

デジタルコンテンツのアノテーション 基盤技術とそれに基づく音楽情報処理 に関する研究

梶 克彦

名古屋大学大学院情報科学研究科

2007年1月

目次

概要	1
第1章 序論	5
1.1 背景	5
1.1.1 アノテーションに関する研究動向	7
1.1.2 音楽に関するメディア形式とアノテーションの研究動向	11
1.1.3 音楽コンテンツの応用に関する研究動向	13
1.2 本論文の目的	14
1.3 本論文の構成	15
第2章 任意のコンテンツに対するアノテーション基盤	17
2.1 任意のコンテンツに対するアノテーションについての問題点	18
2.2 ElementPointer	20
2.2.1 ElementPointer の形式	20
2.2.2 ElementPointer プロセッサ	24
2.3 Annphony	24
2.3.1 構造と解釈のためのアノテーション形式	24
2.3.2 アノテーション定義に基づくアノテーション管理	26
2.3.3 Annphony の機能	32
2.4 アノテーションが持つ諸問題に関する考察	34
2.4.1 埋め込み式アノテーションと外部アノテーション	34
2.4.2 Orphan/Misleading Annotation への対処	35

2.5	実装	35
2.6	Annphony と各システム間の関係	36
2.7	Annphony がもたらす効果	38
2.8	本章のまとめ	39
第3章	音楽アノテーションシステム	41
3.1	音楽アノテーションの獲得	41
3.2	音楽アノテーションシステムの機能	43
3.2.1	Annphony の利用	45
3.2.2	Web ブラウザを通したアノテーション	45
3.2.3	3種類のアノテーションエディタ	46
3.3	連続メディアの構造化支援	56
3.4	システム構成	57
3.5	多様な解釈の顕在化に関する実験	58
3.5.1	実験方法	58
3.5.2	実験の結果と考察	59
3.6	本章のまとめ	63
第4章	楽曲の解釈・構造を用いたアプリケーション	65
4.1	検索・再構成システムのためのアノテーション定義	65
4.2	楽曲検索システム	66
4.2.1	検索システムのインタフェース	66
4.2.2	キーワード検索	67
4.2.3	コード進行に基づく検索	67
4.2.4	個人の感性に適応する検索	68
4.2.5	楽曲の構成に基づく絞込み検索	69
4.3	楽曲再構成システム	70
4.3.1	楽曲再構成システムのインタフェース	70

4.3.2	楽曲再構成システムの実装	71
4.4	本章のまとめ	74
第5章	コミュニケーションメディアとしてのプレイリスト	77
5.1	プレイリスト作成支援システムのためのアノテーション定義	79
5.2	歌詞とアノテーションによる楽曲・ユーザ間の類似度推定	80
5.3	プレイリスト作成支援システム	82
5.3.1	協調フィルタリングによる基プレイリストの絞込み	83
5.3.2	基プレイリストの個人への適応	85
5.3.3	インタラクションによるプレイリスト更新	86
5.3.4	プレイリストの人手による編集	86
5.3.5	インタラクション内容に基づくユーザモデルの更新	88
5.4	プレイリストを介したコミュニケーション	89
5.5	ユーザモデルの適応に関する評価実験	91
5.5.1	実験方法	91
5.5.2	実験の結果と考察	92
5.6	本章のまとめ	93
第6章	関連研究	95
6.1	マルチメディアアノテーション	95
6.2	音楽アノテーション	97
6.3	コミュニケーションシステム	99
第7章	結論	102
7.1	まとめ	102
7.2	今後の課題	105
7.3	将来展望	105
7.3.1	CGMサービスの拡張	106

7.3.2 マッシュアップコンテンツ	107
謝辞	108
参考文献	110
研究業績	117

目 次

1.1	コンテンツとアノテーションの関係	7
1.2	MPEG-7 の記述例	8
1.3	RDF のデータ構造	9
1.4	アノテーションの枠組みの間の壁	10
2.1	同一のリソースに対する解釈の多様性	20
2.2	XML ドキュメントの内部ノードの XPointer による指示	21
2.3	MIDI ファイルの時間区分	23
2.4	Named Graphs(左) と Annphony アノテーション(右)	27
2.5	RDF Schema と RDF の関係	28
2.6	データ型定義の例	29
2.7	”impression” アノテーション型の定義の例	29
2.8	”impression” アノテーション型の定義間の関係	30
2.9	Annphony におけるアノテーションの例	31
2.10	複数のアノテーション対象の指定	31
2.11	Annphony と各システム間の関係	37
2.12	コンテンツからメタコンテンツへ	38
3.1	MIDI の内部構造	43
3.2	MusicXML の内部構造	44
3.3	音楽アノテーションシステムのトップページ	47
3.4	Tune Annotator の画面例	48

3.5	アノテーションメニューの動的な変更	49
3.6	Timeline Annotator の画面例	50
3.7	Timeline Annotator のアノテーションメニュー例	51
3.8	時間的セグメントの一括切り出し	51
3.9	Score Annotator の画面例	53
3.10	選択したオブジェクトへのアノテーション付与	53
3.11	アノテーション表示メニュー部	54
3.12	アノテーションのフィルタリング	55
3.13	音楽アノテーションシステムの構成	58
3.14	理由ごとのハイライト区間数	60
3.15	ユーザ A における理由ごとのハイライト区間数	61
3.16	ユーザ B における理由ごとのハイライト区間数	61
3.17	ハイライトと認識された区間の分布	62
4.1	楽曲検索システムの検索フォーム	67
4.2	楽曲検索システムでの検索結果	68
4.3	再構成前の画面	72
4.4	再構成後の画面	73
4.5	MusicXML における歌詞の記述	74
4.6	MusicXML における楽器パートの記述	75
5.1	コミュニケーションツールとコミュニケーションメディア	78
5.2	歌詞・楽曲情景・鑑賞状況の特徴量空間	81
5.3	プレイリスト作成の流れ	83
5.4	状況入力フォーム	84
5.5	ユーザに提示されるプレイリスト	85
5.6	プレイリスト編集画面	87
5.7	歌詞の特徴量空間の基底変換前 (左) と後 (右)	89

5.8	プレイリスト検索画面	90
5.9	評価実験の結果	92
7.1	ブログでのビデオ・辞典の同時引用	108
7.2	ビデオ用例付き辞典システム	109

表 目 次

2.1	Annphony の実装環境	36
3.1	デジタルコンテンツとしての音楽の主な記述形式	42
3.2	各エディタを利用する前提とアノテーション対象となる要素の粒度	46
5.1	鑑賞状況に関するアノテーション項目	80

概要

近年、音楽・ビデオ・写真などのコンテンツが爆発的に増え続けており、そのコンテンツの種類も多様化してきている。このように Web 上に存在しているコンテンツを単に視聴するだけでなく、ユーザに適したコンテンツの検索・推薦や、コンテンツ変換など、高度に利用したいという欲求が高まっている。そのためには、それぞれのコンテンツの構造や関連性といった意味情報を計算機が把握する必要がある。しかし、コンテンツの種類によって異なる論理構造が存在しており、またコンテンツが持つ潜在的な曖昧性により、複数の解釈が可能な場合があるという問題から、コンテンツを機械的に解析して意味を推論するだけでは、コンテンツの理解には限界がある。そこで、人手によってコンテンツに対するメタ情報を付与するアノテーションの研究が進められている。

本論文では、まず Web 上に存在するコンテンツへのアノテーションのあり方を考察する。一般的なコンテンツを、1. コンテンツそのもの、2. 時間や長さ等に関する連続メディア、3. ビデオのシーンや音楽の音符・小節などの論理的構造を持つ離散的なメディアという 3 種類の側面を持つものと捉えると、それぞれに対するアノテーションを付与するには連続メディアや離散メディアの任意の内部点を指し示す必要がある。そこでコンテンツの任意の内部点を指し示す手法である `ElementPointer` と、RDF を拡張してアノテーション自身が識別子を持つアノテーションを管理する基盤技術として `Annphony` を提案する。`Annphony` が `ElementPointer` を採用することで、コンテンツの内部構造に対する多様な解釈の柔軟な定義・作成・管理を実現する。`Annphony` により、コンテンツの種類によらず、その構造や解釈に基づく検索やコンテンツ変換が可能になるとともに、複数のコンテンツを同時に利

用したアプリケーションの構築が容易になる．Annphony の設計・構築を通して、計算機がコンテンツの構造・解釈・関連性を横断的に扱うために、コンテンツの内部を指し示す手法と、コンテンツの多義性を扱う必要があること、アノテーションとその定義を統合的に扱う必要があることを明らかにする．

一般に音楽は必ずしも解釈を一意に決定できず、その楽曲を捉える人によって異なる解釈を見出すことがしばしばみられる．そこで、このように解釈が多義的になりやすいコンテンツの一つである音楽を題材とし、音楽に関するアノテーションを収集するためのシステムを提案する．本システムはユーザによって異なる多義的な解釈を獲得するため、Web 上の多数のユーザからアノテーションを収集する．収集されたアノテーションは Annphony によって管理する．またコンテンツは、連続メディアとして、離散的メディアとしてなど、それぞれの側面ごとに表示形態が異なり、さらに付与される補足情報の種類も異なってくる．そのためそれぞれに対応する 1. 書誌情報などの楽曲自体に対するアノテーション (Tune Annotator), 2. 連続メディアに対するアノテーション (Timeline Annotator), 3. 音楽の論理構造である楽譜に対するアノテーション (Score Annotator) の 3 種類のエディタを備える．収集すべきアノテーションは、そのアノテーションの応用の形態によって異なるため、Annphony に登録されるアノテーション定義に応じて、収集するアノテーションを変更することが可能である．このように、様々な種類のコンテンツのアノテーションを獲得するためには、コンテンツのメディア形式に応じてアノテーションエディタを構築する必要があることを示す．また評価実験をつづけて、多義的なアノテーションを Web 上のユーザからオンラインで収集することが可能であることを明らかにする．

また楽曲に対する多様な解釈に関するアノテーションを用いることで様々な応用が実現可能であることを示すため、認識系応用として楽曲検索システムを、生成系応用として楽曲再構成システムを構築する．楽曲検索システムは、音符や歌詞など楽曲の論理構造に基づく要素に付与されたアノテーションを用いて、キーワード検索・印象検索・コード進行検索・楽曲の構造に基づく絞込み検索を実現

する．特に印象検索では，検索を行うユーザと嗜好の類似するユーザが付与したアノテーションを優先して検索対象とすることで適切な楽曲を発見する．楽曲再構成システムは，サビ・イントロなどの楽曲の構造に関するアノテーションを用いて，例えば「イントロ-Aメロ-サビ-Bメロ-エンディング」という構造の楽曲を，ユーザの要求に応じて「イントロ-サビ-エンディング」のように変換する．これらの応用は，楽曲の論理構造に対する，ユーザごとに異なる解釈のアノテーションを用いることで可能となる．

近年 Web 上では e-mail・オンラインチャット・ブログ・SNS(Social Networking Service)などのコミュニケーションツールを通じて，文書や絵，写真などのコンテンツがコミュニケーションメディアとして意思の伝達に用いられている．このように，コンテンツは人同士のコミュニケーションにおいて利用されるものであるといえる．そのためコミュニケーションの活発化に応じて，Web 上に存在するコンテンツが爆発的に増加してきている．コンテンツを介したコミュニケーションを支援することが，人同士のコミュニケーションをより円滑にし，同時にコンテンツの作成・利用を促進することにつながると考えられる．そこで，近年携帯型音楽プレイヤーの普及などにより注目されてきているプレイリスト(再生曲目リスト)をコミュニケーションメディアとして利用する．プレイリストを介したコミュニケーションを Playlist-Mediated Communication と名付け，その実現を目指し，プレイリスト作成支援システムを提案する．

本システムは楽曲の特徴量として歌詞に加え，ユーザの楽曲解釈の情報である楽曲情景と鑑賞状況に関するアノテーションを利用することにより，ユーザの嗜好と状況に合わせたプレイリストを作成する．本システムに利用されるアノテーションも，楽曲検索システムや楽曲再構成システムと同様，音楽アノテーションシステムにより収集する．プレイリスト作成の手順は，まず基となるプレイリストを協調フィルタリングにより発見する．次に，よりユーザの嗜好に適合するようトランスコーディングを行い，ユーザにプレイリストを提示する．ユーザはプレイリスト中の各楽曲に対して，嗜好に合っているか，状況に合っているかといっ

た情報をフィードバックすることによりプレイリストが洗練される。同時にシステムはこの情報を用いてユーザプロフィールを更新し、各ユーザの嗜好に適合させる。また、プレイリストを介してコミュニケーションを行うために、プレイリスト中の楽曲に対してライナーノーツ (解説文) を記述したり、他のユーザにプレイリストを推薦する機能を持つ。本システムによって、ユーザの嗜好と状況に適応するコンテンツ推薦のために、コンテンツごとに、ユーザによって異なる解釈に関するアノテーションを収集し利用することが有効であることを明らかにする。また、プレイリストがコミュニケーションを円滑にする可能性があることを示す。

第1章 序論

1.1 背景

現在、Web 上には音楽・ビデオ・写真など膨大な量のコンテンツが溢れている。また、それまではコンテンツを消費する側であったユーザが、ブログやSNS(Social Networking Service) など、CGM(Consumer Generated Media) サービスを用いてユーザ自身によってコンテンツを発信するようになり、コンテンツの増加・多様化が進んでいる。そのため Web のユーザ数や年齢層の拡大により、様々な属性のユーザがコンテンツを扱う機会が増加している。さらに計算機のウェアラブル・ユビキタス化に伴い、コンテンツを生活のあらゆる場面で楽しむ環境が整いつつある。

このような現状では、計算機が行うコンテンツ処理には個人適応と状況適応が必要である。ユーザごとに異なる嗜好を踏まえるとともに、ユーザがどのような環境に置かれているかを考慮することで、その時、そのユーザにとって適切な検索や推薦、コンテンツ変換などのコンテンツ処理が実現されるだろう。

個人適応・状況適応の必要なコンテンツの代表例に音楽が挙げられる。音楽は、ユーザごとにジャンルやアーティストなど好みの違いが顕著に現れるコンテンツである。利用可能な楽曲も、iTunes Store [1] や Amazon [2] からの楽曲購入が容易となり、音楽定額制配信サービス等も現れたため、個人が利用可能なデジタルコンテンツとしての楽曲が大幅に増加した [3][4]。さらに iPod 等のポータブル音楽プレイヤーの出現により、音楽を持ち歩き、いつでもどこでも手軽に音楽を鑑賞できるようになった。また日常生活における音楽との接し方から推測されるように、

例えば「夏の海で大勢で楽しむ時に聴きたい曲」と「クリスマスの夜恋人と聴きたい曲」は大きく異なるだろう。このようにユーザの置かれている状況が、楽曲検索や推薦に強く依存すると考えられる。

従来の音楽情報処理の多くは、音響メディアや MIDI などの構造化メディアなど、単一のメディア形式を対象として研究が進められてきたが、音楽情報処理研究をさらに発展させるためには、複数のメディアの形式を同時に扱わなければならないだろう。例えば楽曲検索を例にとってみると、音メディアから認識可能なテンポや音量等の音響特徴量に加え、テキストである歌詞の情報や、画像であるアルバムのジャケットイメージの情報、映像であるプロモーションビデオの情報など、他のメディアを対象とした検索を実現することで、ユーザが目的の楽曲にたどり着く可能性が高まる。このように、音楽に関する研究を行う際に、他のメディアも横断的に利用することが求められる。また複数のメディアを同時に扱う必要性は、音楽情報処理に限らない。Tim O'Reilly が提唱した Web 2.0 においても、Web 上のサービスやリソースが互いに連動することによってコンテンツ処理全般が発展すると述べられている [5]。

膨大な量のコンテンツからの検索や推薦、複数の種類のコンテンツの組み合わせなどを行うには、計算機の支援が無くては困難である。特に重要なコンテンツの意味内容として、構造・解釈・関連性が挙げられる [6][7]。構造に基づく検索や、要約などのコンテンツ変換を行うためにはそれぞれのコンテンツの構造を考慮しなければならず、人によって異なるユーザの嗜好や感性を扱うために、コンテンツに対する多様な解釈を扱う必要がある。また種類の異なるコンテンツ同士を横断的に扱うために、コンテンツ同士の関連性を把握する必要がある。例えば撮り溜めた写真を用いて自動でアルバムを作成する場合に、写真間の関係や撮影の意図などを無視して、色ヒストグラムの類似する画像を表示しても、ユーザにとって整理されたアルバムを作成することができない。このように、意味的なつながりや個人嗜好にそぐわないコンテンツ処理となってしまう。

そこで Tim Berners-Lee は計算機が Web 上のコンテンツの意味内容を把握する

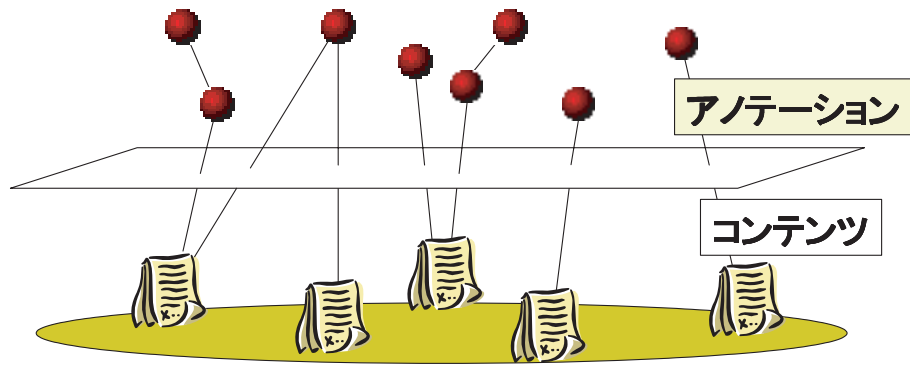


図 1.1: コンテンツとアノテーションの関係

ために、セマンティック・ウェブを提唱した [7]. 適切なアノテーション (メタ情報) を各コンテンツに持たせることによって、コンテンツの意味を計算機が利用可能にするというものである. 現在ではその必要性からアノテーションの管理・収集・利用などに関する研究が進められている [6]. これからアノテーションの研究動向と、個人の嗜好・状況への適応を目指した音楽情報処理に関する研究動向を述べる.

1.1.1 アノテーションに関する研究動向

コンテンツの意味内容を機械的な処理に適用可能な形式で取り込むために、アノテーションに関する研究が進められている. アノテーションとは、図 1.1 に示されるように、あるコンテンツに対して関連付けられるメタ情報である.

アノテーションに関する研究を分類すると、主に 1. 規格化, 2. 収集, 3. 応用の 3 種類に分類されることが考えられる.

アノテーションの規格化 アノテーションの規格化は、コンテンツに付与されるメタ情報の記述能力を左右する重要な点であるといえる. コンテンツの種類に依存したアノテーション形式と、コンテンツの種類に依存しないアノテーション形式に二分される.


```

<?xml version="1.0" encoding="Shift_JIS" ?>
- <Mpeg7 xsi:schemaLocation="urn:mpeg:mpeg7:schema:2001 Mpeg7-
  2001.xsd" xmlns:xsi="http://www.w3.org/2001/XMLSchema-
  instance" xmlns="urn:mpeg:mpeg7:schema:2001"
  xmlns:mpeg7="urn:mpeg:mpeg7:schema:2001">
- <description xsi:type="ContentEntityType">
  - <MultimediaContent xsi:type="ImageType" id="image1">
    <Image>...</Image>
  </MultimediaContent>
</description>
</Mpeg7>

```

図 1.2: MPEG-7 の記述例

コンテンツの種類に依存する形式としては、GDA(Global Document Annotation)[8]や MPEG-7[9] が挙げられる。GDA はテキストに対する構文・意味情報を詳細に記述するための形式である。MPEG-7 はマルチメディア・コンテンツに対するメタデータの表記方法に関する国際標準規格である。正式名称を Multimedia Content Description Interface という。マルチメディア・コンテンツの検索の際に直接の検索対象となる画像・音響特徴データを規格化された手段で表現可能である点が特徴である。MPEG-7 によって画像データのメタデータを記述した例を図 1.2 に挙げる。MPEG-7 は、**Mpeg7** タグから始まる XML 形式で記述され、**description** タグ内に画像から抽出される色ヒストグラム情報や、撮影日時、コンテンツ管理など、規格として定められた属性を入力することができる。

これらの形式は、コンテンツ内部の意味内容に関するアノテーションを記述することを目的としており、それぞれがコンテンツの種類に依存する形式となっている。そのため、横断的にこれらのコンテンツを扱うためには、専門性の高いそれぞれのアノテーション形式を熟知しなければならない。またコンテンツの種類は多様化を続けているため、コンテンツの種類の追加に伴い定義される新しいアノテーション形式に対応しなければならない。

一方で、コンテンツ自体の情報や、コンテンツ同士の関連性などを記述するアノテーション形式として、RDF (Resource Description Framework)[10] が標準化され

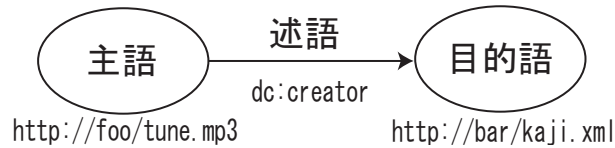


図 1.3: RDF のデータ構造

ている。URI(Uniform Resource Identifier)[11]を持つリソースであれば、コンテンツの形式を問わずアノテーションの対象に指定可能であるため、異種類のコンテンツ間の関連性を表現することができる。従来、コンテンツ間の関連は、HTMLのハイパーリンク構造により導き出されることが一般的であるが、ハイパーリンクには、リンクの意味を記述することができない点が問題であった。RDFのデータ構造は、図1.3に表すように、主語・述語・目的語の組みによって、ラベル付き有向グラフとして表現される。例えばある楽曲の制作者を記述する場合、楽曲のURIを主語に、制作者のURIを目的語に、それらの間の関連性として、制作者であることを示す **dc:creator** という述語を記述する。またアノテーションの述語に相当する部分の柔軟な定義が可能であり、様々な関連性に関する情報を表現することができる。このことから様々な種類のコンテンツ同士の関連の把握を容易にし、多様なアプリケーションに適用可能な形式であるといえる。しかしURIによってコンテンツを指し示すため、それ以上に詳細なコンテンツ内部に関する記述が困難である。そのためコンテンツの種類を横断するアノテーション形式であるRDFはコンテンツの内部構造までを扱うことができない。

図1.4に示すように、コンテンツの内部構造を記述するためのアノテーション形式であるGDAやMPEG-7はコンテンツの種類に依存する。また新規のコンテンツが出現した場合には、そのコンテンツに特化した新しいアノテーション形式が提案されるだろう。一方でコンテンツ間の関連性を記述するためのRDFには、一般にコンテンツの内部構造を扱うことができず、双方の間には大きな壁が存在する。このように多様化するコンテンツとその内部構造や、コンテンツ同士の関連性を統一的に扱う仕組みが求められるが、現在そのような仕組みが存在しないと

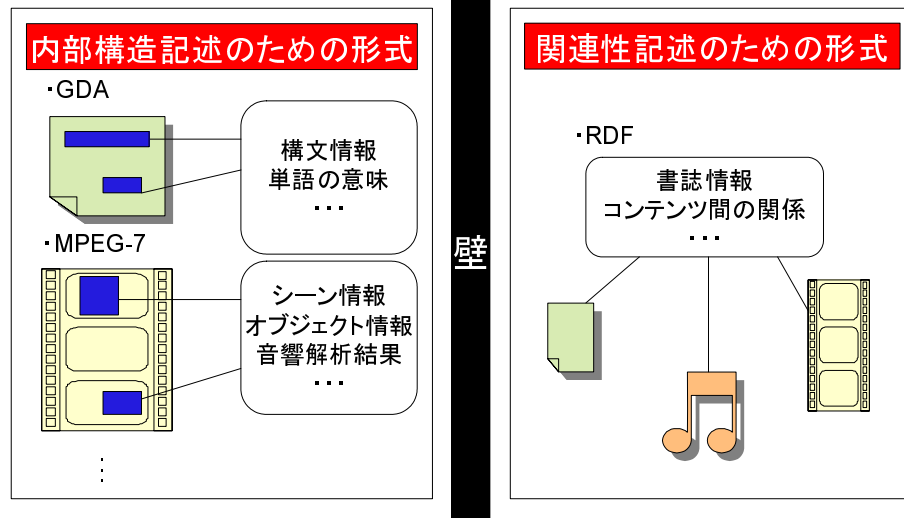


図 1.4: アノテーションの枠組みの間の壁

いう問題がある。

また、アノテーションには計算機によって自動的に生成されるものと人間の手によって作成されるものが存在する。計算機の自動解析によってコンテンツの構造を認識する研究が多く存在するが、一方では自動解析によって得ることの困難な情報を、人間の手によって付与するシステムに関する研究が盛んになってきている [6][12][13]。アノテーションを付与するための特別なエディタを用意し、詳細な構造情報等を付与するというものである。

アノテーションの収集 人手によるアノテーションには、アノテーションの付与のコストという問題が存在する [14]。人手により詳細な情報を付与可能である反面、膨大な量のコンテンツをアノテーションの対象としなければならない。その解決策の一つとして Folksonomy [15] が挙げられる。del.icio.us [16] や Flickr [17] において、Web 上の多くのユーザがそれぞれのコンテンツにタグと呼ばれる分類キーワードを付与することで、そのタグを基に分類やコンテンツへのアクセスを可能にする仕組みを構成している。Web 上の文書や写真といったコンテンツに何

が表現されているかは、計算機がそれを自動解析することが困難であるが、その制作者や閲覧者は容易に理解することができる。そのような情報を幅広いユーザの手によって記述することで、アノテーション付与のコストを軽減することが可能である。このように多くの人の分類作業を蓄積して再利用可能にする仕組みを Folksonomy と呼ぶ。また Web 上の文書に対して、多くのユーザから少しずつ文書情報を収集するシステムも研究が進められており [18][19]、一般大衆から得られる情報を蓄積することで形成される「集合知」に期待が寄せられている。

アノテーションの応用 アノテーションの応用としては、ビデオのシーン検索など、膨大な量のコンテンツの内容を考慮したコンテンツ検索がある。またアノテーションを用いることで、目的に合わせてコンテンツを変換することも可能になる。文書に対する構文や単語の意味に関するアノテーションを用いた応用として、SemCode[6]がある。それは文書の翻訳・要約・音声化・言い換えや、表示端末への適応を行っている。また複数のコンテンツを組み合わせることで新しいコンテンツを生成するための規格化 [20] や研究 [21] が進められている。

