1 *Corpus Mensurablis Musice "Electronicum"*: Toward a Flexible Electronic Representation of Music in Mensural Notation

Theodor Dumitrescu

Jesus College
University of Oxford
Oxford OX1 3DW
UK
theodor.dumitrescu@music.oxford.ac.uk

Abstract

The field of early music editing is beginning to embrace the possibilities offered by computer technology, but may be doing so in ways which compromise the usefulness of musical data for future research and analytical purposes. Adopting the view that 15th- and 16th-century mensural notation is different in significant enough ways from later notational styles that it needs to be transcribed with its own dedicated system, this paper explores the means for designing a new system to make electronic early music editions. The beginnings of a sample system with an accompanying piece of viewing software are presented.

COMPUTING IN MUSICOLOGY 12 (1999-2000), 3–18.

One of the areas of publishing which has traditionally been fraught with debate and controversy is early music editing. The degree of translation involved in converting early sources into modern notation inevitably raises issues of historical accuracy, authenticity, and performance practice for the scholar attempting to compile editions which must serve the needs of modern performers in practice while simultaneously providing clear and accurate readings to facilitate analysis.

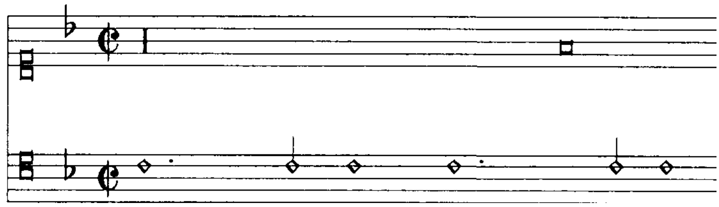
Questions of text underlay, unnotated accidentals, and modern barring, for example, give rise to heated debates in scholarly journals, while performers complain about having to sing from scores with abnormal pitch levels or unaccustomed rhythmic values and barring styles. The focus of many efforts in the reform of editorial policy, to try to cater at the same time to performers and scholars in the creation of diplomatic editions, has led to many and constant compromises for both purposes.

A short excerpt from Pierre de la Rue's motet *Vexilla regis–Passio domini* (Example 1 a-e)[1] can be used to show some of the striking dissimilarities between different transcriptions of a piece of music, as well as the dissimilarity of the original notation from each.

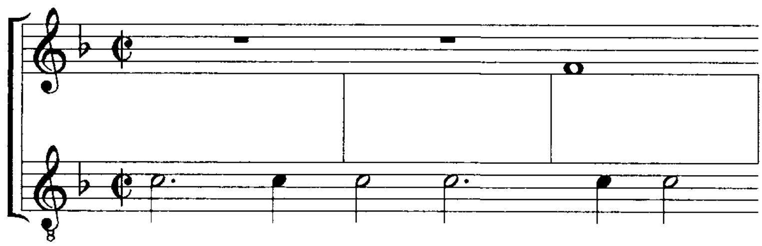
[1] This example is based on a similar one from Bent (1994).

(a)

(b)



(c)



(d)

(e)

Example 1. Five versions of the opening of Pierre de la Rue, *Vexilla regis–Passio domini*: opening of discantus (a) and contratenor (b) in Brussels, Bibliothèque Royale de Belgique, MS 228, ff. 29^{v} and 30^{r}; (c) discantus and contratenor in their original notation put into score; (d, e) discantus and contratenor transcribed into two different styles of modern score.

The further one delves into the problem of creating a *single* diplomatic edition which can serve universal needs, the more insurmountable it becomes. Beyond the issues just described, there are also practical matters of publishing, such as how to create a collected edition of a composer's works from which it is still economically viable for a large choir to perform. Skinner (1998) touches upon these stumbling blocks in a review of the latest volume of the Byrd Edition: "Perhaps in this day of computer-set music scores, Stainer & Bell might consider issuing offprints from the Byrd Edition as practical performing editions (perhaps even offering the music at a variety of pitches, as is the case, for example, with some of Jon Dixon's commendable JOED publications), so that the full measure of Byrd's genius can be enjoyed by more of us."

Perhaps in this day of computer-set music scores, on the other hand, the solution is not for publishers and editors to account for increasingly more minute variations by publishing variant editions at great labor and expense, but instead to look to the advantages of creating electronic editions.

