

# Annotated Music for Retrieval, Reproduction, and Sharing

Keiji Hirata<sup>1</sup>, Shu Matsuda<sup>2</sup>, Katsuhiko Kaji<sup>3</sup>, and Katashi Nagao<sup>3</sup>

<sup>1</sup>NTT Communication Science Laboratories

hirata@brl.ntt.co.jp

<sup>2</sup>Digital Art Creation

shu@dcreation.com

<sup>3</sup>Nagoya University

{kaji,nagao}@nagao.nuie.nagoya-u.ac.jp

## Abstract

*This paper presents a technique for enhancing music information by annotation, and we apply the technique of annotated music to a collaborative music creation system on the Web, Music Resonator (MR). In MR, a user retrieves music fragments from Pool, processes them on Operation Diagram Editor and shares new music fragments with others, just as people readily exchange emails through mobile phones. Since a music fragment in MR is annotated with analysis results based on music theory, a user can perform complicated, skill-based tasks easily and properly. The result of a preliminary evaluation using the current implementation supports the significance of annotated music.*

## 1 Introduction

This paper presents a technique for enhancing music information by annotation and an application of the technique to a collaborative music system on the Web.

An enormous amount of digital on-line content, such as text, images, audio, and music, is produced and consumed on the Internet every second. We currently face a rapid growth of digital content technology that is an attempt to restore social order in terms of content production and consumption. The digital content technology generally covers techniques to create, store, deliver, manipulate, transform, and reuse digital content. The Semantic Web<sup>1</sup> is a famous project on digital content technology, which aims to provide a common framework that allows data to be shared and reused across applications, enterprises, and community boundaries. For that purpose, the Semantic Web adopts an approach that grounds content on human life by annotation and shares content semantically between humans and machines.

*Annotation* in general means metainformation of digital content. In this paper, however, annotation means more than just “data about data” in some restricted format; it emphasizes that it is extra information about the deep, tacit meaning or context of content, which is created by human-machine collaboration (Nagao 2003). It follows that machines become

much better able to process and understand the data that they superficially treat at present. For example, we see a document on a browser merely as a sequence of letters. However, a displayed document is usually enhanced by the HTML tags representing its syntactic structures, behind the screen. Hence, the document can to some extent be automatically and appropriately summarized and reformatted. Furthermore, the GDA tag set<sup>2</sup>, for example, enables the description of linguistic and semantic features of the document, and it generally improves the efficiency and quality of content authoring, i.e. retrieval, processing, and sharing (reuse).

We are interested in enhancing music information by annotation that conveys its musical meaning (called *annotated music*). Unfortunately, conventional working approaches for representing and processing music (Selfridge-Field 2000) mostly focus on surface information, not deep structures. Here, surface information refers to the superimposition and juxtaposition relations between a relevant note and its surrounding notes on a score. Hirata and Matsuda (2003) take the annotated music approach, where a user sees a piano-roll score on a screen, but behind the screen, the score is enhanced by corresponding time-span tree of the Generative Theory of Tonal Music (GTTM) (Lerdahl and Jackendoff 1983). They show the significance of a methodology that uses the knowledge about musical structure and facilitates a complicated task; i.e. discovery of a piece’s structure by melodic similarity checking.

Recently, several collaborative music systems utilizing the Web technology have been developed, wherein users share music fragments and social communication is facilitated. Such music systems are categorized as follows: listening to music and returning reactions (Sgouros 2000), real-time interactive performance (Young 2001; Goto and Neyama 2002), and collective composition (Jordà 2002). In particular, annotated music brings about a great benefit to a collective composition system, since the basic tasks to be performed in such a system are typically retrieval, reproduction, and sharing. To demonstrate and examine the technique of annotated music, we are developing a collaborative music creation system on the Web

<sup>1</sup><http://www.semanticweb.org/>

<sup>2</sup><http://i-content.org/GDA/>

with which even an amateur can create short musical fragments like making a collage, as well as share the fragments with other users.

This paper is organized as follows. First, we discuss existing techniques of annotated music and present a novel annotation technique underlain by music theory. Then, we apply the annotated music technique to a collaborative music creation system on the Web, *Music Resonator*. Finally, we conclude by mentioning future work.

## 2 Annotated Music

We consider what and how to annotate music, and propose a new framework for annotated music by linking.