以上のことから、現状では多様なコンテンツを対象とした場合にはコンテンツの内部に関する情報を扱うことが困難であり、逆にコンテンツの内部構造を扱うためにはコンテンツの種類を横断することが困難であることが分かる。今後は多様なコンテンツを対象とし、かつコンテンツの内部に関する情報を扱うことが求められようになると思われる。

1.1.2 音楽に関するメディア形式とアノテーションの研究動向

音楽は個人の嗜好・状況への適応が強く求められるコンテンツであり、機械的に音楽利用を支援する研究が進められている。以下に音楽情報処理研究の動向を述べる。

音楽メディアは、MP3 等の音響信号などの連続メディア以外に、音楽の離散的な構造を記述する形式が提案されている。特に音楽の詳細な楽譜構造を表現する

形式として、WEDELMUSIC XML[22] と MusicXML[23] がある。これらの形式は楽譜を構成する情報を記述することができ、また XML 形式で記述されているため、計算機は音符や小節、楽器パートなどの詳細な楽曲構造を把握でき、また楽曲の一部を変更するなどのコンテンツ変換の操作が容易に実現可能である。また音楽の構造化に関する研究も進められており [24]、今後も変更・拡張されていくだろう。そのため音楽のみを扱ったコンテンツ処理であっても、このような多様なメディアに横断的に対応しなければならない。

楽曲のタイトルやアーティスト・ジャンルなどのアノテーションを付与する方法として、MP3はID3タグを埋め込む方式をとっている。一方CDDDB[25]やMusicBrainz[26]は楽曲の書誌情報などを外部アノテーションとして一括管理し、そのアノテーションを対象とした楽曲検索が可能である。CDDDBは拡張性に乏しく、新しい属性を追加することができないが、MusicBrainzではRDFをデータ形式として利用しており、新しい属性を定義することで、柔軟に拡張することができる。このような拡張性は、アノテーションの利用法を特定せず、幅広いアプリケーションに適用可能になるという利点がある。一方で、柔軟に定義可能なアノテーションは、あらかじめ定義が固定されたアノテーションと比較して、他人が定義したアノテーションをどのように共有し、利用を支援するかに関して問題があるといえる。

音楽コンテンツの意味内容の中には、自動認識することの困難な情報が多く存在する。例えば楽曲から受ける印象は、テンポや曲調、歌詞の内容、時代背景などを踏まえて判断する必要がある。一般に自動的に解析することが困難である。また音楽は特に芸術作品としての側面を持ち、コンテンツ自体に多義性が残されていることが多く、人が解釈することによって初めてコンテンツの内容を把握することが可能になる場合が多い。そのため音楽は人手によるアノテーション付与が特に有効となる。音楽に関するアノテーションを人手により付与する研究がいくつか存在するが [27][28][29]、コンテンツの内部構造に対するアノテーション付与ができない、人によって異なる解釈を管理することができないなどの問題点が存在し、音楽の意味内容を記述するためには不十分である。

また音楽のメディア形式の多様性や拡張性の点から、図 1.4 に示したアノテーションの壁は、音楽のみを対象とした場合も存在しており、音楽を対象としてこの壁を取り除くことは、一般的なコンテンツに対するアノテーションの壁を取り除くことにもつながると考えられる。

1.1.3 音楽コンテンツの応用に関する研究動向

音楽コンテンツの応用として、楽曲の推薦 [2][30][31] や配信 [1] など、楽曲単位あるいはその集合単位の利用法に関するサービスや研究が盛んに行われている。近年、特に iPod などのポータブル音楽プレイヤーの普及によりプレイリストの有効性が注目されている。また、様々な CGM の中でも、作曲や演奏をしたり、小説を書くといった敷居の高いコンテンツ制作に比べ、作成が容易であるプレイリストは、より手軽な音楽を楽しむ手段として注目されている。

プレイリストとは、ユーザによって事前に選択された曲目リストである。プレイリストを利用することで、一曲ずつ再生する楽曲を決定するといった煩雑な操作を必要とせず、より容易に楽曲が鑑賞できるようになった。そのためプレイリストの自動作成や推薦に関する研究が進められている [32][33][34][35]。これらの研究では、音楽の音響信号の類似度や、ジャンル・アーティストといった基本情報、ユーザの嗜好、またユーザ同士の類似度などを用いることで、それぞれのユーザに適合したプレイリストを作成、推薦している。

音楽以外の他の種類のコンテンツにも、プレイリストと同様の概念が存在する。写真のアルバム、複数人の作品を集めた歌集や詩集、テレビのコマーシャルフィルムを集めた CM 集などである。また、あるテーマに関連したコンテンツを列挙させるプレイリストなど、プレイリストを構成する要素が複数の種類になる場合も考えられる。これらのことから、プレイリストの作成や利用に関する研究は、音楽情報処理の枠を超えて、コンテンツ処理に関する研究全体に対して貢献すると考えられる。

このように、音楽コンテンツを利用したアプリケーションとして、プレイリスト作成の計算機による支援や、作成されたプレイリストの他のユーザへの推薦などの研究を進めることは非常に有意義であり、コンテンツの種類を横断するクロスコンテンツ処理へ発展する可能性もある。

1.2 本論文の目的

以上の背景を踏まえ、本論文では計算機がコンテンツの意味を把握し、様々な応用が実現可能になることを最終的な目標とする。また前提条件として、本研究で扱うコンテンツは Web 上で公開され、利用可能であることと設定する。特に着目するコンテンツとして音楽を選ぶ。音楽は特に個人の嗜好と状況への適応に対する要求が高く、多様なメディア形式や意味内容を統一的に扱う必要のあるコンテンツである。

まず図 1.4 に示した既存のアノテーションの枠組みの間の壁を取り除くために、任意のコンテンツの任意の部分に対する、ユーザの多様な解釈や構造・関連性に関するアノテーションを記述可能にし、アノテーションとその定義を統一的に管理する基盤技術を構築することを目的とする。

また、特に楽曲に対する構造や解釈・関連性に関するアノテーションを獲得することを目的とする。自動認識の困難な楽曲情報を収集するため、Web 上の一般ユーザを対象とした、オンラインでアノテーションを獲得するシステムを提案し、音楽に限らず、他のコンテンツに関するオンラインアノテーション収集システムを実現する際の指針を示す。

さらに、実際に構造や解釈に関するアノテーションを用いた複数の音楽アプリケーションを提案することにより、計算機がユーザのコンテンツ制作・利用を支援するための手法を示し、また音楽以外のコンテンツへの応用可能性を示すことを目的とする。

1.3 本論文の構成

本論文の構成を以下に示す。

まず2章では、コンテンツに対するアノテーションを管理する基盤技術について述べる。Web上の任意のコンテンツに対するアノテーションのための基盤として Annphony と呼ばれるシステムを提案する。コンテンツの内部要素を指し示す形式として ElementPointer を提案し、さらに既存のアノテーション形式である RDF を拡張することで、コンテンツの構造・解釈に関する情報を扱えるようにする。また記述可能なアノテーションの種類を特定せず、柔軟なアノテーション記述を実現するために、RDFのスキーマ言語である RDFS(RDF Schema)[36]によって記述されるアノテーション定義の管理を行っている。

3章では、アノテーションに基づく音楽情報処理を目指し、音楽に関するアノテーションを獲得するためのシステムを提案する。楽曲情報には、自動認識することの困難な情報が多く存在している。それらの情報を収集するため、Web上のユーザを対象とし、人手によるアノテーション収集を支援する。本システムは2章で述べた Annphony をアノテーションの管理に利用することで、楽曲の構造や解釈・関連性に関するアノテーションを収集する。また Annphony で管理されるアノテーション定義を、アノテーション収集のためのシステムに適用することで、収集するアノテーションを柔軟に追加・変更することが可能である。これにより特定のアプリケーションに限定されないアノテーション収集が実現される。

4章、5章では、音楽アノテーションを複数のアプリケーションで利用する手法について述べる。まず4章では楽曲検索・楽曲再構成システムについて述べる。これらのシステムで利用されるアノテーションを定義し、3章の音楽アノテーションシステムに適用することで、印象や楽曲構成などのアノテーションを収集し、利用する。5章では新しい音楽の利用形態として、プレイリストを介したコミュニケーション (Playlist-Mediated Communication) を提案する。

一般的に音楽のプレイリストは作成が手軽であり、プレイリストへの解説や感

想が頻繁にやり取りされるため、Web上のユーザによるプレイリストを中心としてコミュニケーションが成立すると考えられる。そこで、嗜好と状況に適合するプレイリスト作成の支援を行い、さらにそのプレイリストに対する感想や解説を投稿し合うことで、ユーザ同士がオンラインでコミュニケーションを行うシステムを実現する。本システムでは、楽曲そのものから認識することが困難な情報である楽曲情景(楽曲中で表現されている情景)・鑑賞状況(その楽曲を鑑賞したい状況)というアノテーションを音楽アノテーションシステムによって収集し、利用している。

さらに6章では関連研究について述べ、7章で本論文をまとめ、今後の課題について述べる。また将来展望として本論文で提案する音楽を軸としたアプローチが、音楽情報処理以外にも適用可能であることを述べる。

第2章 任意のコンテンツに対するアノテーション基盤

Web上に存在する大量のデジタルコンテンツを有効利用するため、テキストや音楽などのコンテンツに対してメタ情報(アノテーション)を付与し利用する研究が盛んに行われている [6][12]. これらの研究から、コンテンツの意味内容を考慮し、それぞれのユーザに適したコンテンツ処理を行うことの重要性と、そのためにはコンテンツの構造に加えコンテンツの詳細部分にまで踏み込んだ個人の解釈・嗜好情報が必要であることがわかる.

しかし、1.1.1 節で述べたように、現在までに提案されているデジタルコンテンツに対するメタデータの表現形式は、コンテンツの構造を記述するためにコンテンツの種類に強く依存した形式となる仕組みと、異なるコンテンツ間の関連性を記述することができるがコンテンツの内部構造に踏み込むことができない仕組みの二通りに分断されてしまっている. またコンテンツの種類も多様化を続けており、画像・音楽・文書などのコンテンツを横断的に利用し、意味内容と個人性を考慮した応用研究を進めるにあたっての障害となっている.

また、横断的にコンテンツを扱うためには、それぞれのコンテンツの意味内容を記述するためのアノテーション定義を共有しなければならない. 同時に、コンテンツの種類は多様化を続けているため、コンテンツの種類の追加に伴い定義される新しいアノテーション形式に対応しなければならない.

本章では、1. 任意のコンテンツに対するアノテーションを実現するために必要な要素について、2. 我々が構築しているアノテーション基盤 Annphony(アンフォニー)におけるアノテーションとそのスキーマの形式について、3. Annphony に

においてアノテーションやスキーマの利用・共有を支援する機能について述べる。

2.1 任意のコンテンツに対するアノテーションについての問題点

現在デジタルコンテンツに対するアノテーションの記述形式がいくつか提案されている。例えばテキストに対する構文・意味情報を詳細に記述するための GDA(Global Document Annotation)[8] や、テキストコーパスに対するアノテーション形式である CES(Corpus Encoding Standard)[37] などがある。Flickr[17] は Web 上で写真を共有するサービスであるが、独自のアノテーション形式により画像の矩形部分に対するコメント付与を実現している。これらの形式はコンテンツの種類に閉じており、例えば Flickr において記述されたコメントに対して、GDA による詳細な言語構造を記述するといった、あるコンテンツのアノテーションを異種コンテンツに適用することが困難である。このように、現在提案されているコンテンツの内部構造にまで踏み込んだアノテーション形式は互換性が乏しいといえる。

一方で、コンテンツ全般に対するアノテーションの形式としてリソース間の関係をグラフ構造で記述する RDF(Resource Description Framework)[10] では、URI(Uniform Resource Identifier)[11] を持つ対象をリソースと呼び、そのリソースに対するメタ情報を主語・述語・目的語の組で表現する。つまり、どのようなリソースであってもそれを URI で表現することができれば、そのリソースに対するアノテーションを記述することが可能である。

コンテンツの意味情報を扱うためには、そのコンテンツの詳細な部分へのアノテーションを実現しなければならない。なぜなら、コンテンツの構造の一部に対し、その部分がコンテンツの他の部分や、他のコンテンツとどのような関係性を持っているか、またその部分は人によってどう解釈されるかという情報を付与する必要があるからである。そこで、まずコンテンツの構造を表現するために、コンテンツの

任意の箇所を指し示す方式としては、XML の特定のノードを指す XPointer[38] や、ビデオ・オーディオなどの連続メディアの任意の箇所を指し示す URI time interval specification [39] など、いくつかの形式が提案されており、今後も新しいメディア形式が提案されると予想される。しかし、未知のメディア形式を含むそれらの全てを網羅するアノテーション形式は存在しない。さらに、提案されている形式では指し示すことのできない場合が多く存在する。そのため、「MIDI の特定のチャンネルにおける時間範囲」といった、任意のコンテンツの、任意の箇所を指し示すことのできる柔軟な表現形式が必要である。

図 2.1 に示すように、音楽や絵画などの芸術作品には、鑑賞者によって解釈は様々に異なる場合が多い。また楽曲の構造化を行う GTTM(Generative Theory of Tonal Music)[24] では、楽曲の構造を一意に決定することができない場合がある。そのため複数の解釈に基づく構造が存在する可能性がある。そのような多義性のある構造の一部に、関連性や解釈などの意味情報を付与するためには、同一のメディア形式が複数の構造を持つことを許す必要がある。しかし RDF や GDA では、一つのメディア形式に対する複数の構造を扱うような柔軟な関連性記述ができないという問題がある。

また構造や解釈といった様々なアノテーションを定義する際には、既存の定義を利用・拡張することで、アノテーションの再利用性が高まると考えられる。例えば異なるコンテンツ間で共通の属性を定義し、各コンテンツのアノテーションでその属性を利用することで、コンテンツの種類にこだわらずそのプロパティを利用することができるだろう。そのためアノテーションのスキーマの存在が必須となる。

Flickr ではアノテーションを利用するための Flickr API が用意されており、また RDF に関しては Protégé[40] や RDF Gravity[41] など、RDF の記述・検索・可視化を行うアプリケーションが提供されている。様々な種類のアノテーションを幅広く収集し、利用可能にするためには、これらのシステムのように、幅広いユーザが利用可能なアノテーション管理基盤が必要である。しかしこれらのシステム

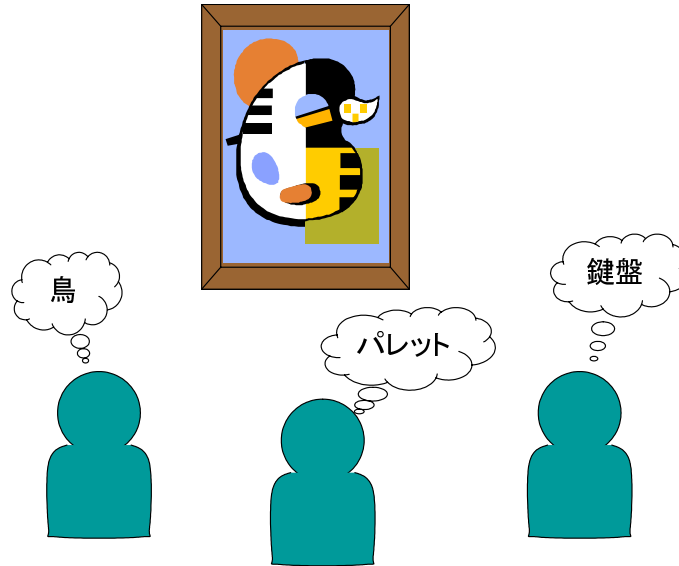


図 2.1: 同一のリソースに対する解釈の多様性

では、任意のコンテンツの内部要素に対するアノテーションを扱えない。

2.2 ElementPointer

前節で述べたように、コンテンツの内部構造に対するアノテーションを扱うためには、そのコンテンツのセグメントを指し示す手段が必要である。任意のメディアの任意のセグメントを指し示す形式として ElementPointer を提案する。

2.2.1 ElementPointer の形式

ElementPointer を XPointer の書式を参考にして定義する。XPointer は以下の形式で XML ドキュメントの任意のノードを指し示す。

[Content]#xpointer([XPath])

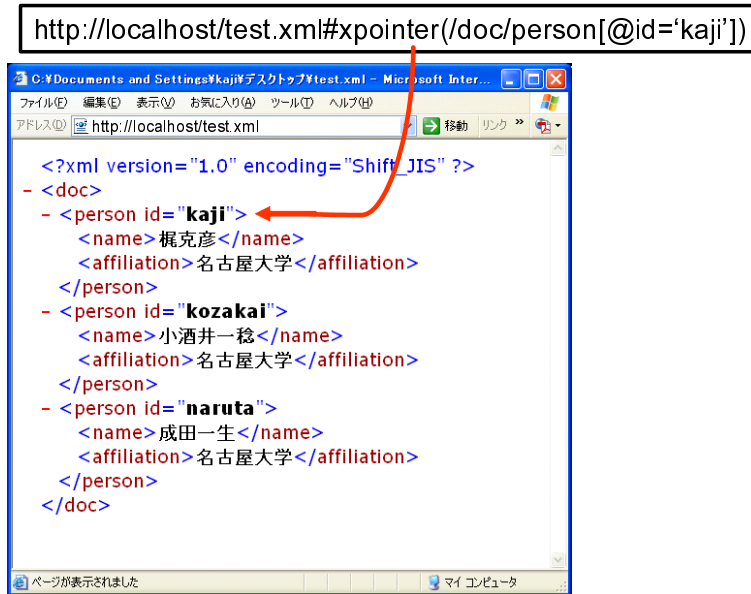


図 2.2: XML ドキュメントの内部ノードの XPointer による指示

コンテンツの URI である [Content] に続き、**#xpointer** 以降 [XPath] に XML ドキュメント内部ノードを指し示す手段として標準化されている XPath [42] を記述することで、内部構造への言及を可能にしている。全ての中括弧は表記には含まない。例えば図 2.2 のように、XML ドキュメント (`http://localhost/test.xml`) の、ルートノード (`doc`) の子ノードである `person` の中で、`id` 属性が `kaji` であるものを指し示す XPointer が示されている。XPointer は一般的に以下のように表現される。

一方で、任意のコンテンツの任意の箇所を指し示す XPath に相当する形式が存在しない。そこで ElementPointer ではコンテンツ内部を指し示すための定義を別途用意し、その定義を引用する。また、例えば図 2.3 でハイライトされている、「MIDI のチャンネル 1 における、楽曲開始後 10 秒から 20 秒までの時間範囲」のようなセグメントを指し示すためには、MIDI のチャンネル番号、開始時間、終了時間という属性名、属性値が必要となる。そこで、ElementPointer は以下のように、定義と、属性名・属性値の組を列挙する形式をとる。

[Content]#epointer([Schema](([Prop1],arg1),([Prop2],arg2),...))

コンテンツの URI である [Content] に続き、以降にフラグメント識別子としてコンテンツのどの部分を指し示すかを記述する。本手法では、コンテンツの内部は、[Schema] で表される ElementPointer 定義の URI 以降に、[Prop1],[Prop2],... で表される有限個の属性の URI とその値を列挙することにより指し示される。これらの属性はコマンドに対する引数の役割を持つ。実際には URI としての妥当性を保つため、[Schema](([Prop1],arg1),([Prop2],arg2)...) の部分を URL エンコーディングする。URL エンコーディングによって、URI に含めることのできない文字を妥当な文字に変換することができる。

アノテーション共有の必要性から、ElementPointer の定義は共有に適した一般的な形式で記述されるべきである。ElementPointer は、コンテンツの任意の内部要素を指し示すことで、コンテンツの構造を表現するための形式であるため、意味的なアノテーションの一つであるといえる。そこで、リソース間の関係を表すためのフレームワークであり、一般的なアノテーション形式として知られている RDF のスキーマ言語である RDFS(RDF Schema)[36] を ElementPointer の定義に採用した。リソース間の関係をグラフ構造として表現する RDF や RDFS を用いることで、その検索言語である SPARQL(SPARQL Query Language for RDF)[43] によって、コンテンツの構造を扱う際に有効なグラフ検索が可能になる。そのため ElementPointer の定義を共有することに適していると考えられる。Annphony においてその定義の検索や利用を支援し、任意のユーザによる ElementPointer の定義の利用を支援する。

ElementPointer の具体例を示す。前述の「MIDI のチャンネル 1 における、楽曲開始後 10 秒から 20 秒までの時間範囲」を指し示すため、MIDI チャンネル別の時間範囲を表す ElementPointer を定義する。本定義で利用される属性とそのデータ型として、MIDI のチャンネル番号 (1 から 16 までの integer 型)、時間区分の開始時間 (double 型)、終了時間 (double 型) を記述する。また時間区分の開始時間、終了時間は楽曲の演奏の開始からの秒数であることを同時に記述する。具体的な RDFS

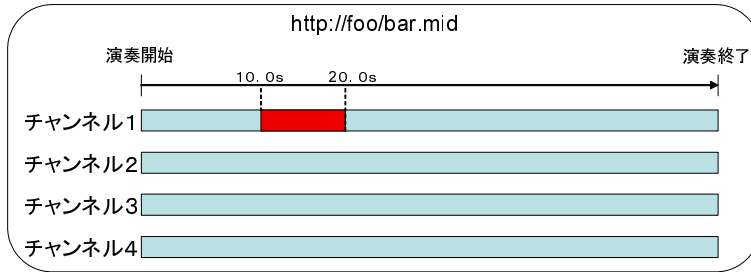


図 2.3: MIDI ファイルの時間区分

の書式については 2.3.2 節において述べる．こうして記述された定義は Web 上に公開するか，Annphony に登録し，URI を持たせることで利用可能になる．便宜的に「MIDI の特定のチャンネルにおける時間範囲」に対する ElementPointer の定義の URI を [Schema]，チャンネル番号，時間区分の開始時間，終了時間の各属性の URI は [Content]，[From]，[To] と表現すると，ElementPointer により，ある MIDI データ (<http://foo/bar.mid>) における，チャンネル番号が 1 番の，10.0 秒から 20.0 秒までの範囲を指し示す ElementPointer は以下のように表現される．

```
http://foo/bar.mid#epointer([Schema]([Channel],1),([From],10.0),([To],20.0)))
```

実際には URI の規格に準拠するため，URL エンコーディングを施す．[Schema]，[Channel]，[From]，[To] に URI を埋め込み，URL エンコーディングを行った具体的な例を以下に示す．

```
http://foo/bar.mid#epointer(http%3A%2F%2Flocalhost%2Fannphony%2F
schema%2Fmidi.rdfs%23midi%28%28http%3A%2F%2Flocalhost%2F
annphony%2Fschema%2Fmidi.rdfs%23channel%2C1%29%2C%28
http%3A%2F%2Flocalhost%2Fannphony%2Fschema%2Fmidi.rdfs%23from%2C
10.0%29%2C%28http%3A%2F%2Flocalhost%2Fannphony%2Fschema%2F
midi.rdfs%23to%2C20.0%29%29)
```


また、例えば画像の矩形範囲を指定するためには、画像の X 座標、Y 座標、幅、高さが決定される必要がある。そこで同様に X 座標、Y 座標、幅、高さの 4 種類の属性を記述し、さらにそれぞれの値のデータ型を指定する。またそれぞれの値は、画像の左上を頂点とし、それぞれの数値がピクセル単位であることを記述することで、画像の矩形範囲を表す `ElementPointer` を定義する。

2.2.2 `ElementPointer` プロセッサ

また `ElementPointer` によって指し示された部分を扱う処理系を同時に作成し、後述する `ElementPointer` の定義へのアノテーションとして関連付けを行えば、Annphony においてそのセグメントのプログラム中での取得・利用が可能になる。`ElementPointer` プロセッサは、指定された部分の存在確認や、実際に指定された部分の取得など、部分同士の演算を行うためのメソッドで構成される。例えば上記の例のように矩形範囲指定により二次元のメディアを指し示す `ElementPointer` プロセッサとして、その矩形範囲の画像を取得して表示するプログラムを用意しておくことで、該当部分が抜き出された画像を容易に扱うことが可能になる。

2.3 Annphony

楽曲に対する複数の解釈を管理するために、任意のデジタルコンテンツに対するアノテーションを扱う基盤である Annphony を構築した。以下に概要を述べる。

2.3.1 構造と解釈のためのアノテーション形式

リソースの関係を有向グラフで表現するアノテーション形式として RDF が提案されている。1.1.1 節で述べたように、URI を持つリソースであれば、コンテンツの形式を問わずアノテーションの対象に指定可能であるため、異種類のコンテンツ間の関連性を表現することができる。しかし 2.1 節で述べたように、RDF は一

意の解釈や定義の記述を目的とした形式であり、多様な解釈を記述することに適していない。一方 RDF を拡張した形式として提案されている Named Graphs [44] は、任意のグラフのまとまりを一意に識別し、他と区別するために、グラフのまとまりに識別子を付与することができる。図 1.3 のように、RDF では「この楽曲は(主語) A さんに(目的語) 作成された(述語)」といった、主語・述語・目的語の 3 つ組みで記述されるが、Named Graphs はそれに加え、そのアノテーションの識別子であるアノテーション URI を持った 4 つ組みで表現される。本形式は一つ一つの解釈が区別されることから、複数の解釈を扱う形式として適していると考えられる。そこで Annphony では Named Graphs をアノテーション形式の基礎に採用した。Web 上で普及している URI の枠組みに基づいてアノテーションを指し示すため、アノテーションの再利用性が高まり、任意のアプリケーションによるアノテーションの利用を可能にする。

Annphony が扱うアノテーションは人により記述されるため、アノテーション自体の情報が不十分であったり、誤っている場合が考えられる。そのため、そのような人が記述した不十分な解釈のアノテーションに対して、そのアノテーションを補足するさらなるアノテーションが必要になるだろう。しかしどのようなアノテーションに対してさらなる情報が必要であるかを事前に予測することは不可能であるため、あらかじめ最小単位のアノテーションに対して識別子が付与される必要がある。Named Graphs では、ある楽曲 T1 と別の楽曲 T2 についての関連性と、楽曲 T3 と楽曲 T4 についての関連性とを同時に記述し、それらをまとめて一つの URI で表現することが可能であったが、T1 と T2 の関連性に対して言及する際に、直接その部分を指し示すことができない。そこで Annphony では任意のグラフのまとまりに対して識別子を付与することができるという Named Graphs の機能に制限を加える。Named Graphs では、Named Graphs では図 2.4 の右側のよう A, B, C の各部分を分離して URI を付与することも可能であるが、左側のよう A, B, C というアノテーションをまとめたものに対して一つの識別子を付与することもできてしまう。この場合識別子が付与されていない A, B, C の各部

分への言及が困難である。Annphony では、図 2.4 の右側のように、それぞれのリソース記述を明確に分離する。具体的には、あるアノテータが、ある主語(リソース)に対して述語・目的語の組を列挙したものを一つのアノテーションとし、それぞれのアノテーションに URI を発行する。

