

## メタデータ制作フレームワーク (Ver.2) の活用と今後

佐野 雅規<sup>†1</sup> 住吉 英樹<sup>†1</sup> 藤井 真人<sup>†1</sup>  
柴田 正啓<sup>†1</sup> 八木 伸行<sup>†1</sup>

本稿では、映像コンテンツのシーン毎に、内容に沿ったメタデータを効率よく付与するためのフレームワークについて紹介する。このフレームワークは、さまざまな情報処理技術を自由に組み合わせて、メタデータ制作を協調的かつ効率的に進めることのできる共通基盤として利用されることを目的としている。また、その設計にあたり、学術分野だけでなく、産業分野にも受け入れやすい枠組みを有するよう配慮をしている。はじめに、本フレームワークの位置づけと、規定項目について説明する。次に、機能検証用に提供しているリファレンスソフトウェアとその利用方法について述べ、最後にこれまでの活動報告と今後の予定について述べる。

## Utilization of Metadata Production Framework (Ver.2) and Its Future Plan

MASANORI SANO,<sup>†1</sup> HIDEKI SUMIYOSHI,<sup>†1</sup>  
MAHITO FUJII,<sup>†1</sup> MASAHIRO SHIBATA<sup>†1</sup>  
and NOBUYUKI YAGI <sup>†1</sup>

In this paper, we describe the latest specifications of the Metadata Production Framework (MPF) which aims to provide common platform for generating content-based metadata for TV programs. In order to effective and collaborative production of the metadata, MPF provides mechanism of combining various kinds of media analysis processes. The mechanism was also designed with benefiting not only academic field but also industrial field. We describe the position MPF is supposed to work and its specifications firstly, and then mention the reference software by which the basic functionality of MPF can be tested over network. Finally the activities done so far and its future works are mentioned.

### 1. はじめに

メタデータ制作フレームワーク (MPF: Metadata Production Framework) は、番組などの映像コンテンツに対し、できるだけ効率よく内容に沿ったメタデータを付与するための枠組みである。基本的なアイデアは、さまざまなメディア処理を組合せることで、できるだけ自動で、かつ精度良いメタデータの生成をめざすものであり、その初版を2006年5月に一般に公開<sup>1)</sup>した。特徴は、複数の機関で独立に研究開発されたメディア処理技術であっても、容易に組み合わせて作業を進められるよう、メタデータ制作フローを整理し、必要なインターフェースを規定したことである。MPFの設計にあたっては、将来の放送局におけるメタデータ制作の全体像を想定しつつ、新しいメタデータ市場の開拓と活性化も視野に入れた。従って、メタデータを制作する放送局や映像コンテンツ提供事業はもちろんのこと、メディア解析技術を提供するメーカーや大学にも、産業的または学術的なメリットがあるよう考慮している。これまでに、さまざまな実証実験を通し、その機能などの検証をしつつ仕様の改良を続けており、現在はバージョン2.02を公開している。更に、MPFの基本機能を検証するために提供しているリファレンスソフトウェアは、メディア解析処理の研究にも役立つよう設計してある。今後は、MPFを基盤として多くの研究者の方々と連携し、効率よいメタデータ制作の環境を実現していきたいと考えている。そこで本稿では、この最新のMPF仕様の骨子を中心に、リファレンスソフトウェアの機能とその利用シナリオ、今後の予定などについて紹介する。

### 2. メタデータ制作フレームワーク (MPF)

#### 2.1 MPFの位置づけと対象メタデータ

図1は、MPFを設計するにあたって想定した、放送局における将来のメタデータ制作フローを示したものである。図に示すように2段階の制作体系を想定し、第一段階 (図左側) では共通に利用できるメタデータの作成、第二段階 (図右側) では、それらを基に各々のアプリケーションに特化したメタデータを作成する。前段におけるメタデータを素材メタデータ、もしくはプリミティブメタデータと呼び、後者をアプリケーションメタデータと呼ぶ。放送局において必要となるアプリケーションメタデータは多種多様であり、その用途毎

