Abstract

The center of attention in this paper is a computer-based music notation application Expressive Notation Package (ENP). ENP is among the various music related software packages developed at Sibelius Academy. ENP is of central importance in the music related applications collected under the umbrella of PWGL (PWGL is a free cross-platform visual programming language with a strong emphasis on music and sound related applications). ENP is an important bridge between the different software components inside PWGL. It allows these packages to inter-operate by using ENP's versatile representation of musical objects. There is a close integration between ENP and PWGLSynth (the synthesis environment of PWGL). The rule-based language called PWGLConstraints, in turn, can be used to calculate (i.e., compose) musical material with the help of ENP. Furthermore, ENP-Script, a music scripting language, can be used to automatically analyze a score using multiple view-points. It is also possible to visualize the analysis information directly on the score. This paper gives an overview of the current status of ENP and gives concrete examples of some of its applications.

1 Background

ENP (Kuuskankare and Laurson 2002) is a music notation program written in Lisp. It is a software package that has a mouse driven graphical user interface based on direct editing. The main motivation behind ENP is to meet the needs of computer aided composition, music analysis and synthesis control. According to Glendon Diener (Diener 1990), music notation programs have three basic types of use: Compositional, archival, and analytic. In its current state ENP can be used:

1. to represent complex music notation including 20th century music
2. to algorithmically generate musical structures: scores, parts, chord, and notes
3. to automatically analyze a musical score using multiple view-points
4. to textually describe a score and the individual notational objects
5. to display music theoretical analysis information, e.g., Schenkerian graphs
6. to manipulate a score through a GUI based on direct editing
7. to use scores as user interface components
8. to visually create custom notational attributes that are immediately usable without recompiling and
9. to add, remove, and modify information using an intelligent scripting language.

ENP is used as a notational front-end in PWGL (Laurson and Kuuskankare 2002). There exist a version for both Mac OS X and Windows XP operating systems. Besides PWGL, there are also other Lisp-based composition environments that provide music notational capabilities. Each of these systems has their own unique features and approach to music representation. OpenMusic’s (Assayag et al. 1999) approach is the simplest: it provides a simplified music notation comparable to its predecessor, PatchWork (Laurson 1996). Only a handful of basic notational attributes are provided and the user interface quite restricted. In the form of CMN (Schottstaedt 1997), Common Music (Taube 1991) provides a powerful music notation system that can represent a wide range of music. However, the interface to music notation in CMN is purely text-based, thus it is only intended for viewing, and not for editing. PWGL’s approach is unique in several different ways. It provides a music notation package with a GUI based on direct editing. It also allows the use of its musical structures (both low-level, e.g., notes, chords; and high-level, i.e., score) in other packages, such as PWGLConstraints (Laurson and Kuuskankare 2005a) and PWGLSynth (Laurson, Norilo, and Kuuskankare 2005). It is also possible to generate and manipulate the musical objects using PWGL’s graphical patch language.
The rest of the paper is organized as follows. First, we give a brief introduction to the notational capabilities of ENP. Next, we enumerate some of the most important applications. The paper ends with some concluding remarks.

2 ENP in Brief

ENP is intended to represent Western musical notation roughly from 17th century onward including 20th century notation. ENP supports an array of notational styles. An ENP score is built out of hierarchical object structures. Typically, an ENP score consists of parts, a part consists of voices, a voice consists of measures, etc. The objects that contain the actual pitch material are chords. Chords also can contain information about additional attributes. These attributes, called ENP-Expressions, are of primary importance in terms of our software. They can be used in a wide range of applications: (1) display music analytical information; (2) attach arbitrary textual annotations to objects; and (3) dynamically inspect and visualize data contained by the notational objects.

Figure 1 gives an example of a score written in traditional metric notation with some standard expression markings, such as accents, slurs, and dynamics. The score shows a typical setting for guitar, where there are multiple voices written in a single staff-system.

Figure 1: An instrumental part containing three voices displayed in a single staff-system. (Isaac Albéniz: Tango, op. 156; transcribed for guitar by Andrés Segovia.)

3 ENP Applications

3.1 Sound Source Control

ENP has already been used as a tool in several different projects as an intelligent sequencer to produce control information to various instrument models realized with the help of PWGLSynth (there is a close integration between ENP and PWGLSynth). One of the unique aspects of ENP is that control information can be drawn directly on the score. Figure 2 shows how control information, specialized to a classical guitar model, can be displayed in an ENP score. An elaborate interpretation is generated with the help of both standard expressions and some special macro-note expressions (Laurson and Kuuskankare 2005b). When the score is played the control information is calculated automatically.

Figure 2: A score written for a virtual guitar containing both standard and non-standard expressions. Also, there are some macro-note expressions to create complex arpeggio gestures.

