

修士論文 2018年度（平成30年度）

ハイパー楽譜システムの研究

慶應義塾大学大学院 政策・メディア研究科

佐竹 紘明

インタラクションデザインプロジェクト

2019年1月

ハイパー楽譜システムの研究

論文要旨

楽譜を簡単に編集したりメモなどの情報を自在に書いたりハイパーリンクを利用したりできる「ハイパー楽譜システム」を提案する。昔から広く使われている楽譜は紙を前提とした静的なメディアであり、(1)簡単に編集できない、(2)メモなどの情報を自在に書けない、(3)参照や管理が面倒、という問題が存在する。計算機上の楽譜ビューアーや楽譜作成システムが広く利用されているが、これらは紙での楽譜の利用形態を再現したに過ぎず、この問題を解決できていない。

「ハイパー楽譜システム」は Wiki と楽譜記述言語の組み合わせによってテキストベースで簡単に楽譜を書いたり、自在にテキストやマルチメディアを埋め込んだり、ハイパーリンクを使って関連する楽譜や情報を簡単に参照できるシステムである。これによって既存の楽譜の問題点が解決されるだけでなく、新しい楽譜の使い方が可能になる。本論文では「ハイパー楽譜システム」の設計や実装、その応用例について述べ、研究の発展性について考察する。

キーワード

音楽、楽譜、Wiki、ハイパーテキスト、IME

慶應義塾大学大学院 政策・メディア研究科

佐竹 紘明

Abstract Of Master's Thesis Academic Year 2018

HyperScore: A Web-based Interactive Music Score

Summary

We propose a new music score system "HyperScore." Using HyperScore, users can quickly write music scores including text, multimedia, Hypertext, etc. Since the music score is printed matter, there are problems such that it cannot be edited, hard to add information other than musical notes, and difficulty of organizing. Although music score viewers and music notation systems are getting popular these days, they cannot solve those problems because such systems merely reproduce the form of the music score on the computer. HyperScore solves problems on the music score with Wiki and musical notation language. Using HyperScore, users can write music scores easily with text, embed any information including multimedia, access to related music scores or other information via Hyperlink. In this paper, we describe the design, implementation, and application of HyperScore.

Keywords

Music, Music Score, Wiki, Hypertext, Input Method Editor

Graduate School of Media and Governance
Keio University

Hiroaki Satake

目次

第1章 序論	3
1.1 研究の動機	4
1.2 研究の目的	4
1.3 本論文の構成	5
第2章 研究背景	7
2.1 楽譜	8
2.2 楽譜の歴史	8
2.2.1 ネウマ譜	8
2.2.2 定量譜	9
2.2.3 奏法譜	9
2.2.4 近代譜	10
2.2.5 現代音楽	10
2.3 テキストの進化	11
2.4 楽譜の問題点	12
2.5 既存の楽譜システム	13
2.5.1 楽譜作成ソフト	13
2.5.2 楽譜ビューアー	14
2.6 まとめ	15
第3章 設計	17
3.1 要件	18
3.2 ハイパー楽譜システム	18
3.2.1 基本構成	18
3.2.2 使い方	22
第4章 実装	25
4.1 アプリケーション構成	26
4.2 楽譜の描画	26
4.3 音符ハイパーリンク機能の実現	27
第5章 楽譜 IME	29
5.1 ABC の問題点	30

5.2	楽譜 IME	30
5.2.1	利用できる機能	30
5.2.2	利用フロー	31
5.3	実装	32
5.3.1	文字入力	32
5.3.2	楽譜の表示	32
5.3.3	楽譜辞書	33
5.4	議論	33
5.4.1	使用感	33
5.4.2	辞書の編集	33
5.4.3	動的な辞書	33
5.5	まとめ	34
第 6 章	応用例	35
6.1	楽器練習	36
6.2	勉強	36
6.3	楽譜共同編集	37
6.4	楽曲の調査/分析	37
6.5	まとめ	38
第 7 章	関連研究	39
7.1	主要な研究領域	40
7.2	楽譜編集/閲覧システム	40
7.2.1	BRASS	40
7.2.2	WIKI: :SCORE	41
7.3	楽譜記述言語	42
7.3.1	MusicXML	42
7.3.2	Lilypond	43
7.3.3	VexTab	44
第 8 章	考察	45
8.1	評価	46
8.1.1	筆者の運用経験	46
8.1.2	感想・意見	47
8.1.3	問題点	47

8.2 考察	48
8.2.1 設計の妥当性	48
8.2.2 解決すべき課題	48
8.2.3 楽譜の問題点の検証	49
第9章 結論	51
9.1 研究の成果	52
9.2 総括	52
謝辞	53
参考文献	54

目次

2.1	ネウマ譜の例	8
2.2	定量譜の例	9
2.3	奏法譜の例	9
2.4	近代譜の例	10
2.5	現代音楽譜の例	11
2.6	ハイパーテキストの例	12
2.7	楽譜上のメモの例	13
2.8	Finale の画面	14
2.9	Piascore の画面	15
3.1	ハイパー楽譜システム上で書かれた楽譜	19
3.2	Scrapbox の画面	20
3.3	関連ページリスト	20
3.4	「ド」の譜例	21
3.5	複雑な楽譜の例	21
3.6	編集中の楽譜	22
3.7	編集後の楽譜	22
3.8	楽譜にテキストや動画が埋め込まれた例	22
3.9	1音目の「C」に「middle C」ページへのリンクが設定されたABC	23
3.10	ハイパーリンクが設定された音符	23
3.11	リンク先の「middle C」ページ	24
4.1	コードブロック記法	26
4.2	ABCとして解釈されるコードブロック	27
4.3	生成された楽譜画像	27
5.1	楽譜IME初期画面	31
5.2	文字入力中の画面	31
5.3	変換候補選択中の画面	32
5.4	変換が完了した画面	32
6.1	技術ページの例	36
6.2	知識ページの例	37
6.3	楽曲ページ	38

7.1	BRASS の画面	41
7.2	WIKI: :SCORE の画面	42
7.3	「ドレミファ」の譜例	42

ソースコード目次

3.1	「ド」を表す ABC	21
5.1	楽譜 IME の辞書データ	33
7.1	ABC における図 7.3 の楽譜の記述例	42
7.2	MusicXML における図 7.3 の楽譜の記述例 (一部)	43
7.3	Lilypond における図 7.3 の楽譜の記述例	44
7.4	VexTab における図 7.3 の楽譜の記述例	44

第1章 序論

本章では本研究の動機と目的、および本論文の構成について述べる。

1.1 研究の動機

楽譜は17世紀ごろに五線譜による近代記譜法のフォーマットが確立してからほとんど形態が変わっておらず、今日においても国際的かつ普遍的な記法として広く利用されている。これは楽譜が優れた表現力を持ち、様々な音楽を記録するのに適していることにほかならない。しかし当初から印刷を前提としたフォーマットであったため、以下のような問題も存在する。

- 簡単に編集できない
- メモなどの情報を自在に書けない
- 参照や管理が面倒

また計算機が普及した現在では楽譜を快適に閲覧したり、美しい楽譜を作成できるソフトウェアも広く利用されているが、PDFのように内容の変更が不可能な形式で楽譜を管理することが一般的であり、紙の楽譜と本質的な特徴は変わっていない。一方、楽譜と同様に紙の上に記録されていたテキストは計算機の登場によって以下のような進化を遂げた。

- コピー/ペースト/Undo/Redoといったテキスト編集支援機能が手軽な編集を可能にした
- IMEといったテキスト入力支援機能が手軽な入力を可能にした
- ハイパーテキストがマルチメディアを活用した分かりやすいドキュメントの作成や、ハイパーリンクを可能にした
- Webが様々なドキュメントへの素早いアクセスを可能にした
- Wikiのようなコラボレーションツールが複数人による共同編集を可能にした

これらの進化によって、紙というメディアの特性上ドキュメントが物理的に独立していて内容の変更も難しいという性質が消えてしまった。かつては楽譜と同じ問題を抱えていたが、計算機によって新しい便利な使い方が発明され、広く普及したのである。楽譜はフォーマットや使い方が数百年変わっていないので、計算機によって進化する余地はまだ存在すると考えられる。

1.2 研究の目的

本研究では、上記のような楽譜が持つ問題点を解決し、これまでの楽譜の在り方にとらわれない次世代の楽譜システム「ハイパー楽譜システム」の構築を目的とする。ハイパーテキストやWikiの手法を取り入れることで、従来の楽譜システムでは実現できなかった快適な楽譜編集・利用環境を実現する。

1.3 本論文の構成

本論文は以下の9章で構成される。

第2章では、本研究の背景をより詳細に分析し、楽譜の問題点を整理する。

第3章では、本論文で提案するシステムの基本構成と使い方について述べる。

第4章では、本論文で提案するシステムの詳細な実装について述べる。

第5章では、本論文で提案するシステムで利用できる楽譜IMEを提案する。

第6章では、本論文で提案するシステムによって実現可能な応用例について述べる。

第7章では、関連する研究を紹介し、それらの特徴や本研究との関連を述べる。

第8章では、筆者による運用経験やユーザーからのフィードバックをまとめ、本論文で提案するシステムの有効性と問題点について述べる。

最後に、第9章で本論文のまとめと結論を述べる。

第2章 研究背景

本章では楽譜の問題点と、楽譜を扱う既存システムの現状を整理する。

2.1 楽譜

楽譜は楽曲を記号や文字によって書き表したものである。一般に五線譜のことを指すことが多いが、ギター演奏用に利用されるタブ譜や日本固有の三味線譜など、用途やジャンル、地域によって様々な種類の楽譜が存在する。西洋音楽をルーツに持つ五線譜は国際的かつ普遍的なものとして認められており、現在は様々な音楽を記録するために広く利用されている。

