

音楽エンタテインメントソフトウェア SoundComplete の 現状報告

平田 圭二

松田 周

NTT コミュニケーション科学基礎研究所 デジタル・アート・クリエーション

hirata@btl.ntt.co.jp

shu@dacreation.com

あらまし

本稿では、音楽の専門的な知識がないユーザでも簡単に楽曲を創作し交換して楽しむことができる音楽エンタテインメントソフトウェア SoundComplete (SC) の設計方針、機能、ユーザインタフェースなどについて述べる。SC はユーザに楽曲 (断片) を加工・編集するモジュール群を提供する。ユーザはそれらをパッチエディタ上で自由に組み合わせ、Web 上からインポートした楽曲 (断片) を出発点として、新しい楽曲の創作を行う。作業の任意時点での楽曲をエクスポートする。SC の主な特徴としては、既存の楽曲 (断片) に少し手を加えて新しい楽曲を制作するという手法、他ユーザの楽曲を容易に入手できる環境、旋律や拍節といった高次レベルの音楽構造を対象とした操作とそれらのパッチエディタ上での柔軟な組合せ機能、が挙げられる。本稿では、SC の操作例を紹介し、そのユーザインタフェースや起動されるモジュールについて説明を加え、最後に今後の課題に触れる。

A Current Status Report on Music Entertainment Software SoundComplete

Keiji Hirata

NTT Communication Science Laboratories

Shu Matsuda

Digital Art Creation

Abstract

This paper presents music entertainment software SoundComplete (SC) and describes its design policy, functions, and user interface; SC can be played with by a music novice to create and exchange new pieces. SC provides a user with a set of modules for manipulating musical fragments. A user freely combines the modules in Patch Editor and creates a new piece of the musical fragments imported from the Web as a starting point. A user exports a music piece at any moment of processing. The main features of SC include: a method that a user creates a new piece by adding small modifications to existing musical fragments, an environment where others' pieces can be easily obtained, and operations applied to high-level musical structures such as melody and metrics and functions making it possible to flexibly combine the operations in the Patch Editor. This paper introduces a sample session of SC, where the SC's user interface and modules invoked are explained. Finally, we mention future work.

1 はじめに

現在我々は、音楽の専門的な知識がないユーザでも簡単に楽曲を創作し交換して楽しむことを目標とする音楽エンタテインメントソフトウェア SoundComplete (SC) を研究開発している。SC はユーザに楽曲 (断片) を加工・編集するモジュール群を提供する。ユーザはそれらをパッチエディタ上で自由に組み合わせ、Web 上からインポートした楽曲 (断片) を出発点として、新しい楽曲の創作を行う。SC の主な特徴は次の3点である: i. 既存の楽曲 (断片) に少し手を加えて新しい楽曲を制作するという手法、ii. 他ユーザの楽曲を容易に入手できる環境としての peer-to-peer 的な通信、iii. アノテーションによって可能となった旋律や拍節といった高次レベルの音楽構造を対象とした操作とそれらのパッチエディタ上での柔軟な組合せ機能。これらの特徴は、ユーザに簡便性と創造性が両立する

ような楽曲創作の環境をもたらすであろう。我々は、この簡便性と創造性の両立がユーザにとってのエンタテインメント性につながると考えている。

本稿では、SC の設計方針、機能、ユーザインタフェースなどについて述べる。第2章では SC の開発に至る背景と関連研究について述べ、第3章で SC の前身となったパピブーンについて触れたあと、第4章で SC の操作例を通じてそのユーザインタフェースや提供されているモジュール機能等について述べ、最後に第5章では今後の課題について述べる。