### 2.1 Analysis Results as Annotations

Conventional indexing schemes for music databases<sup>3</sup> mainly conform to the Dublin Core<sup>4</sup> (and MPEG-7<sup>5</sup> in the near future). The Dublin Core metadata set includes title, creator, subject, and rights, while MPEG-7 is a metadata standard, based on XML technology<sup>6</sup>, for describing features of multimedia content and providing the most comprehensive set of audio-visual description tools. The description handled by MPEG-7 tools basically follows Dublin Core metadata elements: semantic (e.g. the who, what, when, and where information about objects and events), and structural (e.g. color histogram associated with an image or the timbre of a recorded instrument) features of the audio-visual content.

Considering music summarization (Hirata and Matsuda 2003), for example, if a computer cannot recognize musical features of a given melody, neither can it identify which parts are similar to each other, nor know a piece's structure. To solve this problem, a structure analysis based on music theory is required; however, Dublin Core and MPEG-7 are inadequate, since they cannot represent musical features such as grouping structures, stable notes, time-span trees (Lerdahl and Jackendoff 1983), nor implication-realization structures (Narmour 1990). Generally, it is quite difficult to automatically recognize the musical features underlain by music theory, and thus, human-machine collaboration is more or less necessary to create them. Notice that such information deserves to be reused as annotations.

### 2.2 Annotations in XML

How do we add analysis results to music as annotations? There are two ways: embedding (tagging) and linking. Embedding is describing annotations directly within appropriate

<sup>3</sup><http://music-ir.org/evaluation/wp3/>

<sup>4</sup><http://dublincore.org/>

<sup>5</sup><http://www.tnt.uni-hannover.de/project/mpeg/audio/public/mpeg7/w2460.html>

<sup>6</sup>XML (Extensible Markup Language)<sup>7</sup> is the most promising format for the time being. XML is a simple, and very flexible text format for exchanging data on the Web and elsewhere. The markup is information inserted into a document used by computers and takes the form of tags inserted into a text to mark its structure.

tags in music itself. For example, a meta-event of a Standard MIDI File can be used as embedded annotation. Linking is associating annotations with relevant parts of music, typically by XML technology, including XPath (XML Path Language); accordingly, both music (content) and annotations are described in XML.

In terms of music information, we consider that linking is advantageous over embedding, mainly because music itself is inherently subjective and has multiple aspects, and we may obtain more than one analysis result; this naturally leads to multiple annotations.

We propose a framework for annotated music (Fig. 1). The results of structure analysis based on music theory as an-

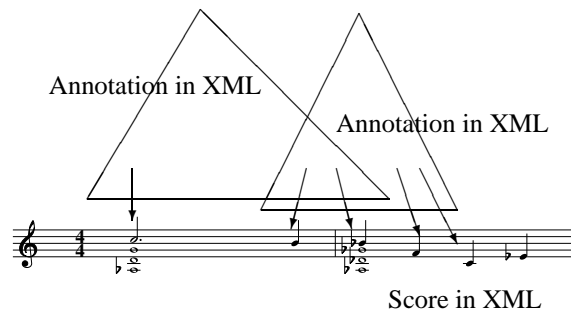


Figure 1: Annotated Music

notation in XML refer to each note in a score also in XML. The features of this framework are: (a) annotations containing the results of structure analysis based on music theory, which are created by human or by human-machine collaboration, (b) descriptive efficiency because a score file is shared, (c) usability of any XML-based music notation formats as they are (e.g., MusicXML<sup>8</sup> and WEDELMUSIC XML<sup>9</sup>), and (d) increased flexibility for attaching annotations. Here, flexibility means that an annotation can be attached to only necessary or desired parts, and annotations to an annotation are also allowed.

## 3 The Music Resonator System

We are developing a server-client system for collaborative music creation, *Music Resonator* (*MR*). Under the system, users may share music fragments, readily as people exchange emails through mobile phones. The length of music fragments varies from 1-bar long to full-piece. *MR* fully benefits from annotated music, because *MR* can support user's complicated, skill-based tasks, such as retrieval, reproduction, and sharing in an environment for collaborative music creation.

### 3.1 System Overview

Figure 2 depicts the system overview of *MR*. A client

<sup>8</sup><http://www.recordare.com/xml.html>

<sup>9</sup><http://www.wedelmusic.org/lang/xmlnot.html>

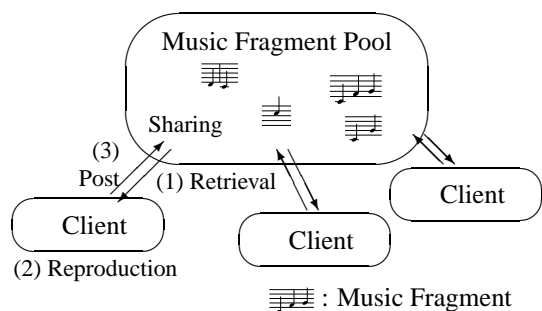


Figure 2: System Overview

asynchronously (1) retrieves music fragments from the Music Fragment Pool, (2) reproduces a new music fragment and (3) posts it to the Pool for later sharing. The Pool always manages music fragments stored in it and provides convenient services to facilitate sharing among users.