音楽のプレイリスト中に含まれる複数の楽曲を対象としてコメントを付与する場合など、複数の楽曲という単一ではないリソースを主語としたアノテーションが想定される。そこで Annphony ではアノテーションの主語として複数のリソースを指定することを許可している。また図 2.4 の左側において A, B, C が何らかの意図を持ってまとめられていたということも、A, B, C という複数のリソースを主語とするアノテーションにより表現可能である。

コンテンツの代表的な構造として、グルーピング構造、ツリー構造、グラフ構造が挙げられる。RDF はリソース間の関係をグラフ構造で表現する形式であり、グラフ構造はツリー構造を内包するため、RDF はこれらの構造を扱うことのできる形式であるといえる。しかし RDF では、主語に直接複数のリソースを指定できないという、グルーピング構造を表現するために問題がある。そこで Annphony ではそれに加え、前述の ElementPointer によるコンテンツのセグメント指定と、主語に複数のリソースの記述を許すアノテーション形式により、コンテンツのグルーピング構造を表現する。例えば楽譜中に表れる音符や休符群をまとめて、そのグループに対してイントロやサビなどの楽曲構成情報などを関連付けることが可能になる。以上より Annphony はコンテンツの代表的な構造に関する情報を扱える基盤である。

2.3.2 アノテーション定義に基づくアノテーション管理

従来のコンテンツの内部構造に言及するアノテーション記述形式 [8][9] は、アノテーション記述のための全ての仕様が固定されており、拡張性が低い。そのためこれらの形式で記述することのできないアノテーションを利用することが困難で

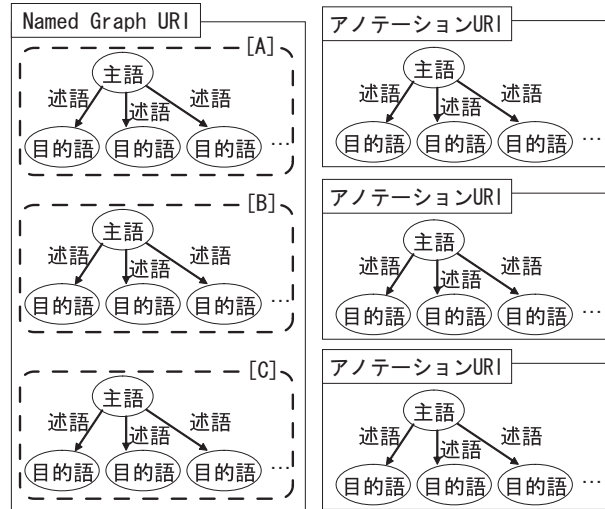


図 2.4: Named Graphs(左) と Annphony アノテーション (右)

ある。そこで、Annphony のアノテーション形式のベースである RDF のスキーマ言語である RDFS を、アノテーション定義のための言語に採用した。取得すべきアノテーションの形式を RDFS で記述することにより、Annphony はそれに基づく検索やアノテーション利用を支援する。

スキーマ (アノテーション定義) とはデータの構造を決定するものである。図 2.5 に RDF におけるスキーマとデータの関係を示す。左側の RDFS では、アノテーションの種類として「楽曲の基本情報」を定義し、その中では、アーティストやタイトルなど複数の属性の記述を許可する。また各属性のデータ型の指定を行う。図 2.5 の右側は、そのスキーマに基づいた RDF である。RDFS で定められた「楽曲の基本情報」というアノテーション定義に則り、アーティストやタイトルなどの属性を指定されたデータ型で記述する。また、どのリソースを対象としたアノテーションであるかを同時に記述する。

Annphony におけるアノテーション定義の記述方法を以下に述べる。例として、事前に決められた「陽気」「刺激的」「厳格」という 3 種類の印象語のいずれかを値として持つ「印象情報」という名前のアノテーションを定義する場合を挙げる。

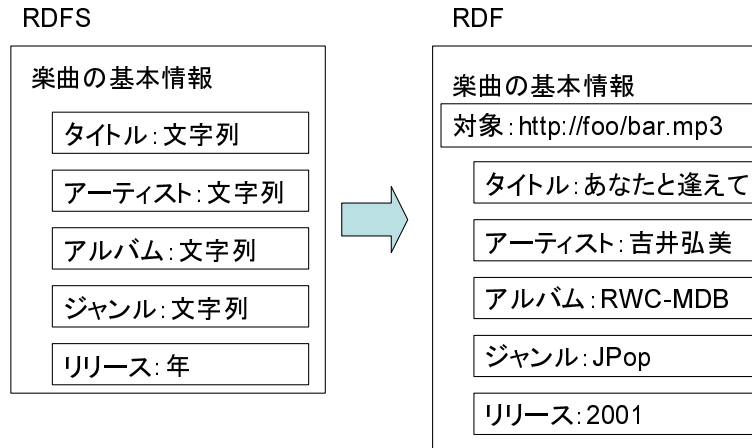


図 2.5: RDF Schema と RDF の関係

まずこれら3種類の印象語のうちいずれかの値のみを許可するデータ型を定義する。データ型の定義はXML Schema [45]により記述される。XML SchemaはXML文書の構造を定義するスキーマ言語の一つであり、柔軟にデータ型を拡張することができる特徴を持つ。一般的にRDFSではXML Schemaにおいて定義されるデータ型が利用されている。図2.6に実際のデータ型の定義を示す。XML Schemaにおいて定義された語彙をxsd:という接頭辞を用いて利用する。データ型の拡張にはsimpleTypeを用いる。この例ではXML Schemaで定義された基本データ型であるstring型をxsd:restrictionという語彙で拡張している。xsd:enumerationにより列挙された3種類の語の中から一つを選択するためのデータ型が定義されている。

次に、これらの印象語を値として持つアノテーションの定義を図2.7のように記述する。RDF, RDFSにより定義されている語彙は、rdf:やrdfs:という接頭辞を用いて利用している。RDFSにおいて新しいアノテーション定義を行うために、まずrdfs:Classを記述する。属性として、その定義を利用するために必要なURIを、rdf:aboutという属性において記述する。rdfs:labelにはその定義の名前を、rdfs:commentにはさらに詳細な説明を自然言語で記述する。次に、rdf:Property

```

<xsd:simpleType name="impressionType">
  <xsd:restriction base="xsd:string">
    <xsd:enumeration value="陽気"/>
    <xsd:enumeration value="刺激的"/>
    <xsd:enumeration value="厳格"/>
  </xsd:restriction>
</xsd:simpleType>

```

図 2.6: データ型定義の例

```

<rdfs:Class rdfs:about="http://foo/newschema.rdfs#impression">
  <rdfs:label>印象情報</rdfs:label>
  <rdfs:comment>定義の説明</rdfs:comment>
</rdfs:Class>
<rdfs:Property rdfs:about="http://foo/newschema.rdfs#impression_term">
  <rdfs:label>印象語</rdfs:label>
  <rdfs:comment>プロパティの説明</rdfs:comment>
  <rdfs:domain rdfs:resource="http://foo/newschema.rdfs#impression"/>
  <rdfs:range rdfs:resource="http://foo/datatype.xsd#impressionType"/>
</rdfs:Property>

```

図 2.7: "impression" アノテーション型の定義の例

において、利用可能な属性を定義する。属性にも同様に、属性の URI、名前、説明を記述する。また、どの定義に対してこの属性が適用されるかを **rdfs:domain** の属性である **rdfs:resource** において指定する。さらに、その属性がどのようなデータ型であるかを **rdfs:range** において指定する。rdfs:Class である **impression**, rdfs:Property である **impression_term**, **impression_term** のデータ型である **impressionType** 間の関係は図 2.8 のようになる。impression_term のドメインとして **impression** が、range として **impressionType** が指定されていることがわかる。ここで指定されているデータ型である **impressionType** は、図 2.6 において定義した 3 種類の印象語からいずれかを選択するデータ型として定義されている。以上で印象情報のアノテーションが定義された。

上記のアノテーション定義に基づいた実際のアノテーションの例を図 2.9 に示す。**an:** は Annphony で定義されている語彙を、**def:** は上記のアノテーション定義を利

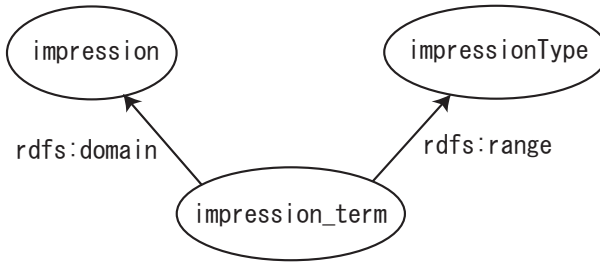


図 2.8: "impression" アノテーション型の定義間の関係

用するための接頭辞である。この例では図 2.9 で定義した印象情報アノテーションを表す **def:impression** が指定され、ある楽曲 (<http://bar/tune.mp3>) に「厳格」というアノテーションを付与している。属性である **rdf:about** の URI は、Annphony が自動的に発行するアノテーションの識別子である。アノテーションの対象となるリソースは **an:target** に、また、アノテータの URI は **an:annotator** において指定されている。楽曲そのものに対してのアノテーションである場合は、**an:target** としてその楽曲の URI を、ある楽曲の区間に対するアノテーションの場合、2.1 節で述べた **ElementPointer** の URI を記述する。またそれらのリソースの集合を対象とする場合、図 2.10 のように、RDF のコレクション表現のための形式である **rdf:Seq**(順序付リスト)、**rdf:Bag**(順序無し集合)、**rdf:Alt**(代替表現リスト) を用いて複数の対象を列挙する。印象語を記述する属性である **def:impression_term** には、図 2.6 において定義された範囲の値を記述する。

このように Annphony はアノテーション定義に基づくアノテーションを管理する。アノテーション定義は任意のユーザにより記述されるため、Annphony によりコンテンツの内部構造や解釈に対する幅広い種類のアノテーションを管理することができる。2.1 節で述べた **ElementPointer** の定義も、本アノテーション定義と同様の形式で記述されることで、有限個の属性を列挙する定義によって任意のコンテンツの任意の部分を指し示すことが可能になる。

```
<rdf:RDF
  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:an="http://foo/annotation.rdfs#"
  xmlns:def="http://foo/newschema.rdfs#">
  <def:impression
    rdf:about="http://foo/annotation?key=1">
    <an:target rdf:resource="http://bar/tune.mp3"/>
    <an:annotator rdf:resource="http://bar/user.xml"/>
    <def:impression_term>厳格</def:impression_term>
  </def:impression>
</rdf:RDF>
```

図 2.9: Annphony におけるアノテーションの例

```
<an:target>
  <rdf:Seq>
    <rdf:li rdf:resource="http://bar/resource1.obj"/>
    <rdf:li rdf:resource="http://bar/resource2.obj"/>
  </rdf:Seq>
</an:target>
```

図 2.10: 複数のアノテーション対象の指定

2.3.3 Annphony の機能

幅広くアノテーションを共有・利用するためには、アノテーションを利用する環境を整備しなければならない。Annphony では、アノテーションやその定義を容易に共有・検索・作成・利用する機能、またコンテンツの構造・解釈を扱うために必要となる機能を提供する。さらに、アノテーションが潜在的に抱える問題に対処するための機能を備えている。以下に Annphony の主要な機能について述べる。

ElementPointer の利用支援

Annphony は ElementPointer を扱う機能を備える。ElementPointer 定義の URI と対象コンテンツの URI を指定し、また ElementPointer の定義に沿った属性の値を全て設定することで ElementPointer URI が生成される。逆に ElementPointer の URI から、コンテンツの URI や各引数の値を取得することも可能である。さらに、ElementPointer 定義に記述される ElementPointer プロセッサを登録し、利用するための機能を備える。

アノテーション定義へのアノテーション

アノテーションの定義が登録される際、定義の作成者はその定義へのアノテーションを同時に記述する。具体的には定義者情報、自然言語による利用例などの説明、「Jazz」、「ニュース」、「判例」など複数のタグによる適用可能なコンテンツの列挙を行う。

どのような場面においてその定義が利用可能であるかという情報は、定義の作成者本人が全て網羅することは困難である場合が多い。そこで Annphony では、定義者以外の複数のユーザがそのスキーマに対して利用例やタグなどのアノテーションを行うことを許している。また ElementPointer の定義へのアノテーションの場合、上記のアノテーションに加え 2.2 節で述べた ElementPointer プロセッサを指定する。

また Annphony はそれらのアノテーションに基づくアノテーション定義の検索機能を備えている。アノテーション定義を発見したいユーザは、キーワードでの検索や、適用するコンテンツのタグ情報から絞り込むことで、目的の定義の発見や、利用・拡張定義が可能となる。

アノテーション定義に基づくアノテーションの検索

幅広いコンテンツへのアノテーションを扱う環境では、動的にアノテーションの定義が増加することが予想されるが、アノテーションの定義が追加される度に、その定義に基づくアノテーションを利用するようアプリケーションを対応させることは非常に高コストになる。

Annphony では、アノテーション定義の継承関係を考慮し、着目した定義の継承元や継承先のアノテーションに関しても検索対象とすることができる。また、ある特定の属性に着目し、その属性が利用されている定義に基づくアノテーション全てを検索対象にするなど、アプリケーションが、動的に追加されたアノテーション定義に基づくアノテーションを利用できる機能を備えている。

コンテンツの構造記述

様々なコンテンツの構造化に関する研究が行われているが、それらに多く見られる手法として、コンテンツのグループ化、または階層化が挙げられる [8][24]。Annphony では ElementPointer の採用により、コンテンツの任意の箇所のグループ化に対応している。また、コンテンツやアノテーションを一つのノードとみなし、それらの関係をツリー形式で記述し、そのツリーを利用する機能を備える。さらに、アノテーションの基本構造として RDF を採用しているため、他のコンテンツとの間の複雑なグラフ構造の関係を扱うことができる。

アノテーション信頼度の導入

アノテータに関する情報は、コンテンツの変換や推薦など、個人の嗜好や解釈を踏まえた応用を実現する際に必須となる。Annphony ではそれぞれのアノテータに URI を持たせ、そのプロファイルをアノテーションとして管理する機能を提供する。

また、あるコンテンツに対し複数の解釈が存在する場合、それぞれの解釈が正確であるか、または信頼できる情報であるかを判断する必要がある。そこで各アノテーションとアノテータに「信頼度」という属性を付与する。信頼度の算出に関しては、文献 [12] において提案された手法を採用した。

アプリケーションはアノテーションの信頼度に基づき、複数のアノテーションの中から最も信頼性の高いアノテーションを採用したり、信頼度が閾値以上である全てのアノテーションを採用したりするなど、複数の解釈の扱いを決定することができる。

2.4 アノテーションが持つ諸問題に関する考察

2.4.1 埋め込み式アノテーションと外部アノテーション

MP3 形式の音楽ファイルにはアーティストやタイトルなどのメタデータをそのファイル自身に埋め込むことができ、Exif(Exchangeable Image File Format) 形式の画像には、撮影日時やその画像についての情報を埋め込むことができる。このようなコンテンツの内部にアノテーションを埋め込む方式の場合、コンテンツとアノテーションが切り離されないため、利用が容易であるが、コンテンツとアノテーションを同時に管理する必要がある。しかし様々な種類のコンテンツを対象としたアノテーションを扱う場合、必ずしもコンテンツとアノテーションが同時に管理されとは限らない。そのため Annphony ではコンテンツとアノテーションを切り離した管理方法を採用している。

2.4.2 Orphan/Misleading Annotation への対処

コンテンツとそのアノテーションが分離されている場合、コンテンツの改変・移動・削除によって、そのコンテンツに関連付けられたアノテーションが Orphan Annotation(指し先の無いアノテーション) や Misleading Annotation (誤解を生むアノテーション) になってしまう危険性が指摘されている [18]. そこで Annphony ではこれらの問題の発生を抑えるために以下の機能を提供する.

Orphan Annotation への対処のため、指し先が存在しないアノテーションを登録することを許可しない. データベースへの登録時に、アノテーションの指し先のリソースへアクセスを試み、存在を確認した後に登録される. 指し先が ElementPointer によって記述された URI の場合は、前述の ElementPointer プロセッサによって該当するコンテンツのセグメントの存在を確認する.

一般的にポータルサイトと呼ばれるページや、Web ニュースのトップページなどは、頻繁に更新される. このように、対象となるコンテンツやアノテーションが編集されることで、そのリソースを指し示しているアノテーションが意味をなさなくなったり、別の意味になったりしてしまうことを Misleading Annotation と呼ぶ. Annphony では、アノテーションが編集・再登録された場合、そのアノテーションに関連するアノテーションを取得し、それらの確認・編集を管理者に促す.

また一定時間ごとに登録されたアノテーションを巡回してコンテンツの編集の有無を確認し、編集や削除が確認されたコンテンツに関連するアノテーションに関しても同様にそれらの確認・編集を促す.

2.5 実装

実装環境を表 2.1 にまとめる. Annphony は様々なアプリケーションやアノテーションエディタによるアノテーションやアノテーション定義の登録や検索の要求などをブラウザなどの Web クライアントから受け付けるため、Web サーバとして実装されている. Annphony は XML-RPC(XML-Remote Procedure Call) [46] によっ

て要求を取得し、検索結果の送信などを行う。また、クライアント側は Annphony APIを用いることでサーバへの要求や、アノテーションの生成を行う。RDF を Java で扱うための API として、Java によるセマンティック Web アプリケーション開発のためのフレームワークである Jena[47] を利用している。

実装言語	Java (JDK 1.5) Java Servlet (Apache Tomcat 5.5)
HTTP サーバ	Apache HTTP Server 2.1
使用ライブラリ	Jena 2.2
データベース	PostgreSQL 8.1
対応環境	Windows, Linux

表 2.1: Annphony の実装環境

2.6 Annphony と各システム間の関係

図 2.11 に Annphony と後述する音楽アノテーションシステム、アプリケーションとの関連を示す。アプリケーション開発者は収集するべきアノテーションの定義を行い、Annphony に登録する。既に必要なアノテーションの定義が別アプリケーションの開発者により登録されている場合は新たに定義を行う必要は無い。次章で述べる音楽アノテーションシステムの各アノテーションエディタは、自身に適用可能であるというアノテーションの付与された全てのアノテーション定義を Annphony から受け取る。アノテーションエディタは取得した定義群から、それぞれの定義がどのようなデータ型の属性を持つかを Annphony API を利用して解析し、さらに、ユーザが入力可能なアノテーションメニューを動的に構成する。メニューの動的な構成法はアノテーションエディタにゆだねられている。例えばアノテーションエディタは、HTML のフォームを自動生成したり、右クリックメニュー

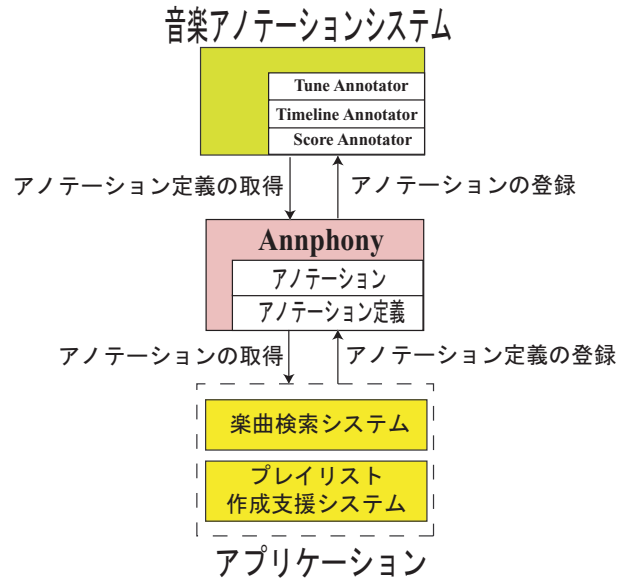


図 2.11: Annphony と各システム間の関係

を変更したりするなどの処理を行う。ユーザからのアノテーションの入力を音楽アノテーションシステムのサーバが受け取ると、Annphony にアノテーションを登録する。アプリケーションは、アノテーションエディタを用いて収集されたアノテーションを Annphony を通して取得し利用する。

Annphony においてアノテーションとその定義を管理することで、従来強い依存関係にあったアノテーション定義とアノテーションエディタ、またアノテーションとアプリケーションをそれぞれ独立させる。Annphony において管理されるアノテーション定義を参照し、動的にアノテーションエディタを構成することで、柔軟なアノテーションの取得が見込まれる。また Annphony において管理されるアノテーションはアプリケーションの種類を問わず利用可能であるため、類似するアノテーションを重複して収集する必要がなくなり、アノテーション作成のコストを抑えられる。

Annphony は音楽以外のコンテンツへのアノテーションも同時に管理することができる。音楽のアプリケーションのために収集されたアノテーションを、別の

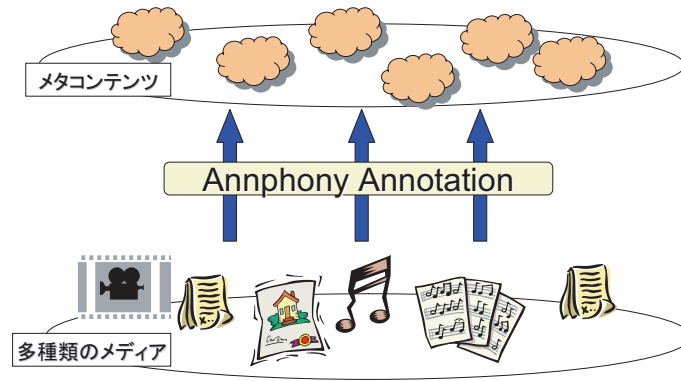


図 2.12: コンテンツからメタコンテンツへ

コンテンツのアプリケーションが利用するなど、コンテンツの種類を横断するアプリケーションの実現を支援する基盤であるといえる。

2.7 Annphony がもたらす効果

メタコンテンツの統合的処理 様々な種類のコンテンツに対して付与された、統一した形式による内部構造及び個人の解釈情報を利用することで、図2.12のようにそれぞれのコンテンツをメディアの種類によらないメタコンテンツとして捉えることが可能になる。今後はメタコンテンツの処理に関する研究が盛んに行われることが期待される。

具体的な応用例としては、例えばビデオのテロップに対するアノテーションに、テキスト処理のために定義されたアノテーション定義を利用するなど、異種コンテンツ間でアノテーションを共有し、互いに利用したり、音楽プレイリストを拡張し、音楽に限らない複合メディアのプレイリストを作成・配信したりするといった、従来困難であった複数のコンテンツの意味内容と個人嗜好を考慮した統合メディア処理が実現されるだろう。

本研究では、特に音楽コンテンツを対象としているが、今後の展望として、現在複数のコンテンツを横断したアプリケーションとして、ビデオと辞典を組み合

わせたシステムを構築中である。本システムについては7.3節において述べる。

オントロジ構築へのアプローチ アノテーションは、人間がコンテンツに対し詳しい意味記述を行うものである。そのため大量のアノテーションを用いてその解析を行うことにより、リソースの分類体系やその関係、推論のためのルールなどを定義するオントロジの構築を実現する可能性を秘めている。

トピックマップ[48]ではリソース全体についてのオントロジ構築を目指しているが、コンテンツの任意の箇所を分節化する ElementPointer は、より詳細なオントロジの構築に役立つと期待される。また図 2.12 に示されるように、アノテーションがコンテンツの種類に依存しない共通の形式で記述されるため、Web における様々なコンテンツにまたがる意味関係を記述されたアノテーションが増加することになるだろう。Annphony が様々なコンテンツ管理システムに組み込まれ、横断的なアノテーションが大量に収集されることで、それを解析し、様々なコンテンツにまたがったオントロジが構築されるだろう。

2.8 本章のまとめ

本章では、任意のデジタルコンテンツに対しコンテンツの構造と個人の解釈に関するアノテーションを作成・利用するアノテーション基盤である Annphony について、アノテーションとそのスキーマの形式と、それらを管理するための機能に関して述べた。

任意のコンテンツの内部を指し示す ElementPointer を提案し、Annphony に適用した。またコンテンツの種類に依存しないアノテーション形式である RDF を拡張することにより、任意のコンテンツの構造や解釈・関連性を統一的に扱うことができるようになった。また Annphony はアノテーション定義の管理を行うことで多様化を続けるコンテンツへのアノテーションを柔軟に追加することができる。Annphony により、コンテンツの内部を指し示す手法の一般化と、任意のコンテン

ツに対するアノテーションの枠組みが実現されたことで、図1.4で示した既存のアノテーション形式の壁を取り払い、横断的にコンテンツの意味内容を扱う環境が実現されたと言えるだろう。

今後はネットワーク上に Annphony が複数存在する分散環境における協調機能やアノテーションのアクセス・編集ポリシーについて検討すると共に、実際にアノテーションを付与した異種コンテンツによる応用を実現していく予定である。

第3章 音楽アノテーションシステム

本章では、音楽に関する多様な解釈を取得するためのシステムとして、音楽アノテーションシステムを提案する。まず音楽アノテーションを取得するために必要な条件を考察し、それに基づいて構築した音楽アノテーションシステムの機能を述べる。

3.1 音楽アノテーションの獲得

音楽は嗜好・状況への適応が強く求められるコンテンツであるため、計算機はそれを支援するためにコンテンツの意味内容に関するアノテーションを獲得する必要がある。

一方で、1.1.2 節でも述べたように、音楽コンテンツの意味内容の中には、自動認識することの困難な情報が多く存在する。例えば楽曲から受ける印象は、テンポや曲調、歌詞の内容、時代背景などを踏まえて判断する必要があり、一般に自動的に解析することが困難である。そのため音楽は特に人手によるアノテーション付与が有効であると考えられる。また同一のリソースに対する異なる解釈を管理することが必要となる。

現在音楽コンテンツを表現するための様々なメディア形式が提案されている。表 3.1 に主なメディア形式を挙げる。現在広く普及している MP3 等の音響メディア以外に、単純な文字列列挙によってメロディや和音を表現する MML や、楽曲データを電子楽器などとやり取りする規格である MIDI、音楽の詳細な楽譜構造を表現する形式である WEDELMUSIC XML や MusicXML が提案されている。MIDI

は図 3.1 に示すように、複数のチャンネルごとにイベントとデルタを記述する形式となる。イベントには、音高や音量など、音を発生させるための情報を、デルタには何秒間待機するかという情報を記述する。また MusicXML は図 3.2 に示す構造を持つ。まずタイトルや作曲者などの楽曲の基本情報を記述し、以降に楽器パートごとの楽譜情報を記述する。各楽器パートは小節ごとに区切られ、小節内では音符情報が列挙される。このように音楽を表現するメディアには、連続メディアと離散的な構造を持ったメディアといった多様な形式が存在する。それぞれのメディアは、最小オブジェクトの単位が音符であったり、サウンドサンプルであったりするなど、粒度も異なる。また今後も新しい形式が提案されることもあるだろう。そのため音楽という単一種類のコンテンツを扱う場合であっても、このような連続メディアと離散的な構造を持ったメディアへの対応や、粒度の異なるオブジェクトへの対応を行わなければならない。

