Musical Analysis
of a Computer Music Project,
by Computer-Assistance,
for Computer Music Applications

Jian-Li Liu
Doctoral Program in Engineering,
University of Tsukuba

Rumi Hiraga
Bunkyo University

Shigeru Igarashi
Department of Computer Science,
University of Tsukuba

Abstract

Daphne is a computer-assisted musical analysis system that analyzes musical structure and other information necessary for computer music applications, especially for performance synthesis. The obtained analysis results are stored in a style which extends the information of a note event so that application systems can share analysis information. Providing sheets of real musical scores, Daphne's design includes a unique function of analysis database which stores analysis result of multiple editions of a piece. Users can make use of the database to consider how performance is generated based on the interpretation of each edition. In this paper, we describe how analysis information is stored and shared among computer music applications — performance synthesis with rules, performance visualization, and automated accompaniment.

1 Introduction

Daphne is a computer-assisted musical analysis system in order to analyze musical structure and other information which is important for performance synthesis in WYSIWYG way [5]. With our music analysis system Daphne, users can obtain the following items either by manually or automatically: musical structure (motif, phrase, and so forth), relationships among occurrences of musical structure, (metric) structural function (anticipative [anacrusis], culminative [tiarra], etc.) [1, 4], harmony and chord progression, tonality, and musical forms.

Daphne displays sheets of real musical scores and sounds actual performance of the analyzed musical piece in order for users to analyze in the similar environment as they do in the real world on their desks and instruments with musical scores and pencils. The obtained results are promptly displayed on the score, at the same time they are internally stored into a file of Daphne's own format.

While both Paleo [6] and Humdrum [3] are music analysis systems which intend to analyze complex items on various kinds of data, Daphne analyzes musical scores of mainly classical and roman piano pieces. Paleo addresses the use of composers and performers by providing sound/music analysis tools and database. Music analysis information obtained through Daphne is shared among systems of computer music project Psyche [2]. With the main theme of performance synthesis with rules, several systems for performance visualization and automated accompaniment have been built in the Psyche project.

Daphne can be a tool for music specialists since its design includes the analysis database for comparing musical interpretation of different editions of musical pieces. Those who are not familiar with the use of a computer can access, retrieve, and compare interpretation, since Daphne provides a user-oriented interface by displaying the same musical scores as published.

The shared data format called UNI and basic classes in Daphne are described as system structure. Then we show how analysis data is used by systems of Psyche.

2 System Structure

2.1 Input/Output

As shown in figures (Figure 3, Figure 4, and Figure 5), users of Daphne can analyze a musical piece on score sheets which are the same as in the published musical scores. In order to provide this
graphical user interface, a bitmap files for score sheets, an event-position file which includes the X-Y positions of notes on the score sheets, and a file for coded score should be prepared prior to an analysis session by a Daphne administrator.

A user specifies a range on the displayed score then an analysis item to analyze either automatically or manually. The obtained results are promptly displayed on the score while they are internally stored into a file of Daphne's own format (UNI format) which applications of Psyche share.

2.2 UNI

UNI format is an extended/extendable note event format based on MIDI messages. Each tuple represents a note event containing both the information of performance which is derived from MIDI file and the information of score analysis obtained through Daphne.

An example of a UNI formatted tuple is as follow. Fields of each note event are labeled for readability:

```
I83K77V64L83W1A2880
MOT11M1RgM2S3P3HRgB1RgB0PS0SEN9gB1RgB0SS0
ACC3UPB0DESS0
```

Here, I, K, V, and L shows information from MIDI data, W and A from a score, from MOT to SS are descriptions of structure analysis results (such as motif, reference motif, and similarity level, etc.), and other fields are for different kinds of structure functions. Besides the readability, the most important characteristic of Uni format is the easiness to extend, so we can add more fields according to the increase of analysis items.

2.3 Classes

Daphne’s GUI is implemented by Visual Basic, while the internal analysis algorithms are written by Visual C++. Figure 1 shows a part of classes implemented in Daphne.

