

MetaSMIL: ストリーム型コンテンツ統合のための記述言語

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 新たなコンテンツ利用の形態として、コンテンツ統合が注目されているが、多くの研究では Aggregator や Author などさまざまなユーザの存在が考慮されていない。我々は統合コンテンツ記述言語 MetaSMIL を提案することでこの問題を解決しようとしている。MetaSMIL は統合スクリプト、統合制約メタデータからなり、それぞれ Aggregator と Author がそれらを記述することで、権利処理を行った上での動的なコンテンツ統合が可能になる。本稿では、MetaSMIL の概要を説明すると共に、MetaSMIL の特徴的技術である、(1) 統合の基本単位の同定、(2) 統合制約、(3) 統合質問、に関する技術について述べる。

キーワード 情報統合、権利処理、統合制約、統合質問

MetaSMIL : A Description Language for Stream-Type Content-Integration

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 Content integration gets many attentions as a new style of content use. However, many of research about content integration do not consider about existence of various users, such as aggregator, author, etc. In order to solve these problems, we propose a new language MetaSMIL. MetaSMIL has two description : integration constraints metadata and integration script. Authors describe integration constraints metadata and aggregators describe integration script by using it. In this paper, we explain about overview of MetaSMIL and its characteristics: (1) unit identification, (2) integration constraints, and (3) integration query.

Key words information integration, digital right management, integration constraints, integration query

1. はじめに

新たなコンテンツ利用の形態として、コンテンツ統合が注目されている。しかし、多くの研究では、さまざまな立場のユーザの存在が考慮されておらず、権利の問題が解決されていない。我々は制作者側のユーザを統合コンテンツの素材に関して権利を持つ Author、統合コンテンツを制作する Aggregator に分類し、これらのユーザに対して、それぞれ統合制約メタデータ記述と統合スクリプト記述を提供することでこの問題の解決を目指している。我々はこれらの2つの記述を可能にする言語として、統合コンテンツ記述言語 MetaSMIL [1] を提案している。本稿では、MetaSMIL の概要を説明すると共に、以下の MetaSMIL の特徴技術について述べ、理論的な検証を行う。

- 基本情報単位の同定
- 統合制約
- 統合質問

また、MetaSMIL の位置付けについて考察する。

本論文の構成を述べる。第2節では MetaSMIL の概要と関連研究、第3節では基本的情報単位の同定について、第4節ではコンテンツの統合制約について、第5節では統合質問について説明する。第6節では MetaSMIL の設計および処理系のプロトタイプシステムについて述べる。第7節では MetaSMIL の位置付けについて考察する。第8節ではまとめと今後の課題について述べる。

2. MetaSMIL の概要と関連研究

2.1 背景

多様かつ膨大な数のコンテンツが利用できるようになり、その中から有用な情報をいかに入手するかが重要視されるようになってきている。Web からの知識を入手する研究としては以下のようなものがある。

- 知識発見 : Web から重要な内容を抽出することを主な目的

とする。

- 情報統合：Web の情報をよりリッチにすることを主な目的とする。

情報統合の研究でも知識発見によって得られた知識を統合するものもあり、これらの研究は互いに排反ではない。我々はこの情報統合の中でさらにコンテンツそのものの位置づけに注目した。コンテンツは情報の一種の表現の手段であり、現在の知識発見では発見しきれないような多くの情報を含んでいると共に、優れた演出技法も含んでいる。そのため、我々はコンテンツそのものを用いた情報統合であるコンテンツ統合に重要性を見出した。コンテンツ統合とは、複数のコンテンツを空間的、時間的に配置することで新たなコンテンツを生成することであると定義する。

このような定義に従うならば、HTML [2] や SMIL [3] もコンテンツ統合言語として位置づけることができる。我々が提案する MetaSMIL と統合を実現する既存の言語との相違点をまとめると表 1 のようになる。

表 1 他の統合言語との比較

	HTML	SMIL	MetaSMIL
複数コンテンツの単純な並列表示	○	○	○
複数コンテンツの同期	×	○	○
検索機能	×	×	○
複数制作者モデルへの対応	×	×	○

HTML ではフレーム機能を用いることにより、複数のコンテンツの同時表示という最も単純なコンテンツ統合が可能である。また、SMIL ではそれに加えて、コンテンツの時間的な同期が表現できる。つまり、これらの 2 つの言語により形式的なコンテンツ統合は可能である。しかし、コンテンツ統合には実際これらでは不十分である。それは、HTML, SMIL が前提としているユーザモデルがコンテンツ統合においては不十分だからである。HTML, SMIL ではコンテンツ制作に用いられる素材に関する権利問題が解決していることを前提としているため、形式的に制作者が単一とみなすことができる。(単一制作者モデル、図 1 左参照。)しかし、実際は素材の制作者、素材に登場する人物、素材を組み合わせる制作者などさまざまな権利保持者が存在し、その権利が複雑に絡み合っている。これを単一制作者モデルで扱うことは非常に困難である。そこで、我々は制作者を以下の 2 つに分けたモデルを導入する。