形式	説明
MP3	音響信号
MML	音階と音長を列挙したテキスト
MIDI	電子楽器間における音楽情報伝達のための規格
MusicXML	楽譜構造を記述する XML
WEDELMUSIC XML	楽譜構造を記述する XML

表 3.1: デジタルコンテンツとしての音楽の主な記述形式

一般的にコンテンツは、ビデオにおけるシーンや音楽における音符・小節など離散的な論理的構造を持っており、コンテンツの意味内容に関するアノテーションを記述するためには、コンテンツの内部構造へのアノテーションが必要となる。しかしこれらの音楽メディアの持つ情報量や楽曲構造はそれぞれ異なる。それぞれアノテーションエディタは、利用する音楽メディアにあわせたインタフェースを提供し、楽曲の内部へのアノテーションを支援する必要があるだろう。例えば

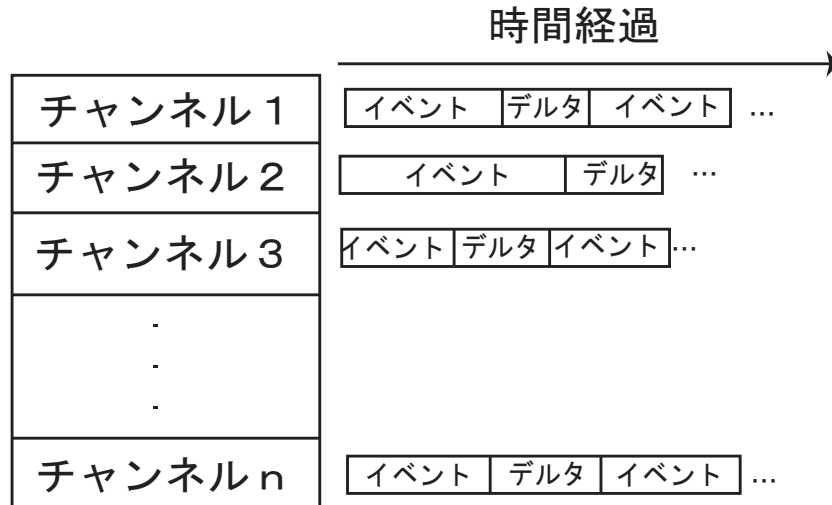


図 3.1: MIDI の内部構造

MusicXML は楽譜情報を保持しているため、アノテーションエディタはユーザに楽譜を提示することで、楽曲の詳細な内部に対するアノテーションが記述可能となる。

また、どのような種類のアノテーションを利用するかは、実現するアプリケーションによって異なるため、音楽アノテーションを収集するためのシステムはアノテーションの種類を固定してはならない。例えばジャンル情報を固定された 10 種類の中から選択するシステムを構築してしまうと、今後新たなジャンルが現れた際、そのジャンルを入力することができない。

3.2 音楽アノテーションシステムの機能

前節からわかるように、音楽アノテーションを獲得するためのシステムは、多様な音楽メディアに対応し、人手によって様々な種類のアノテーションを付与することが必要である。これらの点を踏まえて、オンライン上のユーザから多くの楽曲に関するアノテーションを収集するための音楽アノテーションシステムを構築した。以下に本システムの機能を述べる。

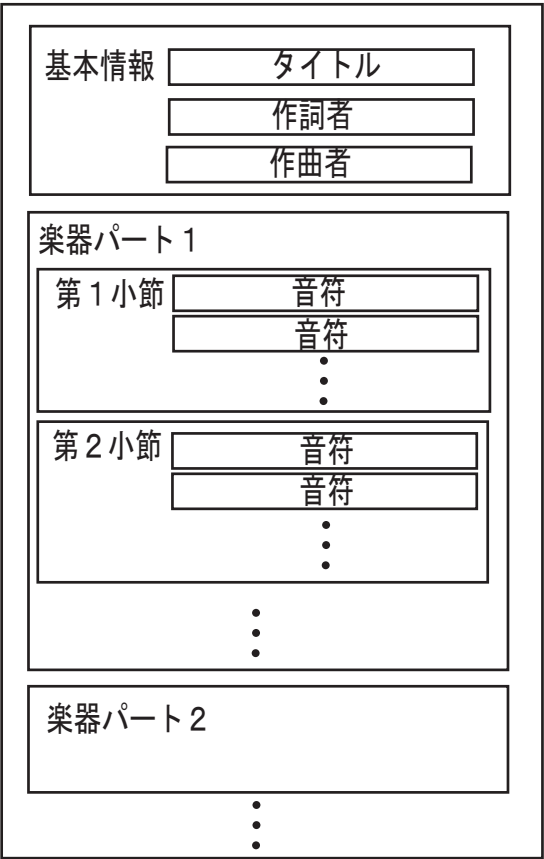


図 3.2: MusicXML の内部構造

3.2.1 Annphony の利用

前章で述べた Annphony の有用性を向上させることと、音楽を具体的な題材として、一般のコンテンツに対するアノテーション獲得・管理するための指針を示すために、アノテーション管理のための基盤として Annphony を利用した。また多様なアノテーションを収集するために、Annphony で管理されるアノテーション定義に基づくアノテーションを収集するシステムとなっている。アノテーション定義とアノテーションシステムを分離し、Annphony において管理されるアノテーション定義に基づいてアノテーションのメニューを動的に構成するため、収集するアノテーションの種類を限定せず、幅広いアプリケーションで利用可能なアノテーション収集を支援する。

3.2.2 Web ブラウザを通したアノテーション

本システムではメタ情報を幅広いユーザから集め、その情報を様々に応用していくことを目的としている。しかしアノテーション獲得のためのシステムがスタンドアロンのシステムであった場合、ソフトウェアのインストール作業やサーバへの接続作業などに手間がかかるためユーザがそのシステムを利用する際の足かせとなり、結果多くの情報を集めることが困難である。

そこで、ネットワークにつながっている人なら煩雑な手続き無しでアノテーションが行えるように、Web ブラウザベースシステムの形態をとった。多数のユーザから少量のライトウェイト・アノテーションを収集することにより、一人一人にかかる人的コストを低く抑える。

多人数のユーザを対象とする場合、アノテーションの信頼性を考慮する必要がある。そのため前準備として、ユーザはあらかじめユーザ登録を行い、本システムを利用する際にはベーシック認証を通してログインする。誰がどのアノテーションを作成したかを全て記述するためである。

3.2.3 3種類のアノテーションエディタ

様々な種類のアノテーションを取得するために、対象となるリソースの粒度に応じたエディタを構築する必要がある。楽曲に対するアノテーション付与の代表例として、Cddb [25] や MusicBrainz [26] が挙げられる。これらのように、書誌情報などの基本情報を楽曲自体に対するアノテーションが必要であり、一方でコンテンツの意味内容に基づく応用のためには、コンテンツの内部構造に関するアノテーションが必要である。

そこで我々は、楽曲の形態や取得するアノテーションの種類に応じて様々な粒度のアノテーション取得を支援するため、1. Tune Annotator, 2. Timeline Annotator, 3. Score Annotator の3種類のエディタを構築した。図 3.3 は音楽アノテーションシステムのトップページである。システムは登録されている楽曲を列挙し、さらにそれぞれの楽曲のメディア形式から、適用可能なアノテーションエディタを識別してそれぞれのアノテーションエディタへのリンクを生成する。それぞれのアノテーションエディタを利用する前提と、付与されるアノテーションの粒度を表 3.2 に示す。ユーザは本ページを通してアノテーションを行う楽曲とエディタを選択する。アノテーション対象となる楽曲は随時追加可能である。

以下にそれぞれのアノテーションエディタの詳細を述べる。

エディタ名	前提	粒度
Tune Annotator	・ URI を持つ ・ 任意のメディア形式	楽曲そのもの
Timeline Annotator	・ URI を持つ ・ 連続メディア	楽曲の時間区分
Score Annotator	・ URI を持つ ・ MusicXML	楽譜中に出現するオブジェクト

表 3.2: 各エディタを利用する前提とアノテーション対象となる要素の粒度

タイトル	アーティスト	アルバム	歌詞	Media Type	Tune Annotator	Timeline Annotator	Score Annotator
あかといんぼ			☐	MOML			
ふるさと			☐	MOML			
春が来た			☐	MOML			
きらきら星			☐	MOML			
こいのぼり			☐	MOML			
チューリップ			☐	MOML			
うみ			☐	MOML			
ぞうさん			☐	MOML			
ちょうちょう	浅川浩美	RWC-MDB-R-2001	☐	MP3			
むすんでひらいて	浅川浩美	RWC-MDB-R-2001	☐	MP3			
赤い靴	浅川浩美	RWC-MDB-R-2001	☐	MP3			
七つの子	浅川浩美	RWC-MDB-R-2001	☐	MP3			
桃太郎	浅川浩美	RWC-MDB-R-2001	☐	MP3			
ジャングルベル	Jeff Manning	RWC-MDB-R-2001	☐	MP3			

- Pop -							
---------	--	--	--	--	--	--	--

タイトル	アーティスト	アルバム	歌詞	Media Type	Tune Annotator	Timeline Annotator	Score Annotator
How Deep Is Your Love	Donna Burke	RWC-MDB-P-2001	☐	MP3			
Doing That Thing	Jeff Manning	RWC-MDB-P-2001	☐	MP3			
Waiting for the moment	Jeff Manning	RWC-MDB-P-2001	☐	MP3			
Angel Baby	Betty	RWC-MDB-P-2001	☐	MP3			
I think of you	Jeff Manning	RWC-MDB-P-2001	☐	MP3			

図 3.3: 音楽アノテーションシステムのトップページ

Tune Annotator

Tune Annotator は書誌情報など楽曲自体に関する情報を収集するためのエディタである。タイトルやアーティスト情報などの、楽曲の基本情報の記述にとどまらず、楽曲の推薦度やアンケートなどの柔軟なアノテーション取得を支援する。図 3.4 に Tune Annotator の画面例を示す。画面の左側には、タイトルやアーティストなどの楽曲の基本情報、再生用のプレイヤーが配置され、右側にはアノテーションメニューが HTML のフォームとして表示され、ユーザはこのフォームを通してアノテーションを記述する。

ユーザは記述するアノテーションの項目を動的に変更することができる。図 3.5 の左側はタイトルやアーティストなどの楽曲の基本情報を入力するためのアノテーションメニューが表示されているが、ユーザが「アノテーションの種類」の部分のプルダウンメニューから、記述したいアノテーションの種類を選択すると、Tune Annotator は図 3.5 の左側のように自動的に該当するアノテーションの属性のフォームに変更する。

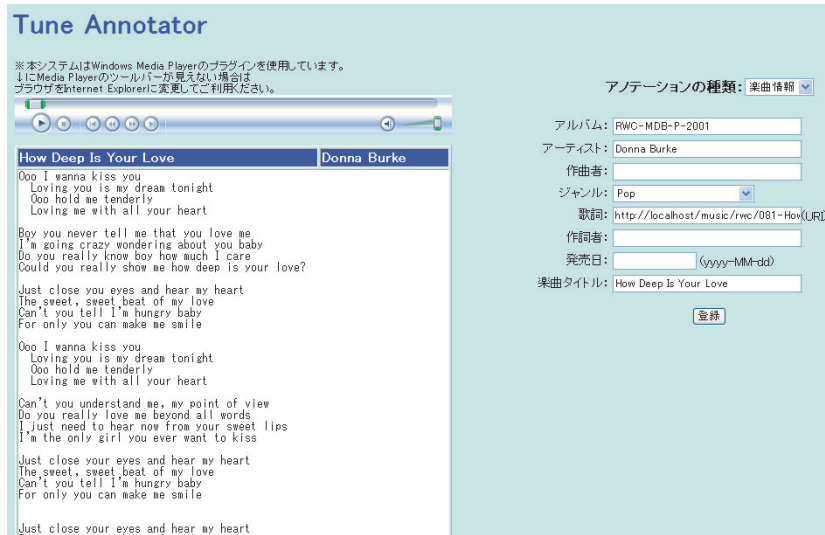


図 3.4: Tune Annotator の画面例

本エディタを利用するための条件は、楽曲が URI で表現されることである。つまり、Web 上に存在する全てのメディア形式の楽曲に対して Tune Annotator を利用したアノテーション収集が可能である。

Timeline Annotator

楽曲の最も単純なセグメンテーション手法は時間による区分であり、エンドユーザにとって直感的に理解可能である。そこで、MP3 などの連続メディアに対する時間的なセグメンテーションと、セグメントに対しアノテーションを付与するエディタである Timeline Annotator を構築した。図 3.6 に Timeline Annotator の画面例を示す。上部の音楽プレイヤーで楽曲を鑑賞しながら、[FROM][TO] ボタンにより開始、終了時間を指定することで時間的なセグメントを切り出す。さらに [ANNOTATE] ボタンを押すと、図 3.7 に示すアノテーションフォームが現れる。ユーザはアノテーションの種類を選択すると、アノテーションの属性の入力フォームを自動的に構成する。ユーザが各項目に入力することでアノテーションを行う。図 3.7 に例示した音符情報は、3.3 節において連続メディアの構造化を行うためのアノテシ

Figure 3.5 illustrates the dynamic change of the annotation menu. The left panel, titled "アノテーションの種類: 楽曲情報", contains fields for Album (RWC-MDB-P-2001), Artist (Donna Burke), Composer, Genre (Pop), Lyrics (http://localhost/music/rwc/081-Hov), Lyricist, Release Date, and Song Title (How Deep Is Your Love). The right panel, titled "アノテーションの種類: 主観情報", contains fields for Impression (悲しい), Rating (★★★★★), and Comment (RWC 音楽DBの楽曲です), along with a "登録" button. An arrow points from the left panel to the right panel.

図 3.5: アノテーションメニューの動的な変更

ンの一つである。

付与されたアノテーションは、図 3.6 の下部にリンクとして表示され、リンクにマウスポインタを当てることで内容の確認を行う。

楽曲の時間区分の表現形式として、前章で述べた `ElementPointer` を利用している。セグメントの開始時間、終了時間という二つの属性を持つ `ElementPointer` を定義することで、楽曲の時間的な区間を URI により表現することが可能になる。`Timeline Annotator` はこの連続メディアの時間区分を表す `ElementPointer` URI を対象としたアノテーションを記述することで、楽曲の内部へのアノテーションを実現している。

全ての小節にアノテーションを付与する場合など、開始時間、終了時間の指定を一定の間隔で行うために、本エディタは一括で時間的セグメンテーションを行う機能を備えている。図 3.8 に表れるセグメント幅と開始時間の秒数を指定して、「自動セグメント切り出し」ボタンを押すことによって、画面下部に一定間隔で切り出されたセグメントが表示される。ユーザはそれぞれのセグメントをクリックし、そのセグメントへのアノテーションを記述する。

本エディタを利用するためには、1. 楽曲が URI を持ち、2. 音楽プレイヤーで再生可能な連続メディアである必要がある。広く普及している MP3 や MIDI などの

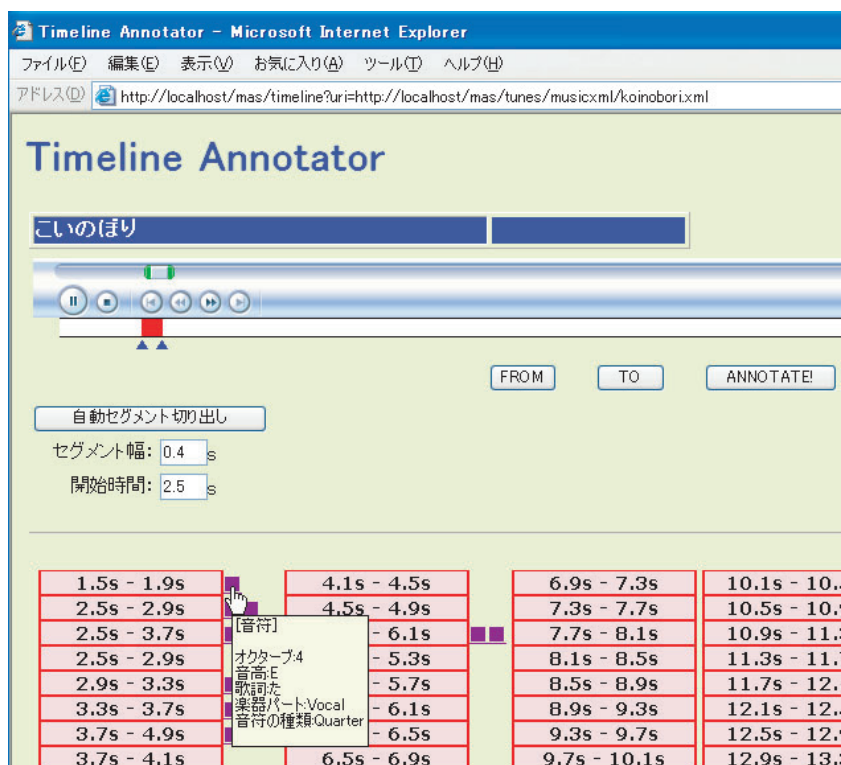


図 3.6: Timeline Annotator の画面例

Timeline Annotator Form - Microsoft Internet Expl...

アノテーションの種類: 音符

楽器パート: Vocal

歌詞: た

音高: C

オクターブ: 4

音符の種類: Quarter

登録

開じる

ページが表示されました

イントラネット

図 3.7: Timeline Annotator のアノテーションメニュー例

自動セグメント切り出し

セグメント幅: 4 s

開始時間: 1.3 s

1.3s - 5.3s	81.3s - 85.3s	161.3s - 165.3s
5.3s - 9.3s	85.3s - 89.3s	165.3s - 169.3s
9.3s - 13.3s	89.3s - 93.3s	169.3s - 173.3s
13.3s - 17.3s	93.3s - 97.3s	173.3s - 177.3s
17.3s - 21.3s	97.3s - 101.3s	177.3s - 181.3s
21.3s - 25.3s	101.3s - 105.3s	181.3s - 185.3s
25.3s - 29.3s	105.3s - 109.3s	185.3s - 189.3s
29.3s - 33.3s	109.3s - 113.3s	189.3s - 193.3s
33.3s - 37.3s	113.3s - 117.3s	193.3s - 197.3s

図 3.8: 時間的セグメントの一括切り出し

メディア形式に対応するため、実用性の高いエディタであるといえる。

Score Annotator

Score Annotator は、楽譜に現れる要素の集合に対するアノテーションを付与するためのエディタである。アノテーションの付与が可能な楽譜の要素は、各楽器パートの音符、歌詞、タイトル、作詞者、作曲者、発想記号である。ユーザには図 3.9 のように楽譜とアノテーションを可視化したオブジェクトが重ねて表示される。ユーザがブラウザに表示された楽譜から、マウスのドラッグにより矩形の範囲指定を行うと、矩形内に存在する要素または要素集合が選択され、赤色で強調表示される。複数の矩形範囲を同時に選択することも可能であるため、楽譜上で遠い位置に存在する要素に対して同時にアノテーションを付与することが可能である。次に図 3.10 のように、矩形上でマウスを右クリックし、出現するメニューからアノテーションの種類を決定し、その後具体的な情報を記述する。

感想や解説などのアノテーションは、人間が記述する情報であるために、それが不十分であったり誤っていたりする場合があるだろう。2.3 節で述べたように、Annphony で管理されるアノテーションには、全て識別子が自動的に付与されるため、アノテーションに対するアノテーションを付与可能である。しかしアノテーションエディタにもアノテーションへのアノテーションを行う仕組みが必要である。そのため本エディタは、アノテーションを補足、修正するために、画面に出現するアノテーションオブジェクトに対してアノテーションを関連付ける機能を持つ。

アノテーションを楽譜上に重ね合わせる場合、アノテーションの量が増えると楽譜が見づらくなってしまうという問題がある。そこで、本エディタでは表示するアノテーションを、アノテータやアノテーションの種類からフィルタリングする機能を持つ。図 3.11 は本エディタの左側に表れるアノテーション表示のメニューである。このメニューから、楽譜上に表示するアノテーションをマウスクリックにより選択すると、表示されるアノテーションをフィルタリングする。図 3.12 は、

社 社 社 社 社 社 社 社 社 社

コイノボリ

端午の節句ですよ

作詞
作曲

なぜカタナ

違

厳格

江戸時代、武士は太刀・太鼓・武者人形を飾り、それに対抗して庶民はこいのぼりを立てた

コード
梶 克彦
kaji@nagao.nui.e.nagoya-u.ac.jp

Voice

C F C

や ね よ り 元 気 た か い こ

元 気 C F C 穏 や か

お お き い ま ご い は お

お母さんけどこいった?

図 3.9: Score Annotator の画面例

アノテーション

play

Search(&F)...

Next Search(&A)

Copy SVG(&C)

Source(&V)

Save As...

Help(&H)

About Adobe SVG Viewer(&B)...

意見・感想
解説

印象

構成

コード

陽気
刺激的
厳格
元気
悲しい
夢想的
穏やか
上品

こ い は

図 3.10: 選択したオブジェクトへのアノテーション付与

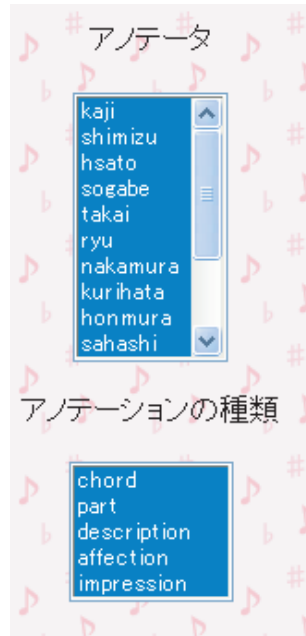


図 3.11: アノテーション表示メニュー部

kaji, shimizu というアノテータが付与した、和音情報と楽曲の大まかな構造に関するアノテーションのみを表示した例である。

本エディタを利用するためには、1. URIを持ち、2. メディア形式が MusicXML [23] である必要がある。MusicXML は楽譜を形成するために十分な情報が記述されており、楽曲の構造を表現する形式であるといえる。音楽的に構造化されたメディアに対するアノテーションを行うことにより、楽曲そのものや時間区分といった大まかな対象ではなく、楽譜という音楽の論理構造に基づいた詳細な対象への言及を実現している。音楽的に意味のある箇所へのアノテーションにより、楽曲の意味内容に基づく、より高度な音楽アプリケーションの実現が期待される。

Score Annotator

Java Sound Synthesizer

play stop auto

アノテータ

kaji
shimizu
hsato
sogabe
takai
ryu
nakamura
kurihata
honmura
sahashi

アノテーションの種類

chord
part
description
affection
impression

[トップページへ](#)
[アノテーション検索](#)
[マニュアル](#)

ふるさと

作詞: 高野辰之
作曲: 岡野貞一

Aメロ

Voice

G D7 G D7

う さ ぎ お い し か の や
い こ さ か ん げ い ま す ち た は
こ ぶ な つ り し か と の か
つ い つ の な ひ し に か え か
ゆ め は い か ま も め ぐ
あ や め ま に あ ぜ お に っ け
Cメロ

D G D G Am D7

図 3.12: アノテーションのフィルタリング

3.3 連続メディアの構造化支援

前節で述べたように、音楽アノテーションシステムは楽曲の意味内容に基づく応用のため、Score Annotator によるアノテーション付与を実現しているが、本エディタは MusicXML という楽曲の論理構造を表す形式でなければ利用できないという大きな制約がある。現在 Web 上に存在するメディアの多くは MP3 などの連続メディアであり、音楽の意味内容を考慮したアプリケーション実現のために、これらのメディアを構造化する必要がある。そこで連続メディアに対する構造化の支援を行う機能を構築した。

楽曲の構造化を行うために、タイトルやアーティストなどの基本情報に関するアノテーション定義に加え、楽譜生成のために十分な情報をアノテーションとして記述するための、楽器パートや発音列 (音高, 音長, 音符のタイプ), 歌詞などのアノテーション定義を用意した。また、それぞれのアノテーション定義を Tune Annotator と Timeline Annotator に適用した。ユーザは楽曲の基本情報を Tune Annotator を用いて記述する。次に Timeline Annotator を用いて、構造化を行う連続メディアを時間的にセグメントし、各セグメントの音符情報や小節の情報を入力する。一つの音符や歌詞の一単語など、一つ一つのアノテーションは楽曲の断片情報を表現しているに過ぎないが、ある楽曲に付与されたこれらのアノテーションの集合から、楽曲全体の楽譜を生成することができる。

次に、音楽アノテーションシステムはこれらのアノテーション集合を解析し、MusicXML とそれに対するアノテーションを以下のように生成する。アノテーション集合から読み取られる、メロディや小節の情報を元に、その楽曲を表す MusicXML を生成し、同時に MusicXML に現れる音符や小節に関する各ノードと、それに対応する連続メディアの時間区分を対応付けるアノテーションを生成する。

連続メディアを構造化し、MusicXML を生成することにより、Score Annotator への適用が可能になる。本機能により、連続メディアを用いた論理構造に基づく応用を実現することができる。

連続メディアの構造化には音楽的な知識が必要なため、専門性が高い。そのため、連続メディアの構造化に関するアノテーションは Web 上から幅広く収集することが困難であると予想される。本システムでは人間が作成するアノテーションに加えて機械が自動的に生成する情報も扱うことができる。

今後はビートトラッキング [49] などの採譜に関する自動処理の結果を取り込み、その結果を人手により修正することで容易に構造化を行うことができるよう機能拡張する予定である。

3.4 システム構成

図 3.13 に音楽アノテーションシステムのシステム構成を示す。アノテーション、アノテーション定義と共に、ユーザ情報を Annphony により管理する。音楽アノテーションシステムは Web システムとして構築されており、サーバはユーザからのアノテーション対象の楽曲とエディタの種類の要求に応じて、Tune Annotator, Timeline Annotator, Score Annotator のいずれかのエディタを提示する。その際システムは対象となる楽曲をアノテーションエディタに取り込み、同時にアノテーションメニューは Annphony のアノテーション定義に基づき生成する。ユーザはそれぞれのアノテーションエディタを通してアノテーションを記述し投稿すると、それを受け取った Web サーバはユーザ、アノテーション対象となる楽曲の部分、アノテーションの種類とその内容を基に Annphony で管理する形式のアノテーションを生成し、Annphony にアノテーションを格納する。

