easily be added. There is a tool for exporting the performance data to a spreadsheet.

4 Rule Definition

Most rules require a context. This may consist of a sequence of tones, each with properties such as pitch, interonset duration, or harmonic analysis, etc. Some rules operate on metrical context and some on both vertical (harmonic) and horizontal (melodic) contexts. This context framework was crucial to the choice of score representation and tools for formulating rules.

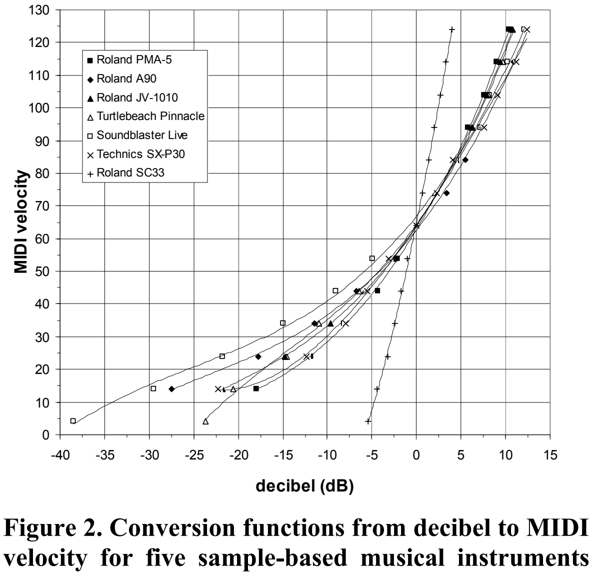


Figure 2. Conversion functions from decibel to MIDI velocity for five sample-based musical instruments and two sound cards (Soundblaster Live and TurtleBeach Pinnacle). The functions are normalized so that 0 dB corresponds to MIDI velocity 64. The curves interpolating the measured values are implemented in Director Musices.

Instead of a complex data structure describing the music, we chose a simple data structure complemented by flexible dynamic viewpoints, i.e., rules can 'look' at the score at different hierarchical levels and in different chunks. For example, instead of notes a track can contain a list of voice segments, each corresponding to a phoneme, such that a note consists of one or several segments. A rule can be applied both at the segment or note level, allowing pronunciation rules to work at the segment level and, at the same time, performance rules at the note level. The performance rules will simply 'see' the track as consisting of a sequence of notes and all accesses to performance variables are the same as for an instrumental track. The different viewpoints are dynamically allocated when a rule is applied, allowing even rule-based selection of chunks. Other typical viewpoint selections are phrases, measures and chord progressions.

Rules are written in Common Lisp syntax. Predefined functions help rule development and all standard functions in Common Lisp are available. Some examples of functions and rules will be given below and in Figure 3.

Rule Top-level Definition

Rules are defined by the normal lisp defun special function

```
(defun <rulename> (<k parameter>
<additional key parameters>) <body>)
```

This function defines a rule with the main rule parameter `k`. Additional parameters are specified using key parameters.

Serial Sequencing Functions

These special functions (lisp macros) will step through the score in chunks as specified by each macro and are used within the body of a rule definition. The macro

```
(each-note-if <conditions> (then <body>))
```

iterates over each note and track of the score and evaluates `<body>` if all conditions are met. Within the body, access functions are used for note variables. The macro

```
(each-segment-if <conditions> (then
<body>))
```

is the same as above, but for segments; has the same function as each-note-if, provided the track is a mono-track. For a voice-track this macro works at a lower level, each segment corresponding to a voice segment or a phoneme. The macro

```
(each-group
  '<group begin condition>
  '<group end condition>
  <body>)
```

first creates a new track consisting of segment group objects (chunks) as specified by the `begin` and `end` conditions and then evaluates `<body>` for each group.

Serial Access Functions

Within the body of the sequencing macros, these functions are used for accessing the variables in each chunk. They are also used for defining contexts. A slowly changing variable (time-shape object) can be applied over an entire chunk.

