

制作者の意図を反映したマルチメディアコンテンツ統合

湯本 高行[†] 馬 強[†] 角谷 和俊[†] 田中 克己[†]

[†] 京都大学大学院情報学研究科社会情報学専攻 〒 606-8501 京都市左京区吉田本町

E-mail: †{yumoto,qiang,sumiya,tanaka}@dl.kuis.kyoto-u.ac.jp

あらまし 複数の情報ソースからのコンテンツの動的統合、複合化は、多様化する利用者の要求を満たすために非常に有効である。しかし、現在、動的コンテンツ統合では、利用者の立場しか考慮されていないことが多く、コンテンツの動的な統合を容易にする実現する言語も存在しない。そこで、本研究ではコンテンツ統合をコンテンツの制作者、利用者、統合スクリプト記述者の立場から考え、これら3者の意図を反映した形で、動的なコンテンツ統合を行うためのスクリプト言語を提案する。このスクリプト言語は統合コンテンツの構造記述と統合意図記述の2つの部分に分かれている。この言語の特徴は、1) 統合コンテンツの構造記述と意図記述の分離、2) 統合プロセスの分割による意図の分散記述の実現、3) 意図を反映した動的な統合コンテンツの実現、の3点である。

キーワード 動的コンテンツ統合、スクリプト言語、意図

Multimedia Contents Integration reflecting Authors' Intention

Takayuki YUMOTO[†], Qiang MA[†], Kazutoshi SUMIYA[†], and Katsumi TANAKA[†]

[†] Department of Social Informatics, Graduate School of Informatics, Kyoto University

Yoshida Honmachi, Sakyo, Kyoto 606-8501, Japan

E-mail: †{yumoto,qiang,sumiya,tanaka}@dl.kuis.kyoto-u.ac.jp

Abstract Dynamic contents integration of multiple information sources is one of the ways to provide more rich contents to satisfy users' diverse demand. In this paper, we propose a framework for dynamic contents integration reflecting intentions of content creator, provider and consumer. We also propose a script language based on this framework. This script language consists of two parts : a) spatiotemporal description scheme and b) intention description scheme. The features of our mechanism can be summarized as 1) separation of content description and structure description, 2) support of distributed intention descriptions by separating integration process, and 3) dynamic content integration reflecting authors' intentions.

Key words Dynamic Contents Integration, Script Language, Intention

1. 序 論

近年、ブロードバンドネットワークの普及、蓄積型テレビの実現により、テキストや画像のみならず、ストリーミング放送のようなビデオコンテンツなど多様なコンテンツが利用できるようになっている。しかし、その利用形態は全く多様化することなく、ひとつのコンテンツをただ閲覧、視聴するだけである。それに対して、新たなコンテンツの閲覧方式として、コンテンツの変換やコンテンツ統合に関する研究が行われている [1], [2]。コンテンツ統合とは関連する複数コンテンツを時間的、空間的に配置することによって同時、順次に閲覧する方式であり、これによって、単一のコンテンツでは貧弱な情報を補強したり、コンテンツの2次利用によるコンテンツの増殖が可能になる。現在、利用者の要求は多様で、利用者の欲する情報は複数のWebサイトに分散しており、その形式もテキストやビデオなど

さまざまである。さらに蓄積型TVの登場によりテレビ映像もWebと結びつけて利用できるようになりつつある。このような状況下で、異なる情報ソースから情報を取得し、利用者の多様な要求を満たすためには、動的なコンテンツ統合が非常に有効である。しかし、動的なコンテンツ統合を実現する研究の多くは、コンテンツにかかわる人物の意図を忠実に反映しておらず、反映していたとしても、利用者の意図だけである。利用者以外にも意図は存在する。CMというコンテンツを例に考えると、CMの供給元は同業他社のCMを10分以内に呈示してほしくないなどの意図が考えられる。また、コンテンツの静的な統合は、SMIL [3] などによって記述可能であるが、動的な統合はSMILで不可能である。既存の他のスクリプト言語の一部では、動的な統合も記述可能であるが、専門知識が必要な上に、各立場の意図を反映した統合のための枠組は存在しない。

そこで、本研究ではコンテンツにかかわる人物がそれぞれの

立場から意図を記述し、それを反映して動的にコンテンツ統合を行うためのスクリプト言語 *Macchiate* を提案する。本研究で提案する言語は統合コンテンツの構造記述部分と統合意図記述部分からなる。コンテンツに対する立場としては、統合スクリプト作者、素材コンテンツ制作者、コンテンツ利用者の3つの立場が考えられ、それぞれ意図に基いて統合プロセス(の一部)を記述することによって各立場の意図を反映させることが可能である。統合スクリプト作者はコンテンツ統合を実現するために、構造記述部分では、統合したコンテンツの構造を記述し、意図記述部分では統合コンテンツの構成要素(Digital Asset、詳しくは後述)が満たすべき条件や、動的に Digital Asset を取得する場合の優先条件などを記述する。さらに素材コンテンツ制作者は自分のコンテンツについて意図記述をしておくことによって、コンテンツ利用者は自分の嗜好を意図として記述しておくこと、統合スクリプト作者の意図記述と合成され、これによって3者の意図を反映したコンテンツの統合が実現される。つまり、*Macchiate* はコンテンツ統合の分散記述のためのフレームワークを提供する言語である。提案する言語の特徴は以下のとおりである。