1.1 Existing Formats

A number of formats already exist which have been used or modified to represent music originally conceived in mensural notation. These include both graphic and symbolic representations, where a *graphic* representation is a format which depends essentially on pictures of musical notation, while a *symbolic* representation is based on interpretation of the notation. Any dividing line between these two styles is fairly arbitrary, and there is no reason that one system should have to limit itself to one representation format, but existing systems can be classified fairly easily on one side or the other. A few systems use graphic representations, while the majority use some form of symbolic representation. A large number of existing representation formats are described in Selfridge-Field (1997). Each system has its own approach and goals for the problem of representing music on a computer, and an examination of the limitations of a few systems can reveal some of the difficulties in designing a usable, balanced representation for early music.

1.1.1 Early Music Editions System

[2] *ludwig.gold.ac.uk/RWibberley/*

Created by Roger Wibberley, the Web site for Early Music Editions[2] allows users to view .GIF image files of modern editions of Tallis and Wesley pieces, one page at a time (Example 2). Essentially, this graphical representation is about as close as one can get to reproducing a printed edition on a computer, with many of the limitations of the standard book (separate pages, textual fixity), and the main advantage being easy wide distribution. The system suffers from the limitations of its graphic format, which eliminate the possibility of customizable display, searching and automated analysis, and any number of other possibilities of symbolic representations.

Example 2. First four bars of Tallis's "Salvator mundi" (1575), as presented on the Early Music Editions Web site. The edition is stored and accessed as a series of image files. (Example edited by Roger Wibberley, (c) 1996. Used by permission.)

1.1.2 MusiXTEX

One of the most effective systems for producing professional-looking scores of music from many centuries is MusiXTEX, a set of macros for the TEX typesetting system (popular with computer scientists). The symbol set includes characters for early notations, allowing one to produce close transcriptions of chant and mensural notation. The problem which this representation poses for analytical purposes stems from its source in typography; the system is more concerned with placing particular symbols on a page than with musical data. As a result, the internal representation is bound closely to the visual display, and the data representing a piece in mensural notation is different from the data representing that piece in modern note values, and so on with any change of visual characteristics.

1.1.3 DARMS

DARMS (Digital Alternate Representation of Musical Scores) is a system which balances external and internal representation elements very well, and has been extended in the past for representing several styles of early notation.

A modified version of DARMS (the "A-R dialect") has been used for many years for data input at A-R Editions, which has produced many modern editions of 15th- and 16th-century music. The representation ends at the level of modern notation, however; a scholar performing research involving the meaning of mensuration signs (or many other aspects of mensural notation) would be unable to use these DARMS files, where the signs have been translated into modern time signatures.

Lynn Trowbridge has also modified DARMS to produce the *Linear Music Input Language* for Renaissance music. This might seem to be the ideal solution for encoding music in mensural notation, but the emphasis of standard DARMS on later music shows some weaknesses for this purpose. To reuse the example of mensuration signs: Trowbridge's system, in dealing with DARMS Meter Codes, must represent the mensurations C2, ₵, and Ɔ all with the same internal code, though these symbols by no means necessarily have the same meaning (nor is the meaning of a mensuration sign necessarily consistent across all instances).[3]

[3] A standard introduction to issues involved in the interpretation of mensuration signs is Busse Berger (1993).

Without wishing to dismiss the usefulness of current systems for representing early music, I would suggest that a newly designed representation format may be the best solution to the problem of creating a sufficiently flexible framework which is also useful for current scholarship. By creating a system with a very particular range of input and a certain set of analytic goals, one can avoid many of the pitfalls inherent in trying to use a system designed for different goals, or one designed to be all-encompassing.

1.2 Designing a New System

How does one go about designing a robust, comprehensive representation for early music which will serve the needs of many kinds of users? In my opinion, the starting point for such a system ought to be the original notation itself. By founding the system in the musical language of its repertory, we remove a layer of translation which is always present in a representation based on our modern notation system. This is not to deny that every transcription contains some element of translation, but there is certainly a difference in level.

For demonstration's sake, let us consider the process of translating from a modern edition back to the original notation. For most editions, it is simply impossible to reconstruct a source accurately from its modern translation, whether due to unspecified reductions and transpositions, ambiguities of mensurations and ligature types, or any number of features affected by transcription.

