Abstract

NoteAbility™ is a comprehensive music notation editor which has been under development for about 6 years. NoteAbility was designed to support the graphical flexibility required by composers of contemporary music as well to provide all the standard music editing procedures such as transposition, part extraction and page reformatting. NoteAbility is able to do this because it includes graphical and non-graphical classes of most music images. NoteAbility is ideally suited to the computer music community because it can represent a wide range of score types (including complex graphical scores) and because it can be used in conjunction with many of the music applications commonly used by composers of computer music.

1. Background

Donald Byrd [Bryd, 1986] stated that music notation editors could be categorized by the degree to which they were Semantic-oriented versus Graphics-oriented. Those that are concerned primarily with the semantics and the structure of music (i.e. Semantic-oriented programs) often struggle with the difficulty of including non-standard music symbols, of representing graphical scores, and of allowing the user to contradict the conventions of Common Music Notation (CMN). Those that are designed with a strong graphics orientation allow graphical flexibility, but may have difficulty performing certain music editing and page formatting functions such as transposition, part extraction and automatic wrapping of measures. Examples of programs that are Semantic-oriented are Finale™ and Encore™, and those that are graphics-oriented are NoteWriter™ [Hamel, 1989] and Score™ [Smith, 1973].

NoteAbility™ [Hamel, 1997] was designed to support both models; it has a well-defined music structure and a knowledge of the standard conventions of music notation, yet it allows a high degree of graphical flexibility and it is easy to contradict or override the syntax of CMN.

NoteAbility was originally developed for NeXTStep computers using Objective C. It was later ported to OpenStep and is currently being ported to Rhapsody. In addition, a scaled-down version of NoteAbility is being written in C++ for Macintosh and Windows platforms.

2. Overview and Design Philosophy of NoteAbility

Some of NoteAbility’s design came from the fact that it was initially developed under NeXTStep; it used display PostScript, and therefore could handle complex graphics easily and with a high degree of precision, it supported multiple pasteboard types and inter-application messaging, and was constructed with an advanced Graphical User Interface (GUI) which include dragging and dropping of icons, user-extensibility and on-line help. Although most of these features are now available on other computer systems, they have been part of the NeXTStep operating system for many years.

In terms of the overall GUI, I adopted the paradigm of a composer working on manuscript paper, so NoteAbility provides the user with blank pages of score paper and allows the composer the flexibility and freedom that would be available if she were working on paper; she can start anywhere on the page and enter images in any order desired. I chose this layout because I felt that was most natural for a composer and I recognize that when you fundamentally change the mode of interaction or the tools used to perform a task, you risk changing the results of the task. My goal was to provide tools to assist composers doing what they already know how to do, not to alter or restrict what they do.

3. Underlying Music Structure

In order to support both the Semantic-oriented and Graphics-oriented models, NoteAbility borrowed the notion that exists in some graphics programs of having multiple layers. The background layer of NoteAbility consists of the Systems, Staves, Measures, Beats and Sub-beats. This layer is always drawn first, and represents the underlying music structure. (Of course, these images may or may not be visible, but they are present none-the-less.) A Page
consists of one or more Systems, each of which consists of a two dimensional grid where beat positions (within measures) form the x-axis and staves (or to be more precise - location relative to the staves) form the y-axis. All images which appear in foreground layers are linked to the background by two associations: every image has at least one beat position in a measure, and every image is located some distance from at least one staff in the system. These associations are essential for changing page or system layout and for cutting and pasting within documents and between documents. While some images have only one beat position and staff association, others (eg. crescendo marks) have two and some (eg. slurs) have three. When the underlying music structure is altered (by manual or automatic adjustment of the beat or staff positions), the surface images are re-aligned to their associated staff and redrawn at their corresponding beat positions. The figure below shows a collection of music images along with their main associations. Although the positions of the Measures, Beats and Staves are different in the two music examples, the image data (and their associations) remain the same.