- 統合コンテンツの構造記述と意図記述の分離
- 統合プロセスの分割による意図の分散記述の実現
- 意図を反映した動的な統合コンテンツの実現

本論文の構成を述べる。第2節では関連研究について述べる。第3節では本研究で仮定する統合に関係したモデルについて、第4節では本研究で提案するスクリプト言語 *Macchiate* について説明する。第5節では統合意図記述の部分についての詳細を述べ、第6節では応用について説明する。第7節では考察と今後の課題について述べる。

2. 関連研究

2.1 コンテンツ生成

コンテンツ生成は素材となるコンテンツから新たなコンテンツをつくり出すものであり、本研究もそれに関する研究である。コンテンツ生成には、1) 処理の種類、2) 素材の情報単位の2つの観点がある。

a) 処理の種類

処理の種類としては、変換と統合がある。

変換は表示形式を変えることによって情報の印象を変化させるものである。灘本らの Web の番組化[4]、マヘンらのビデオの Web 化[5]などがある。これらは、ひとつの Web ページまたはビデオを入力とし、それらの表示形式をかえることによってコンテンツの利用スタイルを変えるというものである。

また、統合は複数のソースから情報を取得し、ひとつのコンテンツを生成することである。馬らの WebTelop[6] や寺田らのアクティブカラオケ[7]では、複数のソースからの情報を同時に呈示する方式することによってコンテンツの統合を行っている。この方式をコンテンツの並列化と呼ぶ。これに対して、湯本らの Contextual News[8] や馬らの Virtual TV Channel[9]では、複数のソースからの情報を順番に呈示することによってコンテンツ統合を行っている。この方式をコンテンツの直列化と

呼ぶ。これらを複雑に利用したものとして、佐藤による Web の自動編集[10]があげられる。本研究では、直列化と並列化を組み合わせることによって統合を表現する。

変換と統合は相反する概念ではなく、それらを同時に使う場合もある。*Macchiate* は、統合を記述する言語であるが、変換を行うための関数も用意しており、変換したコンテンツを統合することが可能である。

b) 素材の情報単位

素材となる情報の単位の大きさにも差がある。WebTelop、Contextual News において処理される単位はビデオコンテンツの一部分や Web ページである。ビデオの Web 化もビデオの一部分を処理している。また、Web の番組化においては Web ページに含まれる画像や文章の段落単位での処理を行っており、Web の自動編集では、これを文単位で行っている。それに対して本研究は、WebTelop や Contextual News と同じようにビデオコンテンツの一部分や Web ページを素材の単位としている。

2.2 マルチメディア記述言語およびツール

SMIL[3]、Flash[11]などのさまざまなマルチメディアコンテンツ記述言語およびツールが存在する。これらはコンテンツのプレゼンテーションを具体的に記述するためのものである。そのため、コンテンツ制作者はプレゼンテーションの細部についてまで記述することを強制されている。また、内容と表示部分が完全に融合しており、ユーザの嗜好を反映させにくい。それに対して *Macchiate* では、内容の部分を表示(構造記述)と分離させ、さらに立場ごとに分散して記述可能な意図として表現している。また、*Macchiate* の構造記述ではかなり抽象的な記述を提供している。これは統合コンテンツの分散記述を容易にするため、統合コンテンツの構造をモデル化した状態で記述しているからである。また、*Macchiate* は XML で設計されているため、細かいプレゼンテーションを記述したい場合は、構造記述の部分のかわりに SMIL などの XML ベースの言語を用いることによって記述できる。

2.3 権利処理

権利処理[12],[13]で用いられる条件としては以下の3つが考えられる。

- (1) 使用するコンテンツがどれであるか。
- (2) 使用する人が誰であるか。
- (3) 同時に使用されるコンテンツがどんなものであるか。

このうち、(1)、(2)は多くの権利処理の条件として使われているが、(3)はあまり使われていない。本研究では、述語を用いることによって(3)を表現し、権利処理条件として用いる。また、権利処理では、矛盾が生じた場合の調停が重要であるが、本研究では、呈示するコンテンツの優先順位の導出過程においてスクリプト作者と利用者のランキング値の重みつきの和をランキング値に用いることにより、両者に譲歩を促すことで、この矛盾の一部の解決を試みている。

3. 統合のためのモデル

3.1 コンテンツのモデル化

本研究では、素材となるコンテンツやその一部分を Digital

Asset(以下, A) と呼ぶ。A は URI で表現できるものとそれを時間的に分割したものと定義する。ここで A を時間分割する関数 $split(A, s, f)$ (s :開始時刻, f :終了時刻) を定義すると, A は $A ::= URI|split(A, s, f)$ と定義できる。A は, 作者, 更新日時, メディアタイプなどの書誌情報, キーワードなどの内容情報, リンク先の URI などをプロパティ *property* として持つ。これを $property(A)$ と表記する。A の間には時間や空間などの関係を定義することが可能である。

3.2 メタデータのモデル化

コンテンツには, その内容を表すためのメタデータが付加されているとする。メタデータは, コンテンツの内容解析の結果やコンテンツ制作者が記述したもの, コンテンツの利用者によるアノテーションなどを統合したものが付加されているとし, 客観的に記述され, 内容的な誤りはないと仮定する。^(注1)

3.3 コンテンツに対する立場のモデル化

統合コンテンツに関係するユーザとして, 以下の3つが考えられる。

- 統合コンテンツ利用者 (以下, 利用者)
- 素材コンテンツ制作者 (以下, 素材制作者)
- 統合スクリプト作者 (以下, スクリプト作者)