The reader with the early notation, on the other hand, can always construct a modern edition out of it. So the original notation, for our purposes, more often yields a richer data set than a translation into modern notation. There is no reason that a system designed solely for the study of early music should have to deal with that music only in terms of a notation suited to a different type of music, nor is it any more difficult for a computer to speak the language of semibreves and prolation than it is for that computer to deal with half notes and triplets.

Once we have decided that our representation format is to be built upon the notational system of the music it represents, we can create a vocabulary and grammar for the system. It is imperative, as with any software system's data representation format, to make these elements as clearly defined and documented as possible.

The grammar is defined formally through a set of terminal and non-terminal productions, to describe a hierarchy of symbols. The exact electronic representation of the symbol set is not a part of the grammar, being dealt with instead at the earlier lexical analysis stage. It is for this reason that the exact form of the stored individual data symbols is irrelevant to the usefulness of the representation format; whether, for example, one uses a text-based file format, a Unicode-type numeric symbol set, or a binary file at the first stage of decoding the data, the lexical analyzer will translate it into an abstract symbol stream for the parser. Each symbol format may have advantages or disadvantages compared to the others (such as direct comprehensibility by a human reader), but in terms of ability to represent our grammar, the formats are orthogonal.

Another issue which must be addressed when creating a new grammar is its possible interface with other representation formats. As the short-comings of earlier, "universal" music representation formats become more apparent to scholars interested in specific areas of research, the creation of many individual systems with very particular aims may be accompanied by the design of "interchange languages." Whether these conversion formats actually make use of data files or exist only at a software level, they look much more promising as a means for covering a large variety of musical styles in electronic representations than the old pipe dream of creating one all-encompassing format to represent every type of music imaginable.

With that said, my own approach is actually to avoid designing a new system to fit with existing ones. Rather, the goal in the initial design stages is to create a format suited as comfortably as possible to the music it must represent, without any compromises and concessions to other notations and styles. When it comes time for the translation from one language to another, the amount of loss in translation will inevitably be a function of the scope of the interchange language and the intersection space of the two languages.

1.3 The CMME Format

Example 3 presents a formal grammar for the alpha-stage CMME system (currently capable of representing a small subset of 16th-century music in mensural notation), as designed with the previous guidelines in mind.

Example 3. Preliminary CMME formal grammar.

piece	→	*generaldata voicedata*
generaldata	→	*title composer*
title	→	**TITLE id**
composer	→	**COMPOSER id**
voicedata	→	*numvoices voicelist*
numvoices	→	**NUMVOICES int**
voicelist	→	*voicelist voice*
voicelist	→	*voice*
voice	→	*voicegeneraldata eventdata*
voicegeneraldata	→	*voicenum name*
voicenum	→	**VOICE int :**
name	→	**NAME id**
eventdata	→	**EVENTDATA :** *eventlist*
eventlist	→	*eventlist event*
eventlist	→	*event*
event	→	*clef*
event	→	*mensuration*
event	→	*rest*

event	→	*note*
event	→	*dot*
event	→	*sharp*
event	→	*custos*
event	→	*lineend*
clef	→	**CLEF NOTELETTER int / PITCH ;**
mensuration	→	**MENSURATION** *mensattriblist* **;**
mensattriblist	→	*mensattriblist* **,** *mensattrib*
mensattriblist	→	*mensattrib*
mensattrib	→	**MENSSIGN**
mensattrib	→	**STROKE**
mensattrib	→	**int**
rest	→	**REST** *restattribs* **;**
restattribs	→	**NOTETYPE ,** *proportion* **, int . int**
note	→	**NOTE** *noteattribs* **;**
noteattribs	→	**NOTETYPE ,** *proportion* **, PITCH**
noteattribs	→	**NOTETYPE ,** *proportion* **, PITCH ,** *noteoptionlist*
noteoptionlist	→	*noteoptionlist* **,** *noteoption*
noteoptionlist	→	*noteoption*
noteoption	→	**LIGATED**
noteoption	→	**COLORED**
dot	→	**DOT int**
sharp	→	**SHARP PITCH**
custos	→	**CUSTOS PITCH**
lineend	→	**LINEEND**
proportion	→	**int / int**

In order to demonstrate the data organization format imposed by this grammar, Example 4 contains a listing for a CMME transcription of Example 1(a, b), using a text-based format for the symbol set.