<sup>†1</sup> NHK 放送技術研究所  
NHK Science and Technology Research Laboratories

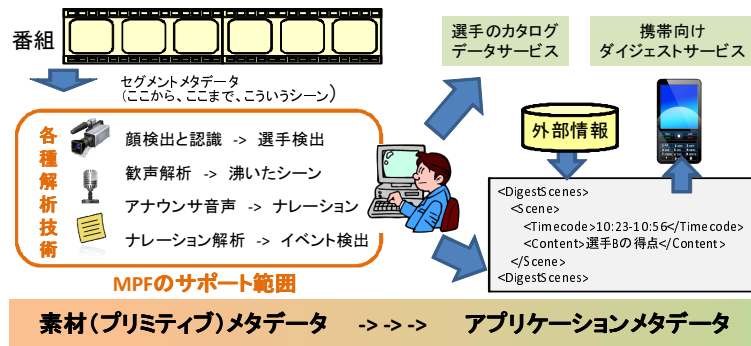


図1 メタデータの制作フロー  
 Fig.1 Metadata production flow

に種類やフォーマット、また内容が異なると考えられる。このようなアプリケーションメタデータを制作する場合に、それぞれを制作する個別のシステムを開発することは膨大なコストを必要とし、逆に1つのシステムとして実現することは、システムの過度の複雑化を招く結果となる。そこで、共通の素材メタデータを作る段階と、これを基にアプリケーションメタデータを制作するという2段階に分けることで、トータルのコストの削減を図る。そして、この制作体系を効率よく稼働させるためには、前段の素材メタデータを、いかにコストをかけず精度良く生成できるかが鍵となる。本稿で紹介するMPFは、この素材メタデータの確度の高い自動生成を実現するための環境を提供することを目的としている。

次に、MPFが生成目的とする素材メタデータについて述べる。一般に、番組に関連するメタデータは、その記述の対象範囲によって2種類に分けることができる。1つは、番組全体にかかるタイトルや制作者など書誌情報にあたるもので、プロダクションメタデータと呼ぶ。これらは通常人手により付与されるものであり、放送局においては番組管理のために付与され、データベースも整っている。これに対し、番組のある区間に対して付与するメタデータをセグメントメタデータと呼ぶ。こちらは付与されている番組もあれば、全く付与されていない番組もある。このセグメントメタデータとして付与される情報には、さまざまな種類が存在し、その付与手法は異なる。例えば、ある区間映像に対する使用方法や著作権などの情報はこの1つであり、これらは人手による入力が必要である。他の情報としては、映像の構図や色合いなどの低次特徴が考えられる。これらはある程度映像解析により自動で

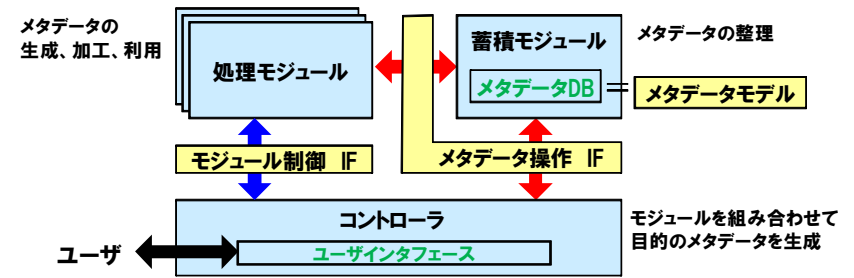


図2 MPFのシステムモデル  
 Fig.2 System model of MPF

抽出することができる。更には、映像中のイベントやシーンの意味などの高次特徴があり、これらは検索などのユーザに最も有用なメタデータであるが、自動抽出は難しく、現在ほとんど付与されていない状況である。MPFでは、この一番最後に挙げた、最も必要であるが、現在あまり付与できていないセグメントメタデータを主な生成対象とし、さまざまなメディア処理を組み合わせることでその自動生成を行う環境を提供する。