ただし, スクリプト作者が, 利用者, 制作者と同一人物である可能性もある。これらの人物は統合コンテンツに対してさまざまな意図を持っている。

利用者

(1) 特定の条件を満たすコンテンツの優先呈示

例: “芸能” に関するコンテンツを優先して呈示してほしい。

(2) 特定の条件を満たすコンテンツの排除

例: 某芸能人に関するコンテンツを見たくない。

素材制作者

自分のコンテンツ利用時の条件

(1) 利用するコンテンツのユニット

例: 同時に別の Web ページ B も使ってほしい。

(2) 統合されるコンテンツについての条件

例: 商用のコンテンツといっしょに使ってほしくない。

スクリプト作者

- スクリプトそのもの

例: Web ページ C と D を同時に見せたい。

3.4 統合のモデル化

コンテンツの統合の目的は, 情報の補完・増強, 多様な利用者の要求の充足などがあげられる。これらを実現するためには, 統合コンテンツの構造と素材となる A の内容が重要である。

3.4.1 統合コンテンツの構造

統合の構造には, 時間的構造/リンク構造と空間的な構造が存在し, その組み合わせによってコンテンツの構造が決定する。時間的構造/リンク構造による統合は以下の2つがある。

- Serialize: コンテンツを順に呈示する。
- Synchronize: コンテンツを同期させて呈示する。

図1は, ノードを素材となる A とし, エッジを時間またはハイパーリンクによる推移として表したものである。

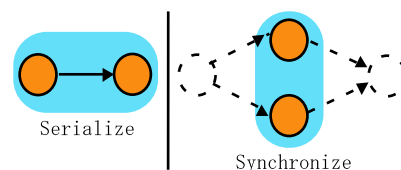


図1 時間/リンクによる統合

空間的な構造であるが, A の相対的な位置関係を, 同一ウィンドウ内にある/ない (neighbor-of/separate), 同一ウィンドウ内であっても同じ位置に存在する (same-place) によって表現する。same-place は, 時間的なレイアウトにおいて同時に表示される場合は, 「一方が他方に含まれる」ことを意味し, 同時には表示されない場合は「同じ場所に位置し, 時間によって切り替わる」と定義する。図2に空間的統合のイメージ図を示す。

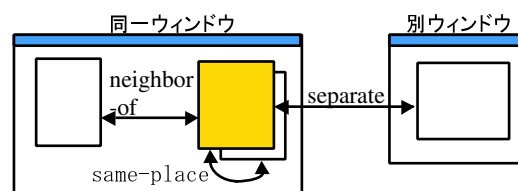


図2 空間的統合

3.4.2 統合プロセスのモデル化

統合の内容を決定する場合, SMIL のように A をあらかじめ静的に定義しておく方式と, 統合する A を動的に決定する方式が存在する。後者では, A を動的に決めるプロセスが非常に重要である。そのプロセスを以下のようにモデル化する。

- (1) Search: 統合する A の候補集合の取得
- (2) Restrict: 統合される A が満たすべき条件の検証
- (3) Rank: 統合される A の候補のランキングの計算とそれに基づくソーティング
- (4) Select: 統合される A の選出

これを式で表すと次のようになる。

$$A = f_{Select} \circ f_{Rank} \circ f_{Restrict}(S_{Search}) \quad (1)$$

$f_{process}$ ^(注2) は各プロセスに対応するフィルタリング関数であり, S_{Search} は Search 関数によって取得されるリストである。前述の利用者の意図 (2), 素材制作者の意図 (1),(2) はプロセス (2) に関係する。また, 利用者の意図 (1) は, プロセス (3) に関係する。ここで, 利用者は呈示されるコンテンツ一般に対して, 素材制作者は自分のコンテンツに対して, 意図を記述しておく。プロセス (2),(3) を修正することで利用者や素材制作者の意図を統合に反映させることができる。意図の反映は, 以下のようフィルタリング関数を修正することによって行う。

$$f_{Restrict} \rightarrow f_{Restrict}^{(u)} \circ f_{Restrict}^{(aa)} \circ f_{Restrict} \quad (2)$$

(注1): メタデータの正当性については非常に大きな問題であるが, 本研究では扱わない。

(注2): ただし, $process = Restrict|Rank|Select$ とする。

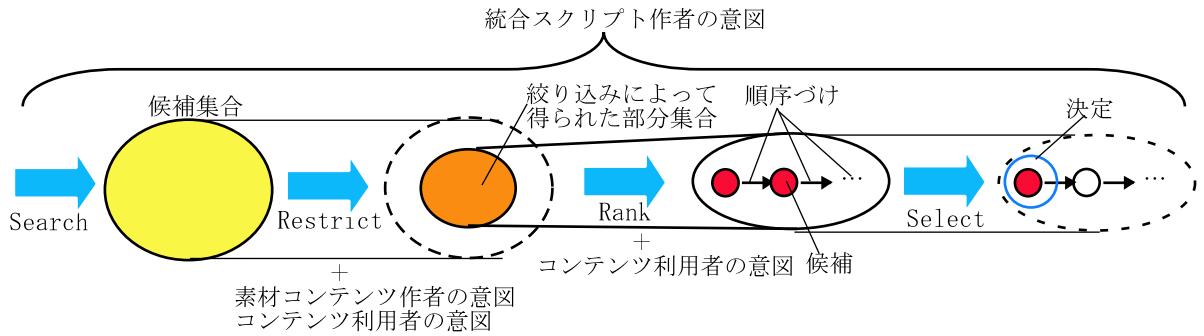


図3 統合相手選択のプロセス

$$f_{Rank} \rightarrow f'_{Rank} \quad (3)$$

ただし、 $f_{Restrict}^{(u)}$, $f_{Restrict}^{(aa)}$ はそれぞれ利用者、素材制作者によって記述された Restrict 関数であり、 f'_{Rank} は、ランキング値にスクリプト作者の Ranking 関数 R_{sa} のかわりに、 $(1-\alpha)R_{sa} + \alpha R_u$ を採用した Ranking 関数である。ただし、 R_u は利用者の Ranking 関数である。つまり、プロセスを以上の4つのモジュールで表現することによってそれぞれの立場ごとに独立して存在する意図をスクリプト作者の意図を中心として、利用者、素材制作者の意図を統合することが可能である。そのイメージを図3に示す。