2.2 楽譜の歴史

本節では今日利用されている楽譜が生まれるまでの進化の歴史を概観する [22]。

2.2.1 ネウマ譜

ネウマ譜 (図 2.1) は旋律の上行下行の動きをネウマと呼ばれる線状の記号で表示する記譜法で、9 世紀頃から中世キリスト教会の典礼音楽であったグレゴリオ聖歌の記譜のために用いられはじめた。10-11 世紀ごろからは横線 (譜線) を添えて、音程関係を明確にする試みが行われた。譜線が添えられている限りは音高が明確だが、リズムの表示はあいまいである。



図 2.1: ネウマ譜の例

2.2.2 定量譜

複数の奏者がそれぞれ異なった声部を同時に演奏する多声音楽が生まれると、リズム表示があいまいな従来のネウマ譜では不十分となった。13世紀後半になると、音の長短を音符の形状で区別する定量譜(図2.2)が利用されるようになった。

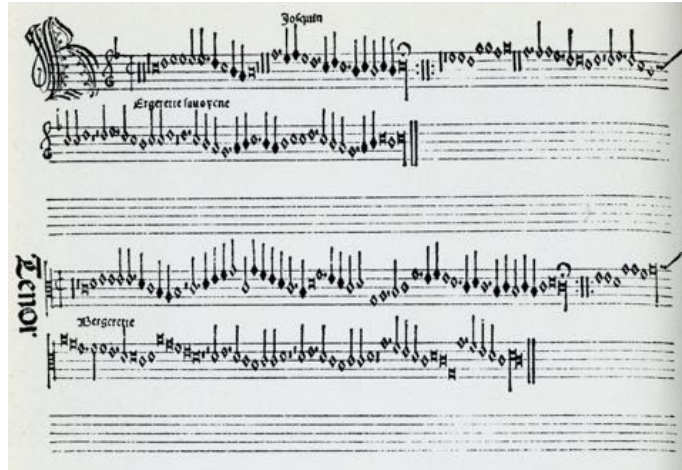


図 2.2: 定量譜の例

2.2.3 奏法譜

15世紀ごろからは特定の楽器の演奏法を具体的に示す奏法譜(図2.3)も広く利用された。楽器ごとに固有の奏法譜をもち、現代でも「タブ譜」としてギター演奏に利用されている。

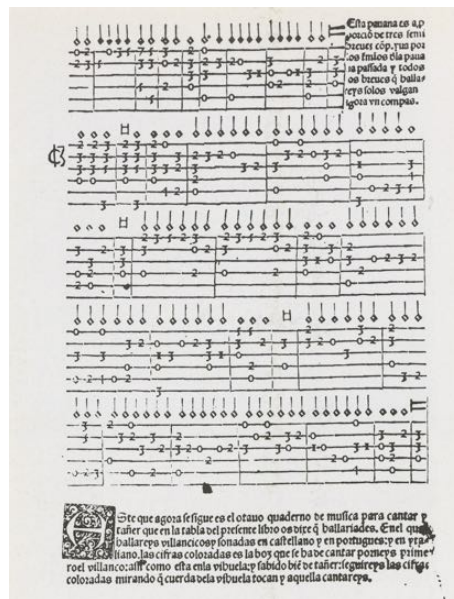


図 2.3: 奏法譜の例

2.2.4 近代譜

17世紀初頭から現在までに世界で最も普及している楽譜の形態(図2.4)で、鍵盤楽器用の奏法譜をルーツに持つ。

- 五線による音高表示
- 音符の「はた」によるリズム表示
- 小節線の使用

といった特徴がある。

3

CONCERTO I:



The image shows a page of a musical score titled "CONCERTO I:". It features five staves of music. The top staff is labeled "Violino concertato." and the second staff is "Violino I.". The third staff is "Violino II." and the fourth is "Viola.". The bottom staff is "Continuo.". The music is written on five-line staves with notes and rests, illustrating the characteristics of modern notation.

図 2.4: 近代譜の例

2.2.5 現代音楽

五線譜は鍵盤楽器をベースにした西洋音楽に最適化された記譜法のため、半音以下の繊細な音程のゆれや微妙なニュアンスを表現できないという制約がある。現代の作曲家たちはそれぞれ独自の方法で克服しようと試みている(図2.5)。

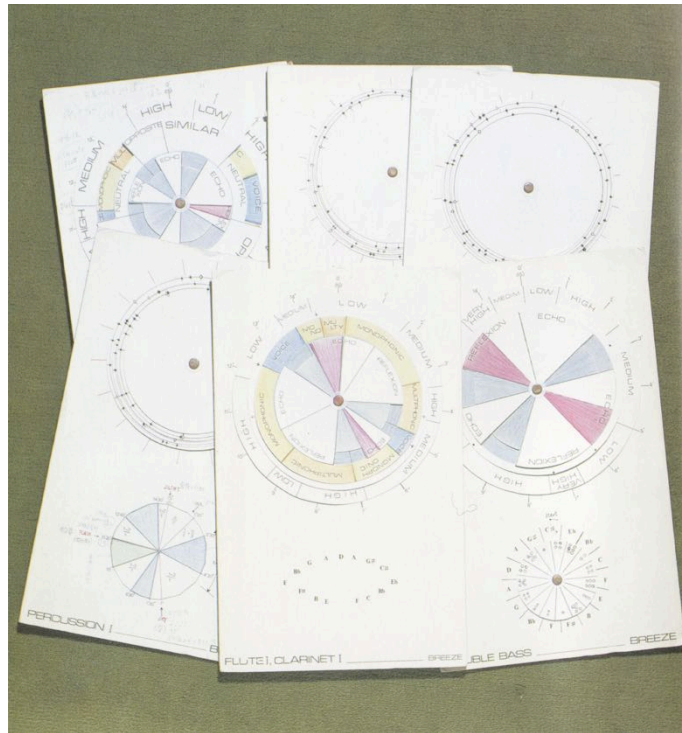


図 2.5: 現代音楽譜の例

2.3 テキストの進化

楽譜同様に紙の上で利用されていたテキストの進化にも注目する。計算機が普及する以前のテキストは印刷物であるという制約から以下のような問題が存在した。

- 簡単に編集できない
- せいぜい挿絵や写真程度の情報しか一緒に記録できない
- ドキュメントが複数存在するとき参照や管理が面倒

しかし現在では計算機の進化によりこれらの問題は完全に解決されている。

- 手軽な編集
 計算機上でテキストを扱えるようになり、コピー/ペースト/Undo/Redoといったテキスト編集支援機能を搭載した各種エディタによって手軽な編集が可能になった。また印刷レイアウトを簡単に構成できるワープロソフトによって、美しいレイアウトの文章を簡単に作れるようになった。
- 手軽な入力
 日本語入力などの文字変換をサポートするシステム (IME) などによって手軽な入力が可能になった。
- マルチメディアの活用
 画像/音声/動画といったマルチメディアを自在に埋め込むことができるハイパーテキ

スト (図 2.6) によって、情報量の多いドキュメントの作成が可能になった。

- ハイパーリンク
ハイパーテキストでは文章間リンクを示すハイパーリンク機能が利用でき、関連情報への素早いアクセスが可能になった。
- Web
インターネット技術の進化と Web の普及によって、世界中に存在する様々なドキュメントへ瞬時にアクセスすることが可能になった。
- 共同編集
Wiki のようなコラボレーションツールによって、場所や人数に制約を受けない共同編集が可能になった。

図 2.6: ハイパーテキストの例

2.4 楽譜の問題点

一方、楽譜の性質は計算機の普及後も印刷物の時代から進化しておらず、以下のような問題が存在する。

1. 簡単に編集できない

現在流通している楽譜は紙や PDF 形式での配布が一般的であり、内容の変更は前提とされていない。また自分で楽譜を書く場合は楽譜作成ソフトを利用できるが、印刷を前提としたレイアウトでの編集を強制されるため、一部だけアレンジしたい・1フレーズだけメモしたいといった用途には向いていない。

2. メモなどの情報を自在に書けない
楽譜を利用するには、

- 演奏時に気付いたこと
- 演奏時に気をつけるべきこと
- 先生に習ったこと

などのメモを書き込みながら使うのが一般的である(図 2.7)。メモ書きできるのは余白スペースだけであり、楽譜自体に編集を加えてレイアウトを変更したり、音声や動画といった文字以外の情報を埋め込むことも不可能なので、記述できる内容は必然的に限られる。

3. 参照や管理が面倒

紙や PDF はそれぞれが独立しているため、手元の楽譜が増えると管理に苦勞する。ファイル名でしか楽譜の内容を判断できないため、作曲者別にフォルダを分割するといったファイルの階層的な整理が必要である。また楽器構成や年代といった別の側面から参照したい場合には別途索引を用意しなければならない。

Figure 2.7 shows a musical score for 'Fuga "Great" BWV542' by J.S. Bach, arranged by G.E. Touru. The score is written for Bass and includes several handwritten annotations in Japanese. The annotations include performance instructions such as '8分刻はリズム感の後' (After the 8-measure rhythm), 'メロは全体に適用' (Apply melody to the whole), and '8分のリズム感も失うと危い' (Dangerous to lose the 8-measure rhythm). There are also fingering numbers and circled numbers (e.g., 5, 9, 10) indicating specific notes or measures. The score is titled 'Fuga "Great" BWV542' and includes the composer's name 'J.S. Bach' and the arranger's name 'G. E. arr. Touru. K'.