The macro for given by

```
(this <variable>)
(next <variable>)
(prev <variable>)
```

returns the specified variable of the current, next, or previous chunk, while

```
(set-this <variable> <value>)
(set-next <variable> <value>)
(set-prev <variable> <value>)
```

assigns the specified value to the variable in the current, next, or previous chunk.

5 User Interface

Figure 4 shows the main windows in the Windows version of Director Musices. The track variables of the score are shown in the second window from the top. Here basic features such as track volume or MIDI program number can be edited.

A performance is defined by selecting rules and rule parameters in a rule palette window. Rule effects are additive, i.e., if a rule is applied twice, the change of the performance variables will be twice as large. Several rule palette windows can be open at the same time, thus allowing easy comparison of different performances. All performance variables can be shown graphically together with the music notation, see Figure 4. The time axis can either be real time or score time.

In addition, the Windows version contains an editable score window where all variables can be edited and displayed along with the music notation. This facilitates the adding of extra information to the score, such as phrase markers.

Rule palettes

In Director Musices, rules can be organized in so called rule palettes (see Figure 4). These can be saved for use in future working sessions. Rule palettes are stored in text files that can be easily edited, i.e. it is possible to add or delete rules. In some cases, such as

```
(defun phrase-rule (k)          ;a complete rule for lengthening notes
  (each-note-if                 ;before phrase-start markers
    (not (last?))
    (next 'phrase-start)
    (then
      (add-this 'dr (* 40 k))  ;40 ms lengthening if k=1
      )))

(each-track                     ;a rule fragment that increases the duration
  (set-this-dr                  ; for the whole track with 20 %
    (* (this-dr) 1.2) ))

(each-note-if                   ;process this note if:
  (< (this ‘dr) 500)            ;it is shorter than 500 ms
  (> (this ‘f0) (prev ‘f0))     ;and if the pitch is higher than previous
  (then
      ...

(each-group                     ;process phrase by phrase
  ‘(this ‘phrase-start)         ;group beginning
  ‘ (or (last?) (next 'phrase-start)) ;group end or the last chunk
  (then
    (set-this                          ;increase the sound level with an
      ‘sl                              ;envelope over the phrase
      (make-time-shape ...
```

Figure 3. Examples of performance rules.

in the GERM model (Juslin, Friberg and Bresin, In press), it is desirable to make use of several rule palettes at the same time. In the GERM model there are four rule palettes: one for each of the four compoments of the model, (1) Generative grammar, (2) Emotion, (3) Random deviations, (4) Motion component. Rules are applied by using the buttons “Init & Apply” for the first rule palette and the button “Apply” for the remaining rule palettes. In this way the effects produced by each rule palette are added to those produced by previous rule palettes.

6 Links

Further information about the Director Musices program can be found at:
http://www.speech.kth.se/music/performance

7 Acknowledgments

This paper is a revised and updated version of a previous paper published in the Computer Music Journal in year 2000 that was mainly written by Anders Friberg (Friberg, Colombo, Frydén and Sundberg, 2000). Lars Frydén, Johan Sundberg, and Anders Friberg developed most of the rules. Roberto Bresin contributed the articulation rules. Roberto Bresin and Anders Friberg developed the macro-rules for emotional performance. AF wrote most of the kernel code and the Macintosh version. VC developed most of the user interface code for Windows. The project was supported by The Bank of Sweden Tercentenary Foundation.

The authors would like to thank the organizers of RENCON 2002 for inviting Roberto Bresin and for making this paper possible.

References

Bresin, R. (1993). MELODIA: a program for performance rules testing, teaching, and piano score performance. *In Proceedings of X Colloquio di Informatica Musicale*, Milano, 325-327.

Bresin, R. (2001). Articulation rules for automatic music performance. *In Proceedings of International Computer Music Conference - ICMC2001*, Havana, San Francisco: International Computer Music Association, 294-297.

Bresin, R. and A. Friberg (1997). A multimedia environment for interactive music performance. *In Proceedings of Proceedings of KANSEI - The Technology of Emotion AIMI International Workshop*, Genova, 64-67.