2 背景と関連研究

2.1 コンテンツ流通

近年、エンドユーザ側でより大量のデータを消費・制作することが可能になり、エンドユーザ間でのデー

タ交換が盛んになってきた。その理由として、ファイル交換ソフトの普及、メディア加工ツールの発達、エンタテインメント技術への関心の高まりが挙げられる。ファイル交換ソフトとは、peer-to-peer (P2P) 通信で音楽ファイルや画像ファイル等を共有するもので、Napster, Winny, WinMX などがある。メディア加工ツールには、画像データを扱う Photoshop, Illustrator, Web デザインをする Homepage Builder, 楽譜エディタの Finale があり、ワードプロセッサの Word 等もその一種と言える。エンタテインメント技術 [19] は、コンピュータ初心者に対する敷居を下げるのに効果的である。例えば、音楽エンタテインメント分野において、偶然出会った人どうしがその場で音楽的なメッセージを交換して楽しむ CosTune [20] や、片手を振るだけで楽曲の表情付けが制御できる iFP [21] は、誰でも容易に操作できることを主眼としている。擬音/擬態語を入力するだけで実際の楽曲に変換されるストン音楽 [16] のコミュニティでは、一般ユーザからの作品投稿・発表が活発に行われていた。また、現在は Blog (Web Log) も流行しており、これらの現象から、エンドユーザは潜在的にコンテンツ提示や受容に強い欲求を持っていることが示唆される。

2.2 コンテンツ制作

前述のような状況を踏まえ、次に重要になってくるのが、大量の創造性に富む魅力的なコンテンツを簡便に創出・供給する手法やシステムであろう。応用領域を限定した専門家向けシステムはある程度成功を納めているものの [25]、現状で大衆ユーザがコンテンツ創作する手法には、システムがユーザに提供するプリミティブを利用する手法 (Photoshop や Illustrator など) 及び発想のヒントになるであろう刺激データをユーザに提示する手法の 2 つが一般的であろう。

前者の手法では、例えば描画ソフトであれば、定規、コンパス、ボスタライズ、モザイク処理といった基本的な描画手段やフィルタ/特殊効果の機能だけが提供されている状況である。楽譜エディタであれば、音符、休符を五線譜上に置く機能が提供されているだけである。これらの手段は、操作適用の結果がその対象や文脈にあまり依存しないという意味で低レベルであると言ってよいだろう。後者の手法の研究例 [18] では、アートのデザイン作業において発想のヒントになるであろう画像データを提示する際、デザイン知識の文脈性、信頼性、作業への没入感が重要であるという結果が得られている。しかし、ユーザにとっての簡便さという観点からは、いずれの手法も基本的には作品をゼロから構成するものであり、コンテンツ創作過程を高次レベルにおいて直接的には支援していない。

2.3 既存コンテンツの利用

我々は、簡便性と創造性を兼ね備えた手法の候補として、素材や既存事例に (小さな) 変更を加えてコン

テンツ創作する手法に注目する。そのようなシステム例として、Web 上の画像やテキストを対象とするシステム (Hunter Gatherer, Collage Machine) や写真や音楽を対象としたシステム (S&A, SIC, iFP (上述)) がある。

Hunter Gatherer は、すでに存在する Web 上の画像やテキストの必要な部分だけを、本来の作業を一時中断することなく非常に簡単な操作で切り貼りするシステムである [23, 24]。この時、対象コンテンツである画像やテキストは HTML や XML で記述 (知識表現) されているので、その切り出す部分は XPath を用いて指定する。

CollageMachine は、Finke らの creative cognition [4] の理論に基づき、ユーザが嗜好する Web 上の画像やテキストを自動的にかつ継続的に集めコラージュを作成するシステムである [13, 14]。コラージュは、モノを置く文脈を再定義し直してそこに新しい意味を生み出す現代芸術での基本テクニックの 1 つである。コラージュ自身が作品となるだけでなく、創発的、偶然的なコラージュがユーザの新しい発想を促す場合もある。

S&A (Stylization and Abstraction) は、視線によるユーザとのインタラクションを通じて、与えられた写真の構造を明確化する手法である [3]。まず、与えられた写真をその色情報やエッジ情報を元に適当な領域に分割し階層構造を求めておく。次に、ユーザ注視の情報から、その注目部分を識別し強調したり、注目していない部分を線画として簡略化することで、新しい画像の表現スタイルを実現する。

SIC (Synergistic Image Creator) は、入力画像からベクトル形式の特徴量を抽出し、その入力画像を筆触と呼ばれる微少均一色領域に還元する [12]。そして、その筆触情報を元にしてアルゴリズム的に新しい絵画のレンダリングを行う。

以上のシステムが、素材に変更を加えてコンテンツ創作するという意味で、適切に機能している理由は、操作対象であるコンテンツの意味的な構造 (HTML/XML の構文的情報、注視部分、筆触) を考慮して素材を分解・再構成しているからである。コンテンツの意味構造を単位として様々な操作を適用するので、無意味な分解と無意味な結果出力が少ないと考えられる。