## 2.2 システムモデル

図2は、MPFにおけるメタデータ制作システムのモデルである。モジュール群とそれらを制御するためのコントローラの2種類で構成する。モジュールは更に2種類あり、1つは蓄積モジュールと呼ばれ、メタデータのためのデータベース（メタデータDB）を内包し、生成したメタデータを蓄積して管理し、コントローラもしくは他のモジュールからの要求に応じて、メタデータDBを検索あるいは更新する。もう1つのモジュールは処理モジュールと呼ばれ、MPFにおいて最も重要な位置づけにあり、メタデータ制作に関わるさまざまな処理機能を提供する。この処理には、メタデータの生成、加工、利用など、メタデータのライフサイクルに絡んだ全ての操作が含まれる。コントローラは、ユーザとシステム間の仲介を行い、目的のメタデータを生成するために、必要なモジュール群を制御するものである。更に、処理モジュールとのやり取り以外には、生成したメタデータの保管や活用のために、蓄積モジュールとのやり取りも発生する。

このようにMPFは、メタデータ制作に関わるさまざまな処理を、統一されたインタフェースを持つモジュールとして作成しておき、それらを組み合わせることで、目的のメタデータを制作していくというコンセプトに基づいている。従って、メタデータの制作を連携して行う方法として、作成したメタデータを交換するだけでなく、個々の処理自体も交換するこ

とが可能である。そして、このシステムモデルを支障なく機能させるためには、次に述べるメタデータの表現（メタデータモデル）と、モジュールやコントローラの間でのデータのやり取り（インタフェース IF）について規定する必要がある。

### 2.3 MPF 仕様の規定項目

#### 2.3.1 メタデータモデル

MPF のメタデータモデルには、メタデータの国際標準である MPEG-7<sup>2)</sup> のサブセットを採用した。従って、MPF のメタデータは XML 形式で表現し、その構造などは XML スキーマにより定義した。ただし、一部スキーマでは表現できない制約も課しており、それらは文章により規定した。MPEG-7 からのサブセットの選択方法については、当初より、番組を記述するための最小限のものにとどめ、実証実験を進めていく過程で、必要な場合には順次拡張するという方針で進めてきた。このため MPF バージョン 1 では、映像や音の低次特徴は対象外とし、基本的にテキストで記述するメタデータのみを対象とした。現在の MPF バージョン 2 では、映像や音の低次特徴を含め、MPEG-7 で規定されていないメタデータについても、外部ファイルに保存し、そこへのポインターを保持するという形で拡張している。

図 3 は、MPF のメタデータモデルと、システムモデルの核である処理モジュールの動作をあわせて示したものである。MPF では、単一の映像コンテンツ（番組）を対象としており、図中央がそのメタデータの構造を示している。先に説明したプロダクションメタデータは「基本情報」に、また多種多様なセグメントメタデータは「セグメントユニット」として、その生成方法により「セグメントブロック」としてグループ化する。1つの番組メタデータの中に、セグメントブロックはいくつでも生成することができ、そのセグメントブロック内にも必要なだけセグメントユニットを生成することができる。作成方法の例としては、ショット区間や発話区間などをあげることができ、これらは処理モジュールの出力結果に相当する。

ここで処理モジュールを用いたセグメントメタデータの生成について説明する。サッカーの中継番組を解析対象の例とする。図 3 左側にある抽出モジュール A は、番組の音響を解析して「盛り上がった区間」を検出し、その検出区間をセグメントブロック A にまとめて格納する。同様に、処理モジュール B は、アナウンサの発話を認識し、1文1文をセグメントユニットとして、セグメントブロック B に格納する。最後に、図右側の処理モジュール C が、これら 2つのモジュールの結果を用い、各々の盛り上がった区間に前後 10 秒を加えた新しい区間を作成し、そこに、その時刻に含まれるアナウンサの発話内容を言語解析して抽出した発生イベント（例：シュート）を付与する。このシナリオは、3つの処理モジュール