4. スクリプト言語 *Macchiate*

動的なコンテンツ統合を可能にするスクリプト言語 *Macchiate* を設計する。この言語は、統合コンテンツの構造記述部分と意図記述部分からなる。統合スクリプト作者は、構造記述によって統合コンテンツの構造を、意図記述によって統合コンテンツを構成する A についての意図を参照的に記述する。また、素材制作者は自分のコンテンツについて、コンテンツ利用者は利用するコンテンツ一般について、同様に意図を記述することによって、統合コンテンツに意図を反映させることができる。構造記述部分と意図記述部分を分離することによって、統合スクリプト作者、素材制作者、コンテンツ利用者の3者の意図を同じ方式でかつ参照的に記述できる。つまり、統合スクリプト作者の書いたスクリプトは単体でその役割を果たすが、参照的かつ分散して記述された素材制作者、コンテンツ利用者の意図を統合スクリプト作者の意図に取り込むことによって、立場の異なる意図を反映することが可能である。立場と記述内容の関係を下表1に示す。

表1 ユーザと記述内容

ユーザ	構造記述	意図記述	対象
スクリプト作者		すべて	統合コンテンツ
利用者	x	Restrict, Rank のみ	統合コンテンツ
素材制作者	x	Restrict のみ	自分のコンテンツ

4.1 構造記述

構造記述では、時間的/空間的レイアウトの簡単な記述を主な目的としている。このために時間的/空間的なレイアウトについては、時間的統合と空間的な統合を組み合わせた抽象的な記述を採用している。また、それだけではなく、以下のように反復

記法などを用いた簡略化された記述も可能である。

<for var="変数" max="回数">

以下に構造記述部分 (*Structure*) の簡易文法のBNF表現を示す。

```

Structure
 ::= A | < Method > (Structure)+ < /Method >
    | < for var = "変数" (max = "回数")? >
      Structure < /for >
Method ::= serialize|synchronize
        neighbor - of|separate|same - place
  
```

4.2 意図記述

呈示されるコンテンツを記述するために候補集合の取得とそれらに対するフィルタリングを行う。この記述の特徴は以下のようにフィルタリングを複数のフィルタリング関数の合成関数として定式化している点である。具体的な記述は、意図の対象の指定と4種類の意図の記述からなる。4種類の意図の記述は以下のように行う。

- (1) Search : Search 関数の種類とその引数の指定
- (2) Restrict : 満たすべき条件を表す論理関数 (bool 型)
- (3) Rank : Ranking 関数 R (値域: 0 以上 1 以下) の記述
- (4) Select : Select 関数の引数と変換の有無

Restrict プロセスで用いる論理関数を $L(A)$, Rank プロセスで用いるランキング値を $R(A)$ で表すと、前述のフィルタリング関数との関係は次のようになる。

$$f_{Restrict}(S) = \{A | L(A) = "true", A \in S\} \quad (4)$$

$$f_{Rank}(S) = (A_1, A_2, \dots, A_n)$$

$$(ただし, i < j \Rightarrow R(A_i) \geq R(A_j)) \quad (5)$$

意図記述では、上記の4つの意図の記述とその対象を記述する。以下に意図記述部分 (*Intention*) の簡易文法をBNFで示す。

```

Intention
 ::= < target > 意図記述の対象 < /target >
    < search > Search 関数 < /search >
    (< restrict > Restrict 関数 < /restrict >)?
    (< rank > Rank 関数 < /rank >)?
    (< select > Select 関数 < /select >)?
  
