A WEB-based Approach to Music Notation using GUIDO

Kai Renz and Holger H. Hoos

Darmstadt University of Technology
Wilhelminenstr.7, D-64283 Darmstadt, Germany
email: {renz,hoos}@iti.informatik.tu-darmstadt.de

Abstract. This paper describes how flexible, online conventional music notation (CMN) can be obtained using the powerful music description language GUIDO in conjunction with the WEB-based GUIDO NoteServer. This fast, flexible, easy to use, and portable approach to platform-independent music notation can be used for a broad range of applications, for example to display conventional music notation on WEB-pages or to realise score displays in musical applications.

Introduction and Background

Today's conventional music notation can be traced back to Guido d'Arezzo (ca. 992-1050), the inventor of the staff system for notating music. Even though some graphical and musical symbols have been added or changed as musical composition and performance techniques evolved, the basic concept of representing duration and pitch in a two-dimensional graphical way has remained the same. It is well known that conventional music notation is not well suited for representing music within computers and much work has been done to find other suitable digital representations of musical material (see, e.g., [Selfridge; 1997], [Grande; 1997], [Hewlett, Selfridge; 1997], [Sloan; 1997], [Rotterdam; 1998]). So far, for various reasons, none of these formats have become true interchange formats, even though some them can be used to exchange musical as well as notational (e.g. graphical) information between applications and/or operating systems. However, we find that existing representations are either too weak to encode all the required information (like, for example, MIDI), or they are too complex (like, for example, SMDL).

The recently developed music representation language GUIDO [Hoos *et.al.*; 1998a] is a new approach to music representation which is capable of transporting musical as well as graphical information. The basic idea behind the GUIDO design is *representational adequacy* which means that simple musical concepts are represented in a simple way and only complex notions require more complex representations. GUIDO is divided into three layers: Basic GUIDO covers simple musical and notational concepts; Advanced GUIDO supports exact formatting and layout specifications as well as more advanced musical concepts; and Extended GUIDO provides support for musical concepts beyond conventional music notation. GUIDO is designed to be an open standard which can be easily extended to incorperate new musical concepts. It is currently supported by NoteAbility, a professional music notation program written by Keith Hamel [Hamel; 1997]; by the SALIERI system, an interactive system for analysis, composition, and transformation of music on a score-level basis [Hoos *et.al.*; 1998b]; and by Elody, a compositional system based on lambda calculus [Orlarey *et.al.*; 1997]. Other musical applications supporting GUIDO are currently being developed; there are also plans for realising GUIDO support for OpenMusic [Assayag *et.al.*; 1997].

For a given piece of music, the GUIDO NoteServer transforms a GUIDO Music Notation (GMN) description into a graphical score. The output is either a bitmap-like format supported by all WEB-browsers (like GIF or JPEG), or a postscript file for usage in conjunction with other software, like word processors or publishing applications. This approach is very flexible, as it works sufficiently fast to support interactive changes of musical material with almost instantaneous display of the results. Thus, the GUIDO NoteServer offers dynamic notation of musical material in an adequate and easy-to-use way.

There has been some previous work on supporting conventional music notation on the WEB. Earlier HTML-based formats for music description have the disadvantage of being too weak for encoding all the relevant notational information, while requiring to much overhead for representing simple musical information. Musical scores written in this fashion were neither intuitive to read nor easy to edit. The new

XML (eXtensible Markup Language) is meant to replace HTML as the predominant WEB language. Some work has been done for using the extension capabilities of XML to support music representation [Rotterdam; 1998]; nevertheless, XML has not been designed with the specific requirements of music representation in mind, and documents describing even simple musical pieces in XML tend to be difficult to read and are in no means representationally adequate.

SMDL (Standard Music Description Language) [Sloan; 1997] is another approach to create a comprehensive music description language and has been designed as an interchange format which is capable of transporting musical information over the internet. Some work has been done to develop WEB-services that are capable of displaying SMDL files. However, to our knowledge, none of the existing applications is capable of handling the full SMDL specification, partly because SMDL is such a large and complex standard.