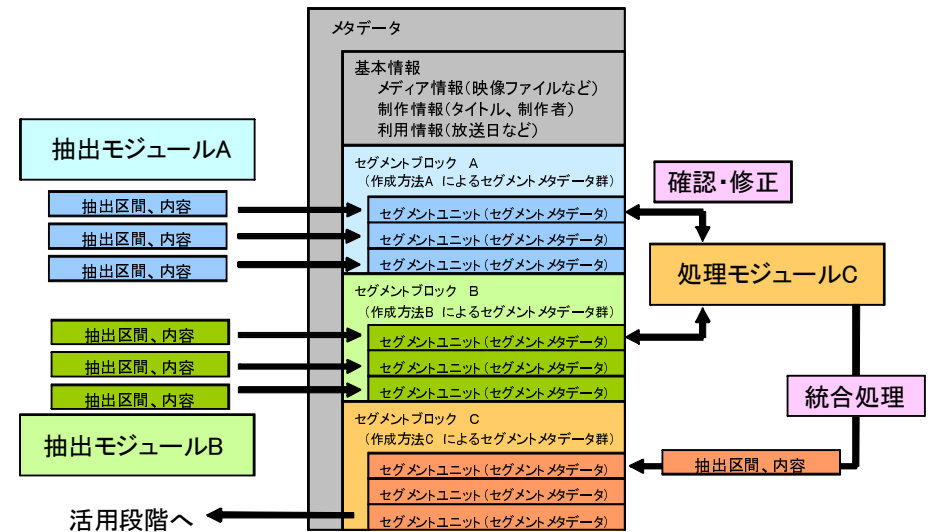


図 3 MPF のメタデータモデル  
 Fig. 3 Metadata model of MPF

を組合せた一連の処理で、ダイジェスト作成のためのメタデータを生成する例である。このようにセグメントブロックを単位としたモジュール処理を組合せ、また他のモジュールの結果を再利用しながら、目的のメタデータを生成する。

#### 2.3.2 インタフェース

MPF では、生成したメタデータを送受するためのメタデータ操作インタフェースと、モジュールを制御するためのモジュール制御インタフェースの 2 種類を規定している。どちらも WebServices による実装を基本としているが、モジュール制御インタフェースについては、処理モジュールの開発を容易にする観点から Windows DLL による実装を許している。以下に、それぞれの概要を説明するが、詳細はリファレンスを参照されたい。

##### (1) メタデータ操作インタフェース

メタデータ蓄積モジュールに蓄えられるメタデータの操作を行うためのインタフェースである。使用頻度とシステム実装を簡便にする観点から、高レベルと低レベルの 2 種類のインタフェースを定義しており、全ての蓄積モジュールは高レベルインタフェースの実装を必須としている。高レベルインタフェースは、先に述べたメタデータモデルの中の 5つの構造

要素を操作単位とするものであり、それらは番組全体、セグメントブロック、セグメントユニットの基本的な3要素に加え、番組全体の基本情報と、番組全体やセグメントブロックのヘッダ情報である。また、ネットワークを介した複数のプロセスによる1つのメタデータの更新を可能とするため、書き込み操作権限によるメタデータ操作を用意している。一方、低レベルインタフェースとしては、XMLで記述されるメタデータに自由にアクセス可能なクエリ規格として、W3CのXQuery<sup>3)</sup>を採用する。また、XML要素の更新などについては、現在勧告案であるXQuery Update Facilityの採用を予定している。

## (2) モジュール制御インタフェース

モジュール制御インタフェースは、処理モジュールに必ず実装するものであり、MPFでは以下の4種類の関数群のみを規定し、シンプルな構成としている。

- MPFInitializeModule( ) モジュールの初期化
- MPFGetModuleInfo( ) モジュールのプロパティと処理に必要なパラメータの取得
- MPFSetModuleInfo( ) モジュールが処理に必要とするパラメータの設定
- MPFDoProcess( ) モジュールの動作制御(開始, 停止)

