

H-002

実運用に向けたメタデータ制作フレームワークの リファレンスソフトウェアの改良

Improvements of MPF (Metadata Production Framework) Reference Software Toward a Real Use Case

佐野 雅規[†] 住吉 英樹[†] 藤井 真人[†] 柴田 正啓[†] 八木 伸行[†]
Masanori Sano Hideki Sumiyoshi Mahito Fujii Masahiro Shibata Nobuyuki Yagi

1. はじめに

映像コンテンツに対して、内容記述メタデータを効率良く付与するための共通基盤として、メタデータ制作フレームワーク(MPF)を提案[1]してきた。また、本フレームワークの基本機能を検証するためのリファレンスソフトウェアをあわせて公開[1]している。しかし、これらのソフトウェアは、大量のコンテンツを対象とする実運用を試験するには、一連の解析処理の自動化(バッチ処理)と、自動生成したメタデータを効率良く修正する仕組みに欠けていた。そこで、既存のツールに改修を加え、この2つの機能を追加実現したので報告する。

2. MPFのリファレンスソフトウェアと問題点

MPFの公開にあたっては、仕様書だけでなく、MPFに準拠したさまざまなリファレンスソフトウェアを配布している。これらは商用でなければ、自由に利用することができる。図1は、リファレンスソフトウェアの中核であるメタデータエディタの画面である。本エディタでは、映像コンテンツを1つ選択し、これに複数のメディア解析モジュールを適用させることで、ここからここまでという時間区間の情報と、当該区間の内容や特徴にかかわる情報から構成される「セグメントメタデータ」を生成する。図1下側の3つの緑色の丸印が、それぞれ解析処理モジュールに相当し(例えば、ショット分割、音声認識、シーン分割)、各解析により生成された情報を含むセグメントが、右側に四角形として表示されている。このセグメントメタデータをクリックすると、画面左上の赤枠部分に、当該セグメントメタデータの内容情報が表示され、編集することが可能となっている。画面右上では、該当する映像部分を再生することができ、映像内容を確認しながらのメタデータ修正を可能としている。

本エディタを用いた大量の映像コンテンツへのメタデータ付与を考えると、次の2点が問題となっていた。1つは、メタデータを付与する映像は、1つ1つ手でエディタに読み込む必要があり、更に各種処理モジュールも手動で選択し実行させなければならない点である。もう1つは、セグメントメタデータを修正するGUI部分の操作には、MPEG-7のデータ構造の知識が必要な点である。これは、現在の編集用GUI(図1の赤枠部分)が、MPEG-7のメタデータをそのままの木構造で表現しているためである。そこで、新しくバッチ処理機能を追加し、編集モジュールのインタフェース(IF)を規定することで、自由な編集用GUIをメタデータエディタ上にプラグイン可能とした。

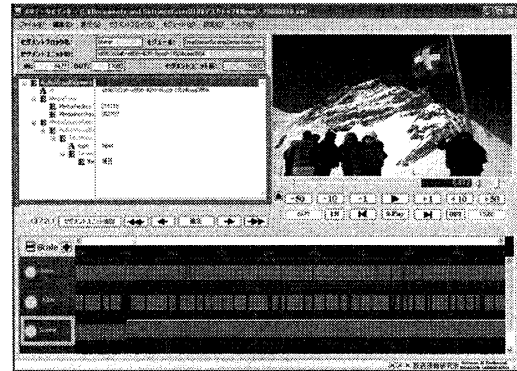


図1 メタデータエディタ

3. バッチ処理機能の追加

MPFにおける柔軟なバッチ処理を実現するために、一連の処理を記述するバッチ処理ファイルの仕様と、これを読み込んで実際の処理を行うバッチ処理プログラムを、メタデータエディタとは別に開発した。

このバッチ処理ファイルには、処理対象とする映像の情報、一連の処理モジュール群とそのパラメータの情報、生成したメタデータの格納場所の情報、ファイル名やパス名の指定に利用するエイリアス情報の4つが含まれる。処理対象とする映像群の指定は、各映像の絶対パスをすべて記述したテキストファイルか、対象映像群が格納されたディレクトリを選択する。また、生成したメタデータの格納については、指定したフォルダ内にまとめて格納するか、それぞれを個別の場所に保存するかを指定する。エイリアス機能は、入力となる映像ファイル名のうち、数字で指定する部分文字列(何番目から何番目)をエイリアスとして宣言するものであり、バッチ処理ファイルの入出力部分のほか、各モジュールのパラメータ内でも利用できるようにした。これはバッチ処理において、入力ファイルの存在場所やファイル名そのものが、ある一定の規則性を有することが多く、これらを有効に利用できるように付加した機能である。

バッチ処理ファイルは、次の2通りの方法で作成する。1つはメタデータエディタを用いる方法であり、エディタ上で各種処理モジュールを十分にテストし、最終的に利用するモジュールとそのパラメータが決定した時点で、メニューから生成することができる。もう1つは、バッチ処理プログラム本体を用いる方法であり、新規に1から作成することも、エディタから出力したファイルを編集することもできる。以上の仕組みにより、複数の映像を対象とした一連の処理を順次実行していく環境を構築することができる。

[†] NHK 放送技術研究所
NHK Science and Technology Research Laboratories

4. 編集用 GUI のプラグイン化

4.1 編集モジュールのインタフェース(IF)