Another music description language focussing on the graphical aspect is *cmn* (Common Music Notation) [Schottstaedt; 1997], a LISP-based system which generates a postscript output containing conventional music notation from musical and graphical information that is described using LISP syntax. To our knowledge, *cmn* is not available as an online-service and depends heavily on the underlying LISP system. The user must have sufficient knowledge of LISP to use it; *cmn* is also primarily designed to produce graphical output, whereas GUIDO is meant to represent graphical as well as musical information.

The GUIDO NoteServer

Concept The GUIDO NoteServer (GNS) is based on the following concept: A musical piece in GUIDO Music Notation (GMN)[1] is transferred to the server using either the Common Gateway Interface (CGI) or by directly embedding a JAVA-applet on a WEB-page or through a direct call using an extended URL, which can easily be constructed using JAVA or PERL. The GNS then parses the input and returns a graphics file containing the given piece in conventional music notation. As of this writing, GIF support is already being provided, while postscript output will be realised soon. Additional parameters (like size, width and number of staves on a page, page number) determine the general graphical appearance of the output.

Implementation The GUIDO NoteServer is implemented in two layers. The first layer is a PERL script which handles the input from the WWW-Server. The script checks the parameters and than calls the GMN2GIF-executable to create the resulting GIF-file. The executable is written in C++ and uses the same sources as the stand-alone NoteViewer tool for GUIDO Music Notation.[2] This way, changes and improvements are automatically available to all graphical GUIDO applications. The underlying architecture is an object-oriented framework based on two sets of class hierarchies which are called "abstract representation" and "graphical representation", respectively. The former is an object-oriented structure representing the piece specified in GUIDO Music Notation; the latter is constructed from the former and represents, also in an object-oriented manner, the resulting graphical output. This approach allows to distinguish between graphical changes, such as positioning or spacing of notes, and structural changes, like changing the clef or meter of the piece. On this basis, user-interaction can also be easily realised.

Accessing the GUIDO NoteServer

At the moment, we support three different modes of accessing the GUIDO NoteServer; these are illustrated in Figure 1 and will be described in the following.

Using the CGI-Interface The predominant mode of access which is primarily used when first experimenting with GUIDO is the CGI-Interface[3] depicted in Figure 2: the left side shows the input form, while the corresponding output can be seen at the right. Using this interface, the user is able to specify all relevant parameters and to experiment with different settings. We are currently working on an alternate input mechanism based on a JAVA-applet; the overall goal is to provide a graphical user interface including

[1] A specification of GUIDO can be obtained at `http://www.informatik.tu-darmstadt.de/AFS/CM/GUIDO`
[2] The stand-alone NoteViewer can be otained at our website.
[3] Publicly available at `http://hp3.iti.informatik.tu-darmstadt.de/salieri/nview/noteserver.html`