```

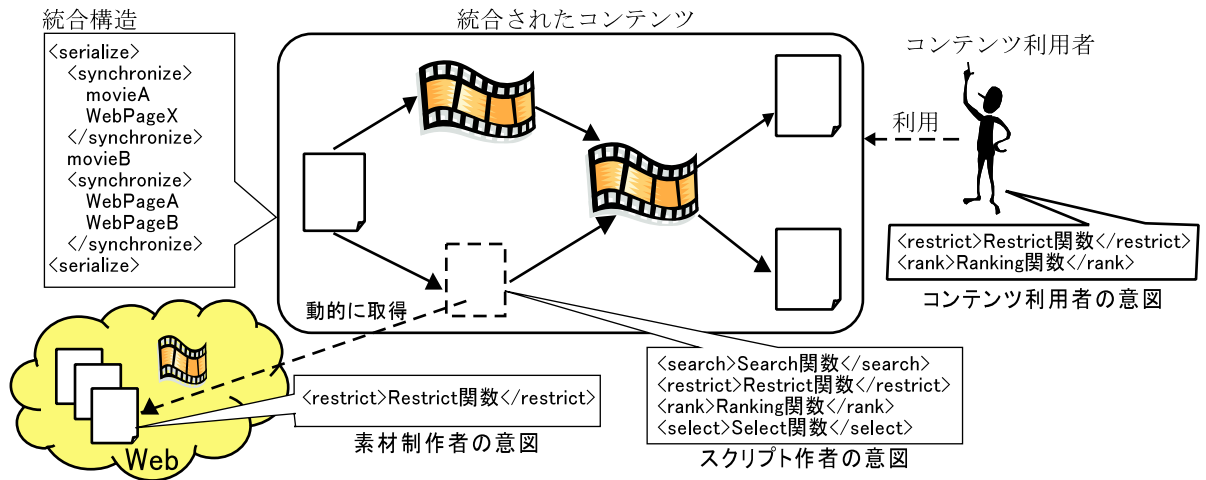


図4 記述イメージ

記述のイメージを図4に示す。これらの意図は本言語で標準で用意されている Primitive 関数を用いて記述する。Primitive 関数には以下のようなものが用意されている。

- 内容に関する関数
- 時間に関する関数
- コンテンツ取得に関する関数
- メディア変換に関する関数

4.2.1 内容に関する関数

- $keywords(A)$: 主要なキーワードを返す。
- $tf(k)$: キーワード k の tf の値を返す。
- $idf(k, \Omega)$: キーワード k の idf の値を返す。
- $sim(A_1, A_2)$: A_1, A_2 の類似度を返す。

ただし、 A がビデオコンテンツやその一部の場合はメタデータ M_A を引数にする。

4.2.2 時間に関する関数

主にビデオコンテンツに関して以下のような関数が定義されている。

- $dur(A)$: 時間の長さ
- $st(A), fin(A)$: 開始時間, 終了時間
- $diff(t_1, t_2)$: 時間差 $t_2 - t_1$

ただし、Web コンテンツでは、 $dur(A) = \infty$ と定義し、時系列を持つ A と同期化されたときのみ、 $dur(A) \neq \infty$ とする。

4.2.3 コンテンツ取得に関する関数

$getURI, getVideoSegment$ の2つの関数が用意されている。

$getURI(domain, keywords, number)$

Web の検索エンジンによって、キーワード集合 $keywords$ で $domain$ で指定されたドメインから $number$ で指定された数の URI を取得する。

$getVideoSegment(uri)$

引数に単一の URI または URI 集合をとる。引数が単一の URI であつビデオコンテンツのメタデータ [14], [15] の場合は、そのメタデータを利用してビデオセグメント集合を取得する。ビデオコンテンツのメタデータには、手動で付加された、またはビデオ解析によって生成されたセグメントの情報やキーワードの

情報が格納されているとする。(注3)

また、引数が単一の URI であるが、メタデータではない場合は、空集合を返す。引数が URI 集合 $\{URI_1, URI_2, \dots, URI_n\}$ の場合は、以下が成り立つとする。

$$getVideoSegment(\{URI_1, URI_2, \dots, URI_n\}) = \bigcup_{i=1}^n getVideoSegment(URI_i)$$

$getVideoSegment(getURI(domain, keywords, number))$ によって、Web 上の複数のビデオコンテンツからビデオセグメントを取得することができる。

4.2.4 メディア変換に関する関数

Web コンテンツをビデオコンテンツに変換する $Web2Video$ 関数とビデオコンテンツを Web コンテンツに変換する $Video2Web$ 関数が用意されている。

また、四則演算や max, min 関数などの基本的な算術関数も用意されている。

5. 意図記述の詳細

意図記述の4つの意図について詳細を述べる。ただし、意図の中で使われる関数、演算子、関係は実際のスクリプト言語 *Macchiate* 中では XML タグとして表現される。

5.1 Search

A のリストを取得する Search 関数には、 $getURI, getVideoSegment$ があり、 $getURI$ の出力は、構成要素は検索エンジンの出力結果によって、 $getVideoSegment$ の構成要素はメタデータの構造に応じてランキングされている。Rank プロセスでランキングの計算が行われた場合は、Rank プロセスでのランキングが優先される。

5.2 Restrict

この部分では、論理演算子を用いて、統合される A が満たすべき条件を論理関数を用いて記述する。 A の候補集合の要素の各々について、対応する素材制作者の意図の適用、統合スクリ

(注3): メタデータかどうかの判定は拡張子やタグ名などによって判定する

プト作者の意図 (Restrict 部分) による絞り込み, コンテンツ利用者の意図 (Restrict 部分) による絞り込みの順に意図の適用する. 統合には素材制作者が定義した A(これを A_{aa} で表す) を使用するべきであるが, これが統合スクリプト作者が定義した A(これを A_{sa} で表す) と一致しない場合, A_{sa} に対応する A_{aa} を発見し, 現在の候補を A_{aa} に置換する. その後, 条件による絞り込みが行われる.

最適な A の発見

A をビデオコンテンツの一部としたとき, 開始/終了時間を $st(A), fin(A)$, 時間の長さを $dur(A)$ とおく. また, 素材制作者が定義した A の集合を S_{aa} とおく. 最適な A を「 A_{sa} の区間を含み, 集合 S_{aa} に含まれ, dur が最小のもの」と定義し, それを返す関数を $Match_{aa}(A_{sa})$ と定義する. この値は, 下線部を満たすものうちの dur が最小のものである. 下線部を満たすものを最適 A 候補と呼び, そのリスト L を以下の要領で作成する.