図 3.13 では、同一の楽曲に対するアノテーションを Tune Annotator と Timeline Annotator という二つのエディタにより取得している。このように、同一の楽曲を対象とし、その楽曲自体に対するアノテーションや楽曲内部のアノテーションなど粒度の異なるアノテーションを取得することが可能であるため、より多角的な楽曲の意味内容の把握が可能となる。

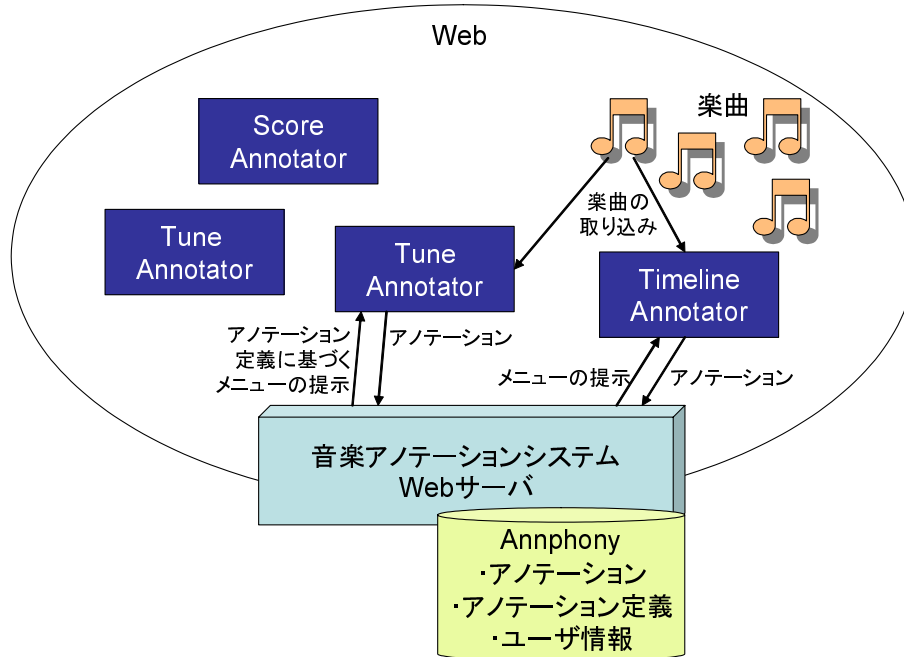


図 3.13: 音楽アノテーションシステムの構成

3.5 多様な解釈の顕在化に関する実験

ユーザや楽曲の特性によって解釈に多様性が存在し、個人に適応するアプリケーションを実現するために解釈の多様性と個人性を扱う必要があること、またそのような解釈の多様性を音楽アノテーションシステムによって顕在化させることが可能であることを実証することを目的とした。

3.5.1 実験方法

「ハイライト」の認識に関する実験を行った。楽曲のどの部分を、どのような理由でハイライトと認識するかが解釈に相当する。本実験では、ハイライトの定義を「楽曲の最も印象的な部分」とした。以下に実験の詳細を述べる。またハイライトと認識する理由の一つに「サビである」という理由が予想されるが、本実験ではサビの定義を「楽曲の中で繰り返しの最も多い区間」としている。

実験に利用する楽曲は、過去5年におけるオリコン年間売り上げの上位10曲ずつ、計50曲であり、被験者は20代の男女10名である。それぞれの被験者は、時間区分に対するアノテーションエディタである Timeline Annotator を用い、各楽曲について、ハイライトであると認識した時間区分を切り出し、同時にその部分をハイライトであると認識した理由を自然言語により記述する。ハイライト区間は、各楽曲につき一箇所であり、切り出しの長さに制限を設けていない。50曲の楽曲に対して500箇所のハイライト区間と、645個の理由を収集した。ハイライトであると認識した理由には複数の記述を許しているため、理由の合計は500個を超えている。

3.5.2 実験の結果と考察

収集されたアノテーションを基に、ハイライトと認識した理由を手で分類したところ、以下の9種類に分類された。

- サビである
- 聴いたことがある
- 歌詞が良い
- 歌い方が特徴的
- 盛り上がるから
- メロディが良い
- リズム・テンポが良い
- 個人的な思い出
- 演奏が特徴的

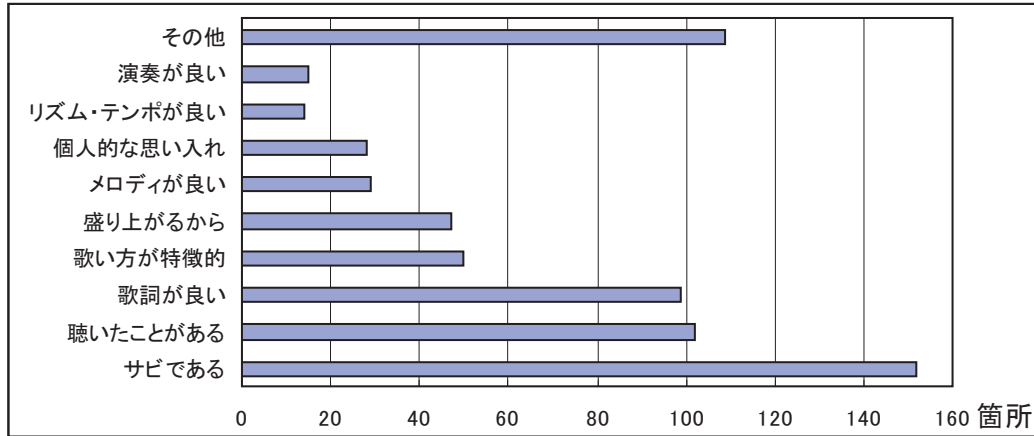


図 3.14: 理由ごとのハイライト区間数

図 3.14 はハイライトと認識した理由ごとのハイライト区間の数を表している。従来からハイライトであることが多いとされる「サビである」という理由以外にも、「聴いたことがある」「歌詞が特徴的」「歌い方が特徴的」という理由に基づく認識が多いことが分かる。

次にユーザごとのハイライト認識理由の個人性について考察する。図 3.15 にユーザ A が、図 3.16 にユーザ B がハイライトだと認識した理由の割合を示す。ユーザ A は「サビである」「聴いたことがある」という理由でハイライトであると認識された箇所が多く、一方ユーザ B は「歌詞が良い」、「歌い方が特徴的」という理由でハイライトであると認識する可能性が高いことが読み取れる。このようにユーザごとのアノテーションを解析することにより、それぞれのユーザの個人性を浮き上がらせることができることを確認した。

次に切り出されたハイライト区間についての考察を行う。図 3.17 に、ある楽曲のサビ区間と理由別のハイライトを可視化した図を示す。楽曲中で、多くの被験者がハイライトだと認識した区間ほど濃い色で表されている。一段目には理由に関わらずハイライトだと認識された区間を表示し、二段目にはサビであるという理由で、三段目には歌詞が特徴的であるという理由で認識されたハイライト区間

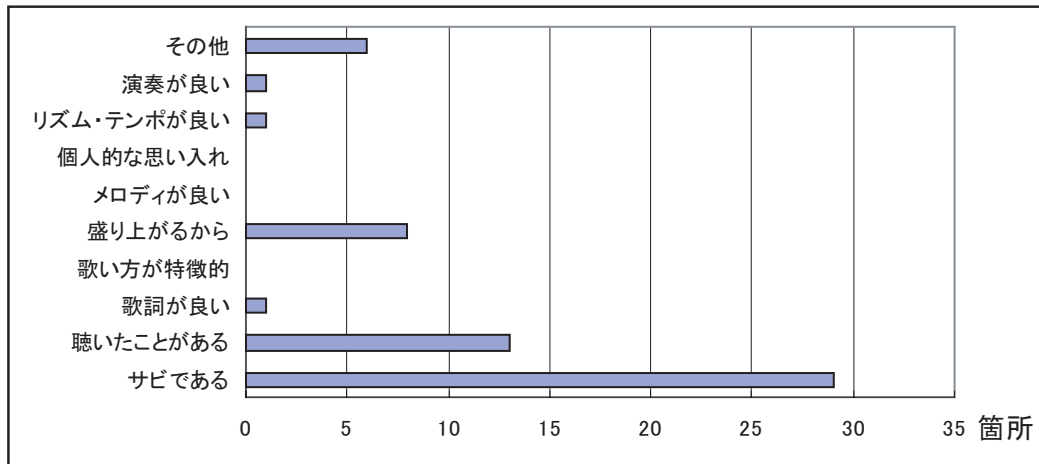


図 3.15: ユーザ A における理由ごとのハイライト区間数

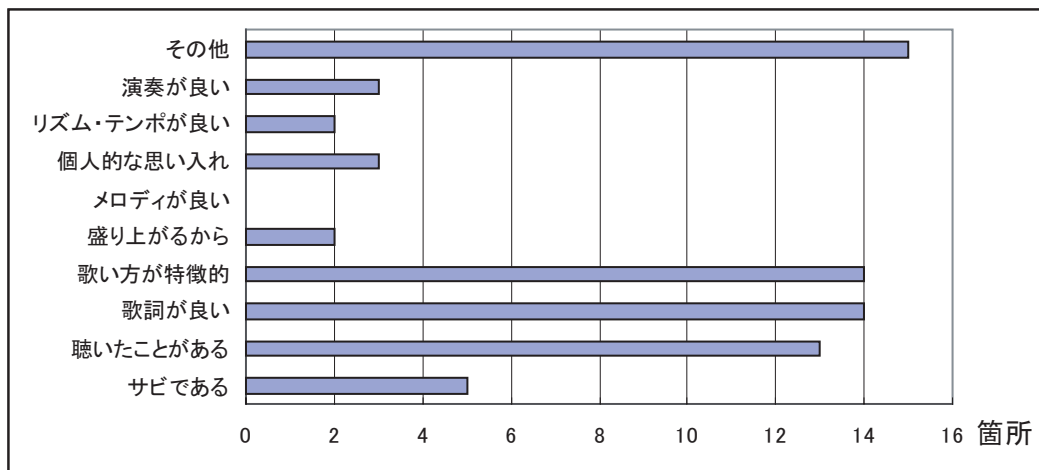


図 3.16: ユーザ B における理由ごとのハイライト区間数

宇多田ヒカル-traveling

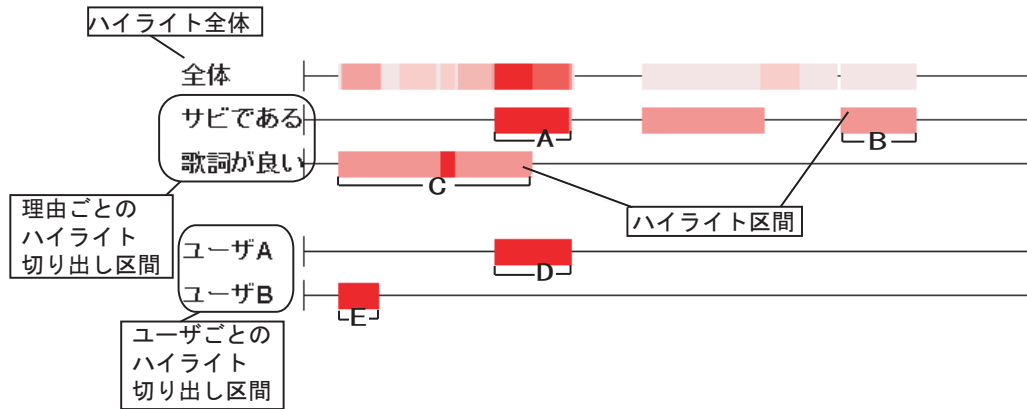


図 3.17: ハイライトと認識された区間の分布

を表示している。一段目から、ハイライトであると認識された区間は楽曲の多くの区間に渡っていることが分かる。また二段目と三段目には重複する区間が少なく、大きな違いがあることが分かる。このことからハイライトと認識する理由によって切り出される区間の多様性を、アノテーションの分類という統計的な処理によって浮き上がらせることができたといえる。

楽曲の解釈に関する多様性や個人性を利用することにより、それぞれのユーザーに適応するアプリケーションの実現が可能になると考えられる。例えば楽曲のハイライトを試聴支援に用いる場合、サビを高い確率でハイライトと認識するユーザー A にとっては、図 3.17 の区間 A, B を中心に試聴させることが効果的であり、歌詞に基づきハイライトを認識するユーザー B にとっては区間 C を中心に試聴を促すことが効果的であると考えられる。実際に図 3.17 にはユーザー A が本実験で切り出したハイライト区間 D は、ユーザー A にとって試聴支援に効果的であると思われる区間 A, B に含まれており、またユーザー B のハイライト区間 E も同様に、試聴支援に効果的であると思われる区間 C に含まれている。

本実験により、楽曲の内部構造に関する解釈の多様性と各ユーザの個人性を、本システムで収集されるアノテーションを解析することによって浮き上がらせることが可能であること、またこうして導き出された個人性をアプリケーションに適用可能であることが確認された。

3.6 本章のまとめ

本章では、音楽を題材として、Web 上のユーザから様々な解釈を獲得する音楽アノテーションシステムを構築した。本システムでは、多様な種類の音楽メディアに対応するために連続メディアの構造化をサポートし、楽曲のメディア形式と取得するアノテーションの種類に関連して、1: 書誌情報などの楽曲自体へのアノテーションエディタ (Tune Annotator), 2: 連続メディアに対するアノテーションエディタ (Timeline Annotator), 3: 楽曲の内部構造に対するアノテーションエディタ (Score Annotator) の3種類のエディタを備え、様々な粒度のアノテーション収集を支援する。またアノテーション定義とアノテーションエディタを独立させることにより、様々なアプリケーションに適用可能なアノテーションを柔軟に収集することができる。

また、本システムはMP3等の音響メディアに対しても、Tune Annotator と Timeline Annotator を用いてこれらのメディアの構造化を支援し、音響メディアに対応する MusicXML を生成する機能を備える。MusicXML を生成することで、音響メディアに対しても Score Annotator による楽譜による音符や歌詞の一部など楽曲の詳細な要素へのアノテーション付与が可能になる。

本システムを用いることで、楽曲の内部構造に関する解釈の多様性と各ユーザの個人性を、本システムで収集されるアノテーションを解析することによって浮き上がらせることが可能であることを実験によって確認した。

今後の課題としては、楽曲に対する多様な解釈の傾向を分析し、他者の解釈への依存度や解釈の変化の時間的推移などの特徴を明らかにし、多様な解釈を総括

する解釈を導くこと，また音楽以外のコンテンツに対しても複数の解釈を収集し，コンテンツの種類を横断するアプリケーションを構築することなどが挙げられる．

第4章 楽曲の解釈・構造を用いたアプリケーション

音楽アノテーションシステムにより、様々なアプリケーションに適用可能な音楽の意味内容に関するアノテーションを Web 上から収集することが容易になった。楽曲に対する多様な解釈を用いることで、ユーザの個人性に基づくアプリケーションを実現可能であることを確認するために、楽曲検索システムと楽曲再構成システムを構築した。以下に詳細を述べる。

4.1 検索・再構成システムのためのアノテーション定義

楽曲の論理構造に基づく検索のために、Score Annotator を用いて楽譜に出現する要素集合に対するアノテーションを収集した。アノテーションの定義として、以下の5種類を設定した。

- 意見・感想
- 解説
- 印象
- 曲の構成
- コード

解説や、意見・感想は、楽曲をキーワードで検索する際に有効であると考えられる情報である。そこでこれらの情報を自然言語による記述により収集する。ま

た、ユーザによる解釈のばらつきが多くみられ、個人の感性に適応する楽曲検索に利用可能であると考えられる情報として、楽曲を鑑賞した際に受ける印象が挙げられる。そこで、楽曲の各部分に対し、ユーザが楽曲を鑑賞して受けた印象を Hevner [50] が提唱した8種類の印象語(陽気, 刺激的, 厳格, 元気, 悲しい, 夢想的, 穏やか, 上品)から選択し、収集することにした。また、音楽の構造に基づく検索に有効であると考えられる、楽曲の構成の情報を収集する。楽曲の構成は、ポップスなどに見られるイントロやサビといった、その曲の大まかな構成を記述するためのアノテーションであり、試聴支援[51]などに利用される価値の高い情報である。コードは楽曲の各セグメントがどのような和音構成であるかを、基音・サフィックス・ベース音の組により記述する。コードは、同一の楽曲であっても出版社によって異なる箇所があることから、複数の解釈が可能な情報の一つと言えるだろう。

楽曲の論理構造に基づく応用を実現するために、ここでは音楽メディアの形式が MusicXML である楽曲に限定し、アノテーションを収集した。

4.2 楽曲検索システム

楽曲の構造や多様な解釈のアノテーションを利用することで、個人の感性に合わせた楽曲の内部検索が可能になる。そこで我々は、サビやイントロなどの楽曲の論理構造に基づき、ユーザの感性に合わせて楽曲の印象検索を行うシステムを構築した。

4.2.1 検索システムのインタフェース

ユーザが本検索システムにログインすると、図 4.1 のように検索フォームが提示される。検索フォームの「アノテーションを用いた検索」では、左側のリストボックスから楽曲の構造を、右側のリストボックスからキーワードの記述、印象や楽



図 4.1: 楽曲検索システムの検索フォーム

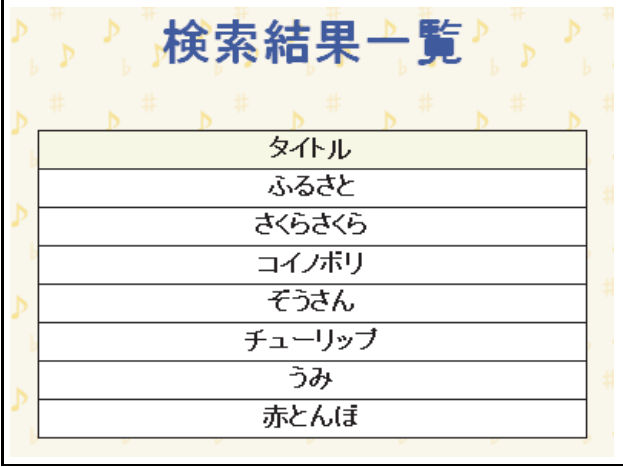
曲構造、コードを指定することにより、ユーザの検索要求がシステムに送信される。検索結果は図 4.2 のように、検索ランクが上位の楽曲から順に列挙される。

4.2.2 キーワード検索

タイトル、作詞者、作曲者、印象、意見・感想、解説の情報に関してキーワード検索を行うことができる。楽曲の基本情報であるタイトルや作詞・作曲者情報からの検索以外に、Web 上のユーザが付与したアノテーションに基づく検索が実現されている。楽曲の基本情報以外の、楽曲の鑑賞を通じて得られた感想などによる検索は、ユーザが目的の楽曲にたどり着く支援となると考えられる。

4.2.3 コード進行に基づく検索

楽曲の論理構造に基づく検索の例として、コード進行検索を実装した。本検索機能は作曲の際に参考となる楽曲を参照したり、ある楽曲のコード進行と類似す



タイトル
ふるさと
さくらさくら
コイノボリ
ぞうさん
チューリップ
うみ
赤とんぼ

図 4.2: 楽曲検索システムでの検索結果

る楽曲を検出して楽曲同士のミキシングを行うなど、幅広いアプリケーションでの利用が期待される。

検索要求としてコード進行を入力すると、システムはまず各コードの基音とベース音を相対値に変換する。例えば「C G/B Dm C」というコード進行の場合は「0,-5/-1,2m,0」となる。次にそれぞれの楽曲のコード進行に対して DP マッチングを行い、検索要求との距離を計算し、距離の近い順に正答の候補とする。検索ランクは DP マッチングの距離の逆数を正規化した値を利用しているため、コード進行が完全一致する楽曲だけでなく、類似するコード進行が出現する楽曲が検索結果として提示される。

4.2.4 個人の感性に適応する検索

楽曲に対する多様な解釈を利用し、個人の感性に適応する検索機能として、印象に基づく楽曲検索機能を構築した。印象情報はユーザ間の競合がしばしば見られる情報であるが、類似する感性を持つユーザ同士は、同一楽曲やセグメントから類似する印象を受ける。そこで本検索システムでは、検索するユーザと、ある閾値以上感性の類似するユーザが付与した印象アノテーションのみを利用し、楽

曲検索を行う。それぞれのユーザがどのような感性であるかというユーザモデルとユーザ間の類似度の閾値は、5章において述べるプレイリスト作成支援システムのものをを用いた。

一般的な楽曲検索の場合、検索結果のランクは検索要求との適合度のみによってランキングされるが、本検索システムではアノテータに合わせたアノテーションのフィルタリングを行うことで、例えばある二つの楽曲に、等しい数の「悲しい」という印象が付与されていた場合、嗜好の類似したユーザによるアノテーションが多い楽曲の方が上位にランクされる。ユーザの感性に合わせた検索結果のランキングを行うことで、それぞれのユーザが要求する楽曲を適切に提示する。

4.2.5 楽曲の構成に基づく絞込み検索

サビやイントロなどの楽曲の構成は、エンドユーザが容易に認識可能な楽曲の内部構造であるといえる。そこで本検索システムでは、多くのアノテーション収集を期待される楽曲の構成情報を用いて、絞込み検索を行う機能を実装した。

「サビが悲しい曲」という検索要求を受け取った場合、「サビ」が絞込みを行う楽曲の構成であり、検索対象は「悲しい」という印象である。まず、楽曲の構成以外の検索要求である「悲しい」について、前述の印象に基づく検索を行う。この時の検索結果のランクを計算する。この時の検索結果のランクを priorrank とする。次に、絞込む楽曲の構成「サビ」の部分に対する「悲しい」などの含有率 contains を計算し、含有率に基づき絞込み検索のランクを決定する。

「サビ」のアノテーションが付与されている音符、休符などの音楽オブジェクトが $o_i, o_j, \dots$ の要素であり、その要素数は k であるとする。また、「悲しい」というアノテーションが付与されている音楽オブジェクトは、 o_l から o_m までで、その要素数は n であるとし、これら二つのアノテーションが共通して付与されているオブジェクトの要素数は p であるとする、含有率 contains は以下の式で求める。

$$\text{contains} = \max\left(\frac{p}{k}, \frac{p}{n}\right)$$

「サビ」の部分に多く「悲しい」の要素が含まれている場合、または「悲しい」と関連付けられた要素が多く「サビ」に含まれている場合に contains が高くなる。最終的な絞り込みの検索ランクは以下の式で表される。

$$\text{rank} = \text{contains} \times \text{priorrank}$$

本機能により、楽曲の論理的なセグメントの検索が可能となり、引用・要約など様々なアプリケーションにおいてそのセグメントを利用可能になる。

本節で述べた個人適応された楽曲の内部検索は、楽曲要約や編曲など様々な音楽アプリケーションにおいて、利用する楽曲やその部分を選出する導入部として利用可能であると考えられる。

4.3 楽曲再構成システム

本研究で実装する再構成システムは、MusicXML が元から持っている歌詞・楽器パートの情報に加え、ユーザからアノテーションとして収集した楽曲の構成情報を用いて、楽曲の鑑賞したい部分を選択することにより、曲を再構成し、ユーザに提示するシステムである。例えばサビだけ抜き出したいといった要求や、「イントロ-サビ-エンディング」というように、曲を要約したいといった要求に応える。以下に本システムのインタフェースと実装を述べる。

4.3.1 楽曲再構成システムのインタフェース

再構成システムを実現するために、曲の構成というアノテーションが必要となるが、前述の音楽検索システムのため、既に曲の構成のアノテーションが作成さ

れているので、簡単のため、再構成用に新しくアノテーション定義XMLを記述せず、検索用に生成されたアノテーションを利用する。

ユーザは再構成を行いたい曲を選択することで再構成システムにログインする。図4.3はログイン直後の再構成システムの画像である。本再構成システムは、曲の構成、楽器パート、歌詞による音楽コンテンツの再構成を行う。

左のフレームが再構成システムのメニューとなっている。曲の構成による再構成を行いたい場合は、＜パート＞の部分にある曲の構成から、表示したい部分を選択する。歌詞の有無を変更したい場合は、＜歌詞＞の部分のセレクトボックスにチェックを行い、楽器パートを選びたい場合は、＜楽器パート＞の部分にあるチョイスボックスから、表示したい楽器パートを選択する。選択が終了したら「再構成」のボタンをクリックすると、曲の構成、歌詞の有無、楽器パートの選択を反映させた楽譜が表示される。

また、こうして再構成された音楽コンテンツの MusicXML をダウンロードすることも可能である。

上記のように、歌詞、楽器パート、曲の構成により再構成した MusicXML が生成される。この MusicXML からブラウザに表示できる SVG(Scalable Vector Graphics)[52] 形式の楽譜を生成しブラウザに表示し、同時に、再構成済みの MusicXML をファイルとしてサーバに保存し、ユーザはそれをダウンロードすることができる。再構成後の画面は図4.4のようになる。図中の左側に表示される、再構成メニューから、A メロ (verse-a) とサビ (chorus) の二つのパートが選択されており、それらのパートのみを抜き出して短縮された楽譜が表示されている。

4.3.2 楽曲再構成システムの実装

MusicXML には、歌詞や、楽器パートの情報が含まれているため、歌詞と楽器パートによる再構成は、元の音楽コンテンツである MusicXML のみを利用して実現する。



図 4.3: 再構成前の画面

歌詞は MusicXML では図 4.5 のように音符である note タグの内部に lyric タグとして記述されている。

歌詞の表示をしない場合は、MusicXML の各 lyric タグを、親である note タグ内から削除する。

また楽器パートは、MusicXML では図 4.6 のように、part-list として定義し、さらにそれぞれの楽器パートについて演奏情報が記述される。楽器パートによる再構成では、P1, P2 といった楽器パートの ID を取得し、MusicXML の part-list タグから残しておく ID 以外の score-part タグを削除し、さらに part-list 以降に記述されている part タグも、残しておく ID のもの以外は削除する。

アノテーションである楽曲の構成情報に基づく再構成について説明する。例えば音楽コンテンツが「きらきら星」であり、その楽曲を「イントロ-サビ-エンディング」という再構成するよう要求された場合、まずシステムはアノテーションが格納されたデータベースから、「きらきら星」の内部要素に対して付与されたアノ



図 4.4: 再構成後の画面

テーションのうち、「イントロ」「サビ」「エンディング」という楽曲構成のアノテーションを全て検索する。次にそれらのアノテーションの指し先である楽譜中の要素を全て抜き出し、「イントロ-サビ-エンディング」という順番にそれらを列挙した新しい MusicXML を作成し、再構成した楽譜をユーザに提示する。