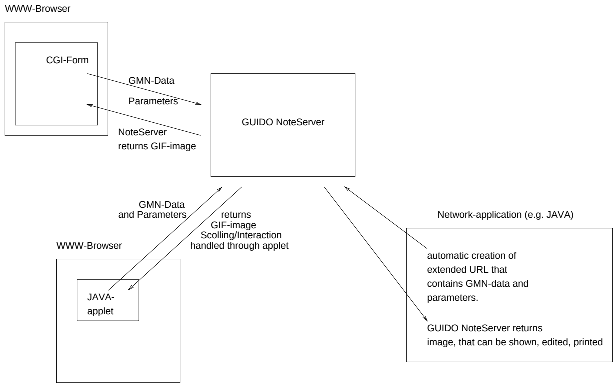

Figure 1: The different ways to access the GUIDO NoteServer

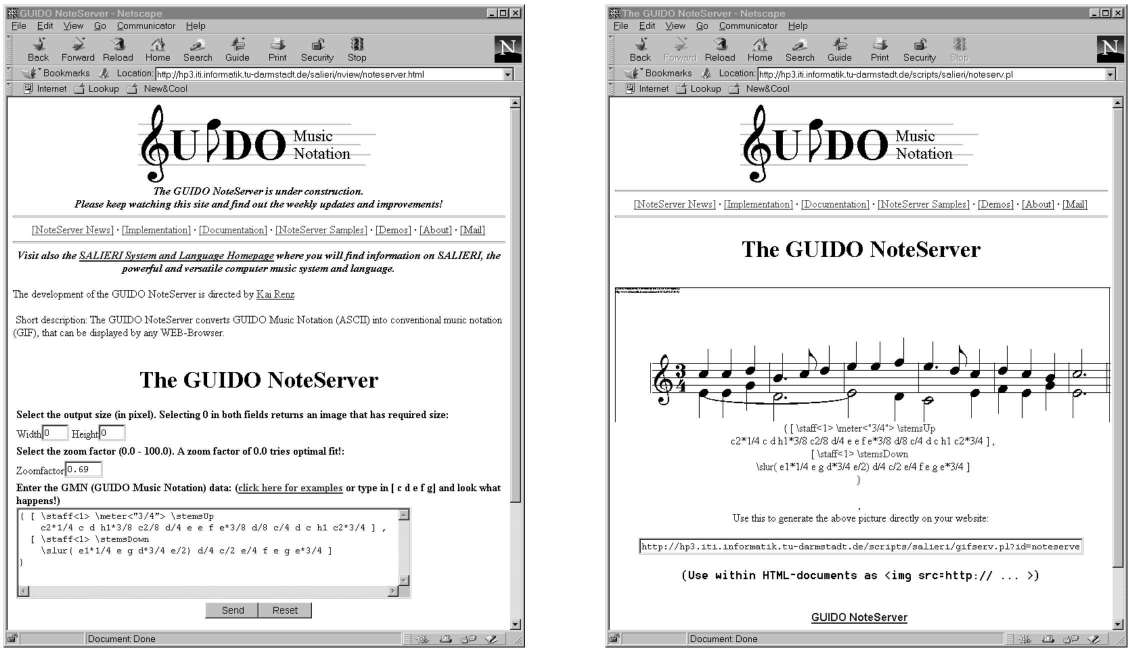

Figure 2: Using the CGI-Interface of the GUIDO NoteServer

a virtual keyboard and other means of interactive input of musical material.

Using a JAVA-Applet Another means of access can be realised by simply embedding a JAVA-applet in a WEB-page. The GUIDO Music Notation input is given as a parameter to the applet which handles the interaction with the GUIDO NoteServer. The returned image is scrollable within the specified area. An example of this approach can be seen in Figure 3 showing the HTML-source (left side) and the resulting WEB-page (right side).

Direct access using extended URLs The most challenging access to the GUIDO NoteServer works as follows: the user (or application) creates an extended URL containing all parameters in a HTTP-encoded representation. This URL serves as a reference to an image and can be easily embedded in WEB-pages or applications. More information on this mode of access which is used, for example, by the Elody system [Orlarey *et.al.*; 1997], can be obtained at the GUIDO NoteServer WEB-page.

```
<html> <body>

<applet codebase=
   "http://hp3.iti.informatik.tu-darmstadt.de/salieri/nview/test"
  code="GuidoSimpleApplet"    width=650    height=450 >
  <PARAM NAME=noten VALUE='
  { [ \staff<1> \meter<"3/4"> \stemsUp
    c2*1/4 c d h1*3/8 c2/8 d/4 e e f e*3/8 d/8 c/4 d c h1 c2*3/4 ] ,
    [ \staff<1> \stemsDown
    \slur( e1*1/4 e g d*3/4 e/2) d/4 c/2 e/4 f e g e*3/4 ],
    [ \staff<2> \clef<"bass"> \meter<"3/4">
     c0*3/4 g-1 c0*2/4 f-1*1/4 c0*2/4
     a-1/4 f g/2 c0*3/4 ] }  '>
</applet>

</body> </html>
```