2.4 アノテーション

コンテンツの意味構造を、機械可読な情報として積極的に記述する枠組がアノテーションである。Web 上のコンテンツ (オンラインコンテンツ) に対しアノテーションを付加することで、その構造を考慮した知的な蓄積、交換、再利用、加工、変換が可能になってきている [17, 6]。

アノテーションを用いると、専門知識を持たない (コンテンツの意味構造を知らない) ユーザでもシステム支援により高次レベルの処理が可能になる。我々は、このアノテーションによる高次レベル処理を、簡便性と創造性を兼ね備え、素材に変更を加えるコン

テンツ創作過程を直接的に支援する枠組として有望視している。そして、簡便性と創造性の両立がエンタテインメント性をもたらすと考えている。

3 パピブーン

音楽エンタテインメントソフトウェア *SoundComplete* (SC) は、音楽要約システム「パピブーン」[9, 10]の機能や仕組の一部を受け継いでいるので、本章ではパピブーンを概観する。

パピブーンは要約による速聴を行うシステムである。要約は個人によって様々な解が得られる正解の無いタスクであり、既存の楽曲や他人が作成した素材等に(少し)手を加えるという手軽なタスクでもある。パピブーンでは、楽曲を表現するデータ(SMF, タイムスパン木, 時間構造)が1つのXMLファイルとして記述されており、意味構造(タイムスパン木, 時間構造)を単位として要約を実行するよう制限されている。これより、パピブーンでは、初心者でもある一定の質を達成しつつ、簡便にアイデアや創造性を要約結果に盛り込むことが可能となった。

パピブーンは起動時にまず、タイムスパン木に基づく楽曲各部分どうしの総当たりの類似度判定の前処理を行う。その後、ユーザとのインタラクションは次のような流れで進む(図1): 類似度計算の結果から楽曲構造の発見(ステップ1, 2), 残すあるいは削除する楽曲部分の同定(ステップ3), 要約として残す楽曲部分の接続(ステップ4), 上記の繰り返し(ステップ5, 6, 7)である。

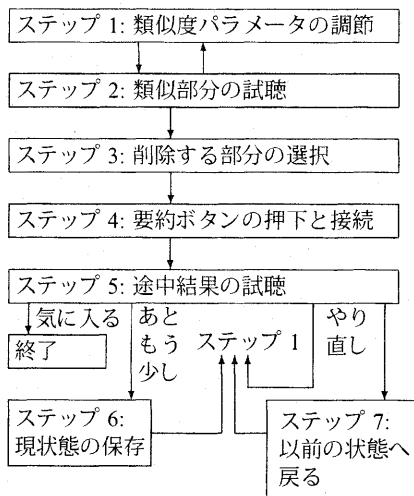


Figure 1: パピブーンにおけるユーザとのインタラクション

起動時の前処理では、楽曲のタイムスパン木を参照しながら楽曲の各部分どうしの類似度を計算する。各部分の旋律(ポリフォニー)の類似度判定のアルゴ

リズムには、タイムスパン木どうしの最小上界(least upper bound)を用いる。

ステップ1, 2にてユーザは3つの類似度パラメータを変化させ、タイムスパン木重視, 時間構造重視などの重み付けを指示する。探し出された類似楽曲部分はまとめてウィンドウ上に表示される。ユーザは類似度パラメータを繰り返し調節し類似した楽曲部分を確定し、楽曲構造を明らかにしていく。これは動的問合せ(dynamic query)[1]と同様である。ステップ3にてユーザが要約創作のために削除すべき楽曲部分を同定し、ステップ4にて要約ボタンを押下する。この時、パピブーンは削除されなかった残り部分を接続する。

ステップ5にてユーザは途中結果を試聴する。もし途中結果がユーザの好みに合致すると要約は終了する。そうでない場合は類似度判定のパラメータを調節するステップ1に戻る。ステップ6でユーザが明示的にその状態を保存した時のみ(例えば図2), 後でそのステップ6の状態に戻ることができる(undo処理)。保存されるのは要約の状態(削除された部分)