$st(A_{aa}) \leq st(A_{sa})$ を満たす $A_{aa} \in S_{aa}$ の各々に対して, $fin(A_{aa}) > st(A_{sa})$ の場合, 以下の式で求めた A を L に追加する.

$$A = \begin{cases} A_{aa} & (\text{if } fin(A_{aa}) \geq fin(A_{sa})) \\ A_{aa} \cup Match_{aa}(A_{sa(2)}) & (\text{if } fin(A_{aa}) < fin(A_{sa})) \end{cases}$$

ただし, $A_{sa(1)} = split(A_{sa}, st(A_{sa}), fin(A_{aa}))$,

$A_{sa(2)} = split(A_{sa}, fin(A_{aa}), fin(A_{sa}))$ である.

式の意味であるが, $fin(A_{aa}) \geq fin(A_{sa})$ (図 5 の a) の場合, $A_{aa} \cap A_{sa} = A_{aa}$ より $A_{aa} \in L$ は自明である. また, $fin(A_{aa}) < fin(A_{sa})$ (図 5 の b) の場合は $fin(A_{aa})$ で A_{sa} を分割すると, $t \leq fin(A_{aa})$ の部分は $fin(A_{aa}) \geq fin(A_{sa(1)})$ となり, 図 5 に帰着するので, A_{aa} が候補になる. $t > fin(A_{aa})$ については, $Match_{aa}(A_{sa(2)})$ を求め, それを連結する.

このようにして求めた L の中で $dur(A)$ が最小となる A を $Match_{aa}(A_{sa})$ とする.

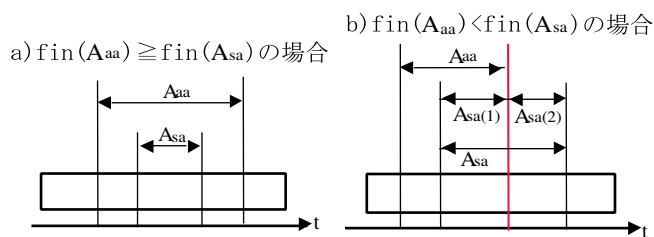


図 5 最適 A 候補の導出

5.3 Rank

Primitive 関数を合成することによって正規化された Ranking 関数 R (値域: $0 \leq R \leq 1$) を決定する. また, スクリプト作者の意図による Ranking 関数 R_{sa} とコンテンツ利用者の意図による Ranking 関数 R_u を合成し, $R = (1 - \alpha)R_{sa} + \alpha R_u$ ($0 \leq \alpha \leq 1$) とすることによってコンテンツ利用者の意図も反映することが可能となる. α は統合スクリプト作者がコンテンツ利用者に譲歩する度合いであり, これが大きいほど利用者の意図を大きく

反映することができる. Ranking 関数を正規化しておくことによって, R_u と R_{sa} を合成するときにそれらの影響の大きさを対等に評価でき, α のみによってスクリプト作者と利用者の意図のバランスをとることが可能になる.

5.4 Select

Select 関数によって, Rank で決定された順位から取得する A を決定する. Select 関数 $Select(rank, number)$ は引数に取得する A の順位 rank と返す A の数 number を取り, A またはその集合を返す.

また, Select 関数の返り値を Video2Web, Web2Video 関数の引数にすることによって Web コンテンツまたはビデオコンテンツに変換することが可能である.

6. 実装とその応用

6.1 実装

Macchiate の処理系としては, Macchiate から HTML コンテンツを生成するシステムを開発中である. これは以下のような 2 つのサブシステムからなる.

- レイアウト変換システム (以下, 変換システム)
- 呈示コンテンツ決定システム (以下, 決定システム)

変換システムでは, 構造記述から HTML のタグ構造への変換を行う. このとき, プレースホルダーが存在した場合は, 決定システムからの出力を利用する. 決定システムでは, スクリプト作者と利用者の意図記述部分からベースとなるフィルタリング関数を生成する. その後, コンテンツ選択時に, コンテンツと対応づけて蓄積されている素材制作者の意図を選択プロセスに反映させながら, フィルタリングを行う. これを図に表すと図 6 のようになる.

6.2 応用

6.2.1 Macchiate を利用したアプリケーション

ビデオの Web による補完

WebTelop [6] は, ビデオのシーンごとに関連した Web ページをビデオに同期して並べて呈示するアプリケーションである. これを近似的に記述すると以下のようなになる. 構造記述は以下のようなになる.

```
<serialize>
webpage_0
<for var="i"><synchronize><separate>
webpage_i video_i
</separate></synchronize></for>
</serialize>
```

これを図に表すと図 7 のようになる.

$scene_i$ に対する意図はメタデータからビデオのセグメントを順に取得することである. これは以下のように表す.

```
<target>scene_i </target>
<search>
getVideoSegment(getURI(Any,keywords(metadata_i),100))
</search>
<select>select(i,1)</select>
```