- Author：素材に関する権利保持者
- Aggregator：コンテンツを実際に制作するユーザ

これを図示したものが図 1 右である。

このように Author と Aggregator の間で調停を行いながら、いかに意味的な統合を実現するかが重要である。

2.2 目的とアプローチ

我々は Author と Aggregator の間で調停する技術として、以下に注目した。

- 基本情報単位の同定
- 統合制約
- 統合質問

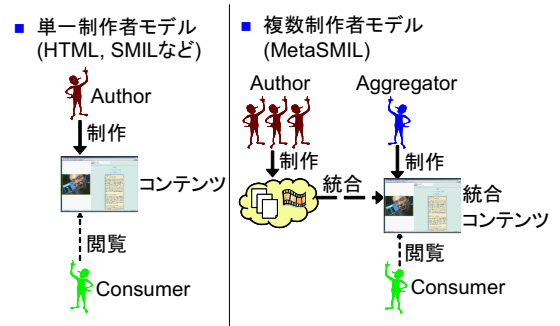


図 1 ユーザモデルの違い

Author は自分のコンテンツの利用条件を統合制約として記述し、Aggregator は統合質問を記述する。統合は統合質問を処理することによって行われ、統合に用いられるユニットには統合制約が反映される。また、ユニットの同定は統合質問に統合制約が加わることによって行われる。

これらのスキームを実現するために我々は MetaSMIL [1] を提案している。MetaSMIL は 2 つの記述から成り立っている。

- (1) 統合制約メタデータ：Author が自分のコンテンツが統合に使われるときの制約を記述する。
- (2) 統合スクリプト：Aggregator が統合コンテンツを記述するのに用いる。

前者は Author が自分のコンテンツを統合する条件についてメタデータとして記述を行う。後者は、Aggregator が統合コンテンツを記述するためのものであり、SMIL のようにストリーム型のコンテンツを記述でき、さらに検索機能も利用できる。MetaSMIL では図 2 のように、Author, Aggregator がそれぞれの記述を MetaSMIL の処理系に入力することで、処理系が検索・権利処理を行い、SMIL コンテンツが生成されることを想定している。

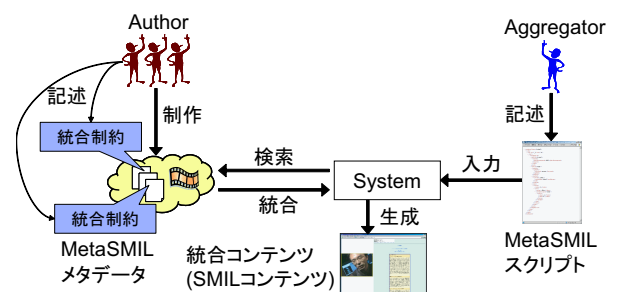


図 2 MetaSMIL のイメージ

それぞれの記述では以下のようなことが実現できる。

- 統合制約メタデータの例: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 3 参照)
- 統合スクリプトの例:ビデオと Web ページの動的統合
馬らはニュース映像に同期させて、関連する Web ページを動的に検索、呈示する WebTelop [4] を提案している。MetaSMIL の統合スクリプトによって、WebTelop のアルゴリズムの簡易

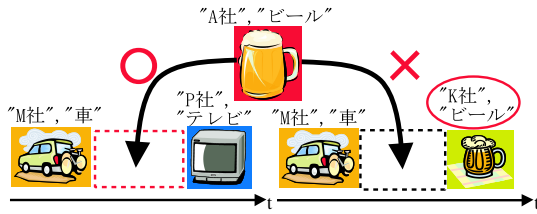


図3 隣接するCMに対する制約

版が表現でき、ビデオとWebページの動的統合、表示することが可能である。

2.3 関連研究

まず、MetaSMILの統合制約メタデータ記述機能についてであるが、デジタル権利処理技術としてMPEG-21 [5], Creative Commons [6] などコンテンツの(2次)利用のための条件の記述方式が提案されている。これらでは、コンテンツ統合時の統合される相手に対する条件などが記述できない。また、統合の単位についても「任意の30秒以内の区間」など柔軟な定義ができない。

次に、MetaSMILの統合スクリプト機能についてであるが、単にURIを指定するだけの検索機能を持たない統合を実現する手法は、HTML, SMIL などさまざまである。また、キーワードを具体的に記述しておく方式の統合は、Multimedia Formatting Vocabulary [7] によっても実現できる。しかし、陽にキーワードを与えることなく、統合の相手のコンテンツから問合せを生成し、検索を行う機能する機能は馬らによるWebTelop [4] や Henzinger らによる Query-Free News Search [8] として実現されているが、これらはアプリケーションとして実現されている。制作者がその機能を使って独自のアプリケーションを設計できるようなフレームワークはMetaSMIL 以外には提供されていない。