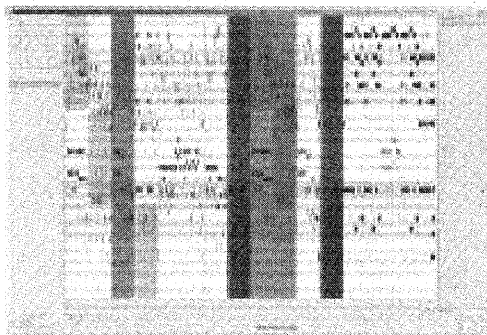


Figure 2: パピブーンの途中状態(ステップ6)

と類似度判定のパラメータであり、それが線型な操作履歴として保存される。ステップ2や3の途中状態を保存することはできない。パピブーンのユーザは単独で要約を創作するよう想定されているので、ステップ1と2の作業は一体化しており、戻る状態はユーザ自身が予め指定しておく必要がある。

4 音楽エンタテインメントソフトウェア *SoundComplete*

音楽エンタテインメントシステム *SoundComplete* (SC) は、音楽の専門的な知識がないユーザでも簡便に楽曲を創作し交換して楽しむためのソフトウェアである(図3)。その特徴を3つの観点から述べる。

(1) エンドユーザ間での楽曲断片交換: 創作途中の任意の段階で楽曲を Web 上にエクスポートすることができる。また創作の素材となる楽曲をいつでも Web

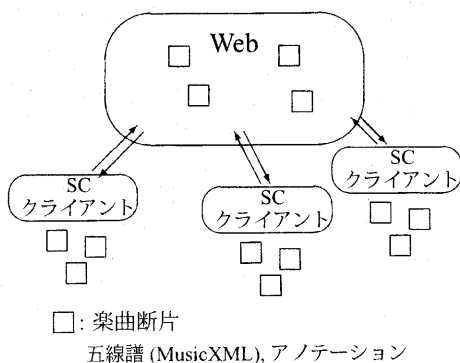


Figure 3: システム概略

上からインポートすることができる。これよりユーザ間 (SC クライアント間) での P2P 的な楽曲の交換が可能となる。

(2) 楽曲の創作: 音楽要約ソフトウェア「パピブーン」(3 節) には楽曲の構造を解析する機能、楽曲の部分を選択する機能、楽曲断片を接続する機能が組込まれていた。SC では、それら機能を、ユーザが自由に組み合わせられる 3 つの独立したモジュールとして提供する。これらモジュールを組み合わせるためのパッチエディタは、パッチ言語 MAX [22] 風の外観をしており、楽曲の創作履歴を可視化する。これよりユーザは簡便な操作で楽曲の創作が可能となる。

(3) 意味構造のアノテーション: 楽曲情報は、五線譜の情報とその意味構造 (音楽理論 GTTM のタイムスパン木 [15] と時間構造の情報 [11]) から成り、五線譜の情報は MusicXML で記述し、その意味構造は MusicXML へのアノテーション (XML で記述) として表現する。創作途中の楽曲でも Web 上にエクスポートするには常に正しいアノテーションが付加される。この仕組みより、旋律レベル、拍節レベル (高次レベル) でのシステム支援が可能となる。

アノテーションは五線譜情報の任意部分を柔軟に参照できなければならない。それには、五線譜情報を XML で記述し、アノテーションは XPath 等を利用するのが現実的であろう。現在、五線譜情報を XML で記述する規格として MusicXML [5], WEDELMusicXML [2] 等が提案されている。SoundComplete では、その相互運用性、ツール等の開発・配布状況を考慮して MusicXML を採用した。

4.1 操作例

本節では現在開発中の SC の操作例を紹介し、そのユーザインタフェースや起動されるモジュールを説明する。

パッチエディタ: 図 4 に SC の起動直後のパッチエディタを示す。パッチエディタ中のノードをクリック

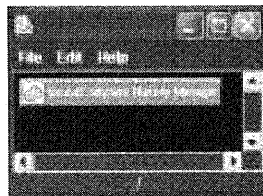


Figure 4: 起動直後のパッチエディタ

すると、その時点で実行可能なモジュールがメニューとして提示される。図 4 の時点では楽曲断片のインポートのみが実行可能なので、「楽曲断片のインポート」という項目だけが含まれたメニューが表示される。そのメニュー項目を選択すると、楽曲データのロケーションを入力するためのウィンドウが現れる。図 5 は楽曲 (Bach's Menuetto in G-dur) をインポートした直後のパッチである。図のように、個々の操作を