コントローラは、これらのインタフェースを介し必要なモジュールを動作させて、目的のメタデータを生成する。処理モジュールを動作させるための一般的な手順は次のようになる。はじめにMPFGetModuleInfo関数により、モジュールに関連する情報を取得する。ここには、モジュールの処理内容や、制作者、バージョンなどのプロパティ情報とともに、実際にモジュールが処理を行う上で必要となるパラメータのリストが記述されている。従って、プロパティ情報はモジュールの選択時に利用し、パラメータ情報はモジュール選択後に利用することになる。利用するモジュールが決定したら、MPFInitializeModule関数で、そのモジュールの初期化を行う。続いて、先に取得したパラメータリストに必要なパラメータを埋め込み、MPFSetModuleInfo関数を用いてその情報をモジュールに渡す。最後に、MPFDoProcess関数を通して、モジュールの動作を開始する。以上が一連の操作であり、モジュールを制御するコントローラには、これらのやり取りを実装しなければならない。

## 3. リファレンスソフトウェア

### 3.1 構成

MPFの仕様書はWebページ<sup>1)</sup>よりダウンロードできるが、この中には関連するリファレンスソフトウェアとそのマニュアルが含まれている。これらは、MPFの基本動作を確認するために無償で配布しているものであり、商用でなければ自由に利用することができる。

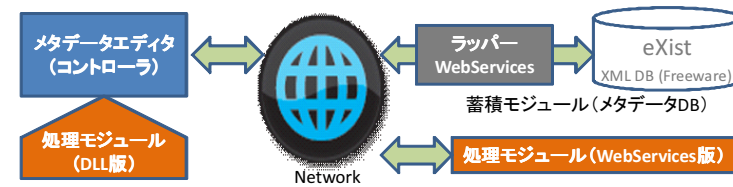


図4 リファレンスソフトウェアによる実験環境

Fig. 4 Experimental environment based on reference software

提供しているソフトウェアは3種類あり、コントローラの機能を有するメタデータエディタ、蓄積モジュールを構築するためのラッパープログラム、そしてサンプルとしての各種処理モジュールである。図4は、これらリファレンスソフトウェアによる実験環境を示したものである。ソフトウェアは必要なものを組合せて使用することができ、その組合せによりテストできるMPFの機能が異なる。以下では、主な組合せ5つについて説明する。

(1) メタデータエディタのみ： PC1台を想定した、一番コンパクトなメタデータ生成システムである。ただし、セグメントメタデータの付与はマニュアル操作であり、生成したメタデータは外部ファイルとしてしか保存できない。

(2) メタデータエディタ+処理モジュール(DLL版)： 2番目にコンパクトな構成であり、ローカルのPC1台で動作する。1の作業に加え、開発した処理モジュール(DLL版)による自動処理のテストを行うことができる。

(3) メタデータエディタ+処理モジュール(DLL版)+蓄積モジュール： 2の機能に加え、作成したメタデータの蓄積場所としてデータベースを用いる構成である。この蓄積モジュールは、ローカルPCに構築して利用することも、ネットワーク経由で別のPCに構築して利用することも可能となっている。

(4) メタデータエディタ+処理モジュール(DLL版+WebServices版)： 2の機能に加え、ネットワーク上に提供された処理モジュールの利用テストを行う構成である。ネットワーク経由の処理の場合には、解析データそのものの授受を行う必要があり、映像や音声などデータ量が大きい場合はかなり処理時間がかかり、現時点ではあまり実用的とはいえない。しかし、言語処理など、テキストデータを送って、そのサマリを返却するようなモジュールは実際の用途として実現性があるといえる。

(5) メタデータエディタ+処理モジュール(DLL版+WebServices版)+蓄積モジュール： MPFの機能をテストする最大の構成である。ローカルにある処理モジュールをはじ