図 2.7: 楽譜上のメモの例

2.5 既存の楽譜システム

現在広く利用されている楽譜利用システムを解説する。

2.5.1 楽譜作成ソフト

Finale¹(図 2.8)、Sibelius²といった楽譜作成ソフトがデファクトスタンダードとして商用・私用問わず広く利用されている。スタンプ感覚で音符を入力できる WYSIWYG な編集環境

¹<https://www.finalemusic.jp/>

²<http://www.sibelius.jp/>

を持ち、以下のような楽譜作成支援機能を利用できる。

- 楽譜スキャン・OCR
- パート譜自動作成
- レイアウトの細かい微調整



図 2.8: Finale の画面

2.5.2 楽譜ビューアー

ほとんどの楽譜はPDFファイルとして配布されるため、閲覧のために標準のPDFビューアーが利用されることが多い。また Piascore³(図 2.9) を代表とするタブレット端末向けの楽譜に特化したビューアーも広く普及しており、

- 楽譜への書き込み
- 自動譜めくり
- メトロノーム

といった機能を利用できる。

³<http://piascor.com/>

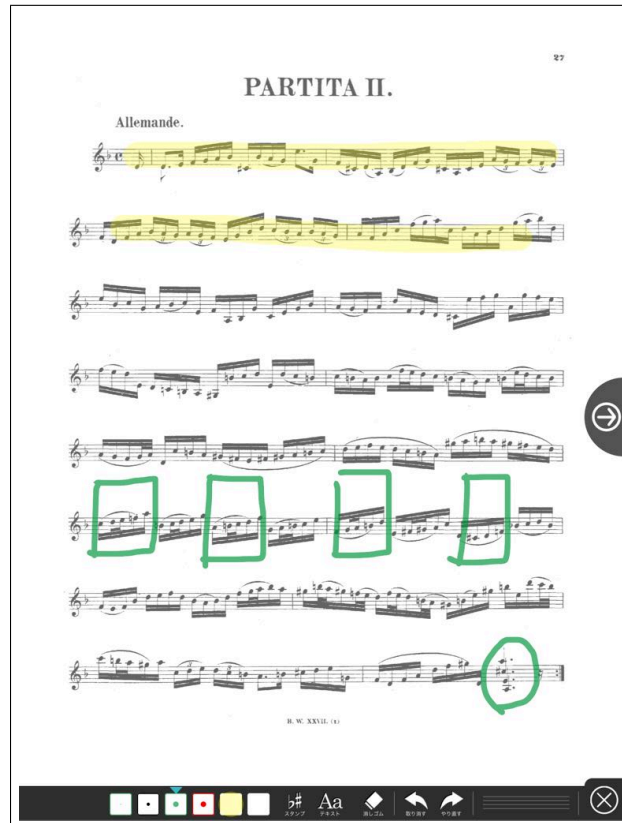


図 2.9: Piascore の画面

2.6 まとめ

楽譜の編集や閲覧を支援するシステムが広く利用されているが、これらは計算機上で楽譜の利用形態を再現したに過ぎず、

1. 簡単に編集できない
2. メモなどの情報を自在に書けない
3. 参照や管理が面倒

という本質的な楽譜の問題は解決されていない。次章では上記のような楽譜が持つ問題点を解決し、これまでの楽譜の在り方にとらわれない次世代の楽譜システム「ハイパー楽譜システム」を提案する。

第3章 設計

本章ではハイパー楽譜システムの要件と設計について述べる。

3.1 要件

前章で整理した楽譜の問題点を踏まえた上で、ハイパー楽譜システムの要件を整理する。

1. 楽譜を簡単に編集できる
メモを取るような気軽さで楽譜を書くことができ、新規作成/既存楽譜の編集両方を簡単に行える。
2. メモなどの情報を自在に書ける
テキストで楽譜を記述でき、ハイパーテキストのようなフォーマットの中で楽譜以外の情報も一緒に管理できる。
3. ハイパーリンクを利用できる
楽譜上の要素やテキストに対してハイパーリンクを設定でき、関連情報に素早くアクセスできる。

この要件を満たす楽譜システムは、ハイパーテキストの編集環境である Wiki と、テキストベースの楽譜記述言語の組み合わせによって実現できる。

3.2 ハイパー楽譜システム

本研究で提案するハイパー楽譜システムの基本構成と使い方を解説する。

3.2.1 基本構成

ハイパー楽譜システム (図 3.1) は Wiki である Scrapbox¹ と、楽譜記述言語の ABC 記譜法²(以下 ABC) によって構成されている。

¹<https://scrapbox.io>

²<http://abcnotation.com>

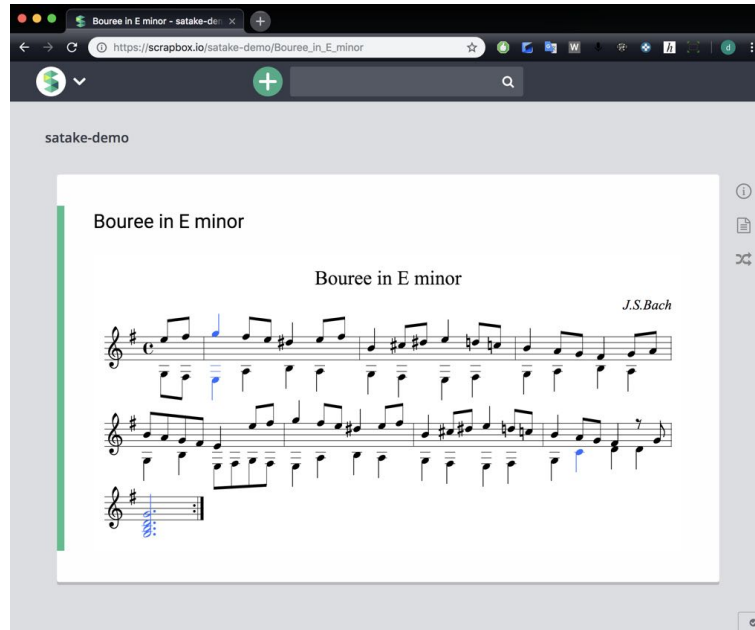


図 3.1: ハイパー楽譜システム上で書かれた楽譜

Scrapbox Scrapbox(図 3.2) は Gyazz[23] をベースにして開発された、Nota³社が運営している Wiki である。他のシステムには存在しない、ハイパー楽譜システムとして利用するのに最適な特徴を持つ。

- シンプルで柔軟な記法をもつ WYSIWYG エディタ
入力/改行/段落/箇条書きといった基本的なテキスト編集を見たまま行える。またリンクやマルチメディアの埋め込みには [] 記法のみで対応でき、他に様々な記法を覚える必要がない。

□ 記法

[ページ名]	同一 Wiki 内ページへのリンク
[URL]	外部リンク
[画像 URL]	画像埋め込み
[音声 URL]	音声埋め込み
[YouTube/VimeoURL]	動画埋め込み
[GoogleMapsURL]	地図埋め込み