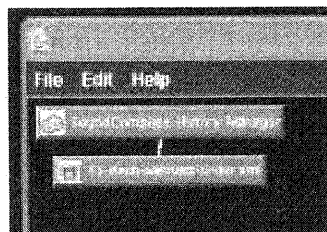


Figure 5: 楽曲のインポート

表すモジュール (箱) が操作実行順に上から下に自動的に接続されていく。パッチエディタ上で、ユーザは任意のモジュールを削除したり変更を加えたりすることもできる。もし途中のモジュールを削除すると、それより下のモジュールも一斉に削除される。

楽曲データをインポートすると、次に実行可能なモジュールは次の 4 つである: 1. 楽曲データのエクスポート (途中状態を他のユーザに受け渡す時に実行する; 図 15), 2. 要約モジュール (パピブーン相当; 図 7), 3. タイムスパン木の選択 (図 9, 10), 4. タイムスパン木の圧縮。

ただし、タイムスパン木を圧縮するためには、事前にタイムスパン木の一部分を選択しておく必要がある。図 5 の時点ではまだどの部分も選択されていないので、本操作例では上記 3 の選択モジュールの実行を経た後で起動される (図 12, 13)。

要約モジュール: メニュー項目から要約モジュールを選択すると、要約モジュールのウィンドウが現れる。要約モジュールは、前処理としての楽曲構成判

定器 (図 6) と楽曲の部分の選択・削除を行う本体モジュール (図 7) から成る。楽曲構成判定器は、パピ



Figure 6: 楽曲構成判定器の起動

プーンにおけるタイムスパン木に基づく楽曲各部分どうしの総当たりの類似度判定に相当し、本体モジュールはステップ 1~5 に相当する (3 節)。

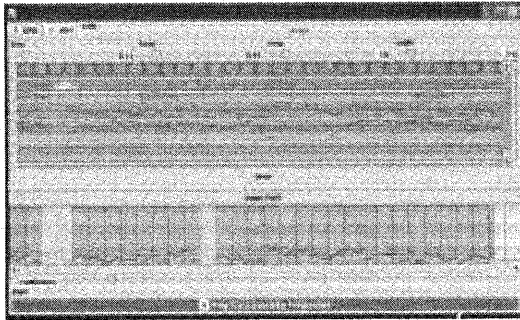


Figure 7: 要約本体モジュール

要約モジュール本体 (図 7) は、主に以下の 5 点に関してパピプーンを改良したものである: 1. 楽曲構成判定器と本体の分離 (ユーザが操作するパラメータ数の減少), 2. focus+context 風表示 (上サブウィンドウが focus で下サブウィンドウが context. とともにピアノロール譜), 3. 楽曲構成の表示法改良 (タイムスパン木単位と小節単位), 4. 選択範囲の反転機能 (全ての楽曲部分が選択可能に), 5. 選択された部分タイムスパン木の head 音の位置表示の付加。

図 8 は要約モジュール本体を起動中のパッチエディタの様子を示している。図中、上から 3 番めの箱 Papipuu Done が楽曲構成判定器の実行を、その下の Papipuu editor (文字が灰色のため見難い) が本体モジュールの実行を表している。しかし現在は、本体モジュールの出力を楽曲データとして出力する機能が未実装なため、その下にさらにモジュールを接続することができない。

タイムスパン木の選択: インポートした楽曲データを受けてタイムスパン木の選択を行うモジュール (TS-Picker) を起動し、そのエンディング付近を選択した様子を図 9 に示す。

TS-Picker も、要約モジュールと同様に、focus+context 風表示 (上サブウィンドウが focus で下サブウィンド

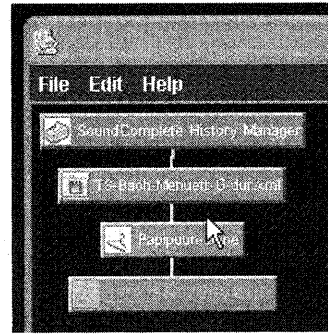


Figure 8: 要約本体モジュール実行中のパッチエディタ

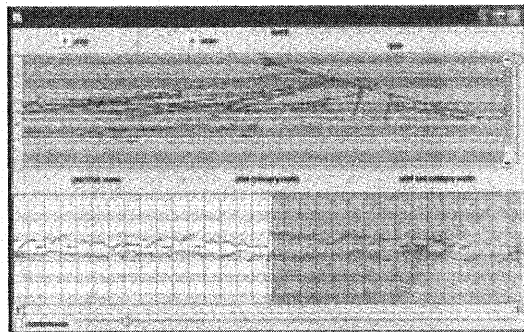


Figure 9: TS-Picker によるタイムスパン木の選択 (エンディング付近)