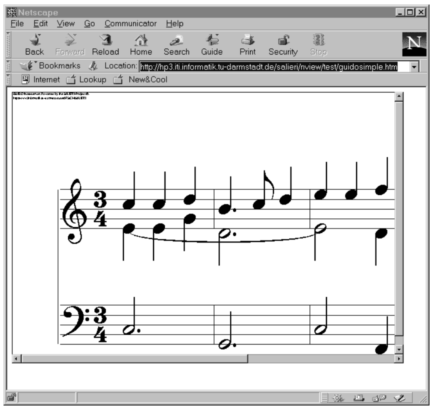

Figure 3: Embedding the JAVA-applet in an HTML-document

Conclusions and Future Work

We are constantly improving the GUIDO NoteServer; the primary goal behind this development is to provide a service which supports all Basic and Advanced GUIDO tags and can be used freely throughout the community. In the near future, we will be able to provide postscript output and, at the same time, offer a version of the GUIDO NoteServer and stand-alone NoteViewer for Macintosh computers. In the course of this development we plan to make the NoteViewer/NoteServer sources publicly available; thus, interested parties will be able to port the GUIDO NoteServer to other platforms and to provide support for other graphic output formats.

References

[Assayag *et.al.*; 1997] Gerard Assayag, Carlos Agon, Joshua Fineberg, Peter Hanappe; An Object Oriented Visual Environment For Musical Composition; Proceedings ICMC-97

[Grande; 1997] Cindy Grande; The Notation Interchange File Format: A Windows-Compliant Approach; *in:* Beyond MIDI, The Handbook of Musical Codes; Eleanor Selfridge-Field (ed.); pp. 491-512; MIT Press; Cambridge, Massachusetts 1997

[Hamel; 1997] Keith Hamel; NoteAbility Reference Manual; Opus 1 Music Inc.; Vancouver, B.C. 1997

[Hewlett, Selfridge; 1997] Walter B. Hewlett, Eleanor Selfridge-Field; MIDI; *in:* Beyond MIDI; Eleanor Selfridge-Field (ed.); pp. 41-72; MIT Press; Cambridge, Massachusetts 1997

[Hoos *et.al.*; 1998a] Holger H. Hoos, Keith Hamel, Kai Renz, Jürgen Kilian; The GUIDO Notation Format - A Novel Approach for Adequatly Representing Score-Level Music; *in:* Proc. of ICMC-98

[Hoos *et.al.*; 1998b] Holger Hoos, Jürgen Kilian, Kai Renz, Thomas Helbich; SALIERI – A general, interactive Computer Music System; *in:* Proceedings of ICMC-98

[Orlarey *et.al.*; 1997] Yann Orlarey, Dominique Fober, Stéphane Letz; L'environnement de composition musicale Elody; *in:* Proc. of JIM'97; pp. 122-136; 1997

[Rotterdam; 1998] Jeroen van Rotterdam; MusicML - A XML Experience; at `http://www.tcf.nl/trends/`

[Schottstaedt; 1997] Bill Schottstaedt; Common Music Notation; *in:* Beyond MIDI; E. Selfridge-Field (ed.); pp. 217-221; MIT Press; Cambridge, Massachusetts 1997

[Selfridge; 1997] Eleanor Selfridge-Field; DARMS, Its Dialects, and Its Uses; *in:* Beyond MIDI; E. Selfridge-Field (ed.); pp. 163-174; MIT Press; Cambridge, Massachusetts 1997

[Sloan; 1997] Donald Sloan; HyTime and Standard Music Description Language: A Document-Description Approach; *in:* Beyond MIDI; E. Selfridge-Field (ed.); pp. 469-490; MIT Press; Cambridge, Massachusetts 1997