- 関連ページリスト
Scrapbox ページの下部には
 - 別ページへのリンク
 - 別ページからのリンク
 - リンク先ページがリンクしているページ

³<https://www.notainc.com/ja>

といった関連ページリスト (図 3.3) が表示され、どのような情報と関連するのか一目瞭然に分かる。

- リアルタイム共同編集

Scrapbox では複数人による共同編集だけでなく、Google ドキュメント⁴のようなリアルタイム編集作業をコンフリクトすることなく行うことが可能である。

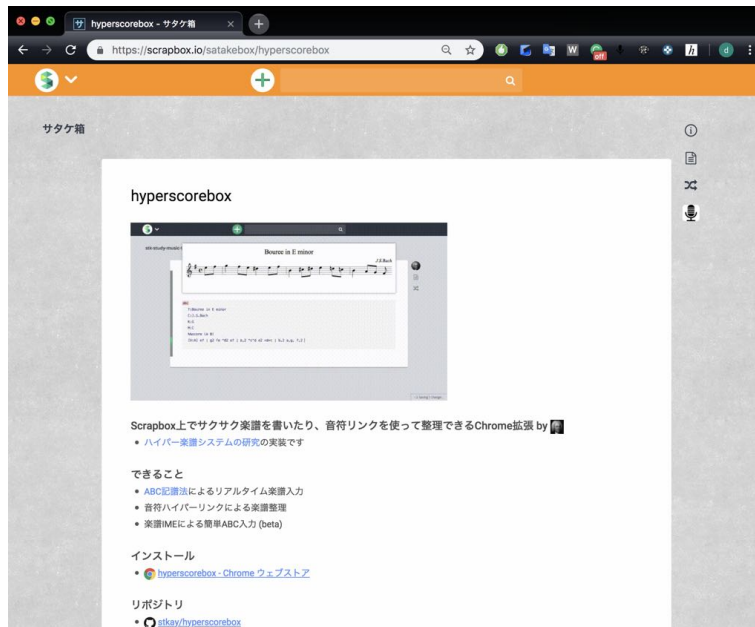


図 3.2: Scrapbox の画面

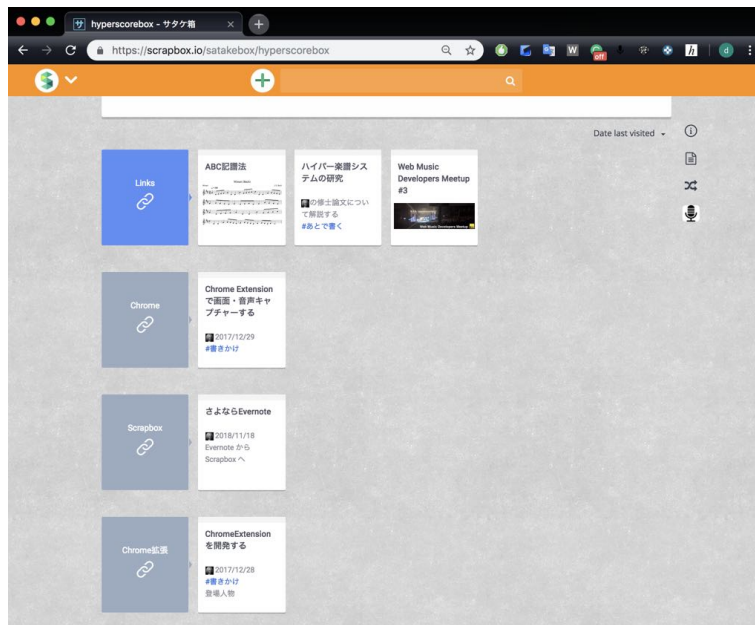


図 3.3: 関連ページリスト

⁴https://www.google.com/intl/ja_jp/docs/about

ABC ABCはプレーンテキスト形式の楽譜記述言語で、cが「ド」を表すシンプルな記法(ソースコード 3.1)を持つ。大規模で複雑な楽譜(図 3.5)の記述も可能で、幅広いシーンで利用可能である。利用者コミュニティ⁵が活発でバージョンのアップデートが頻繁に行われていたり、ユーザーによる譜例が多く公開されていることから信頼できるフォーマットである。ブラウザからはJavaScriptライブラリのabcjs⁶が利用でき、既存の楽譜システムと比較しても遜色のない綺麗な楽譜を出力することが可能である。

ソースコード 3.1: 「ド」を表す ABC

1 c



図 3.4: 「ド」の譜例



図 3.5: 複雑な楽譜の例

⁵<https://groups.yahoo.com/neo/groups/abcusers/info>

⁶<https://abcjs.net/>

3.2.2 使い方

楽譜を書く Scrapbox 上で `code:*.abc` という行に続けて ABC テキストを書くと ABC 記述の真上に楽譜画像が表示され (図 3.6)、楽譜を確認しながらリアルタイムに編集できる。楽譜や ABC の領域外をクリックすると ABC 記述が隠れ、楽譜のみが表示される (図 3.7)。

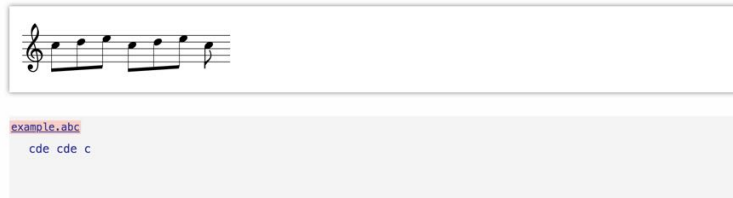


図 3.6: 編集中の楽譜



図 3.7: 編集後の楽譜

音符など以外の情報を書く 楽譜領域以外では標準の Scrapbox 記法を利用できるので、自在にテキストを書いたり、マルチメディアを埋め込むことができる (図 3.8)。

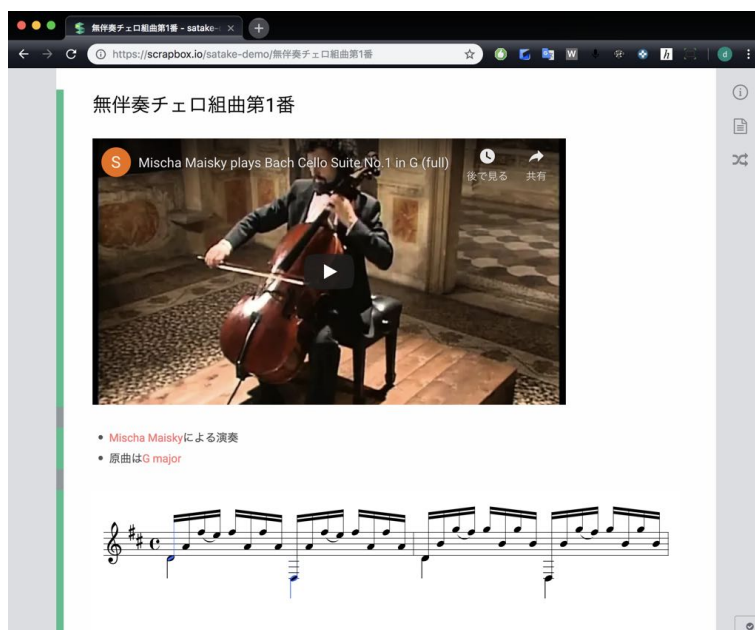


図 3.8: 楽譜にテキストや動画が埋め込まれた例

ハイパーリンクを利用する ABCには無い独自の機能として音符ハイパーリンク機能を利用できる。ABCの末尾に%Links:[リンク 1][リンク 2]…と記述することで対象となる音符にハイパーリンクを設定できる。具体的な設定方法とその例を図3.9に示す。ハイパーリンクが設定された音符は青く表示され(図3.10)、クリックするとリンク先のページへ遷移する(図3.11)。

```
example
M:C
L:1/4
CEGE|GEGz|
%Links:[10 middle_C]
```

↑ 対象音符の位置
↑ リンク先のScrapboxページ名

リンクを設定したい音符のコードブロック内での位置を文字数で指定

図 3.9: 1 音目の「C」に「middle C」ページへのリンクが設定された ABC



図 3.10: ハイパーリンクが設定された音符

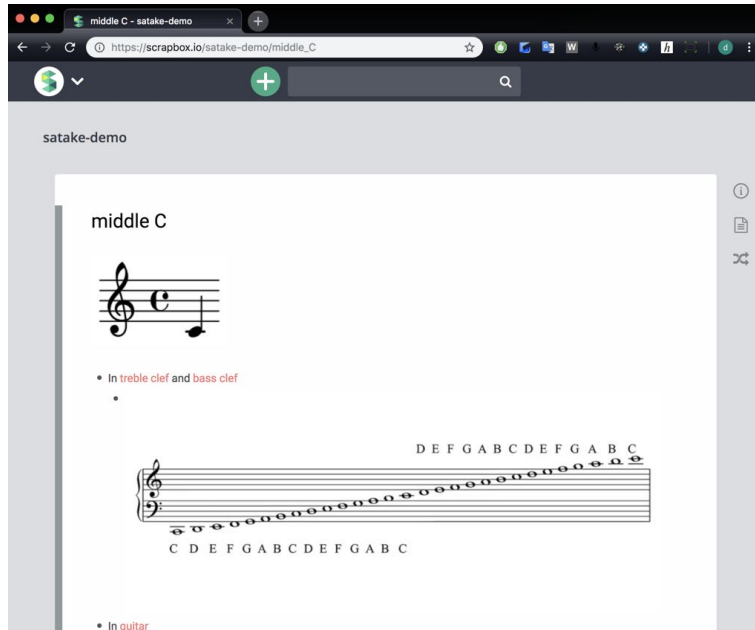


図 3.11: リンク先の「middle C」ページ

第4章 実装

本章では第3章で述べたシステムの設計を受け、ハイパー楽譜システムの実装について述べる。

4.1 アプリケーション構成

ハイパー楽譜システムは Scrapbox 上で利用する Chrome Extension として実装されている¹。Chrome Extension をインストール可能な PC 版 Google Chrome を利用できる環境であれば、OS を問わず利用することができる。Chrome Extension の開発には TypeScript² を、楽譜の出力には abcjs を利用している。

Chrome Extension Chrome Extension は Google Chrome 上にインストール可能なブラウザの機能を拡張できるアプリケーションである。一般的な Web ページと同様に、HTML/CSS/JavaScript のような Web 技術を用いて実装することができ、

- HTML 要素の操作
- カスタム CSS の適用
- HTTP リクエスト
- 外部サービスとの連携

といった機能が実現可能である。アプリケーションは Chrome Web Store³を通じて簡単に公開できる。

4.2 楽譜の描画

本節では楽譜描画の仕組みを述べる。

Scrapbox にはコードハイライトを行うコードブロック記法 (図 4.1) が存在し、"code:ファイル名"と入力することで利用できる。本アプリケーションでは ABC の記述にこのコードブロックを利用している。本アプリケーションによってファイル名の拡張子が*.abc に一致するコードブロック (図 4.2) のみが ABC として解釈され、abcjs によって生成された楽譜画像 (図 4.3) がコードブロックの HTML 要素に対して重畳表示される。

```
example.js
const sayHello = () => {
  console.log("Hello ABC");
}
```

図 4.1: コードブロック記法

¹<https://chrome.google.com/webstore/detail/hyperscorebox/cjlhoobllhkpjjomlijfdblgifcdmoh>

²<https://www.typescriptlang.org/>

³<https://chrome.google.com/webstore/category/extensions?hl=ja>

```
example.abc  
cde cde
```

図 4.2: ABC として解釈されるコードブロック



図 4.3: 生成された楽譜画像

4.3 音符ハイパーリンク機能の実現

本節では音符ハイパーリンク機能の実現方法を述べる。

abcjs は SVG 画像として楽譜を出力しており、一つ一つの音符に対してクリックリスナを設定可能である。Document.location.href オブジェクト⁴を利用してページ遷移を実行するコールバック関数を音符のクリックリスナに設定している。

⁴<https://developer.mozilla.org/ja/docs/Web/API/Location>

第5章 楽譜IME

本章では楽譜入力を支援する楽譜 IME を提案する。

5.1 ABCの問題点

前章までに提案したハイパー楽譜システムでは、Wiki上での楽譜入力を実現するために、テキストベースでの楽譜記述を可能とするABCを利用している。ABCはシンプルだが、