ウが context) を行う。ただし、下には楽曲全体のピアノロール譜が表示され、上には下で選択された部分のタイムスパン木の拡大図が表示される。TS-Picker はパピプーンにおけるステップ 3 に相当する (3 節)。ステップ 1~2 に対応するモジュールは現在未実装である。よって現時点でユーザがタイムスパン木を選択するには、TS-Picker の上サブウィンドウで部分木を試行錯誤的に選択するしかない。

さらに続けてタイムスパン木の先頭付近を選択した様子を図 10 に示す。この時、パッチエディタの操作履歴は図 11 のようになる。楽曲データのインポート後、要約モジュールと TS-Picker に分岐し、エンディング付近の選択 (図中 TS-Picker: 1 node のモジュール) と先頭付近の選択 (図中 TS-Picker: 2 nodes のモジュール) が続く。このように、パッチエディタ上では各操作が 1 つのモジュール (箱) として可視化される。

タイムスパン木の接続と削除: ここで、タイムスパン木の接続を、選択されたタイムスパン部分木を枝として持つような 1 つのタイムスパン木を合成する

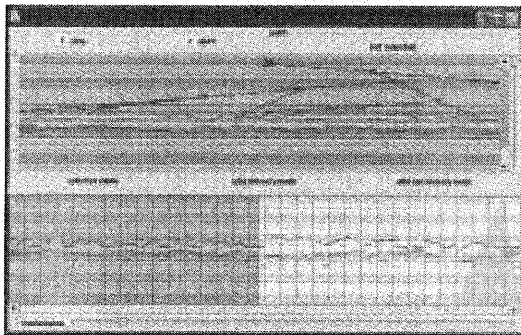


Figure 10: タイムスパン木の選択 (先頭付近)



Figure 11: 操作履歴の分岐

操作と定義する。次に、削除を、選択されたタイムスパン木の部分を単純に削除する操作と定義する。

図 12 の接続モジュールでは、ピアノロール譜の音符が存在している部分が選択された部分であり、図 13 の削除モジュールでは、存在していない部分が選択された部分である。つまり、音符の存在している部分どうしは相補的な関係にある。接続モジュール (図 12) では、先頭付近の部分木とエンディング付近の部分木の間 (ウィンドウ中、灰色で表示) を圧縮したようなタイムスパン木が得られる。同様に、削除モジュール (図 13) では、先頭付近の部分木 (ウィンドウ中、灰色で表示) とエンディング付近の部分木 (表示範囲外) を圧縮したようなタイムスパン木が得られる。部分木と部分木の境界で、接続された楽曲が聴感上不連続に聴こえる場合もあるが、現在特に対策は講じていない。

接続モジュール、削除モジュールともに、部分木間の間隔がある閾値を越える場合に対してのみ接続あるいは削除を実行する。その閾値は、ウィンドウ上方のスライダーによって与えられる。閾値が決定したら、ウィンドウ右下の commit ボタンを押下す