このようにMetaSMILは以下の3点において有用性がある。

- 統合される単位について柔軟な定義が可能である。
- 統合される相手についての条件を制約として記述できる。
- 陽にキーワードを与えることなく、統合の相手のコンテンツから問合せを生成し、検索を行う機能を備えている。

3. 基本情報単位の同定

3.1 ユニットの定義/分類

基本情報単位ユニットを以下のように定義する。

- URIで表現されるリソースとそれを時間的に分割したもの
- それらの集合

さらにユニットは、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u.propertyと表記する。ユニットの間には時間や空間などの関係Rを定義することが可能である。

ユニットには以下の2つ分類基準が存在すると考えられる。

- atomic/compound: 単一のユニット/複数のユニットの合成
- absolute/conditional: 一意性が保証された決定演算/条件指定や質問による一意性が保証されない決定演算

ここで非時系列のユニットではURI、時系列を持つユニットではURIと開始時間、終了時間が唯一に定まる場合、このユニットは一意であると定義する。また、これらの組み合わせによって、以下の4通りに分類できる。

- atomic-absolute:
 - URIと時間指定による一意な単一のユニット
 - 「例:URIが p_1 で表されるファイルで開始時間10s, 終了時間30sまでのもの」
- atomic-conditional:
 - URIと条件指定または質問によって決まる単一のユニット
 - 「例:URIが p_2 で表されるファイルで長さが30s以下の任意の区間」
- compound-absolute:
 - 任意のユニットの列挙によるユニットの集合
 - 「例:URIが p_1, p_2, \dots, p_n で表されるファイル」
- compound-conditional:
 - 条件指定または質問処理によるユニットの集合
 - 「例:キーワードkを持つ任意の区間」

atomic-conditionalではユニットの一意性は保証されない。また、compound-absoluteでは集合を構成するユニットにatomic-conditionalが含まれる場合一意性が保証されない。ただし、atomic-absoluteのみから構成される場合は一意性が保証される。さらに、compound-conditionalでは構成するユニットにかかわらず一意性が保証されない。これらの一意性が保証されないユニット定義は統合質問の処理時のユニットの選択に自由度を与えるものでこれを柔軟なユニットと定義する。

3.2 ユニットの一意な決定

柔軟なユニットは統合質問処理時に一意に決めることができる。その例を示す。まず、一意性が保証されていないユニットとして、Authorは「URIが p_1 で、長さが30s以下の任意のユニット」を定義する。これはatomic-conditional unitにあたる。ここで、URIが p_1 で表されるファイルは図4の上側のように区間毎にキーワードが設定されているとする。Aggregatorは統合質問(の一部)として、「キーワード=「イチロー」」であり、なるべく長さの長いもの」という指定をしたとする。この質問を処理することで図4下側のようにユニットを一意に決めることができる。

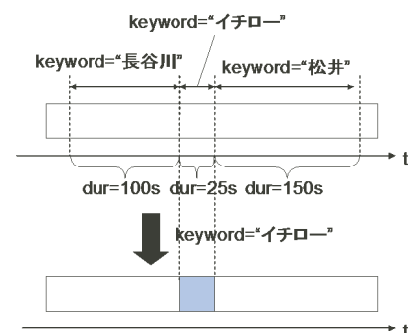


図4 ユニットの一意な決定の例

4. コンテンツの統合制約

統合制約には、以下の2つがある。

- ユニットの同時利用の要求/禁止
- ユニットの条件指定

前者は前述した **compound unit** として定義するかもしれないが以下で述べる **requires/prohibits** 関連を用いて表現する。また、後者はユニット間の関係とプロパティを条件に用いたユニット制約で表現する。

4.1 requires/prohibits 関連

4.1.1 定義

ユニットの同時利用条件/禁止条件を表す関連として、**requires** 関連 (*req*), **prohibits** 関連 (*prh*) を定義する。 x, y をユニットとすると、*req, prh* は以下のようなことを意味する。

- $x req y$: ユニット x を使用する場合は y を含むユニットを同時に使用しなくてはならない。
- $x prh y$: ユニット x を使用する場合は y を含むユニットは同時に使用してはならない。

4.1.2 requires/prohibits 関連の性質

req は以下の関係を満たす。ただし、 x, y, z はユニットであり、 $y' \subseteq y$ である。

$$\text{反射律: } x req x \quad (1)$$

$$\text{推移律: } (x req y) \wedge (y req z) \Rightarrow x req z \quad (2)$$

さらに $x' \subseteq x$ とすると、以下が成り立つ。

$$\text{部分反射律: } x req x' \quad (3)$$

$$\text{部分推移律: } (x req y) \wedge (y' req z) \Rightarrow x req z \quad (4)$$

部分推移律の証明を以下に与える。

$$\begin{aligned} & (x req y) \wedge (y' req z) \\ \Rightarrow & (x req y) \wedge (y req y') \wedge (y' req z) \quad (\text{部分反射律}) \\ \Rightarrow & (x req y) \wedge (y req z) \quad (\text{推移律}) \\ \Rightarrow & x req z \quad (\text{推移律}) \end{aligned}$$