1. 記法を覚える必要がある
2. WYSIWYGでない

という問題点が存在する。このうち2.はハイパー楽譜システムのリアルタイムな楽譜描画によって解決されている。1.の解決には、文字変換を行う日本語入力システム(IME)的アプローチが有効であると考えられる。本章では楽譜IMEを提案し、既存システムと比較しても遜色ない快適な楽譜入力を実現する。

5.2 楽譜IME

楽譜IMEは楽譜辞書を利用した変換機能によりABC特有の記法や様々な楽譜の入力をサポートする。これは前章で実装したハイパー楽譜システムの一部として利用できる。本節では主な機能や利用フローを解説する。

5.2.1 利用できる機能

1. ABC入力支援機能

- ドレミ変換機能
「ドレミ」のようなカタカナ音名をABCに変換する。ABCで利用される英語音名に慣れていなくても楽譜入力が可能になる。
- 記号変換機能
「♯ ♭ ♮」といった臨時記号をABCに変換する。ABCでは♯ ♭ ♮をそれぞれ ^ _ = と表記するが、見慣れた記号をそのまま利用できることで、感覚的な入力が可能になる。
- オクターブ変更機能
上下キーによって音符のオクターブを変更する。ABCではオクターブ上を'、下を,と表記するが、矢印キーを利用することで感覚的な入力が可能になる。
- MIDIデバイス入力機能
MIDIキーボードといったデバイスから直接音符を入力する。既存の楽譜作成ソフトでは一般的な機能であり、鍵盤を利用することで感覚的な入力が可能になる。

2. 楽譜入力支援機能

楽譜辞書を利用した変換機能によって楽譜の入力を支援する。例えばCメジャーの和音(ドミソ)を入力したいときは、cと入力するだけで変換が可能になる。ABCでは和

音を入力するとき [ceg] のように構成音を全て記述する必要があるが、よく使う三和音などはその名前から変換できるようにすることで、楽譜入力の手間を軽減する。

5.2.2 利用フロー

1. 楽譜 IME の起動

コードブロック内で Esc キーを押すと楽譜 IME が起動する (図 5.1)。

2. 文字入力

文字入力すると、入力中の ABC テキストからリアルタイムに楽譜がプレビューされる (図 5.2)。変換候補も同様に楽譜として表示される。ここでは "c" という入力に対して、C メジャー、C マイナーの和音が候補として提示されている。

3. 変換候補の選択

文字入力中に Space キーを押すと変換候補を選択できる (図 5.3)。

4. 変換候補の入力

変換候補を選択した状態で Enter キーを押すと、その楽譜を入力できる (図 5.4)。



図 5.1: 楽譜 IME 初期画面



図 5.2: 文字入力中の画面



図 5.3: 変換候補選択中の画面



図 5.4: 変換が完了した画面

5.3 実装

楽譜 IME は JavaScript を用いてハイパー楽譜システムの内部に実装されており、ブラウザ上で利用可能である。本節では楽譜 IME のアプリケーション構成を解説する。

5.3.1 文字入力

楽譜 IME は OS から利用できる IME とは異なり、アプリケーション自身がブラウザからキーイベントを取得する。具体的には、ブラウザが発行するキー入力イベントが Scrapbox に渡される前に横取りする形で受け取り、各種変換処理を実行している。楽譜 IME 上で変換処理が確定されると、テキストエリア上の文章を操作する `Document.execCommand()` 関数¹を利用してブラウザに文字入力を指示する。

5.3.2 楽譜の表示

楽譜 IME 起動時は楽譜プレビュー・変換候補表示用の Div 要素を Scrapbox の画面上に重畳表示し、ハイパー楽譜システム同様 abcjs を用いて楽譜を描画している。

¹<https://developer.mozilla.org/ja/docs/Web/API/Document/execCommand>

5.3.3 楽譜辞書

辞書データはソースコード 5.1 のような JSON 形式で保持している。辞書エントリの第 1 要素は楽譜の読み、第 2 要素は変換される ABC テキストを示している。このような構造は既存の日本語入力システム [24] と同様のものである。

ソースコード 5.1: 楽譜 IME の辞書データ

```
1 [
2   ["C", "[ceg]"],
3   ["Cm", "[c_eg]"],
4   ...
5 ]
```

5.4 議論

本節では楽譜 IME の利用において気付いた点などを述べる。

5.4.1 使用感

楽譜 IME は既存の IME と同様のインターフェースを提供しており、入力の対象が楽譜となっても候補が表示されたり候補を選択したりする感覚は共通であることから、快適に楽譜を入力できた。一方で、楽譜辞書のエントリ数が少ないため、思うような変換ができない場面もあった。

5.4.2 辞書の編集

現在、楽譜 IME の辞書は筆者自身が編集している。和音を中心にエントリを追加しているが、幅広いシーンで変換機能を利用するためには辞書の充実が求められる。しかし、コードネームやスケールといった一般的な音楽知識でさえジャンルや文脈によって名前が異なることがあり、開発者 1 人で網羅的な辞書を作成するのは大変困難な作業である。単語と楽譜が対応するような辞書は存在しないので、Lexierra[24] のように Wiki 上で辞書を編集する手法が有効であると考えられる。Wiki を利用することでソーシャルに辞書を編集できるため、膨大なエントリを協力して編集したり、ユーザーそれぞれが必要なエントリを自由に追加するといったことが可能になる。また Scrapbox 上で辞書を構築すればハイパー楽譜システム・楽譜 IME を利用できるため、効率的な楽譜辞書編集が可能となる。

5.4.3 動的な辞書

ページ内外の別の楽譜をそのまま利用したいシーンがあった。コピー&ペーストでも対応できるが、事前に用意した楽譜辞書だけでなく、Scrapbox プロジェクト内の楽譜から動的に

辞書を構築できれば、より柔軟な楽譜入力を実現できると考えられる。また前述のソーシャルな辞書編集環境が実現された場合、ユーザーそれぞれが必要なエント리를自由に追加できるが、あるユーザーにとっては不要なエント리가変換候補として提案されてしまう可能性がある。用途に合わせて利用する辞書を切り替えたり、エント리를絞り込むことができれば解決できると考えられる。

5.5 まとめ

ハイパー楽譜システム上で利用可能な楽譜 IME を提案した。日本語入力システムの仕組みを ABC に適用し、楽譜辞書によって ABC 入力を支援するシステムを実現した。Wiki によるソーシャルな辞書編集環境や動的な辞書を実装し、あらゆる楽譜入力シーンで利用できる IME として拡張を行っていく予定である。

第6章 応用例

本章では、ハイパー楽譜システムによって実現可能な応用例について述べる。

6.1 楽器練習

楽器練習の際は気付いたこと・習ったことなどをメモしながら楽譜を使うのが一般的である。しかし楽譜上でメモ可能な領域は限られており、自由に書くことはできない。また楽譜は曲単位で独立しているためメモした情報が各楽譜に散逸してしまったり、そもそもどこに書いたか分からなくなってしまう問題もある。本システムではレイアウトの制約を受けず自在にメモを書いたり、技術要素毎にページ (図 6.1) を用意して情報を一元管理することができる。

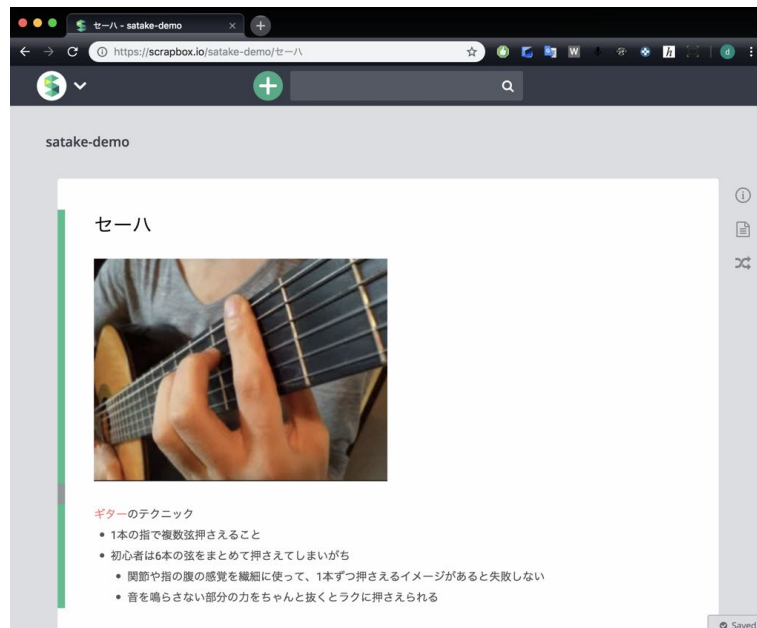


図 6.1: 技術ページの例

6.2 勉強

楽典や音楽理論といった知識は専門書のように楽譜とは独立した場所に書かれることが多く、学習者が関心のある楽曲との関連性を把握するのが難しい。本システムでは、楽典や音楽理論といった知識を楽譜と同じ Scrapbox プロジェクトに記録しておく (図 6.2) ことで、実際の楽曲と結びついた勉強環境を実現できる。それぞれのページでは楽譜/テキスト/マルチメディアを用いて分かりやすく記述できるだけでなく、ハイパーリンクによって楽曲との関連性を可視化し、勉強/演奏をシームレスに繋げることができる。そしてページが追加されるうちにたくさんの知識や楽曲がハイパーリンクで結びつき、Scrapbox プロジェクト全体が自分専用の教材として利用できるようになる。本システムによって、楽譜とテキストのみで表現力が乏しい/実際の楽曲との関連性が分からないといった既存の教材の問題点を解決しながら、勉強するだけで教材が作られていくという新しい勉強法を実現できる。