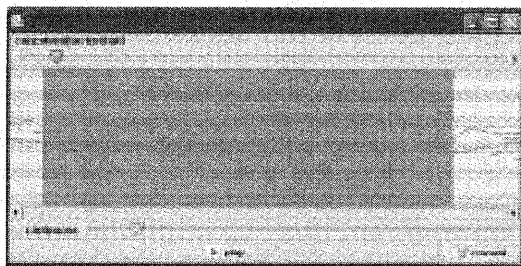


Figure 12: 選択されたタイムスパン部分木の接続

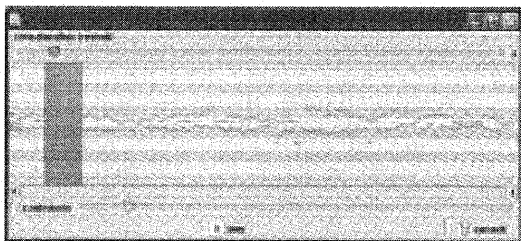


Figure 13: 選択されたタイムスパン部分木の削除

る。この時、圧縮した新しいタイムスパン木が生成されるが、現在そのタイムスパン木に対応する短い MusicXML を生成する機能は未実装である。

接続モジュールと削除モジュールの実行が終了した時点でのパッチエディタの様子を図 14 に示す。接

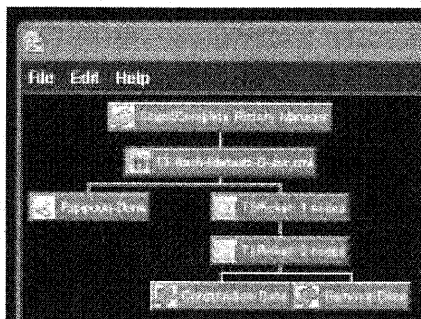


Figure 14: タイムスパン木の接続と削除後

続モジュールと削除モジュールを実行した所で分岐が生じている。

エクスポート： ユーザは任意の時点で任意のモジュール直下にエクスポートモジュールを付加することができる (図 15)。任意の時点でエクスポートでき

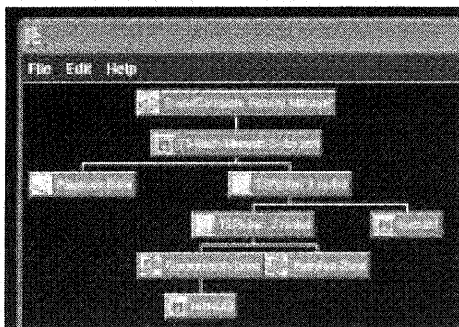


Figure 15: 任意の状態でのエクスポート

るために記録しておくべきデータは (少なくとも), a. 五線譜の情報 (MusicXML), b. その意味構造 (タイムスパン木と時間構造の情報) のアノテーション, c. TS-Picker で選択された部分の情報 (XPath), d. 類似度パラメータ (楽曲構成に従ってタイムスパン木を選択する場合) である。しかし現在の SC でエクスポートされる情報は a., b. のみなので, 一旦エクスポートすると, c., d. の情報は失われてしまう。

以上のように, 楽曲データのエクスポートとインポートを含め, パッチエディタ上でモジュール (箱) を配線することがパピプーんのステップ 5~7 に相当する。

5 おわりに

現在開発中の SoundComplete¹ について, その実装予定の部分も含めて, その設計方針, 機能, 操作例等について述べた。

当面の課題は以下を含む:

- タイムスパン木の選択を行うモジュール (TS-Picker) への入力を生成する楽曲構成判定モジュールの実装。
- エクスポートの際, 現在出力されるデータ (五線譜の情報とその意味構造のアノテーション) に加えて, TS-Picker で選択された部分の情報 (XPath) と類似度パラメータを追加。
- 接続モジュールや削除モジュールで新しいタイムスパン木が生成された時, それに対応する (短い) 新しい MusicXML を生成する機能の実装。

¹SoundComplete という名称について, ここでは二つの意味が込められていると述べるだけにとどめる。

- TS-Picker, 接続モジュール, 削除モジュールにおいて, 楽譜のピアノロール表示だけでなく五線譜表示も可能。

- 他の高次レベル操作の提供。例えば, 補間, 滑らかな接続 (部分木どうしの境界が聴感上滑らかに聴こえるような接続), 引用 (切り出し/埋め込み), 模倣など。

さらに, 将来的な課題は以下を含む:

- 他ユーザの楽曲断片をインポートする際に起動される楽曲検索機能。
- より豊富な内容のアノテーション及びその適用・管理の導入。例えば, 歌詞, 楽曲インデックス情報, アノテーション著者/日時等の内容を含むことで, 豊富な高次レベル操作やアノテーション適用時の認証が可能になる。
- 現在セキュリティや著作権保護については考慮していないが, 実用化の際には何らかの対策を講じる必要がある。

本稿では, 簡便性と創造性の両立がエンタテインメント性をもたらすと述べた。大衆ユーザのエンタテインメントにおいては, さらに, 得られるコンテンツの質をある一定レベル以上に保証することも大切であろう。我々は, アノテーションで表現される音楽知識がユーザの創作する楽曲の質を大きく左右すると考えているので, 今後も引き続き, 音楽知識表現の課題に取り組んでいきたい。