また、 $\forall x' \subseteq x, x' req x$ が成り立つとき、 x は統合の最小ユニットであると定義できる。

続いて、*prh* の性質について述べる。以下が成り立つ。

$$x' prh y \Rightarrow x prh y \quad (5)$$

$$(x prh y) \wedge (x prh y') \Rightarrow x prh y' \quad (6)$$

$$(x req y) \wedge (y prh z) \Rightarrow x prh z \quad (7)$$

$$(x req y) \wedge (y' prh z) \Rightarrow x prh z \quad (8)$$

式 (8) の証明を以下に与える。

$$\begin{aligned} & (x req y) \wedge (y' prh z) \\ \Rightarrow & (x req y) \wedge (y req y') \wedge (y' prh z) \quad (\text{部分反射律}) \\ \Rightarrow & (x req y) \wedge (y prh z) \quad ((8)) \\ \Rightarrow & x prh z \quad ((8)) \end{aligned}$$

定義より、以下は常に成り立たない。

$$x prh x' \quad (9)$$

$$(x req y') \wedge (x prh y) \quad (10)$$

v, w をユニットとしたとき、 $v req w$ または $v prh w$ の形式の条件の集合 F が与えられたときに、その条件に式 (1)–(8) を適用して、(9) または (10) が導出された場合は x は統合に用いることができないことを意味している。

4.1.3 統合可能性の検証

v, w をユニットとしたとき、 $v req w$ または $v prh w$ の形式の条件の集合 F の下でユニット x, y が統合可能であるという関係は、「 F が $x prh y$ を論理的に含意しない」と定義し、 $x r y$ で表す。また、統合不可能を「 F が $x prh y$ を論理的に含意する」と定義し、 $x \bar{r} y$ で表す。ここで、 $x r y$ が成り立つかどうかを判定する方法について述べる。以下のように定義する。

$$\begin{aligned} & req^*(u) \\ = & \{v | u req v \text{ が論理的に含意される, } v \text{ はユニット} \} \quad (11) \end{aligned}$$

$$\begin{aligned} & prh^*(u) \\ = & \{u | u prh v \text{ または } v prh u \text{ が論理的に含意される, } \\ & \quad v \text{ はユニット} \} \quad (12) \end{aligned}$$

ただし、 u はユニットとする。式 (11), (12) の定義を用いて、 $req^*(x), req^*(y), prh^*(x), prh^*(y)$ を計算し、包含関係が以下のいずれかを満たすときは $x \bar{r} y$ である。

$$\begin{aligned} & req^*(x) \cap prh^*(x) \neq \phi, req^*(y) \cap prh^*(y) \neq \phi, \\ & req^*(x) \cap prh^*(y) \neq \phi, req^*(y) \cap prh^*(x) \neq \phi \end{aligned}$$

さらに以下のことが言える。

$$\begin{aligned} & req^*(x) \cap prh^*(x) \neq \phi \Rightarrow \forall v, x \bar{r} v \\ & req^*(y) \cap prh^*(y) \neq \phi \Rightarrow \forall v, y \bar{r} v \end{aligned}$$

4.2 ユニット制約

前節で述べた **requires/prohibits** 関連では、具体的なユニットを対象にしていたが、特定の条件を満たすユニットを **requires/prohibits** 関連の対象にすることも可能である。これらの条件はユニット間の空間的・時間的な関連とプロパティを条件に用いて表現する。

空間的な関連ではユニット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 (**overlap**) によって表現する。図 5 に空間的関連のイメージ図を示す。

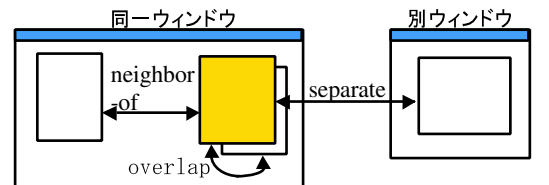


図 5 空間的統合

時間的な関連では、ユニットの時間的な位置関係を Allen の時区間関連 [9] に時間差パラメータ $diff$ を設けた関連によって表現する。また、プロパティとしては、以下のようなものを用意している。

- $keyword$: 主要なキーワード
- $author$: 著者
- dur : 時間の長さ
- ...

これらのプロパティに加えて、以下のような関数を定義する。

- $sim(u_1, u_2)$: u_1, u_2 の類似度を返す。
- $st(u), fin(u)$: 開始時間, 終了時間
- $diff(t_1, t_2)$: 時間差 $t_2 - t_1$
- ...

