

SOFTWARE FOR TEACHING MUSIC THEORY

Heinrich Taube, William Andrew Burnson

University of Illinois at Champaign-Urbana
Department of Music Composition and Theory, United States
taube@illinois.edu

ABSTRACT

We discuss a software development project named *Chorale Composer* that seeks to create a cohesive student environment for learning techniques in harmonic analysis and counterpoint. Through the use of sophisticated four-part algorithmic analysis, student-input music is immediately scanned for anomalous voice leading and chord construction. The algorithms employed are rigorously model-based and do not employ heuristic patterns or statistical guessing. The analytical engine can detect and generate figured bass, sonority, and roman numeral analyses, and can identify keys (and therefore modulations) by examining leading-tone and bass line resolutions. The software can also be used to analyze existing literature such as the Bach Chorales and to detect stylistic points of interest where the music appears to deviate from the pedagogical “rules of counterpoint.” The software is ideal for classroom discussion and practice. It can also assign generative homework exercises to test student comprehension of relevant concepts. The software aims to supplement college-level tonal music theory curricula.

1. INTRODUCTION

Chorale Composer¹ is an interface for teaching four-part tonal music theory. As an analytical reference tool, it contains nearly all of Bach’s 371 Chorales with options for displaying various types of analysis. The ability to rapidly browse these chorales is a useful pedagogical tool for both students in the classroom and at home. Students can complete generated exercises which are automatically annotated for corrections by the algorithmic analysis engine that accompanies the interface. Additionally when reference literature is edited using the built-in editor, the music is reanalyzed in tandem so that instructors or students can view real-time consequences of various compositional alternatives. Instructors may also choose to allow whether the students see their mistakes as they do each exercise (guided completion), or whether they are graded after (in an exam scenario, for example).

¹To download the latest release of Chorale Composer, go to <http://camil.music.uiuc.edu/> and click on the link to Software, then to Chorale Composer. Finally, select an appropriate distribution.

The interface takes a different approach to computer-based music editing in that ease of use is paramount to the success of the software in the classroom. Thus, the interaction is simple and the display minimalist. All manipulation of the score is done using the mouse.

Several C++ source code libraries comprise the environment. The windowing environment is provided by JUCE,² a cross-platform and open source object-oriented GUI shell. The notational display is achieved through *Belle, Bonne, Sage*,³ encoding of musical objects through MENC (*Music ENCoding*);⁴ and the analysis through a proprietary library *Music Theory Workbench*.^[3] These last three libraries were developed specifically for their practical application in Chorale Composer, but they each have potentials outside of the present environment.

Chorale Composer uses two different licenses: one version uses the proprietary analytical engine and is suitable for using in classroom settings ideally as replacement to paper-based workbooks, and the other version contains all the same editing features and musical literature, but does not contain any means to automatically analyze the music. The latter version is licensed under the GNU General Public License.⁵

2. INTERFACE

2.1. Navigation

Score navigation is accomplished by directly manipulating the score with the mouse. Instead of using scroll bars and cumbersome zoom tools, users click on the score to move the pages around. For example, to focus on a different part of the score, the user clicks and drags the score to the desired location. Zooming in and out is accomplished by double clicking and moving the mouse up (to zoom in) or down (to zoom out). Chorale Composer will also optimize the viewing area to ensure that the maximum amount of score is viewable on the screen.

²See: <http://rawmaterialsoftware.com/juce>

³See: <http://bellebonnesage.sf.net/>

⁴See: <http://camil.music.uiuc.edu/>

⁵See: <http://www.gnu.org/licenses/gpl.html>

2.2. Editing

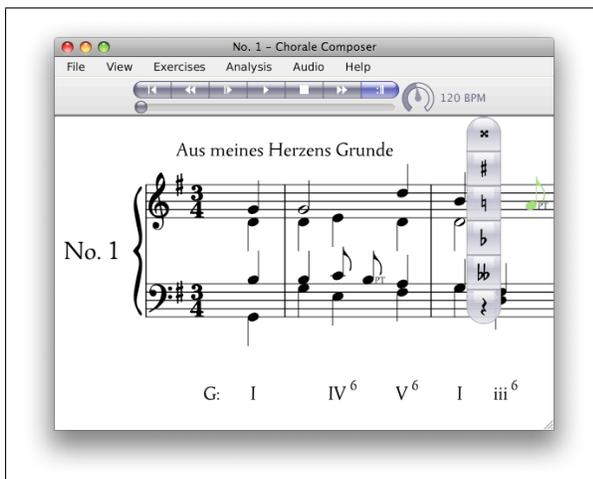


Figure 1. A typical session in Chorale Composer.