Bresin, R. and A. Friberg (2000). “Emotional Coloring of Computer-Controlled Music Performances.” *Computer Music Journal*, **24**(4): 44-63.

Friberg, A. (1991). “Generative Rules for Music Performance: A Formal Description of a Rule System.” *Computer Music Journal*, **15**(2): 56- 71.

Friberg, A. (1995). Matching the rule parameters of Phrase arch to performances of Träumerei: A preliminary study. *In Proceedings of KTH symposium on Grammars for music performance*, Stockholm, KTH, 37-44.

Friberg, A. (1995). *A Quantitative Rule System for Musical Performance*. doctoral dissertation, Speech Music and Hearing. Stockholm, KTH., http://www.speech.kth.se/music/publications/thesisaf/sammfa2nd.htm.

Friberg, A., R. Bresin, L. Frydén and J. Sundberg (1998). “Musical punctuation on the microlevel: Automatic identification and performance of small melodic units." *Journal of New Music Research*, **27**(3): 271-292.

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

Friberg, A., L. Frydén, L.-G. Bodin and J. Sundberg (1991). "Performance Rules for Computer-Controlled Contemporary Keyboard Music." *Computer Music Journal*, **15**(2): 49-55.

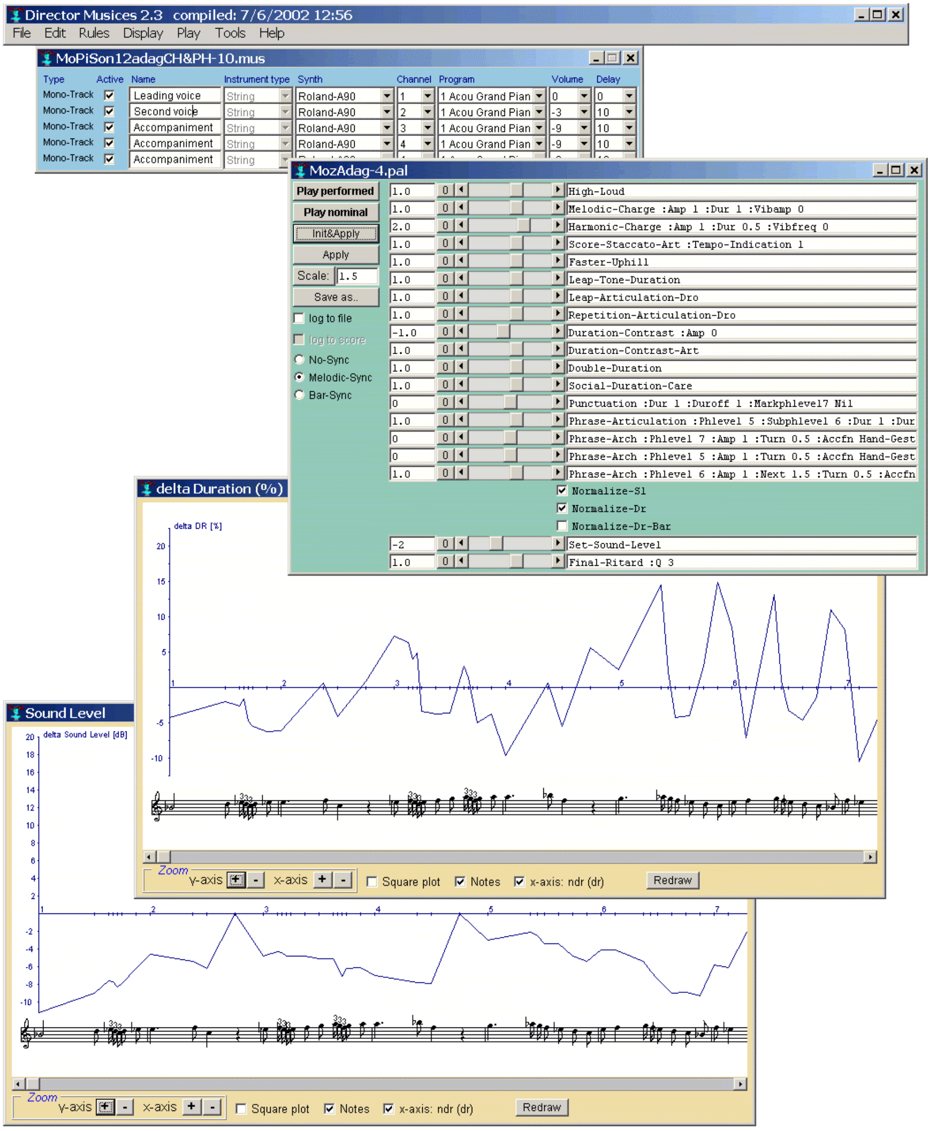
Friberg, A. and J. Sundberg (1999). "Does music performance allude to locomotion? A model of final ritardandi derived from measurements of stopping runners." *Journal of Acoustical Society of America*, **105**(3): 1469-1484.