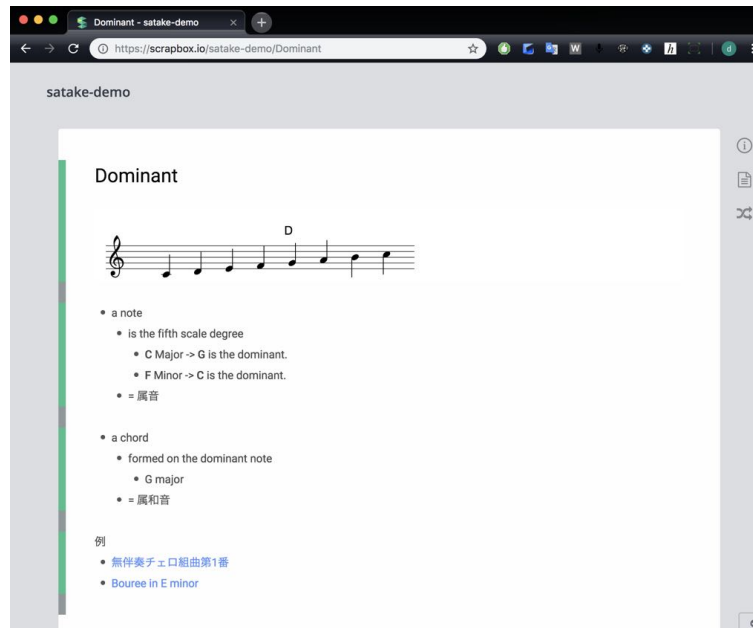


図 6.2: 知識ページの例

6.3 楽譜共同編集

本システムでは楽譜をテキストとして記述しているため、Scrapbox の強力な共同編集機能を活用し、柔軟な楽譜編集を実現できる。例えば採譜 (耳コピ) のような作業をパートや楽章毎に分担することで、大規模な楽譜でも一人一人の負担を軽減することができる。作曲/編曲といった用途では、作者と演奏者が同じページ上で作業することで、楽譜の変更や演奏者のフィードバックを瞬時に共有することができる。

6.4 楽曲の調査/分析

筆者が所属する増井研究室ではあらゆる情報共有に Scrapbox が利用されており、その中には研究に関する話題だけでなく、アニメや音楽といった趣味の情報も多く書かれている。ハイパー楽譜システムの運用時に、あるアニメ主題歌のメロディやコード進行がとても独創的であったために研究室内で話題になり、Scrapbox 上で情報収集や分析が行われるようになった。その楽曲のページには YouTube 動画や音楽家による考察、作曲者自身のツイートなどがまとめられると同時に、ハイパー楽譜システムを活用して採譜が行われた (図 6.3)。音やリズムが適切かというような議論を活発に行いながら楽譜が編集され、楽曲に対する理解が深まったと同時に、和気あいあいと議論しながら曲を分析するという音楽の楽しみ方を発見できた。

もしハイパー楽譜システムが存在しなければ採譜自体が行われなかったと考えられる。Scrapbox 上で楽譜を利用するには楽譜を画像化する必要があり変更が難しくなってしまう

し、既存の楽譜作成システムは美しいレイアウトで清書するための環境なので、楽譜を試行錯誤しながら書いたり、断片的にメモ書きするような用途には適さない。

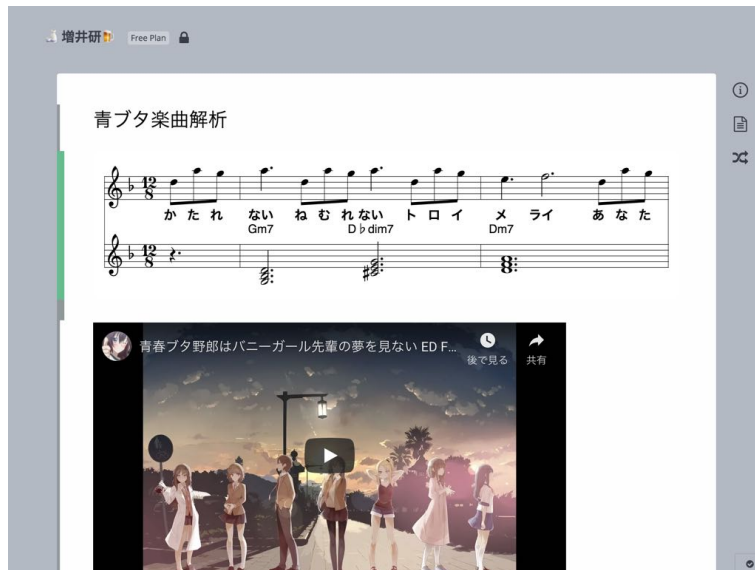


図 6.3: 楽曲ページ

6.5 まとめ

本章では、本システムによって実現可能な応用例について述べた。Wiki と楽譜記述言語の組み合わせによって、今までにない新しい楽譜利用環境の実現が可能である。本章で述べた応用例に限らず、様々な応用が可能と考えられる。

第7章 関連研究

本章では楽譜関連研究を紹介し、それらの特徴や本研究との関連性について示す。

7.1 主要な研究領域

楽譜に関連する主要な研究領域を紹介する。

1. スコアアラインメント

楽譜上のどの部分を演奏しているのか認識するためのスコアアラインメント技術が研究されている [9][11][8][3]。応用例として計算機による伴奏システム [15][19]、音楽鑑賞サポートシステム [16]、楽器練習サポートシステムなどが挙げられる [10]。

2. ジェスチャー楽譜入力インターフェース

ジェスチャーを利用した楽譜入力インターフェースが提案されている [12][6][7]。これにより、計算機上でペンやポインティングデバイスを使って、楽譜を手書き入力することができる。

3. 楽譜認識/再生システム

画像から楽譜情報を読み取るための楽譜認識技術が研究されている [5][17][18]。これを応用して、カメラを搭載したデバイスによって印刷された楽譜を再生するシステムが提案されている [21][4]。

7.2 楽譜編集/閲覧システム

一般的に利用されている楽譜システムとは異なる、紙の楽譜の再現にとどまらない特徴を持った楽譜編集/閲覧システムを紹介する。

7.2.1 BRASS

Watanabe らが提案した BRASS[20](図 7.1) は、楽曲の全体構造を把握しながら楽譜を閲覧できるインタフェースによって、楽曲学習を支援する。複数ページにまたがる規模の大きな楽譜では楽曲全体を把握するのが困難であるが、楽譜を圧縮表示し、音符の数やテンポの大小といったパラメータを濃淡や色によって可視化することで解決している。ここで利用されている、空間を歪ませて注目部分を拡大表示する可視化手法はフォーカス+コンテキスト表示と呼ばれている。本研究ではフォーカス+コンテキスト表示のような可視化の工夫は行っていないが、楽曲理解のために必要なテキスト/マルチメディアを自在に埋め込めることから、学習支援にも利用できる。

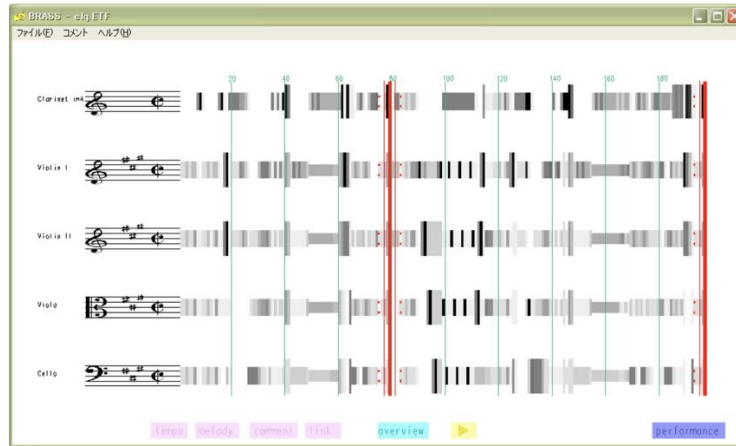


図 7.1: BRASS の画面

7.2.2 WIKI: :SCORE

Almeida らが提案した WIKI: :SCORE^[2](図 7.2) は、Wiki 上で楽譜の共同編集を行えるシステムである。ABC によって楽譜を編集し、PDF/画像/音声といった各種フォーマット出力することができる。複数パート/セクションを持つ規模の大きな楽譜の編集は大変な作業であるが、共同編集や楽譜の出力を行えることで、複数人で協力して実行することが可能である。Wiki と ABC によるシステムであることからハイパー楽譜システムに設計が近いが、以下の点で異なっている。