Figure 1 shows the first Bach chorale with an analysis present below and a few non-chord tone labels for the passing tones (abbreviated PT). The tool palette is shown because the user has just clicked on a note in order to change it accidental.

Score editing is accomplished by dragging notes to the desired pitches. After a note is dragged the tool palette appears, giving the user the option to change the note's accidental. The tool palette disappears after a selection has been made, or when the user has moved the cursor away from the palette.

If the rest button on the tool palette is pressed, then the note turns into a rest. The user can change the rest back into a note by dragging the rest to a desired pitch.

Chorale Composer is unique in that it treats tied notes as single entities. Notes consisting of several tied subordinate notes are considered the same notational object so that changing the pitch of one will cause the tied noteheads to follow. At the time of writing, rhythmic values are fixed to their original values from the source literature, but they will become mutable like accidentals in future releases.

2.3. Playback

Chorale Composer can play back scores using any one of the operating system's enumerated MIDI devices. Future releases will alternatively support synthesized instruments in case the user has poor or inconvenient MIDI support. The system employs a transport toolbar at the top of each window also allowing the user to change the tempo and skip the repeat.

3. ANALYSIS

3.1. Real-Time Analysis

Chorale Composer includes a real-time four-part analysis engine. The analysis is practically instantaneous—typically an entire Bach Chorale in under the time it takes to compute the resulting layout and draw it to the screen. By tightly integrating the analysis with the display, it was not necessary to make the feature a separate tool.

The analysis engine can also generate figured bass and sonority types, as well as identify non-chord tones as well as stylistic departures inconsistent with the rules of contrapuntal part-writing.⁶ When the algorithm encounters a chord that is not a part of its model, the chord is skipped.

At the time of writing, several unresolved issues of analysis and roman numeral display are to be addressed. Since the analysis model uses tonicizations (tonic-dominant relationships) to detect changes in key, secondary functions are currently displayed as brief modulations. Also, there are cases when it is difficult to accurately place the moment of modulation. Pivot chords are generally latent in this model, meaning that a modulation is not confirmed until it is no longer possible to account for a chord in the original key.⁷

Complete the missing lines adhering to the chord progressions written below.

Simple Example

Ex. 1

g i V⁶ i vii^{o6} i⁶ i⁶ V i

Medium Example

Ex. 2

a: V₂⁴ i⁶ vii^{o7} i vii^{o6} i⁶ iv⁶ V

d: v⁶

The image shows two examples of four-part writing assignments. Example 1 is labeled 'Simple Example' and 'Ex. 1'. It shows a musical staff with a treble clef and a bass clef, with a key signature of one flat and a common time signature. The melody is in the treble clef, and the bass line is in the bass clef. Below the staff, the chord progression is given as: g i V⁶ i vii^{o6} i⁶ i⁶ V i. Example 2 is labeled 'Medium Example' and 'Ex. 2'. It shows a similar musical staff setup. Below the staff, the chord progression is given as: a: V₂⁴ i⁶ vii^{o7} i vii^{o6} i⁶ iv⁶ V. Below this, there is a 'd:' section with a chord: v⁶.

Figure 2. An example of a homework assignment with two four-part writing assignments.

⁶For more information pertaining to the algorithms used by the analysis engine, see [3]. For a summary of previous attempts to analyze tonal music see [1].

⁷This can lead to conflict scenarios. Consider moving from the key of B minor (two sharps) to the key of A major (three sharps). A IV⁷ chord followed by a ii chord omitting the fifth in B minor strongly suggests A major; however, due to the lack of an A in the ii chord, there is no leading-tone resolution to confirm the tonic-dominant relationship. The resulting modulation is delayed until a leading-tone is detected even though to many ears the modulation would be perceived in between the IV⁷ and ii chord.