Example 4. CMME transcription of Example 1 (a, b).

```
(* Test file for system prototype *)

(* General data section *)
TITLE    "Vexilla regis-Passio domini"
COMPOSER "Pierre de la Rue"
NUMVOICES 2

(* Voice data section *)

VOICE 1:
  NAME "[Discantus]"

  EVENTDATA:
    CLEF        C1/C3;
    CLEF        B7/B4;
    MENSURATION C,|;
```

```
    REST L,2/1,2.2;
    NOTE B,1/1,F3;

VOICE 2:
  NAME "C[ontratenor]"

  EVENTDATA:
    CLEF        C7/C3;
    CLEF        B6/B3;
    MENSURATION C,|;

    NOTE SB,3/4,C3;
    DOT 4
    NOTE m,1/4,C3;
    NOTE SB,1/2,C3;
    NOTE SB,3/4,C3;
    DOT 4
    NOTE m,1/4,C3;
    NOTE SB,1/2,C3;
```

It may seem surprising at first, in light of comments above on visual representation, to find that the CMME data representation system is in fact very visually based. In this regard, the system is first and foremost a transcription of manuscript pages: the data must be able to provide a visual description of the transcribed page even more than it provides an interpretation.

This distinction can be viewed as a separation of the data format into two layers: a primary *visual layer* which provides direct description of manuscript contents, and a secondary *editorial layer* which provides a particular interpretation of the contents. It is certainly easier to make such a clear division in theory than it is in practice, as is borne out by the ambiguity of many elements of the system as to whether they are description or interpretation.

The first division of the file format is into a general data section and a voice data section. The general data provides information about the piece which is not necessarily contained in the transcribed manuscript (for example, the ascription of *Vexilla regis–Passio domini* to Pierre de la Rue does not exist in the transcription's source, Brussels 228, but is borne out by concordances).

The voice data section contains a list of data sets, with one set for each voice of the polyphonic piece. Voices are divided similarly to the overall file structure, with a general data section and an event list. General data in this case comes from the manuscript page itself, but obviously the system does not provide a mechanism for separating manuscript and editorial general voice data at the current point (see the editorial expansions of voice names in brackets in Example 4).

The bulk of the average CMME file is taken up by the event lists of individual voices. In the event list for a voice, "events" are considered to be the symbols and features on the manuscript staves (and sometimes outside). The event list for any voice is simply a left-to-right listing of these events, reading staves down from top to bottom. In examining some of the individual event types we can see the system's most direct relationship to many elements of mensural notation.

1.3.1 Clefs

Clefs are described in the grammar by a note letter with a number, as well as a pitch. The note letter describes what letter the clef's picture represents (since clefs are stylized letters), and the number describes its location in lines and spaces (1 being the bottom line of the staff, and each line or space going up counting as 1 more); the pitch describes the actual pitch which the clef represents. As noted above, modern "flat signs" in this notation are actually round b clefs, so instead of having a "signature" event, "flat signatures" are transcribed as clefs which are in effect added to the others (although there are some cases as well where the only clefs are round bs).

1.3.2 Mensuration Signs

The event type for mensuration signs contains a number of optional fields describing the visual appearance of the sign, without actually providing any editorial interpretation yet (since relative note lengths are editorial fields within each note event, and are not generated automatically from the mensuration sign). The presence of a symbolic representation of a mensuration sign in a CMME transcription, however, means that a scholar or analysis program has access to that data, and indeed can see it represented on the screen.

1.3.3 Notes and Rests

In the notes and rests, there exist several required fields in addition to some options. All notes and rests have a type and duration; type includes choices for breve, semibreve, etc., which for most notes and rests simply represents a visual description, while duration actually represents an editorial interpretation (in most cases an interpretation which has a strong degree of certainty).

Rests include another visual descriptor of two integers separated by a period, where the first integer represents the starting staff line for the rest, and the second integer represents how many spaces it covers (rests which do not cover complete spaces, such as semibreve rests, simply have 0 for this field).

For notes, a pitch is included, as well as options for ligation and coloration.

Ligatures are groups of notes which are drawn connected together in the manuscript, and have certain rules for rhythmic interpretation. Ligature transcription is at an immature stage in the CMME format, and in fact loses data in translation: the indication LIGATED on a note simply means that the note is connected to the previous one, but does not yet describe the various tail positions, *recta/obliqua* distinctions, and other important features of ligatures.

The other currently available note option is coloration, which indicates that a particular note is filled in black in the manuscript (instead of being left "void"); this also has certain effects on rhythmic interpretation.