### 3.2 Data Structure

Here we introduce Profile XML, which integrates the information related to a music fragment (Fig. 3). A Profile

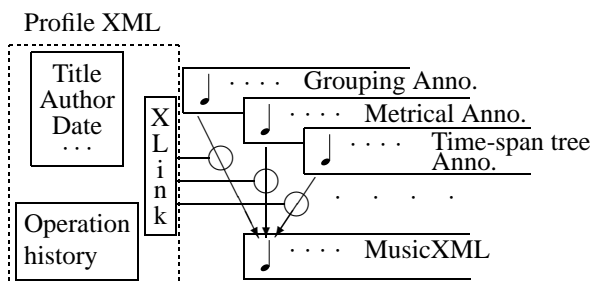


Figure 3: Profile XML, MusicXML, and Annotations

XML file is needed for each music fragment; that is, a music fragment is managed at a Profile-XML basis. Profile XML consists of the following parts: (A) archival information (e.g. title, author, and date), (B) operation history (how a music fragment has been created), and (C) linkage information between a score of the music fragment in MusicXML (content), and annotations of grouping structure, metrical structure, and a time-span tree of a relevant score. We have designed these annotation XMLs for  $\mathcal{MR}$  (Hirata and Aoyagi 2003). Corresponding notes in MusicXML and these annotations are linked to each other by XLink elements in part (C). According to Nagao (2003), part (A) is linguistic annotation, and parts (B) and (C) are commentary annotations.

Operation history represents which music fragments have been imported, and which and how operations have been applied to create a new music fragment. Essentially, operation history is equal to what is displayed on the Operation Diagram Editor (Fig. 5).

### 3.3 System Architecture and Operations

Figure 4 shows the system architecture, in which ODE means Operation Diagram Editor (Fig. 5), a console window of  $\mathcal{MR}$ . On ODE, an operation that is a service provided by

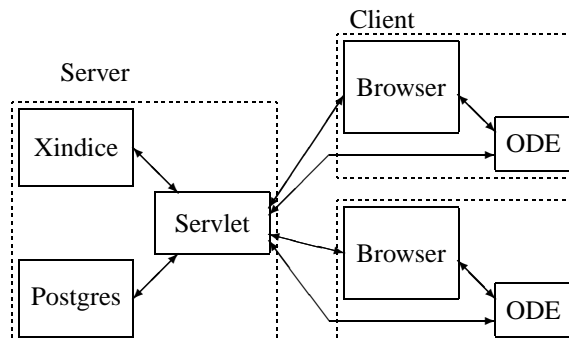


Figure 4: System Architecture

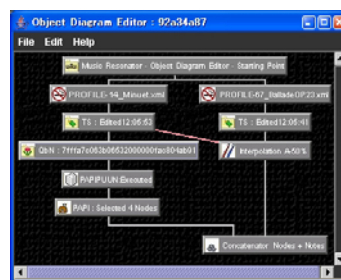


Figure 5: Operation Diagram Editor

$\mathcal{MR}$  is visualized as a box, and boxes are connected from top to bottom in order of execution. At present, operations include import, export, part selection, similarity checking, concatenation, interpolation, and summarization. Connections displayed on ODE can be considered as the creation history of a music fragment.

When a user issues an import operation on ODE, a series of operations to retrieve a music fragment from the Pool starts. Browser is used only when an import operation is issued on ODE, generates a query, and sends it to Server (Fig. 4).  $\mathcal{MR}$  provides three types of query: query by literals (Fig. 6), notes, and a time-span tree. Query-by-literals looks for music fragments in the Pool, in terms of archival information in Profile XML, such as title, author, and date. Furthermore, when more than one answer to a query returns, a user may choose one on Browser, then the selected one is imported back to ODE.