3.2. Literature

Chorale Composer incorporates a database of most of the 371 Bach chorales[2] within the environment for quick reference. Any of these can be edited and played. When the user makes edits to these scores, the analysis updates simultaneously to reflect the musical changes. This feature is a useful resource for instructors and students who would like to experiment with different part-writing alternatives and perhaps even to speculate on the logic governing Bach's compositional choices.

3.3. Homework Generation

The software uses built-in templates that generate homework assignments in pseudo-random keys for students. The main four-part writing exercises are derived from selections of particular chorales and have one or more voices stripped away for the student to fill in (see Figure 2). The analysis and grading engine does not expect the student to match Bach's exact compositional signature, rather it checks to make sure that the student's work is consistent with Bach's style (i.e. student obeys voice leading rules and constructs sensible chords).

Another homework template creates a page of different kinds of cadences and asks the user to compose them each in a particular key. The student clicks on the staff to create notes for each of the four voices. When the student is satisfied, he or she can press a key to have Chorale Composer examine the completed exercises for errors.

In Figure 3 we see an example of an exercise with corrections. For the first chord the user indeed constructs a second inversion V chord, but forgets to spell the 7th. In this case, the analysis engine reports below the roman numeral that the quality is incorrect. In the second chord, we see that the user forgets the 5th scale degree in the B-flat major chord. Since the root, quality, and inversion are correct in spite of the unusual chord spelling the mistake is instead deemed an infraction against part-writing conventions, and so the analysis engine prints a remark above the staff.

Though the exercises for both types of assignments are generated in a pseudo-random key, instructors and students can turn on a classroom synchronization mode that allows students to follow along with a global exercise in the same key.

4. FURTHER CAPABILITIES

4.1. PDF Save and JPEG Export

Chorale Composer does not use a standalone proprietary format to read and write score files. Instead, the information required to reproduce the score within the program is embedded as metadata within PDF files that identically render what is seen in the application window.

Ex. 2

Inverted dominant

Bb: V_3^4 I^6
Quality?

5th?

Figure 3. An example of an annotated cadential exercise.

Since the software uses a device-independent graphics layer, all drawing commands are abstract and serve to relay generalized vector-graphics instructions to renderers for each output medium. For example, when Chorale Composer saves a PDF file, it switches its drawing context to output to a PDF instead of to the screen. By carefully tuning the rendering process for each of the output media, there are no significant graphical differences between the application window and its resulting PDF.

Chorale Composer adheres to a strict subset of PDF—the PDF/X-1a:2001 specification⁸—nominally guaranteeing that the file will remain printable to any PostScript printer (whether physical or virtual). The feature serves to enhance the quality of the output and the long-term portability of the file, as well as to limit the consequences of file format deprecation should older versions of files become unreadable by newer versions of the editing environment.

Since music notation programs are often needed to make supplemental examples for more complex documents such as handouts, Chorale Composer includes a feature that allows for the user to select an area on the page for export to a print-resolution (300 dpi) JPEG. To do this, the user clicks and holds, waits for the cursor to become a cross-hair, and then drags over the desired region. When a suitable region has been selected, the user can then drag-and-drop the area either to his or her desktop or to an external application such as a web browser or office application.

⁸PDF/X-1a:2001 refers to the ISO 15930-1 standard. To read more about what the standard is and how it differs from PDF, read http://planetpdf.com/planetpdf/pdfs/stully_pdfx_underhood.pdf

4.2. Rendering

Chorale Composer uses a sophisticated device-independent graphics library called *Belle, Bonne, Sage* (developed by the authors) for all rendering.

All drawing commands are converted to paths including lines and text. Using paths allows for greater flexibility and quality in the display. For example, all corners are automatically rounded in order to emulate the natural ink bleed in hand-engraved music.