これらによって、ユニットの条件を記述する。例えば、「自分との類似度が 0.5 以上のコンテンツを同時に表示しなくてはならない」という条件は以下のように記述できる。

$$u_1 \text{ req } u \mid u : (sim(u_1, u) \geq 0.5) \wedge (u \text{ neighbor } - \text{ of } u_1)$$

ただし、ユニット u が条件 $conditions$ を満たすことを以下のよう
に表記するものとする。

$$u : conditions \quad (13)$$

また、あるユニット v がある条件 $conditions$ を満たすユニットとの同時利用を必要/禁止するという条件を以下のように記述するものとする。

$$v R u \mid u : conditions \quad (R \text{ は } req \text{ または } prh) \quad (14)$$

$requires/prohibits$ 関連とユニット制約を合わせて、Author が記述する制約は一般に以下の式を基本形とし、その論理和または論理積をとったものになる。

$$u_0 \text{ req } u \mid u : conditions \quad (15)$$

$$u_0 \text{ prh } u \mid u : conditions \quad (16)$$

式 (15), (16) はそれぞれ式 (17), (18) と等価である。

$$(u_0 \text{ req } u_1) \vee \dots \vee (u_0 \text{ req } u_n) \quad (17)$$

$$(u_0 \text{ prh } u_1) \wedge \dots \wedge (u_0 \text{ prh } u_n) \quad (18)$$

ただし、 $u_i (i = 1, \dots, n)$ は $conditions$ を満たす具体的なユニットである。統合可能性を検証する際には式 (15), (16) から式 (17), (18) を求め、具体的なユニットのみの形にしてから 4.1.3 のアルゴリズムで検証を行う。

5. 統合質問

5.1 MetaSMIL の問合せ機能

MetaSMIL では明示的に統合に用いるユニットの URI を指定することなく、クエリを記述することによって、ユニットの選択を行うことができる。MetaSMIL の検索質問では、具体的にキーワードを指定する明示的な検索質問が表現可能である。例えば、 $keyword="イチロー"$ であるユニットを取得する場合は、

引数で指定されたキーワードを既存のサーチエンジンに投げて、コンテンツを取得することなどが可能である。MetaSMIL における問い合わせであるが、基本的には以下の 2 つによって行う。

- 絶対条件の指定：
表示されるコンテンツが必ず満たすべき条件
- 優先条件の指定：
表示されるコンテンツが優先すべき条件

効率的にコンテンツの取得、フィルタリングを行うために絶対条件部分を $Search, Restrict$ に分け、優先条件指定部分を $Rank$ とし、以下の 3 つの部分からなるクエリを用いる。

(1) $Search$: 統合するユニットの候補集合の既存のサーチエンジンを用いた取得

(2) $Restrict$: 統合されるユニットが満たすべき条件の検証

(3) $Rank$: 統合されるユニットの候補のランキングの計算
さらに $Search$ で用いたサーチエンジンの出力順序は、 $Rank$ でランキング関数の値が同じになった場合のランキングの基準に用いられる。これを式で表すと次のようになる。

$$u = f_{Rank}(f_{Restrict}(S_{Search})) \quad (19)$$

$f_{Restrict}, f_{Rank}$ 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。(図 6 参照)

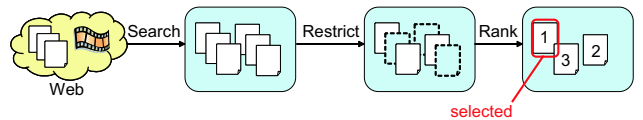


図 6 統合のプロセス

MetaSMIL スクリプトでは表示されるユニットに ID を与えることが可能であり、その ID を用いて、表示されるユニットのプロパティを参照し、問い合わせに利用することができる。ここで例を示す。図 7 では、ユニット u_1, v_1 は並べて、同期して表示されるとする。また、 v_1 の URI は具体的に与えられており、 u_1 には v_1 のキーワードを用いて、検索を行って決定するというクエリが付加されている。また、 u_2 には u_1 と v_3 のプロパティを利用して検索を行うタイプのクエリが付加されている。

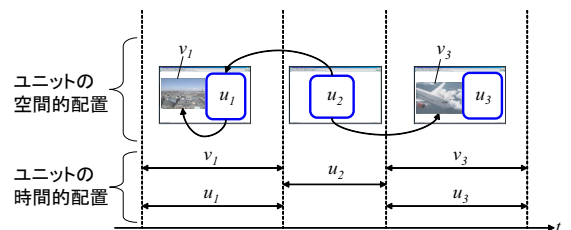


図 7 クエリの例

上記のような記述を用いた場合は、例えば、 v_1 を変更することによって、 u_1 も変化する。つまり、このような記述によって、コンテンツ統合のテンプレートを実現することが可能である。(図 8 参照)