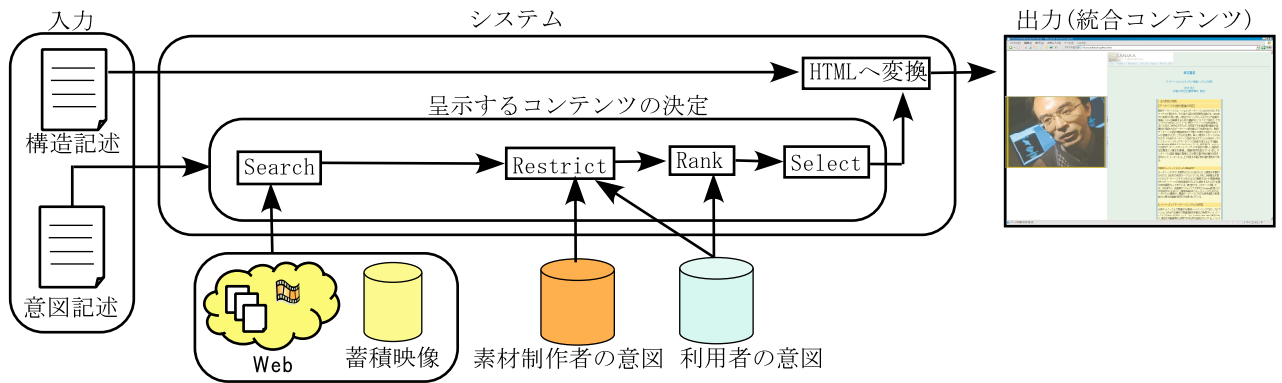


図6 処理系のプロトタイプシステム

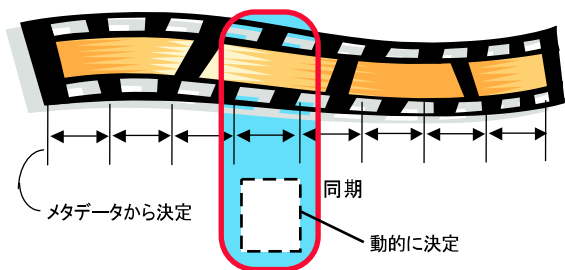


図7 WebTelop の構造

$webpage_i$ に対する意図は以下ようになる。

```

<target>webpagei </target>
<search>getURI(Any,keywords(metadatai),100)</search>
<restrict>mediatype(webpagei)="html"</restrict>
<rank>1-sim(webpagei,webpagei-1)</rank>
<select>select(1,1)</select>

```

<rank> タグ内の記述は、利用者に常に新しい情報を呈示するため、連続して呈示される Web ページの類似度が低いものを優先していることを表している。

Web のビデオによる補完

WebTelop とは逆に、Web ページをビデオによって補完する VideoTelop が定義できる。VideoTelop では、Web ページの内容を補完するビデオセグメントを呈示する。さらに、表示されるビデオセグメントは直前に閲覧した Web ページの内容とも関連が高いものを呈示する。これはコンテンツの切り替えが頻繁に起こると、ビデオの場合は Web ページに比べて、その内容の把握が困難であり、できるだけ内容の大幅な変更を減らすためである。これによって、同じページでもナビゲーションの順序によって異なるコンテンツが呈示される。これを近似的に記述すると以下ようになる。構造記述は以下ようになる。

```

<serialize>
  <for var="i" max="n">
    <synchronize><neighbor-of>
      scenei webpagei
    </neighbor-of></synchronize>
  </for>
</serialize>

```

ただし、 $webpage_i$ ($i \geq 0$) は $webpage_{i-1}$ からのリンクによって決定され、 $webpage_0$ は固定である。これを図に表すと図 8 のようになる。 $video_i$ に対する意図は、長さ 5 分以内であり、

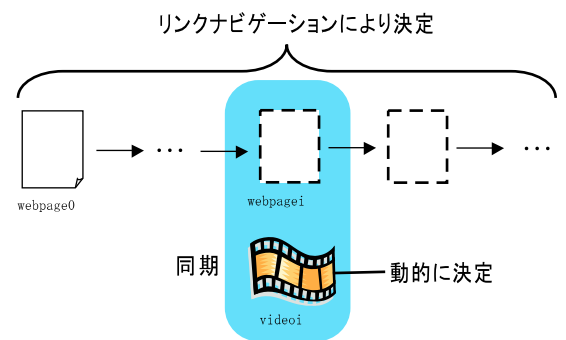


図8 VideoTelop の構造

同時に呈示される $webpage_i$ との類似度と直前に再生される $video_{i-1}$ のメタデータ $metadata_{i-1}$ との類似度が共に高いものを優先することである。これらはそれぞれ <restrict> タグ内と <rank> タグ内に記述される。この意図は以下になる。

```

<target>videoi </target>
<search>
  getVideoSegment(getURI(Any,keywords(metadatai),100))
</search>
<restrict>dur(videoi)≤5 min</restrict>
<rank>
  0.6×sim(webpagei,metadatai)
  +0.4×sim(metadatai,metadatai-1)
</rank>
<select>select(1,1)</select>

```

例えば、ニュースサイト内を巡回すると、ニュースに関連した映像を蓄積型テレビの中から検索し、呈示するなどが考えられる。

6.2.2 その他の意図

素材作者の意図

素材作者の意図はさまざまであるが、ここでは CM を例にとって説明する。

例:CM の強制挿入

あるコンテンツを利用するときは必ずその直後に CM_A を流さなくてはならない。

```
<restrict>
  <asset ID="THIS"><before ID="CM_A"/ ></asset>
</restrict>
```

例:隣接する CM に対する制約

某ビールメーカー A 社の CM を素材として考え、ここに付加する意図を考える。例えば、「某ビールメーカー K 社の CM の後に流してはいけない」という意図が考えられる。(図 9 参照)

```
<restrict><not><asset>
  <keywords>K 社, ビール </keywords>
  <or>
    <before ID="asset1" diff="0s"/>
    <after ID="asset1" diff="0s"/>
  </or>
</asset></not></restrict>
```