め、ネットワーク上の処理モジュール、蓄積モジュールなど、さまざまな機能を利用しながらメタデータを作成することができる。

以降では、個々のソフトウェアについて簡単に説明する。なお、MPF の仕様書を含め、これらソフトウェアとそのドキュメントは、全て日本語版と英語版を用意している。

### 3.2 メタデータエディタ

主にコントローラの役目を果たし、このエディタ上で、さまざまな処理モジュールを選択して組合せ動作させることができる。図 5 が、エディタの操作画面である。画面は大きく 3 部構成となっており、左上が選択されたセグメントメタデータの情報表示部、右上が映像コンテンツ表示部、下半分がセグメントブロックとそれに含まれるセグメントユニットの表示部となっている。基本的な操作は、はじめに映像コンテンツを読み込み、次に画面下半分の部分で、トラック（セグメントブロック）を作成する。このトラックに対して、処理モジュールを 1 つ割り当てることが可能となっている。モジュールを割り当てると、そのモジュールが必要とするパラメータの入力を求められるので、それらを設定して処理の開始を指示すると、トラック内に抽出されたセグメントユニットの出力が描画されていく。別のモジュールを動作させる場合には、別のトラックを作成し、同様な操作を行う。トラック上にあるセグメントユニットをクリックすると、その内容がエディタ画面左上に表示される。ここには選択されたセグメントメタデータの内容が、MPEG-7 に準拠した XML の木構造で表現されており、直接編集が可能となっているので、必要があれば手動で修正する。最終的なメタデータは、外部ファイルとして保存してもよいし、蓄積モジュールが存在するのであれば、そこに保存しても良い。以上が一連の操作の概略である。

次に、メディア解析処理の研究に役立てるためのヒントについて述べる。まず、複数のトラックを使用することで、モジュール化された処理の結果を容易に比較することができる。例えば、同じ処理モジュールを、違ったパラメータで指定し、それぞれのトラックで動作させると、パラメータによる振る舞いの違いが一目瞭然となる。また、同じ目的のモジュール（例えば顔検出）を、異なるアルゴリズムで実装し、それらを 1 つ 1 つのトラックで動作させれば、各アルゴリズムの特性を容易に比較することができる。更に、本エディタ上では、モジュールをカスケードする（他のモジュールの出力結果を入力とする）ことが可能となっている。従って、顔検出の正解データを人手で作成しておけば、このリファレンスデータと選択したトラックの結果データを比較し、正答率などを数値化して外部ファイルに出力するような評価モジュールも開発することができる。一度モジュールを作成してしまえば、異なるアルゴリズムの精度を、パラメータ毎に評価することも容易となる。このほか、グラフを



図 5 メタデータエディタ  
Fig.5 Metadata editor

表示する特別なトラックも実装されており、決められたフォーマットの時刻情報付き数値列を、選択した色や形式で描画することができる。このグラフデータは、処理モジュールが処理を行う際に、外部ファイルとして出力することを想定しており、この機能により、メディア解析処理に関連する時系列データを、視覚的にもわかりやすく確認することができる。以上のように、メディア解析研究者にも配慮された機能が多く実装されている。

### 3.3 蓄積モジュールのためのラッパー

MPF システムモデルにおける蓄積モジュールを構築するためのプログラムである。データベースそのものには、フリーの XML データベースである eXist<sup>4)</sup> を利用する。そしてこのラッパープログラムは、このデータベースに 2.3.2 で定義したメタデータ操作インターフェースを実装したものである。これは WebServices による実装であり、Windows の IIS (Internet Information Server) を対象としたものとなっている。従って、WindowsOS に、eXist, IIS, 本ラッパープログラムをインストールすることで、ネットワークを介してどこからでも利用可能な蓄積モジュールが構築できる。

### 3.4 処理モジュール

MPF の処理モジュールとして、画像、音、その他の処理を行う 5 種類のサンプルプログラムをソースコード付きで提供している。これらには DLL 実装と WebServices 実装の両方のソースコードを含めた。更に、モジュール開発者のために、プログラムの骨格だけを有するスケルトンモジュールもソース付きで含めている。このソースには、プログラム作成のための指示が付与されており、指示に沿って自身の研究開発したメディア処理を埋め込めば、容易に MPF 仕様の処理モジュールを作成することができ、それらはメタデータエディタからすぐ実行してテストすることができる。