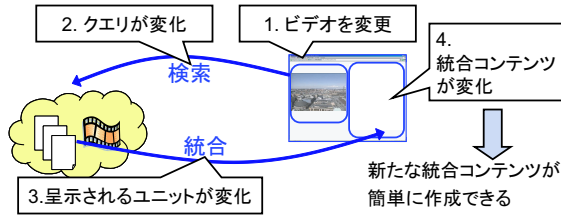


図8 統合のテンプレート

```
<metasmil type="metadata">
<unit (id="id")? >
<target uri="URF" >
(<start>time</start>)?
(<finish>time</finish>)?
(<dur>time</dur>)?
(<keyword>string</keyword>)?
</target>
<restrict>condition</restrict>
</unit>
</metasmil>
```

5.2 統合制約の統合質問への反映

統合質問の Restrict プロセスに統合制約記述を反映させることにより、Author の意図を Aggregator が設定した統合コンテンツに反映させる表示されるコンテンツ (ユニット) を u_{Ag} とおくと、Aggregator が記述する Restrict プロセスは以下のように表現される。

$$u_{Ag} R u \mid u : conditions \quad (R \text{ は } req \text{ または } prh)$$

ここで、Search プロセスで u_{Ag} の候補集合 S_{Search} が得られているとする。ここで $U = S_{Search}$ とおく。 $u_{s(i)} \in S_{Search}$ ($1 \leq i \leq n$) のそれぞれに対して、式 (20) において u_{Ag} を $u_{s(i)}$ におきかえ、

$$u_{s(i)} R u \mid u : conditions \quad (R \text{ は } req \text{ または } prh) \quad (20)$$

とする。この条件と Author が定義した制約を合わせた条件集合、 $u_{s(i)}$ が統合不可能であることを論理的に含意すれば、 $U \leftarrow U \setminus \{u_{s(i)}\}$ とする。

$1 \leq i \leq n$ で以上の手順を行った後に $f_{Restrict}(S_{Search}) = U$ とする。以上のようにして、Author が記述した統合制約が、Aggregator が設計した統合コンテンツに反映される。

また、Author と Aggregator でユニットの定義は一般には一致するとは限らず、その場合上記のアルゴリズムは利用できない。そこでユニットが一致しないときに Aggregator が定義したユニットを Author が定義したユニットのうちに対応させることによって、この問題を解決する。Author が定義したユニットの集合を S_{Au} と定義し、ユニット集合 S のユニットの連結を $\bigcup S$ と定義すると、Aggregator が定義したユニット u_{Ag} には以下を対応させればよい。

$$u_{Au} = \bigcup \{u \mid u \in S_{Au}, u_{Au}.dur = min\} \quad (21)$$

6. MetaSMIL の実装

6.1 言語の設計

MetaSMIL は統合制約メタデータと統合スクリプトからなる述べたが、それらの中で前述の統合制約と統合質問を XML タグで実現する方法について述べる。

6.1.1 統合制約記述

統合制約記述はユニットの同定技術と統合制約を用い、target タグ内にユニットの同定を記述し、restrict タグ内に統合制約を記述する。ユニットの同定としては atomic-absolute, atomic-conditional unit が利用でき、以下のように開始時間 (start)、終了時間 (finish)、時間の長さ (dur)、キーワード (keyword) を指定することができる。

統合制約は 3 節で述べたように u_{Au} に対する制約は以下のように表される。

$$u_{Au} R u \mid u : conditions \quad (\text{ただし, } R \text{ は } req \text{ または } prh)$$

condition は以下のように記述する。

```
condition::=<and>condition*</and> | <or>condition*</or> |
<not>condition</not> | property condition |
<unit>(Relationship or Property)*</unit>
```

req/prh 関連はそれぞれ、<unit><unit> タグ、<not><unit></unit></not> タグに対応し、その中の記述がユニット制約に対応する。

2 節の CM に関する制約の例は以下のように記述できる。

```
<unit id="cm">
<target>URI</target>
<restrict>
<not>
<unit>
<keyword>K社</keyword>
<keyword>ビール</keyword>
<or>
<before id="cm" diff="0s"/>
<after id="cm" diff="0s"/>
</or>
</unit>
</not>
</restrict>
</unit>
```

6.1.2 統合スクリプト記述

統合スクリプトは以下の 2 つの部分からなる。

- レイアウト記述部分
- 統合質問部分

レイアウト記述部分は、図 5 で示されている空間的関連と以下のような時間的関連を用いる。

- seq: 複数のユニットを時間的に連結する。
- par: 複数のユニットを同期させる。

これらは、SMIL でも表現できる時間的/空間的なレイアウトをより直観的に表したものである。これらの関連を用いて、統合スクリプト (script) 以下のように記述する。

```
script::=query<rel>script*</rel>
(rel=par | seq | neighbor-of | overlap | separate)
```

次に統合質問部分 (query) であるが、URI を明示的に記述するかもしくは前節で述べたように 3 つの部分に分けて質問を記述する。

```

query::=<unit>uri</unit> |
  <unit id="id">
    <search>search関数</search>
    <restrict>condition</restrict>
    <rank>rank関数</rank>
  </unit>