- 共同編集のみを問題にしている
- 編集環境が WYSIWYG でない
- 楽譜専用の Wiki である

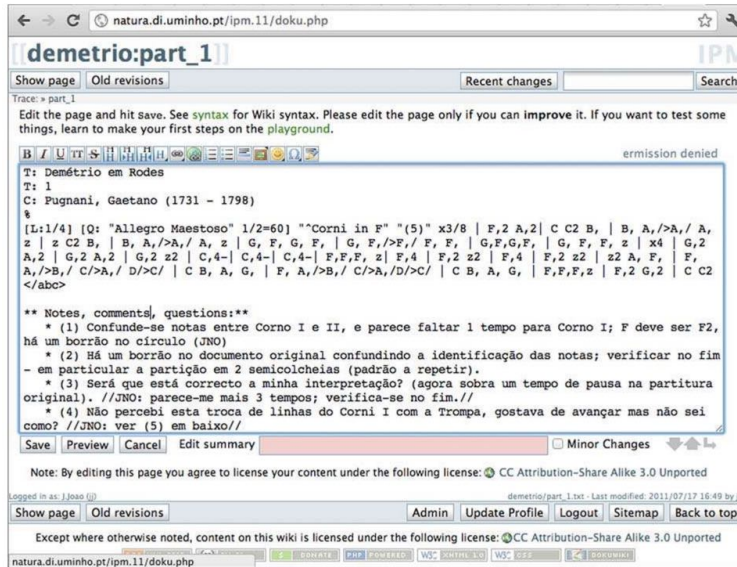


図 7.2: WIKI: :SCORE の画面

7.3 楽譜記述言語

楽譜記述言語は楽譜をベースとした楽曲を記録・共有する目的で利用されており、本研究で利用している ABC 記譜法もその 1 つである。本節では ABC 以外に広く利用されているものを解説する。比較のために、図 7.3 の楽譜を記述する各言語の例を示す。ソースコード 7.1 は ABC における記述例である。



図 7.3: 「ドレミファ」の譜例

ソースコード 7.1: ABC における図 7.3 の楽譜の記述例

```
1 M:4/4
2 L:1/4
3 cdef|
```

7.3.1 MusicXML

MusicXML[13] は XML 形式の楽譜記述言語で、音符や小節といった音楽的要素が階層的に記述される。各種楽譜作成ソフトでも MusicXML によるインポート/エクスポートがサポートされており、楽譜データのやりとりに広く利用されている。

```
1 <score-partwise>
2   <part-list>
3     <score-part id="P1">
4       <part-name>Piano</part-name>
5     </score-part>
6   </part-list>
7   <part id="P1">
8     <measure number="1">
9       <attributes>
10        <divisions>1</divisions>
11        <key>
12          <fifths>0</fifths>
13        </key>
14        <time>
15          <beats>4</beats>
16          <beat-type>4</beat-type>
17        </time>
18        <clef>
19          <sign>G</sign>
20          <line>2</line>
21        </clef>
22      </attributes>
23      <note>
24        <pitch>
25          <step>C</step>
26          <octave>5</octave>
27        </pitch>
28        <duration>1</duration>
29        <type>quarter</type>
30      </note>
31      <note>
32        <pitch>
33          <step>D</step>
34          <octave>5</octave>
35        </pitch>
36        <duration>1</duration>
37        <type>quarter</type>
38      </note>
39      <note>
40        <pitch>
41          <step>E</step>
42          <octave>5</octave>
43        </pitch>
44        <duration>1</duration>
45        <type>quarter</type>
46      </note>
47      <note>
48        <pitch>
49          <step>F</step>
50          <octave>5</octave>
51        </pitch>
52        <duration>1</duration>
53        <type>quarter</type>
54      </note>
55      <barline>
56        <bar-style>light</bar-style>
57      </barline>
58    </measure>
59  </part>
60 </score-partwise>
```

7.3.2 Lilypond

Lilypond[14] はテキスト形式の楽譜記述言語および楽譜出力アプリケーションで、コマンドラインから利用できる。印刷を意識した高度な楽譜作成が可能で、GUI 楽譜作成ソフト

で利用できるようにレイアウト調節パラメータを多く持つ。

ソースコード 7.3: Lilypond における図 7.3 の楽譜の記述例

```
1 {  
2   \time 4/4  
3   c''4 d'' e'' f''  
4 }
```

7.3.3 VexTab

VexTab[1] はテキスト形式の楽譜記述言語および楽譜出力アプリケーションで、JavaScript ライブラリから利用できる。HTML Canvas/SVG に楽譜を出力可能であり利用形態は ABC に最も近いが、記法がやや冗長である。

ソースコード 7.4: VexTab における図 7.3 の楽譜の記述例

```
1 stave  
2 time=4/4  
3 notes :4 C/5D/5E/5F/5|
```

第8章 考察

本章では、ハイパー楽譜システム利用者の意見や自身の運用経験をまとめ、諸問題や新しい可能性について述べる。

8.1 評価

本システムのプロトタイプとなる Chrome 拡張機能「hyperscorebox」を 2018 年 11 月 21 日に Chrome Web Store で公開した¹。2018 年 12 月現在、60 名程度にインストールされている。また、Web 技術や音楽アプリ関連情報を共有する以下の開発者イベントにて hyperscorebox の展示発表を行った。

- WebAudio.tokyo #6² (2018 年 11 月 15 日 株式会社ドリコム)
- HTML5 Conference 2018³ (2018 年 11 月 25 日 電気通信大学)
- Web Music Developers Meetup #3⁴ (2018 年 12 月 21 日 株式会社デジタルハーツ)

本節では、

- 筆者の運用経験
- ユーザーからのフィードバック
- 各種展示発表でのフィードバック

をまとめる。

8.1.1 筆者の運用経験

主に楽器練習/音楽理論の勉強/採譜といった用途で 3ヶ月に渡り利用した。

楽器練習 筆者はクラシックギターの練習に本システムを利用しており、練習中に残した様々なメモを整理して記録できている。以前まではたくさん書き込んでも練習する曲それぞれの楽譜に散逸してしまうため整理や再利用が難しく、結局役に立たないという問題に悩まされていたが、本システムによって解決された。

音楽理論の勉強 既存の教材に強く苦手意識を抱いていたが、自身が演奏している楽曲と知識の関係性がハイパーリンクによって可視化され、自分専用の分かりやすい教材として利用できている。また音声の埋め込みによって実際の演奏を聞きながら学習でき、イメージしづらい概念の理解を深めるのに特に有効である。

採譜 筆者は本システムの利用前まで ABC の利用経験が無かったが、複雑な楽譜でない限りは楽譜 IME の支援によって快適に楽譜編集できている。また採譜した楽曲の音声/動画/作曲家/演奏者といった周辺情報を一元管理でき便利である。

¹<https://chrome.google.com/webstore/detail/hyperscorebox/cjlhooblhkpjjomlijfdblifcdmoh>

²<https://webaudiotokyo.connpass.com/event/103821/>

³<https://events.html5j.org/conference/2018/11/>

⁴<https://connpass.com/event/112300/>

8.1.2 感想・意見

まず、第2.4節で述べた楽譜の問題点に対して多くの賛同を得られた。多くのユーザーは楽譜に不便さを感じていたものの、根本的な解決手段を見つけられない状況に置かれている。その他の感想や意見として主に以下のようなものが挙げられた。

1. 楽譜やコメントを瞬時に共有できる

楽譜や演奏に対する要望を伝えるには「3小節目の1拍裏のドの音を半音上げて」「100小節のシドレミのフレーズをもっと大きく」といった具体的な指示を行う必要がある。本システムでは楽譜と一緒にコメントを書いたり、その場で楽譜を更新して瞬時に共有できるので、コミュニケーションコストを大幅に小さくできるのではないかという好意的な評価を得た。

2. 既存の楽譜作成システムを使うには大げさなシーンで価値がある

既存の楽譜作成システムを使って1.のような楽譜共有を実現する場合、

- (a) ソフトを起動して楽譜入力する
- (b) スクリーンショットを撮影する
- (c) 画像を SNS 等で送信する

という作業が必要である。そもそも楽譜作成システムは断片的な楽譜の作成を前提としていないため、楽譜そのものとは関係ないレイアウトや設定を意識する必要があり、このような用途では大袈裟である。一方本システムでは素早く書いてすぐ共有できるため好評であった。

3. Web リソースを活用できる

紙や PDF 上の楽譜は静的で独立したメディアなので、外部の情報を活用したりリンクすることができない。マルチメディア情報をはじめとする Web 上の様々なリソースを活用しながら楽譜編集するという応用は新鮮なものとして受け止められ、大きな可能性を感じられるという感想を得た。

8.1.3 問題点

問題点として以下のようなものが挙げられた。

1. リンク機能について

複数の音符にハイパーリンクが設定されている場合、すべての音符が1つのリンクを示しているのか、それぞれの音符が別のリンクを示しているのか一目見て理解できないという意見が挙げられた。これはリンク先にかかわらず全て青色の音符として表示しているためであるが、単音に対しても、フレーズや小節に対してもリンクを設定したいので、何らかのインタフェース的工夫が必要である。

2. 楽譜編集について

既存の楽譜を取り込めるような仕組みがほしいという意見が挙げられた。楽譜を自ら編集することを前提としているため、既存の楽曲を利用したい場合は元になる楽譜を別途用意したり、採譜したりして自分で入力する必要がある。楽譜 IME によってある程度入力作業が簡単になるものの、すぐ楽譜を利用したいユーザーにとっては大きな負担になる。

3. 利用環境について

タブレット上でも快適に本システムを利用したいという意見が挙げられた。PC ブラウザ上での利用を前提とした設計のため、タブレットのようなタッチインタフェース端末では利用しづらい。既存の楽譜ビューアーはタブレットで利用されるものがほとんどであり、譜面台の上のような限られたスペースでも快適に利用できるため好まれている。

8.2 考察

8.2.1 設計の妥当性

本システムは既存のものとは全く異なる新しい楽譜の利用形態を持つが、実際に利用したりデモを体験したユーザーからは概ね好意的に受け入れられ、Wiki と楽譜記述言語を組み合わせるといふ本システムの設計指針は正しかったといえる。また本研究で述べた楽譜に対する問題意識にも多くの共感を得られたことから、本システムをベースにして様々な改善や拡張を行うことで、より良い楽譜利用環境を生み出せると考えられる。