Juslin, P. N., A. Friberg and R. Bresin (In press). "Toward a computational model of expression in performance: The GERM model." *Musicae Scientiae*.

Sundberg, J. (1988). Computer synthesis of music performance. *Generative Processes in Music*. J. A. Sloboda. New York, Oxford University Press**:** 52-69.

Sundberg, J. (1993). "How can music be expressive?" *Speech Communication*, **13**: 239-253.

Sundberg, J. (1999). Cognitive Aspects of Music Performance. *Music and signs, Semiotic and Cognitive Studies in Music*. I. Zannos. Bratislava, ASCO Art and Science**:** 219-230.


Director Musices 2.3 compiled: 7/6/2002 12:56
File
Edit
Rules
Display
Play
Tools
Help
MoPiSon12adagCH&PH-10.mus

Type	Active	Name	Instrument type	Synth	Channel	Program	Volume	Delay
Mono-Track	☑	Leading voice	String	Roland-A90	1	1 Acou Grand Pian	0	0
Mono-Track	☑	Second voice	String	Roland-A90	2	1 Acou Grand Pian	-3	10
Mono-Track	☑	Accompaniment	String	Roland-A90	3	1 Acou Grand Pian	-9	10
Mono-Track	☑	Accompaniment	String	Roland-A90	4	1 Acou Grand Pian	-9	10
Mono-Track	☑	Accompaniment	String					

Scale: 1.5

□ log to file

□ log to score

○ No-Sync

● Melodic-Sync

○ Bar-Sync

1.0	0	High-Loud
1.0	0	Melodic-Charge :Amp 1 :Dur 1 :Vibamp 0
2.0	0	Harmonic-Charge :Amp 1 :Dur 0.5 :Vibfreq 0
1.0	0	Score-Staccato-Art :Tempo-Indication 1
1.0	0	Faster-Uphill
1.0	0	Leap-Tone-Duration
1.0	0	Leap-Articulation-Dro
1.0	0	Repetition-Articulation-Dro
-1.0	0	Duration-Contrast :Amp 0
1.0	0	Duration-Contrast-Art
1.0	0	Double-Duration
1.0	0	Social-Duration-Care
0	0	Punctuation :Dur 1 :Duroff 1 :Markphlevel7 Nil
1.0	0	Phrase-Articulation :Phlevel 5 :Subphlevel 6 :Dur 1 :Dur
0	0	Phrase-Arch :Phlevel 7 :Amp 1 :Turn 0.5 :Accfn Hand-Gest
0	0	Phrase-Arch :Phlevel 5 :Amp 1 :Turn 0.5 :Accfn Hand-Gest
1.0	0	Phrase-Arch :Phlevel 6 :Amp 1 :Next 1.5 :Turn 0.5 :Accfn

☑ Normalize-Sl

☑ Normalize-Dr

□ Normalize-Dr-Bar

-2	0	Set-Sound-Level
1.0	0	Final-Ritard :Q 3

Figure 4. Screen shot of Director Musices showing from top to bottom the main window, a track window, a rule palette window and on the bottom the graphs of the duration deviations and sound level deviations resulting from the application of the rules.