メタデータエディタ上に、さまざまな編集用 GUI をプラグイン可能とするため、新しく編集モジュールを定義し、以下に挙げる情報を授受する IF を備えることとした。これにより、目的に沿った効率的な作業を可能とする編集 GUI をエディタ上に容易に組み込むことが可能となる。なお、この IF は、リファレンスソフトウェアであるメタデータエディタに関する規約であり、MPF の仕様ではない。

- 選択中のセグメントメタデータ
- エディタ上の映像の現在時刻 (再生時刻)
- エディタ上の時刻表示モード
- エディタ上の日英言語モード
- セグメントメタデータが変更された通知
- エディタ上の映像パス情報
- エディタ上の映像のフレームレート

4.2 開発した編集モジュールの例

前節に示した IF の検証を行うために、次に挙げる 3 つの編集モジュールを実際に作成し、動作確認を行った。

4.2.1 MPEG-7 木構造の編集モジュール

これまでのメタデータエディタに備わっていた GUI を、そのままモジュール化したものである。MPEG-7 のデータ構造を理解していないと利用が難しい面はあるが、最も自由度が高く、どんな編集も行うことができるという点では、エディタに必須の編集モジュールと考えている。

4.2.2 編集用 GUI 形成 XML による編集モジュール

本モジュールは、編集に必要な GUI を、Windows で提供されるコンポーネントを用いて容易に形成するものである。あらかじめ決められた XML 形式に則り、利用するコンポーネント(図 2 A)と、生成する MPEG-7 メタデータ(図 2 B)の組をファイルに記述しておき、これを読み込むことで GUI を実現する。図 2 は編集用 GUI 形成 XML の記述例であり、例えば<String>はテキストボックスを表し、ID"001"が付与されている。ここに入力された文字列は、図 2 B に記述された"%001%"の部分に置き換えられるようになっており、AudioVisualSegment 下の FreeTextAnnotation 要素として生成される。図 3 は、このような XML ファイルにより形成された画面の例であり、図 1 の赤枠の部分がこれと置き換わる。この仕組みにより、決められた項目を編集するような GUI であれば、事前に編集用 GUI 形成 XML ファイルを作成しておき、これをエディタ上で指定するだけで、必要部分だけがユーザに提示される GUI が実現できる。

4.2.3 代表静止画の編集モジュール

本モジュールは、エディタと編集モジュールの間で規定した IF を全て利用する応用モジュールとして試作した。図 4 がその GUI 画面である。本モジュールは、選択されているセグメントメタデータの時間区間内から、代表静止画を作成/編集/削除するためのものである。画面上部には、当該セグメントメタデータ内に存在する代表静止画のリストが表示され、その 1 つをクリックして選択すると、エディタ右上の映像部分に、その静止画が表示されるようになっている。同時に、その静止画に付与されている注釈のリストが下側に表示され、これらも編集が可能となっている。

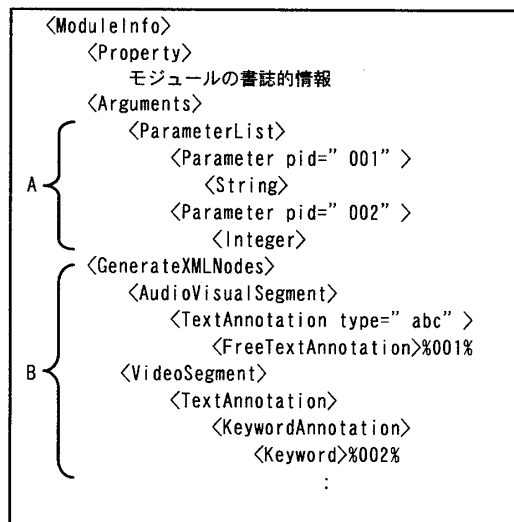


図 2 編集用 GUI 形成 XML の記述例

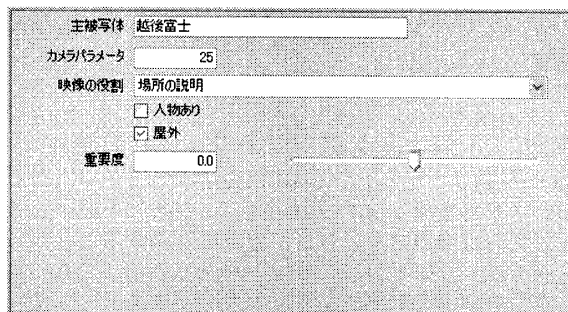


図 3 編集用 GUI 形成 XML による編集画面

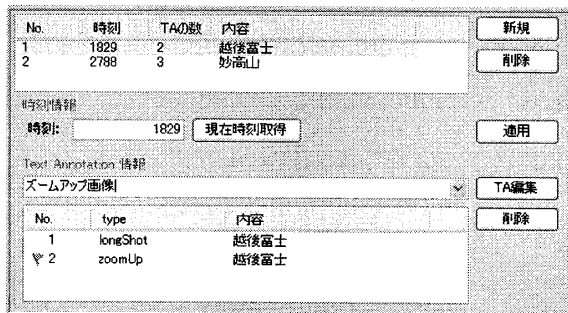


図 4 代表静止画の編集画面

5. まとめ

本稿では、これまで公開してきたメタデータ制作フレームワークのリファレンスソフトウェアについて、実運用を考慮した際の問題点について説明し、これをバッチ処理機能の追加と、編集モジュールのプラグイン化により解決したことを報告した。今後は、賛同いただける研究機関と連携して、更に実践的な利用に向けた機能検証を進めていきたい。

参考文献

- [1]NHK 放送技術研究所, “メタデータ制作フレームワーク”, <http://www.nhk.or.jp/str1/mpf/>