同一の要素に対して、複数のユーザが異なる楽曲構成アノテーションを付与している場合、再構成システムを利用しているユーザの嗜好と類似するユーザが付与したアノテーションを優先させる。多様な解釈から、ユーザにとって適切なアノテーションを選出することで、本システムも楽曲検索システムと同様に、個人の解釈に適応させる。また本システムは、楽曲の論理構造であるサビやイントロなどの大まかな構成、楽器パート、歌詞などの情報を基に、楽曲を容易に再構成することから、コンテンツの意味内容に基づきコンテンツ変換を行うアプリケーションの一つであるといえる。

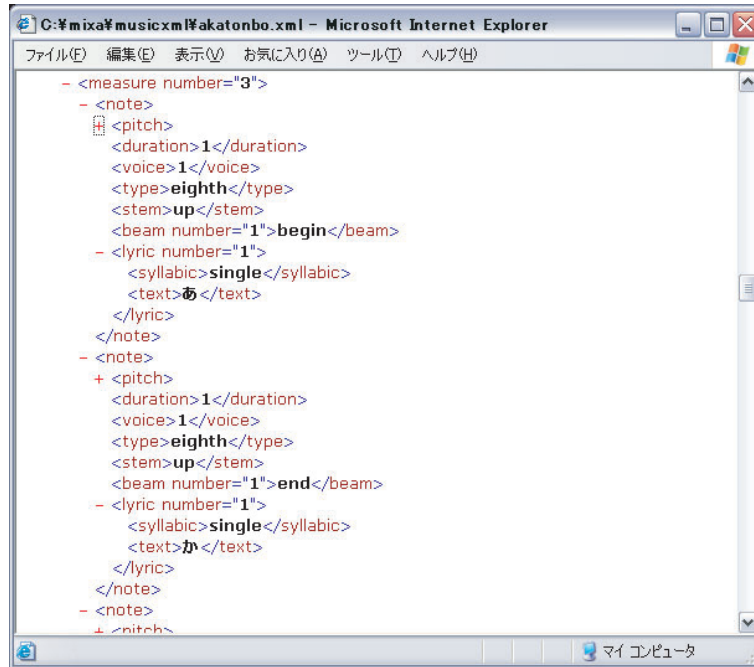


図 4.5: MusicXML における歌詞の記述

4.4 本章のまとめ

音楽アノテーションシステムにより収集が容易になった楽曲アノテーションを用いたアプリケーションとして、楽曲検索システムと楽曲再構成システムを構築した。

まずそれぞれのシステムで利用するアノテーションを Annphony に登録し、音楽アノテーションシステムを通して楽曲の構成や意見などのアノテーションを収集した。楽曲検索システムではそれらの情報を用いて、楽曲の論理構造であるコード進行に基づく検索や、楽曲の構成による絞込み検索を行う機能を持つ。また検索システムを利用するユーザの解釈に類似するユーザが付与したアノテーションを優先的に用いることによって、楽曲の印象検索などをユーザの感性に適応させる。また楽曲再構成システムは、楽曲の論理構造であるサビやイントロなどの大まかな構成、楽器パート、歌詞などの情報を基に、楽曲を容易に再構成するシス

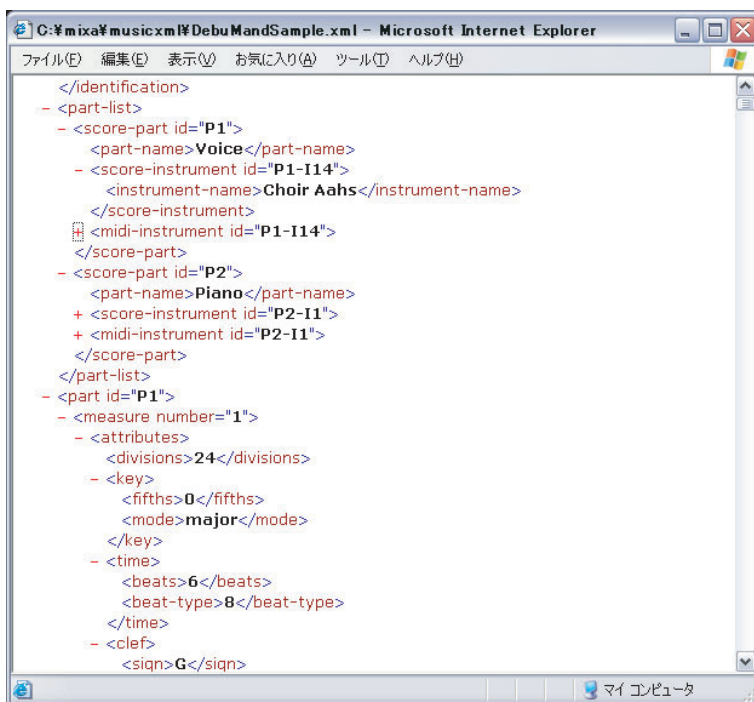


図 4.6: MusicXML における楽器パートの記述

テムであり，楽曲検索システムと同様に，ユーザの解釈と類似するユーザが付与したアノテーションを優先的に利用し，ユーザの感性に適応させる．

これらのシステムから，音楽アノテーションシステムを用いて獲得されたアノテーションを用いて，楽曲の論理構造やユーザの感性に適合させるアプリケーションを実現可能であることが示された．今後の課題としては，これらのシステムを評価するための実験を行い，ユーザの感性に適合させるために必要なシステムの要素を検討することが挙げられる．

第5章 コミュニケーションメディア としてのプレイリスト

近年大容量の携帯音楽プレイヤーが登場し、インターネット上に多数の音楽コンテンツが存在するようになり、いつでもどこでも容易に音楽を楽しむことが可能になった。またネットラジオなどのデジタルコンテンツを自動的に iPod に転送する Podcasting というサービスが現在利用可能である。自分に合ったコンテンツを、インターネット上の膨大なコンテンツの中から自動的に携帯音楽プレイヤーに取り込みたいという要求が高まっている。

そのため楽曲推薦やプレイリストの自動生成に関する研究が盛んに行われている。既に iTunes Store などのインターネットを通じた音楽配信サービスがいくつか利用可能である。また近い将来、定額制サービスが音楽配信にも適用されるだろう。そのような環境下では、多くのユーザ間において、楽曲の ID のみで構成されるプレイリストが共有されることになるだろう。

また現在ブログや SNS により、インターネット上でのテキストや絵、写真などによる活発な個人間コミュニケーションが行われている。音楽は多くの人にとって容易に作成できないメディアであるが、プレイリストはテキストや写真などと同様、容易に作成することができ、また作成者の意図や感情を込めることができる。我々は今後プレイリストがテキスト・絵・写真と同様、重要なユーザ間のコミュニケーションメディアになると考えている。以降プレイリストを介したコミュニケーションを Playlist-Mediated Communication と呼ぶことにする。

従来の楽曲推薦に関する研究は、協調フィルタリングによるプレイリスト生成システム [30][32] やジャンル、アーティストなどの情報を利用したプレイリスト生

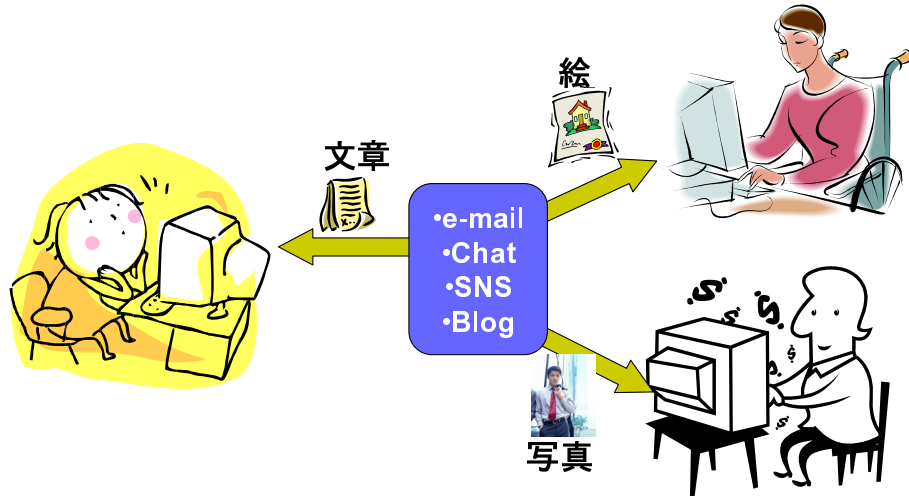


図 5.1: コミュニケーションツールとコミュニケーションメディア

成 [53] などがある。しかし，楽曲推薦やプレイリスト生成に有効である情報の中には収集することが困難であるいくつかの主観的・個人的な情報が存在する [54]。そこで，3章で述べた音楽アノテーションシステムにより，プレイリスト作成に有効な情報を Web 上のユーザから取得し，利用する。

Playlist-Mediated Communication を実現するために必要な要素は，プレイリストの作成・推薦・鑑賞といった活動を活発にすることであると考え，そのためのプレイリスト推薦の手法を提案する。本システムの実現により，プレイリスト作成・推薦の促進を実現することができるであろう。プレイリストをコミュニケーションメディアとして利用されるようにするためには，あるユーザが作成したプレイリストを，他のユーザが容易に参照可能である必要がある。そこで本システムを Web サービスとして設計する。Web サービスであれば，インターネットを介した他のユーザとの円滑な意見交換やプレイリスト推薦の実現が期待される。

本章では Playlist-Mediated Communication を実現するために，構築したプレイリストの作成とコミュニケーションを支援するためのシステムについて述べる。まずプレイリストの作成支援のために必要な楽曲特徴量を考察し，それに基づいてアノテーション定義を用意する。次にそれらのアノテーションを用いて，楽曲群

からプレイリストの作成支援を行う仕組みについて述べる．さらにそうして作成されたプレイリストを介して解説や感想などを交換し合うコミュニケーション機能について述べる．最後に本システムによって作成されるプレイリストがユーザの嗜好や状況に適応しているかどうかを確認するための実験を行う．

5.1 プレイリスト作成支援システムのためのアノテーション定義

ジャンルやアーティスト、歌詞など、一般的に楽曲推薦に利用すると効果的であるとされている楽曲の特徴量がいくつか存在する [55][56]．一方、日常生活における音楽との接し方から推測されるように、リスナの置かれている状況が、推薦すべき楽曲の選択に強く影響すると考えられる．そこでリスナの嗜好と状況に合ったプレイリスト推薦システムを実現するために、楽曲の特徴量として歌詞・表現している情景 (楽曲情景)・聴きたい状況 (鑑賞状況) の3種類を採用した．

楽曲情景は、「片思いの心象を歌った曲」や「卒業・別れの曲」など、その楽曲がどのような情景を表現しているかという情報であり、個人嗜好への適応に有効であると考えられる．鑑賞状況は、「海に大勢で遊びに来ている時」、「恋人と二人のクリスマスの夜」など、その楽曲をどのような状況で聴きたいかという情報であり、状況への適応に有効であると考えられる．楽曲情景、鑑賞状況に関する情報はリスナによる楽曲の解釈に強く関わる情報であるため、楽曲の音響情報を自動解析して得ることが困難である．そこで、3.2.3節で述べた Tune Annotator により各楽曲に対して楽曲情景と鑑賞状況の情報を収集した．

楽曲情景・鑑賞状況アノテーションの定義にあたり、事前の予備実験において、どのような楽曲情景、鑑賞状況が存在するかを調査し、いつ・どこで・どのような心理状態であるかという項目を設定した．表 5.1 に、収集した鑑賞状況アノテーションの項目を示す．例えば「いつ」に関しては、朝、昼、夜、春、夏、秋、冬などの項目が設定されており、それらの中からその楽曲を鑑賞するのに適した時

間・時期を選択する．各楽曲に対するこれらの情報を Annphony にアノテーション定義として登録し，音楽アノテーションシステムの Tune Annotator を用いてアノテーションを収集した．

誰と	一人, 二人, 大勢
いつ	朝, 昼, 夜, 過去, 現在, 未来, 春, 夏, 秋, 冬
場所	家, 外, 移動中, 自然, 都会・街, 外国
心情	元気, 勇気, まったり, 楽しい, 悲しい 怒り, 孤独, 思考中, 真面目, 不真面目 幸せ, 不幸, 失望, 哀愁
その他	恋愛中, 片想い, 失恋, 卒業, 平和, 反社会, 別れ, 日常

表 5.1: 鑑賞状況に関するアノテーション項目

5.2 歌詞とアノテーションによる楽曲・ユーザ間の類似度推定

歌詞の TF*IDF [57], 楽曲情景, 鑑賞状況の3種類をそれぞれの楽曲の特徴量とした． l_{t_i} , c_{t_i} , s_{t_i} を歌詞・楽曲情景・鑑賞状況の各特徴量空間における楽曲 t_i の特徴ベクトルとする． l_{t_i} , c_{t_i} , s_{t_i} は以下のように列ベクトルで表される．

$$l_{t_i} = (f(t_i, x_1), f(t_i, x_2), \dots, f(t_i, x_n))$$

$$c_{t_i} = (f(t_i, y_1), f(t_i, y_2), \dots, f(t_i, y_o))$$

$$s_{t_i} = (f(t_i, z_1), f(t_i, z_2), \dots, f(t_i, z_p))$$

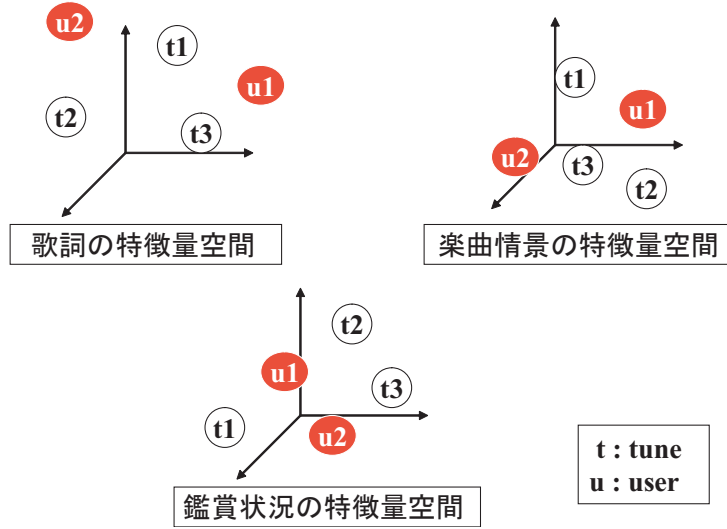


図 5.2: 歌詞・楽曲情景・鑑賞状況の特微量空間

TF*IDF によって求められたキーワードの数は n であり, x_j は j 番目のキーワードを表している. $f(t_i, x_j)$ は楽曲 t_i の歌詞中に, キーワード x_j が含まれていれば 1, 含まれていなければ 0 となる. o, p はそれぞれ楽曲情景, 鑑賞状況アノテーションに設定された項目数であり, y_j, z_j は, それぞれ楽曲情景, 鑑賞状況アノテーションの j 番目の項目を表している. また $f(t_i, y_j), f(t_i, z_j)$ は, 楽曲 t_i に対する楽曲情景, 鑑賞状況アノテーションとして, y_j, z_j の項目が選択されている場合は 1, 選択されていない場合は 0 となる.

各特微量空間における楽曲間の類似度は, 特徴ベクトル同士のコサイン距離により得られる. また歌詞・楽曲情景・鑑賞状況の特微量空間で得られた各コサイン距離の重み付き和を, 最終的な楽曲間の類似度とした.

一方, 各ユーザとプレイリストも各特微量空間にマップされる. ユーザ u_j の特微量は以下の式により求められる.

$$l_{u_j} = \frac{\sum_i l_{t_i}}{i}$$

$$c_{u_j} = \frac{\sum_i c_{t_i}}{i}$$

$$s_{u_j} = (f(u_j, z_1), f(u_j, z_2), \dots, f(u_j, z_p))$$

ユーザ u_j が嗜好に適合するとシステムに申告された t_i で表される楽曲の集合に対して、特徴量の平均をそのユーザの嗜好に最も適合する楽曲であるとし、ユーザは歌詞・楽曲情景の特徴量空間にマップする。さらにユーザ自身が現在置かれている状況をシステムに入力した時点で鑑賞状況の特徴量空間にもユーザがマップされる。プレイリストも同様に、プレイリスト要素である楽曲の鑑賞状況・歌詞ベクトルの平均値と、そのプレイリストが作成された状況から、それぞれの特徴量空間にマップされる。

$$l_{p_k} = \frac{\sum_m l_{t_m}}{m}$$

$$c_{p_k} = \frac{\sum_m c_{t_m}}{m}$$

$$s_{p_k} = \frac{\sum_m s_{t_m}}{m}$$

プレイリスト p_k 中に含まれる楽曲 t_m で表される楽曲の集合に対して、各特徴量の平均をそのプレイリストの特徴量とする。楽曲間の類似度計算と同様に、コサイン距離の重み付き和により、楽曲とユーザ間、ユーザとプレイリスト間などの類似度が算出される。

5.3 プレイリスト作成支援システム

図 5.3 にプレイリスト作成の流れを示す。まずユーザは本システムに、「冬の夜、一人で考え事をしている」といった現在ユーザが置かれている状況を図 5.4 の状況

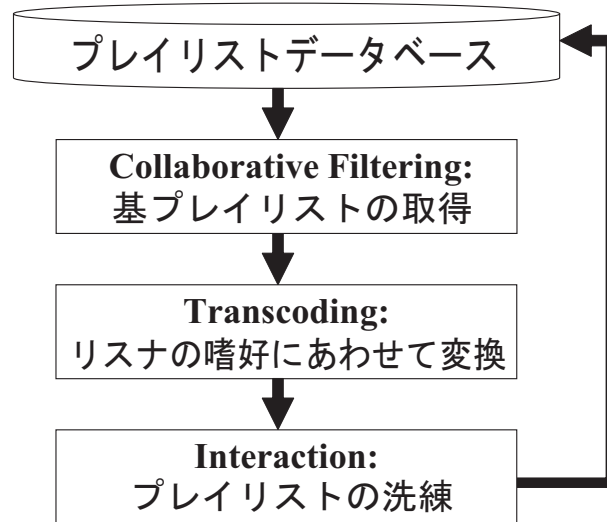


図 5.3: プレイリスト作成の流れ

入力フォームから該当する項目を選択することにより申告する。システムは、データベース中のプレイリストからユーザに適したプレイリストを選出する協調フィルタリングを行う。次に選び出されたプレイリストを、よりリスナの嗜好に適合させるトランスコーディングを施しユーザにプレイリストを提示する。さらにシステムとリスナのインタラクションによりリスナの嗜好に適応していく。以下に詳細を述べる。

5.3.1 協調フィルタリングによる基プレイリストの絞り込み

ユーザの置かれている状況をシステムが受け取ると、まずシステムは過去に作成されたプレイリストの中から、ユーザの嗜好モデルと類似し、さらに現在のユーザの状況と類似する状況で作成された基プレイリストをデータベースから選出する協調フィルタリングを行う。

システムはそれぞれのユーザのプレイリスト作成履歴から、そのユーザの嗜好に合った楽曲集合を保持している。ここで、ユーザにとって最も嗜好に合う楽曲が存在し、その楽曲はユーザの嗜好に合った楽曲集合を元に推測できると仮定した。

あなたは今どんな状況に置かれていませんか？

・誰と一緒に？ ☐ 一人 ☐ 二人 ☐ 大勢
・今はいつ？ ☐ 朝 ☐ 昼 ☐ 夜 ☐ 過去 ☐ 現在 ☐ 未来 ☐ 春 ☐ 夏 ☐ 秋 ☐ 冬
・どこにいる？ ☐ 家 ☐ 外 ☐ 移動中 ☐ 自然 ☐ 都会・街 ☐ 外国
・今どんな状況？ ☐ 恋愛中 ☐ 片想い ☐ 失恋 ☐ 卒業 ☐ 平和 ☐ 反社会 ☐ 別れ ☐ 日常
・どんな気持ち？ ☐ 元気 ☐ 勇気いっぱい ☐ まったり ☐ 楽しい ☐ 悲しい ☐ 怒り ☐ 孤独 ☐ 思考中 ☐ 真面目 ☐ 不真面目 ☐ 幸せ ☐ 不幸 ☐ 失望 ☐ 哀愁
・その他どんな状況かについて 付け足すことがあればご記入ください

送信

図 5.4: 状況入力フォーム

Title of this Playlist:

じめじめしたあなたをさらにじめじめに

recommend this playlist

Please write Liner Note:

入替	好きな曲?	状況に合ってる?	Playlist
☐	☐	☐	奥田民生 - イージュライダー'97
☐	☐	☐	aiko - 天の川
☐	☐	☐	Mr.Children - 星になれたら
☐	☐	☐	松任谷由実 - やさしさに包まれたなら
☐	☐	☐	aiko - ボーイフレンド
☐	☐	☐	Mr.Children - つよがり
☐	☐	☐	aiko - 白い服黒い服
☐	☐	☐	aiko - 桃色
☐	☐	☐	MIT - HORO
☐	☐	☐	Jeff Manning - I think of you

manual edit submit

menu

Other Liner Notes:

このプレイリストは果たして本当にじめじめするのでしょうか？聞いてみましたか？aikoとか松任谷由実とか落ち着いた感じがさわやかでした。むしろじめじめ感が和らぎましたよ。
(by hosono)

Other Liner Notes of Fragment:

まったりしたいときにはこの2曲はぜひセットで聴くと良いです。大御所の二人がじっくりと歌い上げています。さすがです。
(by yamamoto)

aikoの中ではマイナーではありますが、まさに癒しを演出してくれるありがたい曲たちであります。
(by hosono)

図 5.5: ユーザに提示されるプレイリスト

そして歌詞と楽曲情景の特徴量に関してそれぞれユーザの嗜好に合った楽曲集合の平均を計算し、図 5.2 のようにそれぞれの特徴量空間にマップする。また、ユーザの置かれている状況に関しては、本システムの鑑賞状況入力フォームからの入力に基づき、ユーザを鑑賞状況の特徴量空間にマップする。鑑賞状況入力フォームの項目は、5.1 節で述べた各楽曲に対する音楽アノテーションシステムにおける鑑賞状況の項目と同等である。

5.3.2 基プレイリストの個人への適応

次に基プレイリストをさらにユーザの嗜好に適合させるトランスコーディングを施す。具体的には、ユーザが過去に嗜好に合わないシステムにフィードバックした楽曲を取り除き、楽曲データベースから代わりの楽曲を選んで追加する。同時に、ユーザが鑑賞経験のない楽曲を 3 割の割合で含ませている。聴き慣れた楽

曲の中に適度に鑑賞経験のない楽曲を挿入することで、ユーザの嗜好の幅を徐々に広げる効果が期待される。

嗜好に合わない楽曲などの代わりに追加する楽曲は、ユーザの嗜好と、置かれた状況から、各特微量空間におけるユーザと楽曲の類似度を踏まえて選出される。ユーザと楽曲間の類似度は、5.2 節で述べた楽曲間の類似度計算と同様の計算式から導き出される。

5.3.3 インタラクションによるプレイリスト更新

以上の処理を経てユーザには図 5.5 のようにプレイリストが提示される。作成されたプレイリストは左側に、プレイリストを介したコミュニケーションを行う部分は右側に表示される。ユーザはプレイリスト上部に埋め込まれたプレイヤから、通常の音楽プレイヤと同様の操作でプレイリストを楽しむことができる。

ユーザは実際にプレイリストを鑑賞し、各楽曲が嗜好に合っているか、状況に合っているかといった情報をフィードバックする。画面左側に楽曲リストと共に表示される「○」「×」を選択して申告する。また現在のプレイリストから、入れ替えを行いたい楽曲の「入替」のボックスをチェックする。システムはユーザからのフィードバックを受け取り、気に入らない・現在の状況に適していない楽曲の入れ替えを行い再度ユーザにプレイリストを提示する。

このように協調フィルタリングとトランスコーディングによって自動的にプレイリストを生成し、そのプレイリストをユーザとのインタラクションによって洗練していくことで、ユーザが大量の楽曲群からプレイリストとなる楽曲を選択するよりも低いコストでプレイリストを作成することができる。

5.3.4 プレイリストの人手による編集

図 5.5 の左下に現れる”manual edit” ボタンを押すと、図 5.6 のようにユーザ自身が登録楽曲中から選択した楽曲を追加したり、プレイリスト中の楽曲を削除す

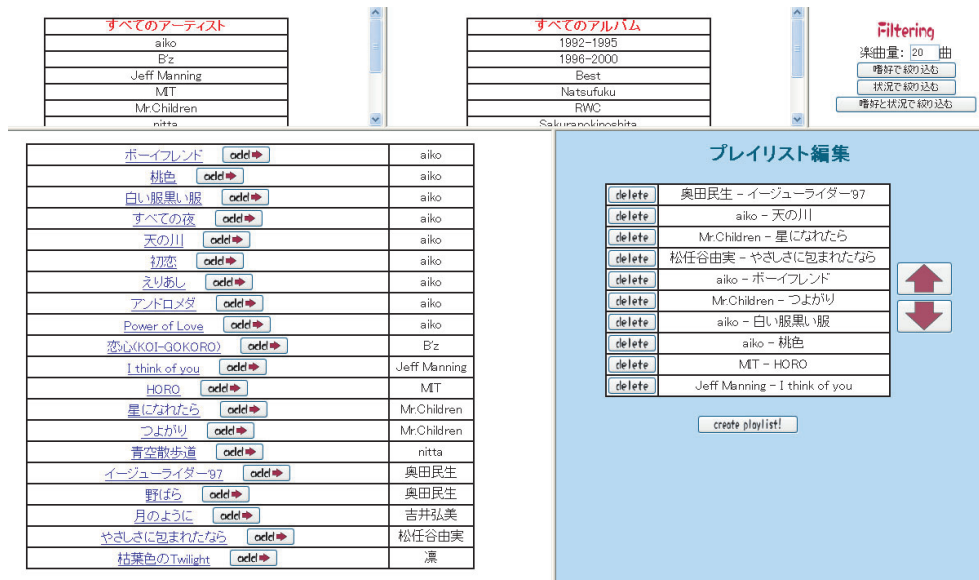


図 5.6: プレイリスト編集画面

るプレイリスト編集ページが表示される。登録されている楽曲のアーティスト、アルバムのリストが上段左側と上段中央に表示され、それらを選択し、プレイリストに追加する楽曲の候補を絞り込むことにより下部左側に選択されたアーティスト・アルバムの楽曲一覧が表示される。ユーザは楽曲タイトルの右に現れる add ボタンを押すと、楽曲一覧の右側に現れるプレイリストに楽曲が追加される。またプレイリストの楽曲タイトルの左に現れる delete ボタンを押すと、編集中のプレイリストから該当する楽曲が削除される。また楽曲を選択し、矢印のボタンを押すことによりプレイリスト中の楽曲の順序を変更することができる。