```

また、関数はタグ名に関数名 (*f_name*) をとり、それらのタグの間に引数を記述する。

```

function::=<f_name>function*</f_name> |
  <value>number or string</value> |
  <variable name="name"/>

```

condition は統合制約と同じ記法を用いる。

2 節で述べた WebTelop の簡易版の例は以下のように記述できる。

```

<metasmil type="script">
  <seq>
    <for var="_i" max="n"> —— 反復記述
      <par>
        <neighbor-of> Video content
          <unit id="scene">
            <search>
              <getvideosegment>URI</getvideosegment>
            </search>
            <rank> <select>_i</select> </rank>
          </unit>
          <unit id="web">
            <search>
              <getURI>
                <keyword>scene</keyword>
              </getURI>
            </search>
            <restrict>
              <unit id="_this">
                <mediatype>html</mediatype>
              </unit>
            </restrict>
            <rank> <date>web</date> </rank>
          </unit>
        </neighbor-of> Web content
      </par>
    </for>
  </seq>
</metasmil>

```

6.2 処理系のプロトタイプ

MetaSMIL の処理系としては、MetaSMIL のスクリプトから SMIL 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 つの部分からなる。

- レイアウトタグ変換部 (以下, LTC)
- 呈示コンテンツ決定部 (以下, PCD)

LTC は空間的/時間的レイアウトに関するタグを SMIL タグに変換する。PCD は検索質問に関するタグから検索プロセスを生成する。また、ユニット毎に定義された制約をとりこみ、その制約のチェックも行う。出力として、URI を返し、LTC に入力する。これを図に表すと図 9 のようになる。

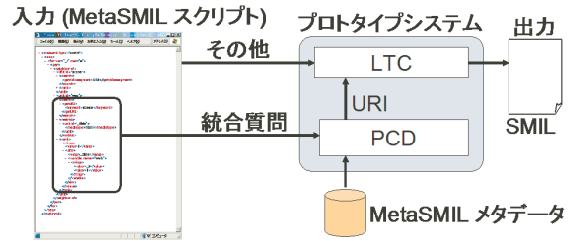


図 9 プロトタイプシステム

7. 考 察

MetaSMIL は以下の 2 つからなる。

- 制約記述言語 (メタデータ)
- マルチメディアコンテンツ記述言語 (スクリプト)

これらのそれぞれに対して他言語との比較を行うと共に、MetaSMIL の位置づけを確認する。

7.1 メタデータ

まず、メタデータについて考察する。他の代表的なメタデータとして、MPEG-7 [10], MPEG-21, Creative Commons と比較したものを図 10 に示す。これは縦軸に内容の記述能力、横軸に権利記述能力を示している。それぞれ直交する軸より下、左側は記述能力がないことを示している。MPEG-7 は内容記述のみが可能で、権利記述はできない。また、MPEG-21 は内容記述と権利記述が可能である。Creative Commons は内容記述はできないが、権利記述は可能である。ただし、MPEG-21, Creative Commons の権利記述いずれもコピーや印刷、再頒布などの従来の利用についてのものに限定されている。それに対して MetaSMIL のメタデータでは統合に関する条件が記述できる。他のメタデータで記述できるものが記述できないという弱点もあるが、MetaSMIL のメタデータは統合に対象を絞っており、既存技術とは競合しない独自のものである。

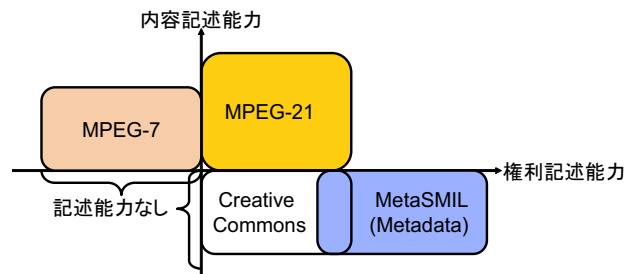


図 10 他のメタデータ記述言語との比較

7.2 スクリプト

続いて、スクリプトであるが、これはレイアウト記述と問い合わせ記述の 2 つからなる。前者は SMIL と競合するものであり、後者は既存の問い合わせ言語と競合するものである。そのため、MetaSMIL のスクリプトと同じ目的を達成するためには既存の技術を拡張するという選択肢も十分に考えられた。それらとの得失について述べる。

7.2.1 SMIL の拡張によるアプローチ

SMIL を拡張して、問い合わせ機能を実現するアプローチを採用しなかった理由について述べる。MetaSMIL は Aggregator, Author の双方のユーザを対象とした言語であるため、スクリプトとメタデータの親和性が高い必要がある。しかし、SMIL はデザイン重視の言語であり、利用条件に関する記述がなされることが前提とされていない。例えば、MetaSMIL のメタデータでは統合の状況を条件として記述することが可能である。しかし、SMIL で記述するような空間的レイアウトでは統合の状況を記述することが非常に困難であり、MetaSMIL のメタデータにおける空間的なレイアウト (overlap, neighbor-of, separate) の方が十分な抽象化がされており、統合の状況を表現しやすくと考えられる。このようなメタデータとの親和性を考えたときに、SMIL を拡張するよりも新たな言語を設計した方が容易であると考えられる。