1.3.4 Dots

Dots are not considered to be attached to particular notes, because they serve a host of purposes in mensural notation, involving imperfection, alteration, division of perfections, and other functions. Often a dot will serve to affect several notes, and it is certainly not considered to be part of a note, but instead an outside agent.

These and just about every other event type of the system serve to demonstrate points where elements of the mensural notation system must be treated differently than they are in modern notation, and differently than most representation systems have facilities to deal with.

As stated above, the current format is by no means complete, and in fact only takes account of a small portion of the 15th- and 16th-century mensural notation system. The current elements need more detailing and correction, and other elements must be added, such as proportions, full notation (as opposed to void) and other coloration types, more special signs, etc.

Just as important to the development of the system and its grammar is the specification and creation of further layers of data and interpretation. A more complete system, for example, would contain for each transcription a color reproduction of its source, physical measurements such as locations of voice parts on choirbook pages (even physical locations of each note!), a layer to interface one transcription with transcriptions of variant versions of the same piece, and other further levels of editorial interpretation. The point is to stray on the side of putting too much information in a type of edition where the display software can ignore what the user doesn't want to know about.

1.4 The Program

There does currently exist a program for viewing transcriptions in the CMME format (using the text-based style of Example 4), which is at a very early stage of development, and only supports a small amount of user control over visual display elements. Figure 1 shows two different views of sections of a complete transcription of Pierre de la Rue's *Vexilla regis–Passio domini*, demonstrating the display of different cleffing and barring styles out of the same data file.

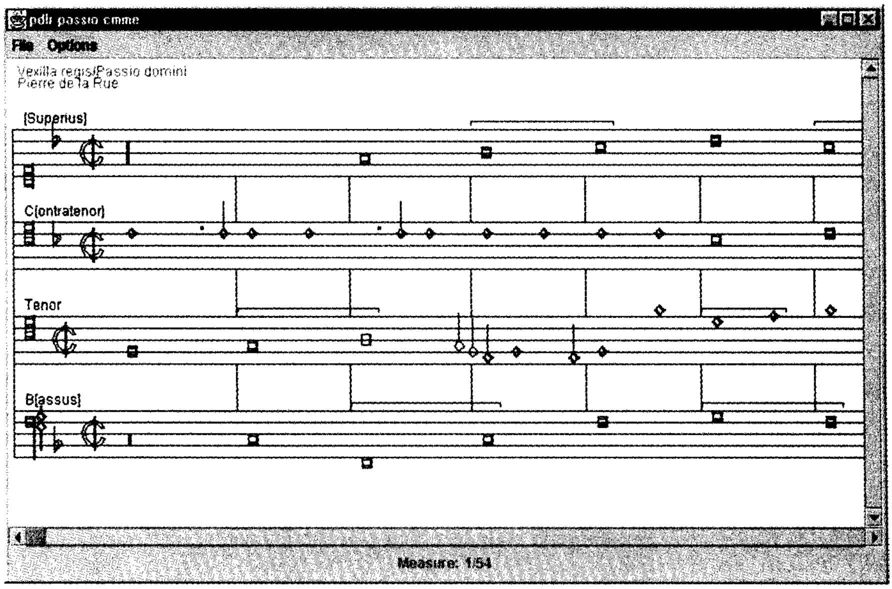
The viewing program exists as a set of Java classes which may be run either in the context of an Applet embedded in a World Wide Web page, or as a stand-alone application to be run by a Java interpreter. The only noticeable difference between the two versions is that the stand-alone application explicitly displays a new window, to simulate the Applet area in a Web page.

The Applet version of the program is located at the following URL:

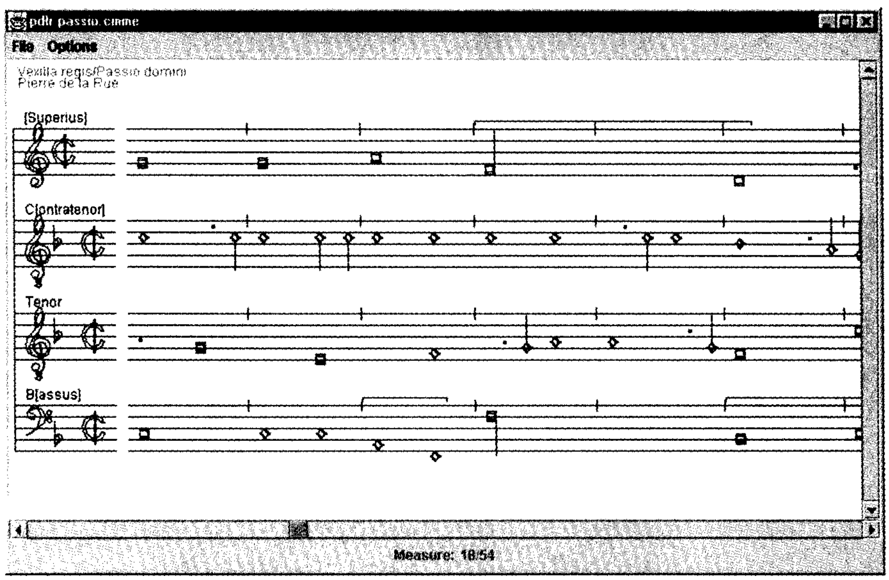
www.cmme.org/test.html

Figure 1. CMME viewing program with different display styles.

(a)



(b)



One peculiarity of the HTML code which loads the Applet is that it currently expects the browser to run the code with the help of the Java Runtime Environment (JRE) 1.2 Plug-in, available freely from Sun Microsystems. This is a temporary measure necessitated by the condition that most Web-browsing programs do not yet natively support version 1.2 of the JRE, which is required to run the CMME viewer. For systems which natively support the JRE v1.2 (such as the `appletviewer` program which comes with v1.2 of the JDK), the program can be loaded from the following URL:

www.cmme.org/test2.html

When the HTML file is loaded by the browser, the CMME viewer should automatically start in the browser window.

The class files for the viewer can also be downloaded and run as an application without a Web browser. The files are available as a Java Archive at:

www.cmme.org/siwtest.jar

The classes may be run either directly in their archived format, or decompressed and expanded into their own directory structure using the `jar` tool. As mentioned above, these classes use the JRE 1.2 specification, thus they will not run using the Java interpreter from a version of the JDK earlier than 1.2.

Additionally, the program uses several third-party libraries which must be set up in the system's CLASSPATH in order to run the CMME viewer as a stand-alone application. The following libraries are required by the program:

Library	Available at:
JLex	*www.cs.princeton.edu/~appel/modern/java/JLex*
Java CUP	*www.cs.princeton.edu/~appel/modern/java/CUP*
Freetype	*ds.dial.pipex.com/town/close/gap59/freetype.shtml*
Lava Rocks	*sharkysoft.com/software/lava/*
JGL	*www.objectspace.com/products/jgl/*
ErrorMsg	*www.cmme.org/errormsg.jar*

When the archives or class files have been installed on the system, the application can be started by running the Java interpreter on the class Main.Main. It should be noted that the archive `siwtest.jar` contains the class files for the application, but no data files. All of the data files which the CMME viewer requires are considered a network resource, and are read by the program from the main CMME server at *www.cmme.org* (thus the application requires a system which can establish an Internet connection).

The current state of the *corpus mensurabilis music electronicum* represents the initial step toward creating a flexible electronic early music edition. Even at this extremely early point in its development, the system already presents some benefits due to its electronic format. The program is available to a wide audience through the popular World Wide Web, and will only become more open to a general audience as Web browsers are updated to keep up with programming technology, and as computer architecture keeps up its trend of developing ever more powerful processors.

The editions available through the database are not fixed to one particular visual representation, but provide the user with a flexible means of configuring the editions to suit particular needs. Correction of errata and updates in the data files entail modifying data which resides only on a central server, automatically eliminating the need for "corrected editions" which must be redistributed to an audience. Most important, the system begins to show a means of structuring information which bypasses many of the foundations of the traditional printed text.

References

Bent, Margaret (1994), "Editing Early Music: The Dilemma of Translation," *Early Music*, xxii/3 (August 1994), 373–392.

Busse Berger, Anna Maria (1993). *Mensuration and Proportion Signs: Origins and Evolution.* Oxford: Clarendon Press.

Selfridge-Field, Eleanor, ed. (1997). *Beyond MIDI: The Handbook of Musical Codes.* Cambridge, MA: MIT Press.

Skinner, David (1998), Review of The Byrd Edition, vii/a, *Early Music*, xxvi/1 (February 1998), 151–152.