参考文献

- [1] C. Ahlberg and B. Shneiderman, Visual Information Seeking: Tight Coupling of Dynamic Query Filters with Starfield Displays, In *Proc. of CHI'94*.
- [2] P. Bellini and P. Nesi, WEDELMUSIC Format: an XML Music Notation Format for Emerging Applications, In *Proc. of First Int'l Conf. on Web Delivering of Music* (2001).
- [3] D. DeCarlo and A. Santella, Stylization and Abstraction of Photographs, In *Proc. of SIGGRAPH 2002*.
- [4] R. Finke, T. Ward, and S. Smith, Creative Cognition, The MIT Press (1992).
- [5] M. Good, Representing Music Using XML, In *Proc. of ISMIR 2000*. Also available at <http://www.recordare.com/xml.html>.
- [6] 萩野他, セマンティック Web とは, 情報処理 Vol.43, No.7, pp.709-717 (2002).
- [7] 平田, 青柳, パービーブーン: 音符レベルでユーザ意図を把握して編曲を行う事例ベースシステム, IPSJ 研究報告, 2000-MUS-37.

- [8] 平田, 平賀, ハービーフン: 2 段階演奏表情付け法によるインクリメンタルな演奏生成システム, *IPSJ 研究報告, 2001-MUS-39*.
- [9] 平田, 松田, パピブーン: GTTM に基づく音楽要約システム, *IPSJ 研究報告, 2002-MUS-46*.
- [10] K. Hirata and S. Matsuda, Interactive Music Summarization based on GTTM, In *Proc. of ISMIR 2002*, pp.86-93.
- [11] K. Hirata and T. Aoyagi, Computational Music Representation based on the Generative Theory of Tonal Music and the Deductive Object-Oriented Database, *Computer Music Journal*, 27(3): 73-89 (2003).
- [12] 笠尾, 中嶋, シナージスティックイメーজクリエイター 描画プロセスを重視した絵画作成システム一, *信学論, J81-D-II, No.4*, pp.671-680 (1998).
- [13] A. Kerne, CollageMachine: An Interactive Agent of Web Recombination, *LEONARDO*, Vol.33, No.5, pp.347-350 (2000).
- [14] A. Kerne, collage machine: interest-driven browsing through streaming collage, In *Proc. of Cast01* (2001).
- [15] F. Lerdahl and R. Jackendoff, *A Generative Theory of Tonal Music*, The MIT Press (1983).
- [16] T. Masui, Music Composition by Onomatopoeia, In *Proc. of the First International Workshop on Entertainment Computing (IWEC2002)*. Also available at <http://pitecan.com/Sutoton/>.
- [17] K. Nagao, *Digital Content Annotation and Transcoding*, Artech House Publishers (2003).
- [18] K. Nakakoji, Y. Yamamoto, and M. Ohira, A Framework that Supports Collective Creativity in Design using Visual Images, In *Proc. of Third Creativity and Cognition Conference*, ACM Press, pp.166-173 (1999).
- [19] R. Nakatsu and J. Hoshino (Eds), *Entertainment Computing*, Kluwer Academic Publishers (2003).
- [20] 西本, 前川, 辻, 間瀬, 夢沼, 装着型楽器 CosTune による「行きずりセッション」の実現, *エンタテインメントコンピューティング 2003 論文集, IPSJ Symposium Series, Vol.2003, No.1*, pp.59-64.
- [21] 奥平, 片寄, 橋田, 音楽演奏インタフェース iFPー演奏表情のリアルタイム操作とビジュアルライゼーション -, *IPSJ 研究報告, 2003-MUS-51*.
- [22] M. Puckette, Combining Event and Signal Processing in the MAX Graphical Programming Environment, *Computer Music Journal*, 15(3): 68-77 (1991).
- [23] m.c. schraefel and Y. Zhu, Interaction Design for Web-Based, Within-Page Collection Making and Management, In *Proc. of 12th ACM Conference on Hypertext and Hypermedia* (2001).
- [24] m.c. schraefel and Y. Zhu, Hunter Gatherer: A Collection Making Tool for the Web, In *Proc. of CHI 2002*.
- [25] B. Shneiderman, Creating Creativity: User Interfaces for Supporting Innovation, *ACM Transactions on Computer-Human Interaction (TOCHI)*, Vol.7 Issue 1 (2000).