8.2.2 解決すべき課題

1. 音符のリンク状況が分からない

音符にリンクが設定されている場合、リンク先を把握するには ABC を確認するか、実際に音符をクリックする必要がある非効率である。Web ブラウザではフォーカスされたリンクに下線を付けたり、画面左下にリンク先アドレスを表示することで解決している。本システムでも同様の手法が有効であると考えられる。

2. 自分で楽譜を書く必要がある

IMSLP⁵といったパブリックドメイン楽譜配布サイトや出版社による販売サイト上には膨大な数の楽譜が存在し、ユーザーは必要とする楽譜を Web から簡単に見つけ出すことができるため、通常自ら楽譜を書く必要はない。本システムでも必要な楽譜をすぐ利用できるようにするために、既存の楽譜ファイルから ABC へコンバートできる機能や、ユーザーが楽譜を配布できる Wiki 上のプラットフォームが必要であると考えられる。

⁵<https://imslp.org>

3. タブレットで使いづらい

楽譜利用環境として主流であるタブレット上でも快適に利用できるようにするために、タッチインターフェースに適した以下のようなインターフェースの工夫が必要であると考えられる。

- ボタン/ペン操作による楽譜入力
- ハンズフリーな楽譜スクロール

8.2.3 楽譜の問題点の検証

本システムにおいて第 2.4 節で述べた問題点が克服されているかどうかを問題点毎に検証する。

- 簡単に編集できない
新規作成/既存楽譜の編集両方を、楽譜 IME の支援を受けながら簡単に行える。
- メモなどの情報を自在に書けない
楽譜と同じ場所にテキスト/マルチメディア情報を自在に書ける。
- 参照や管理が面倒
楽譜内のハイパーリンクによって関連する楽譜や情報を辿って参照できる。

以上のように、第 2.4 節で述べたすべての点に関して問題が解決していることがわかる。

第9章 結論

本章では本研究を総括する。

9.1 研究の成果

本研究では、Wiki と楽譜記述言語を組み合わせた新しい楽譜システム「ハイパー楽譜システム」と、日本語入力システムの的アプローチによる楽譜入力支援システム「楽譜 IME」の提案を行った。

まず第 2 章において、楽譜の問題点をテキストの進化と比較しながら分析した。楽譜を扱う既存システムの現状をとりあげ、計算機が普及した現在も根本的に解決されていないことを示した。

第 3 章では、第 2 章で述べた楽譜の問題点に対する有効的な解決方法を提案した。また、それに基づき本研究で開発した「ハイパー楽譜システム」の基本構成と使い方について述べた。

第 4 章では、「ハイパー楽譜システム」のアプリケーション構成と詳細な実装について述べた。

第 5 章では、テキストによる楽譜入力には限界があることを示し、日本語入力システムの的アプローチによる楽譜入力支援システム「楽譜 IME」を提案した。

第 6 章では、「ハイパー楽譜システム」によって実現可能な応用例について述べた。

第 7 章では、本研究に関連する研究を紹介し、それぞれのアプローチの特徴と問題点を分析した。

第 8 章では、筆者による運用経験やユーザーからのフィードバックをもとに本研究の有効性と問題点を分析した。

9.2 総括

本研究では楽譜を簡単に書いたり音符など以外の情報を自在に書いたりハイパーリンクを利用できる「ハイパー楽譜システム」と ABC の入力を支援する「楽譜 IME」の開発を行った。ハイパー楽譜システムは Wiki と楽譜記述言語の組み合わせによって既存の楽譜の問題点を克服するだけでなく、新しい楽譜の使い方を実現した。楽譜 IME は日本語入力システムの的アプローチを取ることでテキストによる楽譜入力の弱点を克服し、既存の楽譜作成システムに劣らない快適な楽譜入力を可能にした。今後は第 8 章で述べた問題点についての改善や、システムの拡張を行っていく。

謝辞

学部から5年間の長きに渡りご指導を賜りました慶應義塾大学環境情報学部 増井俊之教授に深く感謝いたします。また、本研究の副査としてご意見、ご助言を頂きました小川克彦教授、中西泰人教授に感謝いたします。

学部・修士の4年間を共に過ごし、自身の研究について幅広い議論をしていただいた政策・メディア研究科修士課程の早川匠氏、様々な形でアドバイスをくださった増井俊之研究会OB諸氏に感謝いたします。

参考文献

- [1] VexTab. <http://www.vexflow.com/vextab/>.
- [2] José João Almeida, Nuno Ramos Carvalho, and José Nuno Oliveira. WIKI: :SCORE a collaborative environment for music transcription and publishing. *Inf. Serv. Use*, Vol. 31, No. 3-4, pp. 177–187, July 2011.
- [3] Andreas Arzt, Gerhard Widmer, and Simon Dixon. Automatic page turning for musicians via real-time machine listening. In *Proceedings of the 2008 Conference on ECAI 2008: 18th European Conference on Artificial Intelligence*, pp. 241–245, Amsterdam, The Netherlands, The Netherlands, 2008. IOS Press.
- [4] Tetsuaki Baba, Yuya Kikukawa, Toshiki Yoshiike, Tatsuhiko Suzuki, Rika Shoji, Kumiko Kushiyama, and Makoto Aoki. Gocen: A handwritten notational interface for musical performance and learning music. In *ACM SIGGRAPH 2012 Emerging Technologies*, SIGGRAPH '12, pp. 9:1–9:1, New York, NY, USA, 2012. ACM.
- [5] David Bainbridge and Tim Bell. A music notation construction engine for optical music recognition. *Software: Practice and Experience*, Vol. 33, No. 2, pp. 173–200, 2003.
- [6] Sé bastien Macé, Éric Anquetil, Elodie Garrivier, and Bruno Bossis. A pen-based musical score editor. In *Proceedings of the 2005 International Computer Music Conference, ICMC 2005, Barcelona, Spain, September 4-10, 2005*, 2005.
- [7] William Buxton. Sanand patel and ronald baecher : The evolution of the sssp score editing tools. *Computer Music Journal*, Vol. 3, No. 4, pp. 14–25, 1979.
- [8] A. Cont. A coupled duration-focused architecture for real-time music-to-score alignment. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 32, No. 6, pp. 974–987, June 2010.
- [9] Roger B. Dannenberg. An on-line algorithm for real-time accompaniment. *Proc. of International Cumputer Music Conference (ICMC1984)*, pp. 193–198, 1984.
- [10] Roger B. Dannenberg, Marta Sanchez, Annabelle Joseph, Peter Capell, Robert Joseph, and Ronald Saul. A computer - based multi - media tutor for beginning piano students. *Interface*, Vol. 19, No. 2-3, pp. 155–173, 1990.

- [11] Matthias Dorfer, Florian Henkel, and Gerhard Widmer. Learning to listen, read, and follow: Score following as a reinforcement learning game. In *Proceedings of the 19th International Society for Music Information Retrieval Conference, ISMIR 2018, Paris, France, September 23-27, 2018*, pp. 784–791, 2018.
- [12] Andrew Forsberg, Mark Dieterich, and Robert Zeleznik. The music notepad. In *Proceedings of the 11th Annual ACM Symposium on User Interface Software and Technology, UIST '98*, pp. 203–210, New York, NY, USA, 1998. ACM.
- [13] Michael Good. Musicxml : An internet-friendly format for sheet music. 2002.
- [14] Jan Nieuwenhuizen Han-Wen Nienhuys. Lilypond, a system for automated music engraving. In *Proceedings of the XIV Colloquium on Musical Informatics (XIV CIM 2003), Firenze, Italy, May 8-9-10, 2003*, 2003.
- [15] Akira Maezawa and Kazuhiko Yamamoto. Muens: A multimodal human-machine music ensemble for live concert performance. In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems, CHI '17*, pp. 4290–4301, New York, NY, USA, 2017. ACM.
- [16] M. Prockup, D. Grunberg, A. Hrybyk, and Y. E. Kim. Orchestral performance companion: Using real-time audio to score alignment. *IEEE MultiMedia*, Vol. 20, No. 2, pp. 52–60, April 2013.
- [17] Laurent Pugin. Optical music recognition of early typographic prints using hidden markov models. pp. 53–56, 01 2006.
- [18] A. Rebelo, G. Capela, and Jaime S. Cardoso. Optical recognition of music symbols. *International Journal on Document Analysis and Recognition (IJDAR)*, Vol. 13, No. 1, pp. 19–31, Mar 2010.
- [19] Ian Simon, Dan Morris, and Sumit Basu. Mysong: Automatic accompaniment generation for vocal melodies. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, CHI '08*, pp. 725–734, New York, NY, USA, 2008. ACM.
- [20] Fumiko Watanabe, Rumi Hiraga, and Issei Fujishiro. BRASS: visualizing scores for assisting music learning. In *Proceedings of the 2003 International Computer Music Conference, ICMC 2003, Singapore, September 29 - October 4, 2003*. Michigan Publishing, 2003.
- [21] Yusuke Yamamoto, Hideaki Uchiyama, and Yasuaki Kakehi. onnote: Playing printed music scores as a musical instrument. In *Proceedings of the 24th Annual ACM Symposium on User Interface Software and Technology, UIST '11*, pp. 413–422, New York, NY, USA, 2011. ACM.

- [22] 皆川達夫. 楽譜の歴史. 音楽之友社, 1985.
- [23] 増井俊之. Gyazz-柔軟で強力な万人のための wiki システム. 第 52 回プログラミングシンポジウム予稿集, 第 2011 巻, pp. 43-50, jan 2011.
- [24] 増井俊之. ユニバーサルなテキスト入力システムをめざして. 第 52 回プログラミングシンポジウム予稿集, 第 2011 巻, pp. 1-10, jan 2011.