<table>
<thead>
<tr>
<th>Image</th>
<th>Measure : BeatPos</th>
<th>StaffAssociation : StepsFromStaff</th>
</tr>
</thead>
<tbody>
<tr>
<td>GraceNote (16th)</td>
<td>1 : 0.219</td>
<td>1 : 1</td>
</tr>
<tr>
<td>Slur</td>
<td>1 : 0.241</td>
<td>1 : 0</td>
</tr>
<tr>
<td>GraceNote (16th)</td>
<td>1 : 0.369</td>
<td>1 : 6</td>
</tr>
<tr>
<td>PPP</td>
<td>1 : 0.045</td>
<td>1 : -5</td>
</tr>
<tr>
<td>GraceNote (16th)</td>
<td>1 : 0.549</td>
<td>1 : 7</td>
</tr>
<tr>
<td>GraceNote (16th)</td>
<td>1 : 0.695</td>
<td>1 : 10</td>
</tr>
<tr>
<td>Text (q = 69...)</td>
<td>1 : 0.886</td>
<td>1 : 18</td>
</tr>
<tr>
<td>Piano</td>
<td>1 : 1.034</td>
<td>1 : 14</td>
</tr>
<tr>
<td>Half note</td>
<td>1 : 1.0</td>
<td>1 : 7</td>
</tr>
<tr>
<td>P</td>
<td>2 : 0.884</td>
<td>1 : -5</td>
</tr>
<tr>
<td>Half note</td>
<td>2 : 1.0</td>
<td>1 : 7</td>
</tr>
</tbody>
</table>

Figure. 1 - Music Images with Beat Positions and Staff Associations (Hamel - "I remember...")

4. Graphical Flexibility

In order to accommodate the diverse notational requirements of composers, NoteAbility has been designed to be as flexible as possible. Any music image can be any size, any colour, and most of its attributes can be easily altered. The size and attributes of an image are independent of any images (i.e. a staff or other notes in a chord) that it is associated with or connected to.

Figure. 2. Graphical Flexibility of Music Images (Ravel - "Le Jardin féerique")
To further support graphical flexibility, Structural and Graphical classes of certain images are available. Structural Notes and Rests are associated with relatively simple beat positions in the measure, and although they can be offset from these beat positions, they cannot be freely moved around the page. Graphical Notes and Rests inherit all the attributes of Structural Notes and Rests, yet they can be placed anywhere on the page regardless of the resulting music syntax. In the figure below, the first two measure contain Structural Notes, Rests and Beams and the second two contain Graphical versions of the same images.

In cases where the complexity of the score makes using Structural images awkward, the composer can simply use Graphical versions of the images (which can freely be beamed or associated to Structural images if needed.) In the figure below, Graphical and Structural images have been mixed to create a spatially-notated score.

5. Import and Export Options

NoteAbility is ideally suited to the computer music community because it can represent a wide range of score types and because it can be used in conjunction with many of the music applications commonly used by composers of computer music. Currently, NoteAbility can export scores or score fragments in Standard MIDI format [Hewlett, 1997], in NeXT score format [Jaffe, 1989], as MAX qlists [Puckette, 1986], as MAX explode objects, as Csound scores [Vercoe, 1994], in GUIDO Music Notation format [Hoos, 1997] as well as all standard graphics formats. In many cases, it is possible to copy and paste between NoteAbility and other applications using these formats. A composer using NoteAbility, can select a portion of her score, copy it and paste it directly into a text editor as GUIDO, as a MAX qlist, as a Csound score file, as EPS, or as TIFF. It is also possible to paste GUIDO text strings directly into NoteAbility and have the corresponding music images appear. I chose GUIDO over NIFF [Grande, 1996] as my default interchange format because GUIDO is a text-based (and human-readable) format, because it allows simple music to be represented in a simple and compact way, and because it was easy to implement and to integrate into NoteAbility. In the example below, GUIDO code was copied from a standard text editor and pasted directly into NoteAbility.

% example GUIDO code

\[ \{ \{ \text{\texttt{%\{ \text{\texttt{\% example GUIDO code}}}}\}}\]
6. Future Extensions and Conclusion

Extensions to NoteAbility currently in progress include the conversion of optical scanning files from the OMR system developed by Ichiro Fujinaga [Fujinaga, 1992] and further support for GUIDO Music Notation format. NoteAbility has been designed with the professional composer in mind; it provides a flexible graphical environment along with musical intelligence and sophisticated editing abilities. The program has now been under development for about 6 years, and as it continues to evolve, the feedback and suggestions from composers and other musicians is always welcome.

7. References

For more information on NoteAbility, see http://debussy.music.ubc.ca/~opus1/welcome.html.