プレイリストを人手により作成・編集する際に最も時間を要するのが、膨大な量の楽曲の中からどの楽曲を選択したらよいかを決定する作業である。そこで本システムではその作業の時間を短縮するよう、嗜好と状況に基づいてプレイリストに追加する候補となる楽曲を自動的に絞り込む機能を備える。画面上部右側に現れるフィルタリングのメニューから、絞り込む楽曲数を入力し、ボタンを押すことでユーザの嗜好とユーザの状況に最も適合する楽曲から、指定された数まで

の楽曲を検索し、画面下部左側に表示する。

5.3.5 インタラクション内容に基づくユーザモデルの更新

ユーザの嗜好モデルは、本システムに対して、それぞれの楽曲が嗜好に合うかどうか、また状況に合っているかどうかというフィードバック情報を用いて順次更新される。TF*IDF で得られるキーワード数が x である場合、歌詞の特徴量空間における初期状態の基底は、 x 次元単位行列と同等の標準基底である。ユーザが嗜好に合っているとフィードバックした楽曲群と、合っていないとフィードバックした楽曲群を用い、各ベクトル空間の基底をユーザの嗜好に適応させる。そのためベクトル空間の基底はそれぞれのユーザについて異なる。以下は歌詞の特徴量空間における基底をフィードバックに応じて変化させるための式である。

$$b_u \leftarrow \text{normalize}(b_u + \text{diag}(\delta \frac{sl_f - l_u}{|l_f - l_u|} - \delta \frac{l_d - l_u}{|l_d - l_u|}))$$

b_u はユーザ u の歌詞の特徴量空間における基底である。 l_f はフィードバックによって嗜好に合っているとされた楽曲の歌詞特徴ベクトルの平均であり、 l_d は嗜好に合わないと言われた楽曲の歌詞特徴ベクトルの平均である。 l_u は、ユーザ u がこれまでに嗜好に合っているとフィードバックした楽曲全体の歌詞特徴ベクトルの平均である。一回のフィードバックにおいてどれだけ基底を変化させるかを決定する変換レートは δ で表されている。本システムでは経験的に δ の値を設定している。 diag は列ベクトルを対角行列に変換する関数である。

例えばユーザが「愛」というキーワードの入った楽曲を嗜好に合うとし、「死」というキーワードの入った楽曲を嗜好に合わないと言った場合、上記の基底変換により、歌詞の特徴量空間は図 5.7 に示すように愛の要素が短縮され、死の方向へ拡張される。特徴量空間を嗜好に合った楽曲の方向に短縮し、嗜好に合わない楽曲の方向に拡張することで、ユーザの好む楽曲とユーザの間の距離を縮め、逆にユーザと嗜好に合わない楽曲同士を遠ざける。楽曲情景と鑑賞状況の

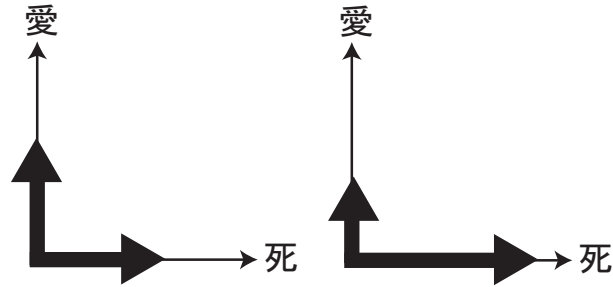


図 5.7: 歌詞の特徴量空間の基底変換前 (左) と後 (右)

特徴量空間についても同様に、アノテーションの項目数を y, z とした場合、 y, z 次元単位行列を初期状態とし、ユーザからのフィードバックごとに基底をユーザに適応させる。ユーザ間の類似度計算の際の基底変換には、プレイリストを作成する主体となるユーザの基底を用いる。

5.4 プレイリストを介したコミュニケーション

後に6.3節で述べるように、コンテンツはコミュニケーションのために必要不可欠な存在である。そのためコンテンツの作成支援を行うことは人間同士のコミュニケーションを支援することにつながる。しかし、計算機はコミュニケーションのためのコンテンツを作成する支援を行うだけではなく、作成されたコンテンツを用いた実際のコミュニケーションに対しても支援を行うことが必要であると考えられる。そこで本システムは前節で作成したプレイリストを中心としたコミュニケーションを行うために、プレイリストを共有し検索する機能、解説や感想などを記述するための機能を備える。以下に詳細を述べる。

ユーザが作成したプレイリストはデータベースに保管され、他のユーザによる検索・鑑賞が可能である。図5.8にプレイリストの検索画面を示す。ユーザはプレイリストに含まれる楽曲の情報や、プレイリストの作成者、プレイリストに対するライナーノーツなどからの検索を行うことができる。

図 5.8: プレイリスト検索画面

ユーザは作成したプレイリストや、検索し鑑賞したプレイリストが気に入った場合、図 5.5 の左側上部に現れるボタンを押すことで、そのプレイリストを他のユーザに推薦することができる。プレイリスト推薦の情報は、基プレイリストを選出する協調フィルタリングの際に、プレイリストの重みとして利用される。多くの嗜好の類似するユーザに支持されるプレイリストは基プレイリストとして選出されやすくなる。

またプレイリストやそれに含まれる複数の楽曲に対するライナーノーツ (解説) や感想の記述が可能である。他のユーザにより記述されたライナーノーツは左側の下部に表示されている。それぞれのライナーノーツをクリックすると、右側の該当する楽曲がハイライトされる。ユーザは図 5.5 の左側に現れる入力フォームから、そのプレイリスト全体に対するライナーノーツを記述する。プレイリスト中の複数の楽曲に対してライナーノーツを記述する場合は、右側のプレイリストから複数の楽曲を選択すると、入力フォームが出現し、選択した楽曲に対するライナーノーツを記述することができるようになる。

以上のようにプレイリストの作成とプレイリストを中心としたコミュニケーショ

ンを支援する機能を備えることで、プレイリストに対する意見交換などのコミュニケーションに触発されて、そのプレイリストから新たなプレイリストが派生させるユーザが増加することになるだろう。多くのプレイリストの派生は、プレイリストを中心としたコミュニケーションの促進につながると考えられる。このように本システムはコンテンツの増加・価値の向上とコンテンツを中心としたコミュニケーションの間の正の循環を促す可能性がある。

5.5 ユーザモデルの適応に関する評価実験

5.5.1 実験方法

評価実験に利用した楽曲は RWC 音楽データベース [58] の POPS100 曲と、一般の POPS130 曲である。事前に各楽曲に対して、5.1 節で述べた楽曲情景・鑑賞状況に関するアノテーションを収集し、その情報を楽曲の特徴量として利用した。

構築したシステムの評価を行うため、7 人の被験者に対して以下の手順で評価実験を行った。まず自由に自分の置かれている状況を設定してもらい、本システムの鑑賞状況入力フォームから入力してもらい、システムは各被験者の嗜好と、入力した鑑賞状況に適したプレイリストを提示する。そこで実際にプレイリストを鑑賞してもらい、各楽曲に対して嗜好に合っていたか、状況に合っていたかの情報をチェックしてもらい、システムにフィードバックしてもらい、システムは受け取った情報を基にプレイリスト作り変え、再びユーザにプレイリストを提示する。以上の行為をプレイリストのすべての楽曲が嗜好・状況共に合ったものになるまで繰り返してもらい、以上のプレイリスト作成プロセスを、3 回繰り返してもらった。

本システムによってユーザの嗜好・状況に合っているプレイリストを作成されているかどうかを確認するため、各プレイリスト作成プロセスにおいて、鑑賞状況入力後提示されるプレイリスト中に、嗜好に合う楽曲・状況に合う楽曲・嗜好と状況に合う楽曲はそれぞれ何曲存在したか、またすべての楽曲が嗜好と状況に合うまで行ったインタラクションの回数を評価した。

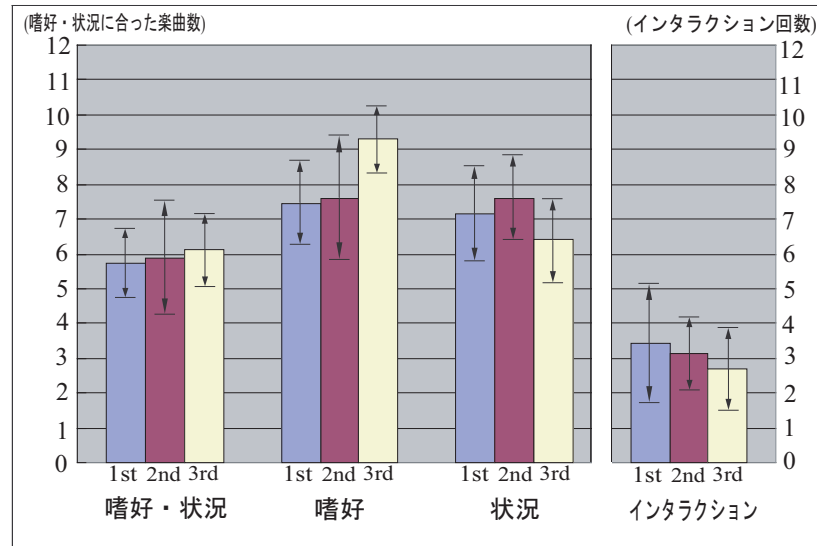


図 5.9: 評価実験の結果

5.5.2 実験の結果と考察

評価実験の結果を図5.9に示す．それぞれのグループは3種類のバーによって構成されており，左から1回目，2回目，3回目のプレイリスト作成プロセスにおける評価値である．各バーの頂点の矢印は各評価値の標準偏差を表している．

図5.9の左から2番目に表される嗜好に合った楽曲数のグラフと，左端の嗜好・状況共に合った楽曲数のグラフからは，プレイリスト作成プロセスを重ねるにつれてユーザに合った楽曲数が増加していることが読み取れる．また右端のインタラクションの回数は，徐々に減少していることがわかる．このことから本システムを長く利用するユーザほど，嗜好と状況に合ったプレイリストを，少ないインタラクションにより得ることができるといえる．

一方，左から3番目のグループに現れている，状況に合った楽曲数のグラフでは，1回目，2回目，3回目とプレイリスト作成を繰り返すことによる増加傾向は見られない．鑑賞状況に合った楽曲提示については，プレイリスト作成を繰り返すことにより改善されているとはいえない結果であった．

この結果は以下の2つの問題点が原因となって起こったと考えられる。一つ目は本システムで入力を要求する鑑賞状況項目の組み合わせが、利用できる楽曲数に比べて非常に多いという点、二つ目に、各楽曲に対して収集した楽曲情景・鑑賞状況のアノテーションは各楽曲に均一に付与されているわけではないという点である。

前者の問題点に対しては、より多くの評価実験を繰り返し、鑑賞状況項目をより効果的な項目に絞り込み、同時にプレイリストの要素として利用可能な楽曲数を追加することで対応できると考えられる。後者の、情報の偏りに関する問題は、一般ユーザを対象としたアノテーション獲得のためのシステムにおいては、コンテンツの人気などにより頻繁に発生する問題である。この問題を解決するために、ユーザをアノテーションが不足しているコンテンツに誘導したり、リスナの鑑賞履歴などを利用することでアノテーションの均一化を図る仕組みが必要となる。

しかし鑑賞履歴のような情報は、アノテーション獲得のためのシステムから付与された情報と比べて一般的に信頼性に劣るため、そういった情報を適用するためには、各アノテーションの重みを考慮する必要がある。

5.6 本章のまとめ

音楽アノテーションシステムにより収集されたアノテーションを用いて、個人の嗜好と状況に適合する音楽アプリケーションを実現した例として、プレイリストにまつわる作成・推薦・鑑賞といった活動を活発にすることで Playlist-Mediated Communication が実現され则认为、プレイリストの作成支援とプレイリストを中心としたコミュニケーションを行うシステムを構築した。

本システムは、ユーザの嗜好と状況に適応するプレイリストの作成支援を行うために、楽曲情景・鑑賞状況に関するアノテーションを定義し、音楽アノテーションシステムに適用した。プレイリストを作成する際には、既存のプレイリスト中から嗜好・状況の類似するプレイリストを選出し、さらにユーザに適応するラン

スコーディングを行いユーザに提示する。ユーザは提示されたプレイリストに対して嗜好・状況に適合しているかどうかをシステムにフィードバックすることで、プレイリストを徐々に洗練していく。またプレイリストに対してライナーノーツや感想を記述することで、プレイリストを中心としたコミュニケーションを促す。

本システムは、評価実験を通して、アノテーションがユーザの心的状態や置かれている状況、またユーザと楽曲間の関係を正確かつ動的に表すために有効であると確認された。嗜好・状況に適応するために、楽曲情景・鑑賞状況アノテーションを利用することが有効であることが明らかになったといえる。同時に、協調フィルタリング・トランスコーディング・インタラクションの組み合わせがユーザの嗜好に合ったプレイリスト推薦に有効であることが明らかとなった。本システムにより、プレイリスト作成と推薦を促す仕組みが実現できたといえる。

第6章 関連研究

本章では、本研究と関連する研究を挙げ、その差を比較することにより本研究の優位性を示す。

6.1 マルチメディアアノテーション

2章でアノテーション基盤である Annphony と関連する研究として、デジタルコンテンツへのアノテーションを対象としたアノテーションの枠組みに関する研究が存在する。

Staab ら [59] は、人物紹介ページに現れる人物の属性や人間関係など、Web コンテンツに存在する様々なオブジェクトのオントロジを構築するため、アノテーションを記述するための仕組みを提案している。Web コンテンツ内のオブジェクトに対するアノテーションはアノテーションエディタである OntoAnnotate を通して記述する。オブジェクトの属性情報やオブジェクト間の関連は RDFS により定義されるため、様々なオブジェクトの情報を記述することができる。

この仕組みは、任意のコンテンツに対して、そのコンテンツを構成するオブジェクトの詳細な情報記述に適していると考えられるが、それらのアノテーションから構成されるオントロジは、コンテンツとは切り離されたオブジェクト間の関係性である。そのため、それぞれのコンテンツがどのような構造をしているか、コンテンツ同士にどのような関連性があるかといった、コンテンツ自体の意味内容を扱うことには適していない。一方 Annphony は、任意のコンテンツの意味内容に基づく処理を実現するために、コンテンツ自体の構造や関連性を記述するため

の仕組みである。

また Barger ら [60] の提案するアノテーションのためのフレームワークでは、外部に用意されたライブラリを参照することで、ドキュメントの一部や画像の一部を指し示す仕組みを備える。指し先の情報は XLink により管理され、その指し示した部分に対するアノテーションを記述することができる。またそのライブラリを利用することによって指し示されたコンテンツの内部をプログラム中で利用することも可能である。これは ElementPointer と ElementPointer プロセッサに相当する仕組みであるといえるが、外部ライブラリの利用支援のための機能がない点が ElementPointer と異なる。

コンテンツ内部の指し示し方は、コンテンツの種類に特化しており、専門性が高い。外部ライブラリの利用支援の仕組みがない場合、それぞれの種類のコンテンツを熟知し、かつその外部ライブラリの存在を知っていなければならず、幅広いユーザが任意のコンテンツの横断的なアプリケーションを構築することができなくなってしまう。一方 ElementPointer の定義やそのプロセッサは、Annphony において管理され、どのようなコンテンツに対してどのような ElementPointer を利用可能であるか、またどのように利用するかという情報を提供する機能を持つ。

またここで挙げた関連研究は、双方とも事実や関係性など、一意に決定される情報のみを扱うことを想定しているため、個人によって異なる多様な解釈を扱うことには適していない。コンテンツを扱うユーザ層の拡大によって、個人の感性への適応した応用が求められているが、これらの枠組みによって記述されたアノテーションを用いて個人の感性に適応することは困難である。一方 Annphony では、個人によって異なる多様な解釈を記述可能であるため、アノテーションをユーザごとに管理したり、特定の傾向を持つアノテーションに分類するなど、個人適応につながる処理が可能である。

6.2 音楽アノテーション

3章で提案した音楽アノテーションシステムに関連する研究として, MoodLogic[27], CUIDADO(Content-based Unified Interfaces and Descriptors for Audio/music Databases available Online)[61], CLAM Annotator[28] を挙げる.

MoodLogic は Web 上のユーザコミュニティと専門家によって楽曲に対するタイトル・アーティストなどの基本情報に加えて, テンポや印象情報を収集し, 「アップテンポなロック」や「ロマンチックなブルース」など, 印象情報に基づく検索やプレイリスト作成を支援する.

音楽アノテーションシステムと MoodLogic の比較について述べる. まず, MoodLogic ではユーザから収集した印象・ジャンルなどの情報を統計し, その楽曲の印象・ジャンルを一意に決定している. 一方音楽アノテーションシステムで収集され, Annphony で管理される複数の解釈に関するアノテーションの利用法は, アプリケーションごとにゆだねられている. MoodLogic と同様に, 統計をとり一意の値を取得することもできるが, それぞれのアノテーションにはアノテータ情報が付与されているため, 本研究のプレイリスト作成支援システムのように, 楽曲の嗜好に関するユーザモデルを用いることで, あるユーザと類似するユーザが付与したアノテーションを優先的に利用したり, 特定のユーザが付与したアノテーションに特に注目するなど, ユーザごとに個人化したアノテーション利用が可能になる.

また MoodLogic では楽曲検索とプレイリスト作成支援にのみ利用されるアノテーションを収集しており, また他のアプリケーションがその情報を利用することは困難である. このことから, アノテーション定義の柔軟性と, アノテーション共有に関して問題があるといえる. 一方音楽アノテーションシステムはアノテーション定義をアプリケーション開発者が定義することで, アノテーション収集のためのエディタが動的に構成されるため, 様々なアプリケーションに適用するアノテーションを収集することが可能である.

CUIDADO は、オンライン音楽データベースの統一的なインタフェースを提案し、楽曲ブラウジング・検索・オーサリングの支援を目指すプロジェクトである。メタデータの記述形式は MPEG-7 を参考にしており、タイトルやアーティストなどの基本情報を収集し統計的なインデキシングを行い、さらにオーディオデータからテンポや音響パワーなどの特徴量を自動抽出し管理する。

アプリケーションの一つとして、音楽制作者のための検索・編集・処理ツールである Sound Palette が提供されている。ユーザはメロディやリズムに基づく類似楽曲検索で得られる楽曲ファイルをインポートし、自動でセグメント化された楽曲の断片を利用することができる。またテンポの異なる二つの楽曲をミックスさせる際に、自動的にテンポを同期させるなど、音楽の意味内容に基づいた編集を支援する。

このプロジェクトでは、楽曲の内部構造に関する情報は自動的に抽出されているが、楽曲編集に有効な情報は、必ずしも自動的に抽出されるとは限らない。そのため音楽アノテーションシステムのように楽曲構造に関するアノテーションを人手により付与する必要があると考えられる。また、本論文のシステムのように本質的に自動的に抽出できない楽曲の特徴を収集するためにはユーザによるアノテーションは不可欠だと思われる。

CLAM Annotator は、楽曲の音響信号の解析結果に関するアノテーションを付与するための仕組みである。楽曲の一部の音程の情報や和音情報、小節などの区間情報、楽曲の構成などの記述が可能である。CLAM Annotator の画面には、音響信号の波形が表示され、波形の一部を選択することで、時間区分に対するアノテーションを記述する。記述可能なアノテーションの種類は固定されておらず、アノテーションの定義を用意することで、様々なデータ型のアノテーションを扱えるようになる。

音楽アノテーションシステムと異なる点として、ユーザに提示する音楽の形式が音響信号の波形情報であることが挙げられる。音楽アノテーションシステムでは、時間区分に対するアノテーションエディタとして Timeline Annotator を構築した

が、ユーザが特定の時間区分を選択するための視覚情報として、CLAM Annotatorのように波形を提示することは有効であると考えられる。しかし CLAM Annotatorは、音メディア以外の形式の楽曲には適用することができない。一方音楽アノテーションシステムでは、MIDI や MusicXML など、音楽メディアの様々な形式に対応している。

また CLAM Annotator は、信号解析の結果などの楽曲に対する事実を記述するためのエディタであるため、人によって異なる解釈を記述することができない点も、音楽アノテーションシステムと異なる。和音情報や楽曲の構成などは、人によって解釈が異なるため、そのような多様な解釈を収集する必要がある。

6.3 コミュニケーションシステム

計算機を通じたコミュニケーションの形態は日々変化しており、その中で最も重要な役割を担っているのがコンテンツであるといえる。Norman[62] は、飛行機におけるパイロットのコミュニケーションの古さについて述べ、共通のタスクを実現しようとするユーザ間では言葉以外のコンテンツによって無意識にやり取りされるコミュニケーションの重要性について述べており、Nelson[63] は、社会的な発展に応じてこのようなコミュニケーションのためのコンテンツは拡張・改良されていくと指摘している。

コミュニケーションコンテンツの作成に関する代表例として、Apple Computerが開発した HyperCard[64] が挙げられる。HyperCard はハイパーテキストのノードにはカードを、カード間をつなぐリンクにボタンを用い、グラフィカルにマルチメディアアプリケーションを作成できるシステムである。現在もインタラクティブ性や、アプリケーション作成の容易さから幅広く普及している。また田中らの IntelligentPad[65] は、計算機をメタメディアとみなし、文化遺伝子(ミーム)を持ったメディアという意味のミームメディアを作成するシステムであり、モックアップシステムを顧客と開発者間で議論しあう際などに、顧客の目前でのデザインを容

易に変更するといったことが可能になる。

このように、コミュニケーションのためにコンテンツは必要不可欠な存在であるといえる。以降では特に楽曲を用いたコミュニケーションシステムについての関連研究について述べる。音楽は言葉として表現できないあいまいな情報を伝えることのできるコンテンツであり、今後そのコミュニケーションへの利用価値の認識が高まると予想される。

5章で提案したプレイリスト作成支援システムのように、ユーザ間で楽曲を推薦し合うための仕組みがいくつか存在する。ここではその代表例として、Last.fm[66]とPANDORA[67]、iMix[68]を挙げ、本研究とのアプローチとの違いを述べる。

PANDORAはシステム上で提示される楽曲が嗜好に合っているかどうかを繰り返し答えることによって、ユーザの嗜好に合わせたラジオ局を開設するシステムである。楽曲の特徴量として、メロディ・ハーモニー・リズム・歌詞・使用している楽器など約400種類を用いるが、それらの情報は音楽の専門家により付与される。

一方Last.fmも、ユーザの嗜好に合わせたラジオ局の開設が可能なシステムである。Last.fmでは、audioscrobbler[69]によって取得・管理されるユーザの日常的な楽曲の鑑賞履歴を基にユーザの嗜好を形成する。PANDORAとは異なり、アーティスト・タイトル・アルバムなど楽曲の基本情報のみを楽曲の特徴量としている。

これらのシステムはSNSと連携し、友人や嗜好の類似するユーザのラジオを楽しむことができる。協調フィルタリングを用いて嗜好の類似するユーザを提示する。嗜好の類似するユーザのラジオを鑑賞することで、自分の嗜好に合った楽曲を多く推薦されることが可能となる。

これらのシステムでは楽曲そのものを推薦することができるが、ユーザ自身が明確に他のユーザへの推薦の意図を伝えることができないという問題点がある。一方プレイリストは、それを作成するユーザが容易に意図を込めることができるメディアであり、プレイリストを推薦しあうことは、それぞれの楽曲へのリンク情報のみを保持していればよいから、楽曲そのものをやり取りする必要が無い。また楽曲そのものの推薦よりも効率が良いため、楽曲推薦の仕組みとしてプレイリ

ストは有効であることがわかる。

プレイリストを共有するサービスの代表的なものに、iMix が挙げられる。iMix では iTunes 上で作成されるプレイリストを、ユーザ間で共有し、評価しあうことができる。しかしプレイリストを作成するための支援が無く、iTunes Store 上に存在する膨大な楽曲の中からプレイリストの要素となる楽曲を選出することが困難である。一方、本研究で提案したプレイリスト作成支援システムでは、ユーザの嗜好やユーザの状況に合わせて楽曲を自動的に絞り込む機能を持っているため、プレイリストに含める楽曲を選出する作業を容易に行うことができる。

またプレイリスト作成支援システムでは、共有されたプレイリストに対する解説やコメントを記述することができ、またプレイリストを派生させて新たなプレイリストを容易に作成することができる。このように新しいプレイリストを作るモチベーションにつなげるための仕組みを備えているため、多くのプレイリストの作成と推薦が行われると考えられる。

第7章 結論

本章では、本論文で紹介した研究成果のまとめを述べ、今後の課題と将来の展望について述べる。

7.1 まとめ

本論文では、計算機がコンテンツの意味を把握し、様々な応用が実現可能になることを最終的な目標とし、特にその研究対象として個人の嗜好と状況への適応の要求が特に高いコンテンツであり、様々なコンテンツの意味内容を同時に扱う必要性がある音楽に着目した。

2章では音楽情報処理の研究を発展させるために必要である任意のコンテンツの意味内容を同時に扱う仕組みとして、任意のデジタルコンテンツに対するアノテーションを管理するための基盤技術である Annphony を提案した。

任意のコンテンツの内部要素を指し示す ElementPointer を提案し、Annphony に適用した。またコンテンツの種類に依存しないアノテーション形式である RDF を拡張することにより、任意のコンテンツの構造や解釈・関連性を統一的に扱うことができるようになった。また Annphony はアノテーション定義の管理を行うことで、ますます多様化するコンテンツへのアノテーションを柔軟に定義することができる。Annphony により、1章で述べた既存のアノテーション形式の間の壁を取り払い、横断的にコンテンツの意味内容を扱う環境が実現された。

3章では、音楽に対する意味内容に関するアノテーションを、Web 上の多くのユーザから獲得する音楽アノテーションシステムを提案した。本システムでは、1:

書誌情報などの楽曲自体へのアノテーションエディタ (Tune Annotator), 2: 連続メディアに対するアノテーションエディタ (Timeline Annotator), 3: 楽曲の内部構造に対するアノテーションエディタ (Score Annotator) の3種類のエディタを使い分けることで、様々な楽曲のメディア形式と、取得するアノテーションの対象の粒度に応じたアノテーション収集を支援する。またアノテーション定義とアノテーションエディタを独立させることにより、様々なアプリケーションに適用可能なアノテーションを柔軟に収集することができる。

またMP3等の音響メディアに対しても、Tune Annotator と Timeline Annotator を用いて音響メディアの構造化を行い、音響メディアに対応する MusicXML を生成する機能を備える。MusicXML を生成することで、音響メディアに対しても Score Annotator による楽譜による音符や歌詞の一部など楽曲の詳細な要素へのアノテーション付与が可能になる。

また、音楽アノテーションシステムを用いることで、多様な解釈が顕在化されることを確認する実験を行い、楽曲の内部構造に関する解釈の多様性と各ユーザの個人性を、本システムで収集されるアノテーションを解析することによって浮き上がらせることが可能であることを確認した。