7.2.2 XQuery の拡張によるアプローチ

問い合わせの記述であるが、既存の問い合わせ言語を利用しなかった理由について述べる。コンテンツ統合では対象が Web 上のコンテンツということで膨大である。そのため、MetaSMIL ではサーチエンジンの出力から候補を取得し、そこから絞り込みを行い、問い合わせの答えとしている。このようなアプローチをしているものはないため、新しい言語を作成したのであるが、我々にはもうひとつの選択肢として XQuery [11] の拡張に注目している。XQuery は XML の問い合わせのための言語であり、一見、MetaSMIL の問い合わせ機能の代替としては不適當であるかのように思える。XQuery で用いられる制約はパス制約であるが、統合の状況を表現するには不十分であると考えられる。しかし、MPEG-7 などの構造化されたメタデータを用いた場合、XQuery のパス制約は非常に有用である。現在の MetaSMIL では内容に関するメタデータは非常に単純なものを想定しており、ビデオに関する検索についても詳細については述べていない。(現在の実装では対応しておらず、言語仕様レベルのみの対応となっている。)そのため、今後、ビデオに関する検索を実装していくためにも、そのため、XQuery の拡張というアプローチについては再考の余地があると考えている。

7.3 MetaSMIL の位置づけ

MetaSMIL はコンテンツ統合を実現するために、Aggregator と Author の双方を支援する言語であり、コンテンツ統合のフレームワークであるとも言える。MetaSMIL はスクリプトとメタデータからなっているため、一見、別々の言語であるかのように思えるかも知れないが、決してそうではない。2つの記述は親和性が高く、また、この2つの記述が両方が揃うことでコンテンツ統合のフレームワークとして機能している。また、MetaSMIL は一部の既存の言語で表現できるものが表現できないため、その点が欠点とされることもある。しかし、MetaSMIL は言語としてもプロトタイプ的な側面があるが、我々が提案する動的なコンテンツ統合を実現するのに十分な能力を持っている。

8. おわりに

本稿では、以下の記述からなる統合コンテンツ記述言語 MetaSMIL について説明した。

- 統合スクリプト記述
- 統制约約メタデータ記述

また、以下の MetaSMIL の特徴技術について述べ、理論的な検証を行った。

- 基本情報単位の同定
- 統制约約
- 統合質問

今後の課題としては次のようなものが考えられる。

- requires/prohibits 公理系の完全性の証明
- ユニットの条件判定の効率のよいアルゴリズムの開発
- ビデオに対する問い合わせ手法についてのさらなる検討

謝 辞

本研究の一部は、平成 15 年度科研費基盤研究 (A)(2) 「モバイル環境におけるコンテンツのマルチモーダル検索・呈示と放送コンテンツ生成」(課題番号: 14208036, 代表: 田中克己)、平成 15 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)、平成 15 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」及び 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] Takayuki Yumoto, Qiang Ma, Kazutoshi Sumiya, and Katsumi Tanaka, "A Dynamic Content Integration Language for Video Data and Web Content", Proceedings of the 4th International Conference on Web Information Systems Engineering (WISE2003), pp.83-92 (2003).
- [2] HyperText Markup Language (HTML) Home Page, <http://www.w3.org/MarkUp/> (2003).
- [3] W3C Synchronized Multimedia Home page, <http://www.w3.org/AudioVideo/> (2003).
- [4] Qiang Ma and Katsumi Tanaka, "WebTelop: Dynamic tv-content augmentation by using web pages", Proceedings of IEEE International Conference on Multimedia and Expo (ICME2003) (II), pp.173-176 (2003).
- [5] MPEG-21 Overview v.4, <http://mpeg.telecomitalia.com/standards/mpeg-21/mpeg-21.htm> (2002).
- [6] Creative Commons, <http://www.creativecommons.org> (2003).
- [7] Jacco van Ossenbruggen, Lynda Hardman, Joost Geurts and Lloyd Rutledge, "Towards a multimedia formatting vocabulary", Proceedings of the Twelfth International World Wide Web Conference (WWW2003), pp.384-393 (2003).
- [8] Monika Henzinger, Bay-Wei Chang, Brian Milch and Sergey Brin, "Query-Free News Search", Proceedings of the Twelfth International World Wide Web Conference (WWW2003), pp.1-10 (2003).
- [9] Allen J. F., "Maintaining Knowledge about Temporal Intervals", Communications of the ACM, Vol.26, No.11, pp.832-843 (1983).
- [10] MPEG-7 Overview (version 8), <http://mpeg.telecomitalia.com/standards/mpeg-7/mpeg-7.htm> (2003).
- [11] XML Query (XQuery), <http://www.w3.org/XML/Query> (2004).