## 4. これまでの活動と今後の予定

MPF は、2006 年 5 月の公開後、プロジェクト参加や展示などの機会を通し、その有用性をアピールしてきた。例えば、2007 年度からの情報大航海プロジェクト<sup>5)</sup>では、映像の意味理解のための共通基盤として提案し、多くのメディア処理をモジュール化した。これらは情報大航海のホームページ<sup>5)</sup>の「CP 共通技術ページ」からダウンロードすることができる。また、EBU (European Broadcasting Union) には、自動コンテンツ解析技術 (音声認識、顔画像認識、テキスト解析等) に基づくメタデータ付与技術の評価を目的とする P/SCAIE (ピー・スカイ)<sup>6)</sup> プロジェクトがある。ここでは、現在評価のための共通のメタデータフォーマットを決めており、そのベースとなる仕様の 1 つとして MPF の検討を進めている。最後に、現在予定している仕様の拡張について以下に述べる。

### (1) 内容記述対象の空間的分割への拡張

現在の内容記述の対象領域は、コンテンツを時分割したものだけである。最近では、映像や画像の空間的一部分を特定し、そこへの情報付与を行うというシナリオも多く、MPF 仕様の拡張の要望がある。MPEG-7 には、元々この機能が含まれているので、それを含む仕様に変更し、さまざまな関連する制約について検討を行う予定である。

### (2) リンクメタデータ付与への拡張

リンクメタデータとは、コンテンツのある部分と、別のある部分の関係性を記述するメタデータである。単一のコンテンツ内だけでなく、複数のコンテンツ間に付与する場合もある。最近の情報抽出の研究分野では、大量のコンテンツを対象として、このようなさまざまな関係を抽出することが盛んに行われている。この分野においても MPF を利用したいという要望があり、仕様の拡張を検討している。現バージョンの MPF は単一の映像だけを対象としているため、これを複数対象とできるように拡張し、さらにリンクメタデータを効率よ

く扱うことができるような仕様の変更を考えている。

### (3) リファレンスソフトウェアの改修

上述した機能拡張に対応するためのソフトウェア改修に加え、次の 2 点の改修を考えている。1 つめは、メタデータエディタ上での、セグメントメタデータを編集する部分の GUI の改修である。現在のメタデータエディタでは、画面左上の木構造で表現された MPEG-7 準拠の XML データを直接操作することができるようになっているが、これには MPEG-7 の知識を必要とする。そこでこの部分を、カスタマイズ可能で取り替え可能なモジュール仕様とすることで、ユーザに MPEG-7 の知識がなくとも、広く利用できるように拡張を考えている。もう 1 つは、現在のメタデータエディタで人手により行っている一連のモジュール実行処理を、バッチ処理で行えるようにするための改修である。

## 5. おわりに

本稿では、映像コンテンツに対し、内容を記述するメタデータを効率よく付与するための枠組み「メタデータ制作フレームワーク」について、その最新の仕様を紹介した。この MPF の利点は、放送局や映像コンテンツ提供事業者にとっては最新技術を次々とモジュールとして組み込むことで、常に最新のメタデータ制作環境を維持することができることである。また、産業的には、各社の独自技術はブラックボックス化されたモジュールとして供給されることになり、比較的受け入れやすい環境であると考えている。更に、学術的には、個々の研究成果がモジュールとなり、研究室などでの技術の蓄積、継承そして利用を容易とし、更なる研究の効率をあげることができるであろう。このように我々は、本フレームワークがメタデータ制作という目的を通して、さまざまなメディア解析、情報処理技術を発展させることに貢献できると期待している。今後は、更に実践的な目標を定めて、賛同して頂ける研究機関と連携した形で、更に研究開発を進めたいと考えている。

## 参 考 文 献

- 1) Metadata Production Framework (MPF). <http://www.nhk.or.jp/strl/mpf/>.
- 2) ISO/IEC 15938: Information technology -Multimedia content description interface-.
- 3) XQuery 1.0: An xml query language. <http://www.w3.org/TR/xquery/>.
- 4) eXist: Open source native XML database. <http://exist.sourceforge.net/>.
- 5) Information Grand Voyage project. <http://www.igvpj.jp/>.
- 6) P/SCAIE project. <http://tech.ebu.ch/groups/pscaie/>.