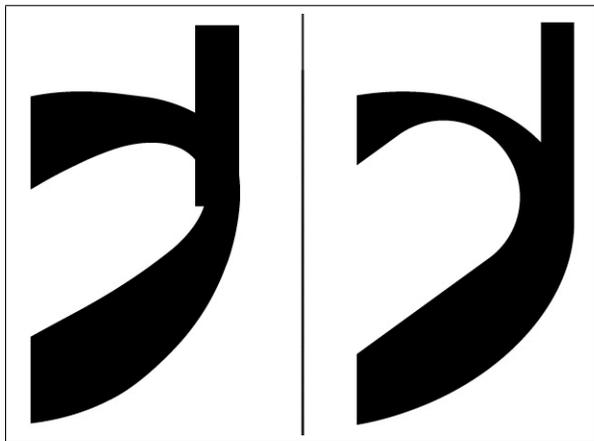


Figure 4. Rendering comparison of half notes in Finale 2008 (left) and Chorale Composer (right).

Many notation programs rely on music fonts for even trivial matters of score geometry. When fonts are combined with other vector or line-based drawing commands, idiosyncrasies may arise. In Figure 4 we see the unexpected result of combining a line with a half note glyph. In the case of Finale,⁹ the stem is so thick that it protrudes into the elliptical center. Chorale Composer solves this problem by calculating the join in such a way that a single path can be used to describe the outer hull of both the notehead and its stem.

Belle, Bonne, Sage differs from preexisting music notation libraries such as Lilypond in that it makes no prior assumptions about the underlying structure of the music or the context in which the music is to be displayed. Lilypond, which is a batch music score generation tool, reads scripts that generate PostScript files (which can then be converted to PDF). Unfortunately, it is not possible to create a GUI that utilizes Lilypond.¹⁰ *Belle, Bonne, Sage* is a more general vector-graphics library placing special focus on music notation, structural flexibility, and real-time interfacing.

⁹Finale is copyright MakeMusic Inc. 2009. The version used to produce the image above was *Finale 2008*, <http://finalemusic.com>, (Eden Prairie, MN: MakeMusic Inc., 2008).

¹⁰However, there are at least a few open source notation programs that can export Lilypond files: *Denemo*, *Rosegarden*, and *NoteEdit*.

4.3. Cross-Platform Accessibility

Chorale Composer is supported on Linux, Mac OS X, and Windows. In addition, the software does not require installation; instead it uses embedded resources within the executable file, including documentation, fonts, and music literature. Since it does not rely on external files, it is not only more stable, but it can be run in a wider range of permissions situations, such as networked university computers, which do not typically permit software installation by end-users.

In one possible usage scenario, a student would watch an instructor demonstrate part-writing concepts with Chorale Composer and then be able to go home or into a lab to complete exercises or experiment with four-part writing on their own. By ensuring cross-platform support, using PDFs to represent score files, and presenting a user-friendly interface to the student, instructors can rapidly teach music theory concepts through the supporting literature.

5. FUTURE WORK

Chorale Composer is in active development and will be expanding in the months to come. Improvements that have been planned are: (1) ability to modify rhythmic values; (2) exporting scores to other file types such as MIDI and MusicXML;¹¹ (3) exporting graphics to other vector-based file formats such as EPS (Encapsulated PostScript) and SVG (Scalable Vector Graphics); (4) ability for instructors to create HTML webpages to explain various music theory concepts with accompanying examples for distribution online; (5) ability for instructors to create homework assignments from scratch based off of more flexible templates; and, (6) creating a server version of the application which can be used to grade and store the results of completed homework assignments.

6. REFERENCES

- [1] J. M. I. Plcido R. Illescas, David Rizo, “Harmonic, melodic, and functional automatic analysis,” in *Proceedings of the 2007 International Computer Music Conference*, 2007, pp. 165–168.
- [2] A. Riemenschneider, *371 Harmonized Chorales and 69 Chorale Melodies with Figured Bass*. New York: G. Shirmer, 1941.
- [3] H. Taube, “Automatic tonal analysis: Toward the implementation of a Music Theory Workbench,” *Computer Music Journal*, vol. 23, no. 4, pp. 18–32, 1999.

¹¹MusicXML is copyright Recordare LLC 2009. For more information about the MusicXML music notation standard, see: <http://www.recordare.com/>