**DEuropa** is a class to handle a score data written by Europa score description language [2]. It loads a score data when a Daphne session starts, then generates objects of the class **DMeasure** and **DNote**. When a voice part is specified, a note on a score is uniquely specified with a tuple of the number of the measure it resides and the offset from the start of the measure. An object of DNote represents a note and that of the class DMeasure corresponds to a measure in a score which has a list of objects of the class DNote in the measure as an attribute.

Objects of the class **DUnit** represent occurrences of musical structure. It has attributes for the start and end positions of an occurrence in a score presented by measures and the offset in the measure, the level of the hierarchy (whether

---

![Figure 1: Classes of Daphne and their relationships](image)

---

**Figure 1: Classes of Daphne and their relationships**

---

**Figure 2: Sharing analysis information**

---

3 Sharing Analysis Data

3.1 Data Sharing among Musical Systems

Currently, the following three types of systems in Psyche share analysis information as in Figure 2.

1. systems of performance synthesis with rules.
2. performance visualization systems.
3. automated accompaniment systems.
3.1.1 Performance Synthesis with Rules

Systems of performance synthesis use rules which are expressed in terms of analyzed items, especially musical structure and the (metric) structural functions (1-6) [concepts in italics are introduced in [4]] by which a typical agogic rule is described as follows: A motif starts (1. inceptive) rather slowly, or long; accelerates in the anacrusis (2. anticipative) to the top whose notes are prolonged, which is composed of the accent (4. initiative) frequently preceded by a 'barra' (3. culminating); simple desinence (5. reactive) follows; and a special time period is taken at end (6. conclusive). Rules are found out by comparing performances by several professional pianists using the visualization as in Figure 3. The expression at the end of a musical structure can be explained with analyzed information of chord progression. Besides inducing performance rules, analyzed data is used at applying rules.

3.1.2 Performance Visualization

Performance visualization enables users to understand performance objectively and to describe relationships between an expression and performance parameters. An example of visualization (Figure 3) is that notes in a phrase (about four measures) are shown as ellipses, starting from the top and moving clockwise, on a polar plane. Multiple parameter values can be shown at the same time on this type of visualization figure. Assigning parameters to the angle of consecutive notes, radii of an ellipse, and the length from the center of the plane to an ellipse can show the relationships among parameter values. Structural function is marked on the figure. Musical structure is the mandatory information for visualization systems.

3.1.3 Automated Accompaniment

The performance for the accompaniment part is analyzed then synthesized referring human player’s rehearsal performance. An accompaniment part does not necessarily always follow human player’s performance. Since the beginning of a musical structure has more meaning, one of our accompaniment systems gets the information of musical structure and the fitting positions before a session starts. Figure 4 shows how to specify a fitting position on a score sheet.

3.2 Analysis Database

In using a performance synthesis system, it is important for a user to realize on which score a
Figure 5: Comparison of Analysis for two Editions

Since different editions reflect different interpretations of each editor (musicologist), musical analysis by Daphne which uses score information generates different analysis results that affect performance synthesis.

Daphne is able to provide score database with analysis information. The comparison of score editions and their analysis results will help understanding the cause in the score and the effect appearing in a performance.

Figure 5 shows two editions (Zen-on and Padelevski) of Mazurka Op. 7-1 by Chopin and their comparisons. The slur put on the Padelevski edition makes a different musical structure from Zen-on’s.

4 Conclusion

We described the internal structure of a computer-assisted music analysis system Daphne and how obtained analysis information is shared among applications of the computer music project Psyche.

So far the analyzed data has been shared well among computer music applications of the project. Although the system is a prototype for research purposes, we shall be able to show its usefulness by solving the following issues.

- enhancing automated/computer-assistant analysis.

Some of the analysis items are not automatically obtained currently. Though it is not our purpose to automate all analysis processes, some computer-assistant interactive functions, for example to propose candidate analysis results, make the system more useful.

- providing more data.

The quantity of data is required for the system to be used as archives.

- giving objects persistency for object-oriented database.

Analysis data are shared by using a file currently. Considering multiple users analyze a musical piece concurrently, database functions will be necessary.

References