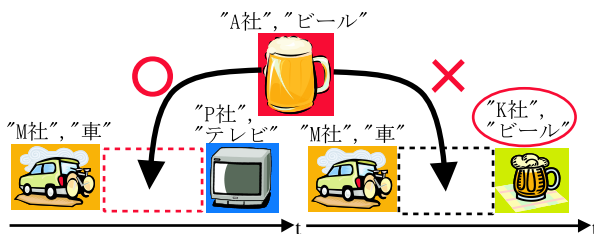


図 9 隣接する CM に対する制約

6.2.3 利用者の意図

例:好みのコンテンツの優先

前述の WebTelop を用いて、メジャーリーグの試合を見るときに、「ichiro」の情報を優先して呈示してほしい場合は以下のように記述する。 $\langle rank \rangle tf(\text{“ichiro”}) \langle /rank \rangle$

従来の WebTelop は利用者の嗜好を反映させることはできなかったが、Macchiate によって実装された WebTelop では利用者の嗜好を反映させることができる。

例:好ましくないコンテンツの排除

コンテンツ利用者の意図として暴力を含むビデオを見たくない場合は以下の記述によって呈示されるビデオを制限することができる。(注4)

```
< restrict > "暴力" ∉ keywords(metadatai) < /restrict >
```

7. 結 論

本論文では、動的なコンテンツ統合のためのスクリプト言語 Macchiate を提案し、その応用例について述べた。提案した言語の特徴は以下のとおりである。

- 統合コンテンツの構造記述と意図記述の分離
- 統合プロセスの分割による意図の分散記述の実現

- 意図を反映した動的な統合コンテンツの実現

また、応用例ではビデオコンテンツ、Web コンテンツの双方において意図を反映したコンテンツの統合を行い、これによって、コンテンツの新たな利用方式を示した。

今後の課題としては、意図、特に利用者の記述方法の簡略化やユーザプロファイルからの抽出と、意図の表現能力の強化が挙げられる。

謝 辞

本研究の一部は、平成 14 年度科研費基盤研究(B)(2)「蓄積型放送のためのパーソナル視聴の研究」(課題番号:14380177, 代表:角谷和俊)及び平成 14 年度基盤技術研究促進事業(民間基盤技術研究支援制度)「クロスメディアコンテンツ基盤技術の研究開発」、21 世紀 COE プログラム「知識社会基盤構築のための情報学拠点形成」によるものです。ここに記して謝意を表すものとします。

文 献

- [1] Hiroyuki Kitagawa, Atsuyuki Morishima, and Hironori Mizuguchi, "Integration of Heterogeneous Information Sources in InfoWeaver", in Advances in Multimedia and Databases for the Next Century - A Swiss/Japanese Perspective -, Y. Masunaga and S. Spaccapietra (eds.), Advanced Database Research and Development Series, Vol. 10, World Scientific, pp. 124-137 (2000).
- [2] Katsumi Tanaka, Akiyo Nadamoto, Machiko Kusahara, Taeko Hattori, Hiroyuki Kondo, Kazutoshi Sumiya, "Back to the TV:Information Visualization Interfaces Based on TV-Program Metaphors", Proc. IEEE International Conference on Multimedia and Expo 2000(ICME2000) pp.1229-1232, (2000).
- [3] W3C Synchronized Multimedia Home page, <http://www.w3.org/AudioVideo/> (2002).
- [4] 灘本明代, 服部多栄子, 近藤宏行, 沢中郁夫, 田中克己, "Web コンテンツの受動的視聴のための自動変換とスクリプト作成マークアップ言語", 情報処理学会論文誌:データベース, Vol.42, No.SIG1(TOD 8), pp.103-116 (2001).
- [5] Mahendren Munisamy, Kazutoshi Sumiya, Katsumi Tanaka, "Automatic Transformation of Video with Meta-Data into Web Contents", 情報処理学会研究報告, Vol.2002, No.67, 2002-DBS-128, pp.359-366 (2002).
- [6] 馬強, 角谷和俊, 田中克己, "WebTelop:放送と Web コンテンツの動的統合システム", 情報処理学会研究報告, Vol.2002, No.67, 2002-DBS-128, pp.169-176 (2002).
- [7] 寺田努, 塚本昌彦, 西尾章治郎, "Active Karaoke:アクティブデータベースを用いたカラオケの背景作成システム", 情報処理学会研究報告 2000-MUS-34, pp.73-78 (2000).
- [8] 湯本高行, 角谷和俊, 田中克己, "ストリームデータのための意味記述言語とその蓄積型テレビへの応用", 情報処理学会研究報告, Vol.2002, No.67, 2002-DBS-128-23, pp.177-184 (2002).
- [9] Qiang Ma, Hiroyuki Kondo, Kazutoshi Sumiya, Katsumi Tanaka, "Virtual Channel:A Filtering, Merging and Presenting Internet Broadcasting Channels", Proceedings of the Workshop on Organizing Web Space (WOWS'99) in conjunction with ACM Digital Libraries, Berkeley, CA, (1999).
- [10] Satoshi Sato, "Automated Editing of Hypertext Resume from the World Wide Web" Proceedings of 2001 Symposium on Applications and the Internet (SAINT 2001), pp.15-22 (2001).
- [11] Macromedia, "Flash オーサリングツール", <http://www.macromedia.com/jp/software/flash/> (2003).
- [12] "MPEG-21 Rights Expression Language FCD" (2002).
- [13] コンテンツ ID フォーラム, "cIDf Specification 2.0" (2002).
- [14] MPEG-7 Overview (version 8), <http://mpeg.telecomitalia.com/standards/mpeg-7/mpeg-7.htm> (2002).
- [15] TV-Anytime Forum: <http://www.tv-anytime.org/> (2002).

(注4): 客観的かつ正当なメタデータが付加されていると仮定する。