Next, for reproduction, a user applies various operations to the imported music fragments on ODE. For each operation issued by a user on ODE, a corresponding window is spawned (Fig. 7). There are two windows for the part selec-

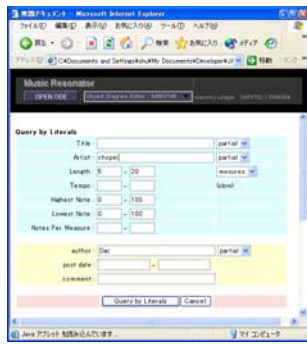


Figure 6: Query by Literals

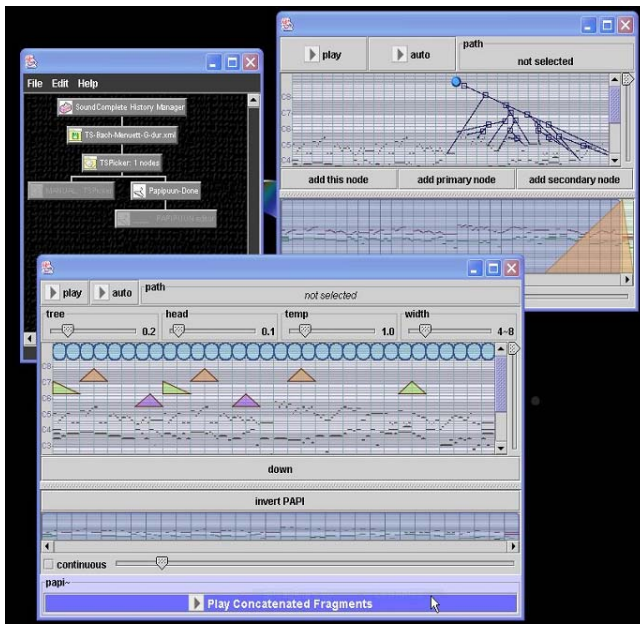


Figure 7: ODE and Operation Windows

tion operation based on the time-span tree (right upper) and the summarization operation (lower) shown in the figure. After completing an operation, the corresponding window is automatically closed, and a new box representing the operation is connected on ODE.

Anytime, a user can store (export) a new music fragment: ODE first generates a new MusicXML file and stores it to Server; next, corresponding annotation XML files are created and stored, and finally, a Profile XML file that includes operation history is created and stored. The current implementation automatically produces as correct annotations as possible (no human-machine collaboration).

Operation history stores all information for exact reproduction of a music fragment, including the values of parameters set by a user during an operation run. Together with the annotated music technique, useful semantic information can be acquired from operation history, such as ancestors of

a music fragment and connection patterns commonly used on ODE.

Within Server (Fig. 4), Servlet transforms data between HTML and XML, and Postgres is introduced merely for efficiency.

## 4 Concluding Remarks

We consider *Music Resonator* to be one of a music creation framework with which a user merely processes an existing music fragment. It is annotation that makes this framework realistic. We conducted a preliminary evaluation using the current implementation with 100 short classical pieces and excerpts as an initial Music Fragment Pool. The subjects offered the following favorable impressions: processing existing music fragments is similar to collage and DJ, the quality of an initial Music Fragment Pool seems important, even amateur users will be able to create natural music, and communication by short musical messages is enjoyable. We plan to design a more precise, larger-scale evaluation.

Future work will include: building more efficient GUI of ODE and query forms and implementing relevant services to promote the sharing of music fragments among users, such as visualizing dependencies between music fragments and data mining of stored music fragments.

## References

- Goto, M. and R. Neyama (2002). Open RemoteGIG: An Open-to-the-Public Distributed Session System Overcoming Network Latency. *SIG Technical Report 43(2)*, 299–309. In Japanese.
- Hirata, K. and T. Aoyagi (2003). Computational Music Representation based on the Generative Theory of Tonal Music and the Deductive Object-Oriented Database. *Computer Music Journal* 27(3), 73–89.
- Hirata, K. and S. Matsuda (2003). Interactive Music Summarization based on Generative Theory of Tonal Music. *Journal of New Music Research* 32(2), 165–177.
- Jordà, S. (2002). FMOL: Toward User-Friendly, Sophisticated New Musical Instruments. *Journal of New Music Research* 26(3), 23–39.
- Lerdahl, F. and R. Jackendoff (1983). *A Generative Theory of Tonal Music*. The MIT Press.
- Nagao, K. (2003). *Digital Content Annotation and Transcoding*. Boston · London: Artech House Publishers.
- Narmour, E. (1990). *The Analysis and Cognition of Basic Melodic Structures - The Implication-Realization Model*. The University of Chicago Press.
- Selfridge-Field, E. (2000). *Beyond MIDI*. The MIT Press.
- Sgouros, N. M. (2000). Detection, Analysis and Rendering of Audience Reactions in Distributed Multimedia Performances. In *Proc of ACM Multimedia 2000*.
- Young, J. P. (2001). Using the Web for Live Interactive Music. In *Proc of ICMC 2001*.