3.2 Computer-assisted Composition

ENP has been used as a computer-assisted composition tool along with PWGLConstraints. For example, in his work, Hodoit Ergo, composer Kimmo Kuitunen uses ENP to visualize the ideas behind large-scale durations (Kuitunen, Kuuskankare, and Laurson 2005). Figure 3 shows one potential solution to a compositional constraints problem prepared by Magnus Lindberg. The pitch material of this excerpt is calculated with the help of ENP and PWGLConstraints.

Figure 3: An example constraints problem prepared by Magnus Lindberg. The pitch material shown in the ENP score is calculated using PWGLConstraints.

3.3 Computer-assisted Music Analysis

There are several sophisticated music analysis applications available (e.g., HumDrum, Huron (2002); athenaCL,
Ariza (2005)) that use text based representation as a starting point. This kind of an approach is usually adequate for analysis questions that are statistical in nature. However, text-based approach makes verifying the result more difficult because the analysis result is provided without the corresponding notational counterpart. In ENP, in turn, it is possible to use a set of specialized graphical devices to visualize the analysis information directly in the score. An ENP score can be processed with ENP-script (Kuuskankare and Laurson 2004) that can access a rich set of information, such as, pitch, duration, dynamics, expressions, timing. The user can also map over different levels of notational objects, such as, measures, beats, chords, and notes in the score. This kind of an approach resembles the container-iterator idea found, for example, in CPNView (Donncha O’Maidin 1999; Donncha O’Maidin and Margaret Cahill 2001). Figure 4 shows the starting-point for our scripting example (an excerpt of a piece by Arnold Schoenberg without any articulations or other markings). Here, we use an ENP-script to analyze and manipulate the score using two different stand-points. First, we analyze the pitch and rhythm information in order to fill the score with some of the articulations found in the original score (e.g., in the original score there is always a slur between two notes if they form a larger interval than an octave). In this example we concentrate only on a few types of articulations: grave accents, slurs and crescendos. Second, we use the script to find and mark the 12-tone set forms of the passage.

![Figure 4: Suite für Klavier, op. 25 by Arnold Schoenberg containing only the pitch and rhythm information.](image)

Below we give a part of the script that is used to perform the analysis. An ENP-script is typically constructed out of a number of so called scripting rules. One scripting rule generally deals with one aspects, for example, in this case the first scripting rule inserts the slurs in the resulting score and the second scripting rule, in turn, inserts the accents (see Figure 5). The set forms are marked with special ENP-expressions called Groups. Groups can be used to mark a continuous collection of notational objects in the score and give them a common identity. In this case the common identity is the set form, e.g., 'P4R', 'P10'.

```plaintext
; 1. scripting rule
(?? ?1 ?2
  (if
    (and
      (> (abs (- (m ?2) (m ?1))) 12)
      (not (prev-rest? ?2)))
    (add-expression 'slur ?1 ?2))
  "slur")

; 2. scripting rule
(?? ?1
  (if
    (= (beaming-level ?1) 2)
    (add-expression 'accent-grave ?1))
  "accent")

; 3. scripting rule
  (if
    (and (= (beaming-level ?1) 2)
         (= (beaming-level ?6) 2))
  "crescendo")
```

![Figure 5: The ENP score given in Figure 4 filled with some analysis information (set forms) and articulations with the help of ENP-script.](image)

### 4 Conclusions

This paper provided an overview of the current state of Expressive Notation Package (ENP), a music notation program used in PWGL. We covered some of ENP’s exist-

---

1The naming convention used here is taken from the New Grove Dictionary (Sadie and Tyrrell 2001) where the different aspects of the series are indicated with P (prime), R (retrograde), I (inversion), and RI (retrograde-inversion).
ing applications dealing with computer assisted-composition, computer-assisted music analysis, and sound source control.

It is convenient to have a system with which analytical information can be represented directly as part of music notation. This kind or representation, however, is suitable for only certain kind of data. For example, statistical analysis information is virtually impossible to display this way. Thus, ENP should be integrated with some statistical package such as $R^2$ (Cribari-Neto and Zarkos 1999), to allow the possibility to represent any kind of statistical information that can be retrieved from an ENP score (e.g., pitch histograms).

Computer-assisted Instruction (CAI) is another interesting area in terms of ENP. CAI refers to a system of educational instruction performed almost entirely by computer. There are many advantages to using computers in educational instruction: they provide instant interaction with a student and also allow students to proceed at their own pace. The facilities in ENP, including its integration with PWGLConstraints and PWGLSynth, make it an ideal CAI tool. Sophisticated tutors for basic music theory, solfège, post-tonal music theory, interactive harmony teachers, ear trainers and other assignments could be created with the help of ENP.

For the needs of CAI ENP should also provide specialized Content Authoring Tools (CAT). CAT provide teachers with the means to create new educational material by enabling to arrange different kinds of visual and textual components to create interactive training content.

Acknowledgment

The work of Mikael Laurson has been supported by the Academy of Finland (SA 105557).

References


---

2$R$ is a language and environment for statistical computing and graphics.