4章では音楽アノテーションシステムにより収集が容易になった楽曲アノテーションを用いたアプリケーションとして、楽曲検索システムと楽曲再構成システムを提案した。

まずそれぞれのシステムで利用するアノテーションを Annphony に登録し、音楽アノテーションシステムを通して楽曲の構成や意見などのアノテーションを収集した。楽曲検索システムではそれらの情報を用いて、楽曲の論理構造であるコード進行に基づく検索や、楽曲の構成に基づく検索結果の絞込みを行う機能を持つ。また検索システムを利用するユーザの解釈に類似するユーザが付与したアノテーションを優先的に用いることによって、楽曲の印象検索などをユーザの感性に適応させる。また楽曲再構成システムは、楽曲の論理構造であるサビやイントロなどの大まかな構成、楽器パート、歌詞などの情報を基に、楽曲を容易に再構成す

るシステムであり、楽曲検索システムと同様に、ユーザの解釈と類似するユーザが付与したアノテーションを優先的に利用し、ユーザの感性に適応させることもできる。

これらのシステムから、音楽アノテーションシステムを用いて獲得されたアノテーションを用いて、楽曲の論理構造やユーザの感性に適合させるアプリケーションが実現可能であることを示した。

5章では、プレイリストにまつわる作成・推薦・鑑賞といった活動を活発にすることで、プレイリストを介したコミュニケーション (Playlist-Mediated Communication) が実現され则认为、プレイリストの作成支援とプレイリストを中心としたコミュニケーションを行うシステムを提案した。

本システムは、ユーザの嗜好と状況に適応するプレイリストの作成支援を行うために、楽曲情景・鑑賞状況に関するアノテーションを定義し、音楽アノテーションシステムに適用した。プレイリストを作成する際には、既存のプレイリスト中から嗜好・状況の類似するプレイリストを選別し、さらにユーザに適応させるトランスコーディングを行いユーザに提示する。ユーザは提示されたプレイリストに対して嗜好・状況に適合しているかどうかをシステムにフィードバックすることで、プレイリストを徐々に洗練していく。またプレイリストに対してライナーノーツや感想を記述することで、プレイリストを中心としたコミュニケーションを促す。

本システムは、評価実験を通して、アノテーションがユーザの心的状態や置かれている状況、またユーザと楽曲間の関係を正確かつ動的に表すために有効であると確認された。同時に、協調フィルタリング・トランスコーディング・インタラクションの組み合わせがユーザの嗜好に合ったプレイリスト推薦に有効であると確認できた。本システムは、プレイリスト作成と推薦を促す主要な枠組みであるといえる。

これは、音楽アノテーションシステムにより収集されたアノテーションを用いて、個人の嗜好と状況に適合するアプリケーションを実現した例の一つである。

本研究により、計算機が横断的なコンテンツのアノテーションを統一的に扱うことで、音楽情報処理の研究を発展させること、また音楽以外のコンテンツへも適用されることを示した。計算機によるデジタルコンテンツの意味内容を把握し、様々な応用を実現可能な環境に一步近づいたといえる。

7.2 今後の課題

今後の課題として、2章で述べた Annphony を一般公開し、様々なコンテンツ管理システムに適用される環境を構築することが挙げられる。様々な種類のコンテンツに対するアノテーションを収集し、そのアノテーションを共有することで、コンテンツの種類に横断的なアプリケーションを実現することが可能となる。

また3.3節で述べた連続メディアの構造化支援のために、ビートトラッキングなどの採譜のための研究を取り入れ、自動採譜結果を人手で修正することができるように拡張する予定である。本機能により、連続メディアから離散的な構造を持ったメディアへの変換のためにかかる人的コストを大幅に軽減させることができると期待できる。

また、4章で述べた楽曲の検索・再構成システムの評価を行い、ユーザの感性に適合させるために必要なシステムの要素を検討することが挙げられる。また、5章で述べたプレイリスト作成支援システムを用いることによってプレイリストの作成やコミュニケーションが活性化することを確認するための評価実験を行う必要がある。

7.3 将来展望

音楽に限らない、異種コンテンツ同士を横断的に扱った具体的な応用を検討した[70]。

複数のコンテンツを同時に利用する際に最も問題となるのは、どの組み合わせでコンテンツを利用すればよいかという点である。様々な種類のコンテンツを組み合わせるためには、関連性の認められるコンテンツ同士を選出することが必要となる。そこで我々は、Nelson[63] もその重要性を指摘する引用情報に着目した。あるコンテンツ内で同時に引用される複数コンテンツ間には明らかに意味的な関連性を認めることができることから、引用・被引用情報が有効であると考えられる。またコンテンツの部分引用の情報は、コンテンツの意味的な構造把握に有用な情報となるだろう。

また近年ブログやビデオ投稿システムなど、消費者がコンテンツを作成し投稿する CGM(Consumer Generated Media) サービスが盛んになってきており、今後 Web 上のコンテンツの多くは CGM サービスによって管理されることになると予想される。しかし現在の CGM サービスはコンテンツの部分的な引用・被引用を行うために十分な機能を備えていない。

そこで、CGM サービスに Annphony を適用し、コンテンツの任意の箇所の部分引用を可能にし、その引用・被引用情報を扱う。具体的には引用先の CGM サービスとしてブログに、引用元の CGM サービスとしてビデオ共有システム Synvie[71] と、試作中（一般には未公開）の認知科学辞典 [72]Web 版に Annphony を適用し、部分引用を可能にする。またブログで同時引用されるビデオと辞典の項目を用いて、事例ベース辞典システムを実現することを検討した。

7.3.1 CGM サービスの拡張

今後 Web 上のコンテンツの多くは CGM サービスによって管理されることになると予想されるため、Annphony を CGM サービスに適用することで、様々なコンテンツ同士の引用と、その引用情報の共有を可能にする。

例えば、引用元の CGM サービスとしてビデオ共有システム Synvie とオンライン辞典システムである認知科学辞典 Web 版に Annphony を適用する。コンテンツの種類は CGM サービスによって異なるため、コンテンツ内部を指し示す Element-

Pointer の定義は CGM サービスごとに用意する．ビデオ共有システムではビデオの時間区分を指し示すためには，開始・終了時間が必要であるため，その旨を ElementPointer の定義に記述した．またオンライン辞典システムの各専門用語は，文書の言語構造を表現するための言語である GDA(Global Document Annotation)[8] 形式で保持されているため，単語・文節・文など，文書の論理構造に基づく部分を指し示すための定義を記述する．

また引用先となる CGM サービスとして，ブログを採用した．ブログシステムに Annphony を適用し，ブログにビデオや辞典などの他のコンテンツを部分引用した記事を投稿する際に，その部分引用の情報を Annphony に保存する．図 7.1 はブログにおいてビデオ・辞典を引用した記事を投稿した画面例である．図では，ある分野の詳細な解説として，参考となるビデオと辞典の一部を引用している．このように同時に引用されるビデオと辞典には，明らかな関連性を認めることができる．

一般的にユーザは，あるコンテンツと他の様々なコンテンツとの多項関係を完全に把握してはいないため，一人のユーザによって網羅的な関連性を記述することは困難である．そのため，一般ユーザが記述可能な同時引用の情報は二項，三項関係といった少数の関連性にとどまると考えられる．しかし CGM サービスを通じて同時引用の情報を広く収集することにより，コンテンツ間の多項関係に拡張することができる．

7.3.2 マッシュアップコンテンツ

関連する複数コンテンツを自動的に統合して新たなコンテンツを生成することをマッシュアップと言う．Web コンテンツを用いてマッシュアップコンテンツを自動生成する例として，オンライン辞典システムを拡張し，ビデオ用例付き辞典システムを構築する．本システムでは，各専門用語の項に，ブログにおいて同時引用された，用例の一つと見なされるビデオを提示することで，その専門用語の理解を深めることができる．

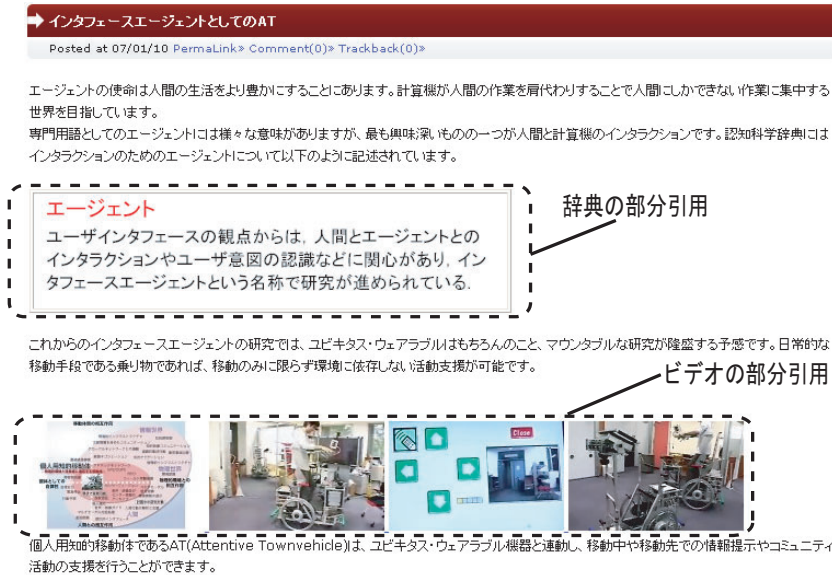


図 7.1: ブログでのビデオ・辞典の同時引用

キーワード検索の要求を本システムが受け取ると、オンライン辞典システム・ビデオ共有システムに対して、検索要求のキーワードが存在する辞典の項目・ビデオを検索する。次にそれらがブログにおいて過去に同時引用されているかを検索し、同時に引用されたことのあるビデオの一部と辞典項目の一部の組を列挙する。またブログに対してもキーワード検索を行い、検出されたブログエントリ内で同時引用されているビデオの一部と辞典項目の一部の組を列挙する。

こうして列挙されたビデオの一部と辞典の項目の一部の組を用い、辞典の項目中の同時引用された文以降に、ビデオへのリンクを埋め込む。図 7.2 はビデオが埋め込まれた辞典の項目の例である。辞典の項目中に具体的な例であるビデオを埋め込むことで、その項目をより深く理解することができる。

このように同時引用の情報を用いることにより、ビデオやオンライン辞典の項目といった異種類のコンテンツ同士を結びつけることで、付加価値を持ったコンテンツを作成することが可能である。これも本研究の成果から自然に導かれる結論の一つである。

ヒューマンインタフェース

エージェント

えーじえんと

agent

仮想的な主体によって人間の情報処理を代行させるシステム、およびそれを実現する技術。エージェントには行為者と代理人の2つの意味があり、自律的な情報処理システムという側面と、人間の命令を解釈して実行するインタフェースという側面がある。人間のエージェントとコンピュータプログラムとしてのエージェントを区別するために、機械のエージェントをソフトウェアエージェントと呼ぶこともある。

エージェントには様々な研究分野が関連しているが、それぞれの研究の目的や背景に応じて、追求すべき特性が異なる。例えば、人工知能においては、エージェントの自律的あるいは合理的行動に主な関心があり、自律的エージェントや合理的エージェントなどの名称で呼ばれる。ユーザインタフェースの観点からは、人間とエージェントとのインタラクションやユーザ意図の認識などに関心があり、インタフェースエージェントという名称で研究が進められている。



また、エージェントをデータや手続きなどがパッケージ化されたソフトウェアモジュールとして捉え、プログラムの実行環境がネットワーク上に分散されていても、それぞれの実行環境に移動しながら、継続した処理を行えるようにしたものとして、モバイルエージェントと呼ばれるものもある。エージェントは情報世界において自律的に行動する主体であり、物理世界において同様に行動するロボットと対比されることが多いが、現在では両者の区別はそれほど明確ではない。

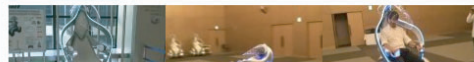


図 7.2: ビデオ用例付き辞典システム

謝辞

本論文は、名古屋大学大学院情報科学研究科 長尾研究室に在籍していた期間中に行われた研究です。本研究を遂行するにあたり、多くの方々の御指導、ご支援を頂きました。ここに心から感謝の意を表します。

長尾確教授には、自由な研究環境を提供して頂き、また日頃より懇切丁寧な御指導と御鞭撻を頂きました。さらに、研究者としての姿勢や考え方など、今後の研究活動を行っていく上でも有益となるような、多くの貴重なご意見を頂きました。

大平茂輝助手には、日頃の研究に関する様々な相談に常に親身になってご助言、ご支援を頂きました。

NTT コミュニケーション科学基礎研究所の平田圭二氏には、本研究の初期の段階より、研究方針や論文執筆に関する御指導を頂きました。

武田一哉教授には、本論文の審査の場を通じて、幅広い角度からの御助言、御示唆を頂き、本論文をまとめる上で有益なものとなりました。

長尾研究室秘書の金子幸子さんには学生生活ならびに研究活動のための様々なサポートを頂きました。また長尾研究室の皆様には本研究を進めるにあたり有益な議論と様々な励ましを頂きました。

本論文をまとめることができたのは、このような多くの方々の御支援のおかげです。ここに、改めて心から御礼申し上げます。

参考文献

- [1] Apple Computer, Inc. iTunes Store.
<http://www.apple.com/jp/itunes/music/>.
- [2] Amazon. <http://www.amazon.co.jp/>.
- [3] Yahoo! Music Unlimited. <http://music.yahoo.com/>.
- [4] Napster Japan. <http://www.napster.jp/>.
- [5] T. O'Reilly. What Is Web 2.0 -Design Patterns and Business Models for the Next Generation of Software-.
<http://www.oreillynet.com/pub/a/oreilly/tim/news/2005/09/30/what-is-web-20.html>, 2005.
- [6] K. Nagao. *Digital content annotation and transcoding*. Artech House Publishers, London, 2003.
- [7] T. Berners-Lee, J. Hendler, O. Lassila. The semantic web. *Scientific American*, 2001.
- [8] 橋田浩一. GDA:意味的修飾に基づく多用途の知的コンテンツ. 人工知能学会誌, Vol.13, No.4, pp. 528–535, 1998.
- [9] P. Salembier and J. R. Smith. MPEG-7 Multimedia Description Schemes. *IEEE Transactions on Circuits and Systems for Video Technology* Vol. 11, No. 6, 2001.

- [10] W3C. Resource Description Framework. <http://www.w3.org/RDF/>, 1999.
- [11] T. Berners-Lee, R. Fielding and L. Masinter. Uniform Resource Identifiers (URI): Generic Syntax. *RFC 2396*, 1998.
- [12] 山本大介, 長尾 確. 閲覧者によるオンラインビデオコンテンツへのアノテーションとその応用. 人工知能学会論文誌, Vol.20, No.1, pp. 67–75, 2005.
- [13] M. Kipp. Anvil - A Generic Annotation Tool for Multimodal Dialogue. In *Proceedings of the 7th European Conference on Speech Communication and Technology (Eurospeech)*, pp. 1367–1370, 2001.
- [14] I. Dagan, S. P. Engelson. Committee-Based Sampling For Training Probabilistic Classifiers. In *Proceedings of International Conference on Machine Learning*, pp. 150–157, 1995.
- [15] A. Mathes. Folksonomies - Cooperative Classification and Communication Through Shared Metadata. <http://www.adammathes.com/academic/computer-mediated-communication/folksonomies.html>, 2004.
- [16] del.icio.us. <http://del.icio.us/>.
- [17] Flickr. <http://www.flickr.com/>.
- [18] K. Marja-Riitta, P. Eric, R.S. Ralph. Annotea: An Open RDF Infrastructure for Shared Web Annotations. In *Proceedings of International World Wide Web Conference (WWW)*, 2001.
- [19] 坂本 竜基, 伊藤 禎宣. annpoly. <http://blog.phase8.jp/annpoly/>, 2005.
- [20] W3C. Synchronized Multimedia Integration Language (SMIL 2.1). <http://www.w3.org/TR/SMIL/>, 2005.

- [21] S. Boll, W. Klas, Jochen Wandel. *A Cross Media Adaptation Strategy for Multimedia Presentations*. ACM Multimedia No.1, 1999.
- [22] P. Bellini, P. Nesi. WEDELMUSIC format: An XML music notation format for emerging applications. In *Proceedings International Conference on WEB Delivering of Music (WEDELMUSIC)*, pp. 79–86, 2001.
- [23] M. Good. MusicXML in Practice: Issues in Translation and Analysis. In *Proceedings of International Conference Musical Application using XML*, pp. 47–54, 2002.
- [24] R. Jackendoff, F. Lerdahl. *A Generative Theory of Tonal Music*. MIT Press, 1996.
- [25] Gracenote. CDDb. http://www.gracenote.com/gn_products/cddb.html.
- [26] MusicBrainz. <http://www.musicbrainz.org/>.
- [27] MoodLogic. <http://www.moodlogic.com/>.
- [28] X. Amatriain, J. Massaguer, D. Garcia, I. Mosquera. The CLAM Annotator: A Cross-platform Audio Descriptors Editing Tool. In *Proceedings of International Conference on Music Information Retrieval (ISMIR)*, 2005.
- [29] P. Herrera, O. Celma, J. Massaguer, P. Cano, E. Gomez, F. Gouyon, M. Koppenberger, D. G. Garcia, J. Mahedero and N. Wack. MUCOSA: A Music Content Semantic Annotator. In *Proceedings of International Conference on Music Information Retrieval (ISMIR)*, 2005.
- [30] U. Shardanand, P. Maes. Social Information Filtering: Algorithms for Automating “Word of Mouth”. In *Proceedings of Conference on Human Factors in Computing Systems (CHI)*, 1995.

- [31] A. Field, P. Hartei, W. Mooij. Personal DJ, an Architecture for Personalised Content Derivery. In *Proceedings of World Wide Web Conference (WWW)*, 2001.
- [32] M. Anderson, M. Ball, H. Boley, S. Greene, N. Howse, D. Lemire, and S. McGrath. Racofi: A rule-applying collaborative filtering system. In *Proceedings of Center for Ocean-Land-Atmosphere Studies*. IEEE/WIC, October 2003.
- [33] B. Logan. Content-Based Playlist Generation: Exploratory Experiments. In *Proceedings of International Conference on Music Information Retrieval (ISMIR)*, 2002.
- [34] M. Alghoniemy, A. H. Tewfik. A Network Flow Model For Playlist Generation. In *Proceedings of IEEE International Conference on Multimedia and Expo (ICME)*, 2001.
- [35] J. J. Aucouturier, F. Pachet. Scaling Up Music Playlist Generation. In *Proceedings of IEEE International Conference on Multimedia and Expo (ICME)*, 2002.
- [36] W3C. RDF Schema. <http://www.w3.org/TR/rdf-schema/>, 2004.
- [37] MULTEXT Project. Corpus Encoding Standard. <http://www.cs.vassar.edu/CES/>.
- [38] W3C. XML Pointer Language. <http://www.w3.org/TR/WD-xptr>, 2001.
- [39] S. Pfeiffer, C. Parker, A. Pang. Specifying time intervals in URI queries and fragments of time-based Web resources. http://www.annodex.net/TR/URI_fragments.txt, 2003.
- [40] Protégé. <http://protege.stanford.edu/>.

- [41] G. Sunil, W. Rupert. RDF Gravity (RDF Graph Visualization Tool).
<http://semweb.salzburgresearch.at/apps/rdf-gravity/index.html>.
- [42] W3C. XML Path Language. <http://www.w3.org/TR/xpath.html>, 1999.
- [43] W3C. SPARQL Query Language for RDF.
<http://www.w3.org/TR/rdf-sparql-query/>, 2006.
- [44] W3C. Named Graphs. <http://www.w3.org/2004/03/trix/>, 2004.
- [45] W3C. XML Schema. <http://www.w3.org/XML/Schema>, 2001.
- [46] Userland Software. XML-RPC. <http://www.xmlrpc.com/>, 1999.
- [47] HP. Jena - A Semantic Web Framework for Java.
<http://jena.sourceforge.net/>, 2006.
- [48] SC34/WG3. Topic Maps. *ISO/IEC 13240*, 2002.
- [49] 後藤 真孝, 村岡 洋一. 音楽音響信号を対象としたビートトラッキングシステム-小節線の検出と打楽器音の有無に応じた音楽的知識の選択-. 情報処理学会音楽情報科学研究会 研究報告, 97-MUS-21-8, Vol.97, No.67, 1997.
- [50] K. Hevner. Experimental Studies of the Elements of Expression in Music. *American journal of psychology*, Vol. 48, pp. 246–268, 1936.
- [51] 後藤 真孝. SmartMusicKIOSK: サビ出し機能付き音楽試聴機. 情報処理学会論文誌, Vol.44, No.11, pp. 2737–2747, 2003.
- [52] W3C. Scalable Vector Graphics(SVG). <http://www.w3.org/Graphics/SVG/>, 2003.
- [53] S. Pauws, B. Eggen. PATS: Realization and user evaluation of an automatic playlist generator. In *Proceedings of International Conference on Music Information Retrieval (ISMIR)*, October 2002.

- [54] J. H. Lee, J. S. Downie. Survey of Music Information Needs, Uses, and Seeking Behaviours: Preliminary Findings. In *Proceedings of International Conference on Music Information Retrieval (ISMIR)*, 2004.
- [55] E. Daniel, B. Whitman, A. Berenzweig, S. Lawrence. The Quest for Ground Truth in Musical Artist Similarity. In *Proceedings of International Conference on Music Information Retrieval (ISMIR)*, 2002.
- [56] B. Logan, A. Kositsky, P. Moreno. Semantic Analysis of Song Lyrics. In *Proceedings of IEEE International Conference on Multimedia and Expo (ICME)*, 2004.
- [57] G. Salton, C. S. Yang. On the specification of term values in automatic indexing. *Journal of Documentation*, pp. 29:351–372, 1973.
- [58] 後藤 真孝, 橋口 博樹, 西村 拓一, 岡 隆一. RWC 研究用音楽データベース: ポピュラー音楽データベースと著作権切れ音楽データベース. 情報処理学会 音楽情報科学研究会 研究報告 2001-MUS-42-6, pp. 35–42, 2001.
- [59] S. Staab, A. Maedche, S. Handschuh. An Annotation Framework for the Semantic Web. In *Proceedings of the First Workshop on Multimedia Annotation*, 2001.
- [60] D. Barger, A. Gupta, A.J. B. Brush. A Common Annotation Framework. *Microsoft Research MSR-TR-2001-108*, 2001.
- [61] H. Vinet, P. Herrera, F. F. Pachet. The CUIDADO Project. In *Proceedings of International Conference on Music Information Retrieval (ISMIR)*, 2002.
- [62] D. A. Norman. *Things That Make Us Smart*. Perseus Books, 1993.
- [63] T. Nelson. *Computer Lib/Dream Machines*. マイクロソフト出版, 1987.

- [64] D. Goodman. *The Complete HyperCard Handbook*. Bantam Books, 1987.
- [65] 田中 譲. ミームメディア・アーキテクチャIntelligentPad とその応用. 情報処理学会論文誌, Vol.38, No.3, pp. 222–231, 1997.
- [66] Last.fm. <http://jp.last.fm/>.
- [67] PANDORA. <http://www.pandora.com/>.
- [68] Apple Computer, Inc. iMix. <http://www.apple.com/jp/itunes/playlists/>.
- [69] audioscrobbler. <http://www.audioscrobbler.net/>.
- [70] 梶克彦, 長尾確. 部分引用の管理に基づく Web コンテンツのマッシュアップ. 情報処理学会第 69 回全国大会, 2007.
- [71] 山本大介, 清水敏之, 大平茂輝, 長尾確. Synvie: ブログの仕組みを利用したマルチメディアコンテンツ配信システム. 情報処理学会第 68 回全国大会, 2006.
- [72] 日本認知科学会. 認知科学辞典. 共立出版, 2002.

研究業績

学会誌等

- 梶克彦, 長尾確. 楽曲に対する多様な解釈を扱う音楽アノテーションシステム, 情報処理学会論文誌, Vol.48, No.1, pp. xx-xx, 2007.
- 梶克彦, 長尾確. Annphony:メタコンテンツ処理のためのプラットフォーム, 情報科学技術レターズ, Vol.5, pp. 381-384, 2006.
- 梶克彦, 平田圭二, 長尾確. コミュニケーションメディアとしてのプレイリストを目指して, 情報科学技術レターズ, Vol.4, pp. 115-118, 2005.

国際会議

- Katsuhiko Kaji, Keiji Hirata, Katashi Nagao. A Music Recommendation System Based on Annotations about Listeners' Preferences and Situations, In Proceedings of Automated Production of Cross Media Content for Multi-channel Distribution (AXMEDIS), 2005.
- Katsuhiko Kaji, Katashi Nagao. MiXA: Musical Annotation System, In Proceedings of International Semantic Web Conference(ISWC), 2004.
- Katashi Nagao, Katsuhiko Kaji, Daisuke Yamamoto, Hironori Tomobe. Discussion Mining:Annotation-Based Knowledge Discovery from Real World

Activities, In Proceedings of The Fifth Pacific-Rim Conference on Multimedia (PCM), 2004.

研究会

- 梶克彦, 長尾確. 任意のデジタルコンテンツに対するアノテーションプラットフォーム電子情報通信学会 Web インテリジェンスとインタラクション研究会, WI2-2006-27, 2006.
- 梶克彦, 平田圭二, 長尾確. 状況と嗜好に関するアノテーションに基づくオンライン楽曲推薦システム, 情報処理学会 音楽情報科学研究会, Vol.2004, No.127, pp. 33-38, 2004.

その他の発表

- 梶克彦, 長尾確. Annphony:メタコンテンツ処理のためのプラットフォーム, 第5回情報科学技術フォーラム (FIT), 2006.
- 梶克彦, 長尾確. 任意のデジタルコンテンツに対するアノテーションプラットフォーム, 第68回情報処理学会全国大会, 2006.
- 梶克彦, 平田圭二, 長尾確. コミュニケーションメディアとしてのプレイリストを目指して, 第4回情報科学技術フォーラム (FIT), 2005.
- 梶克彦, 平田圭二, 長尾確. 状況と嗜好に関するアノテーションを利用したプレイリスト生成システム, 第67回情報処理学会全国大会, 2005.
- 梶克彦, 長尾確. 音楽におけるアノテーションとその応用, 第3回情報科学技術フォーラム (FIT), 2004.

- 梶克彦, 長尾確. 音楽におけるアノテーションとその応用, 第 66 回情報処理学会全国大会, 2004.
- 梶克彦, 長尾確. 音楽におけるアノテーションとそれに基づく検索・要約, 第 65 回情報処理学会全国大会, 2003.

受賞

第 4 回情報科学技術フォーラム (FIT) においてヤングリサーチャー賞受賞