

ビデオハイパーメディアにおけるシナリオ構造の拡張

坂田 哲夫 佐藤 哲司

NTT 情報通信研究所

{sakata, satoh}@isl.ntt.jp

〒 238-03 横須賀市武 1-2356

あらまし VHM(ビデオハイパーメディア)は、映像・音声等のメディアデータを断片化してノードとし、シナリオと呼ぶデータ構造に格納することでインタラクティブなハイパーメディアを実現する。従来のVHMでは、シナリオ再生時の画面レイアウトをアプリケーションが決めているため、レイアウトデザインの柔軟性が不足していた。また、再生時のさまざまな環境に応じてレイアウトを指定することも困難で、シナリオの可用性を制約していた。本稿では、シナリオ再生時のレイアウト単位を記述するデータ構造シーンを追加し、シーン毎にレイアウト候補から適切なものを選択することでこの問題を解決する手法を提案する。

キーワード ハイパーメディア, マルチメディアデータベース

An expansion of scenario structure in Video Hypermedia

Tetsuo SAKATA , Tetsuji SATOH

NTT Information and Communication Laboratories

{sakata, satoh}@isl.ntt.jp

1-2356 Take Yokosuka-Shi Kanagawa 238-03 Japan

Abstract VHM (video hypermedia) is an interactive hypermedia of which nodes are defined from fragments of media data such as video or sound and stored in a scenario. In the former VHM system, a layout of a played back scenario is described in an application program, so that the layout is fixed and, it restricts scenario's availability among various play-back environments. In this paper, we propose a method to select suitable layout from candidates defined in new data structure 'scene' in scenario.

Key words Hypermedia, Multimedia Database

1 はじめに

近年、ハードウェアの低廉化・通信の高速化などを背景に、映像・音声など複数の種類のメディアをつかって情報を表現するマルチメディア情報システムの研究開発が盛んに行なわれている。

筆者らが提案しているビデオハイパーメディア (VHM) は、実写映像のデータを断片(カット)化して蓄積・管理しておき、再生時に再構成することで、ユーザインタラクティブなハイパーメディアを実現している [1]。

VHM では、マルチメディア情報はシナリオという構造で表現される。シナリオは、カットを並列・選択の関係を表現するリンクで結び付けるグラフ構造をもっている。シナリオのリンクに付与されている選択条件を用いることで、ユーザインタラクションと、再生時の条件に応じた提示データの切替とを実現する。

これらの機能を用いて、ビジュアル電子図書館のようなアプリケーションが実現されている [2]。しかし、このシナリオには、再生時のレイアウトの指定がないため、次のような問題を抱えている。

- シナリオ再生時のレイアウトはアプリケーションに組み込まれており、柔軟性に欠ける。
- 再生環境に応じたレイアウトをシナリオからは指定できないので、異なる環境でのシナリオの可用性が不十分である。

本稿では、シナリオを拡張してレイアウト情報を記述可能とし、状況に応じて変更される提示データを適切にレイアウトすることで、レイアウトの柔軟性とシナリオの可用性を改善する手法を提案する。

2 シナリオ拡張の方針

従来のシナリオの不足点 VHM のシナリオは、条件つきリンクを用いてカットどうしの並列関係・選択関係を記述している。これによって、ユーザインタラクションと環境への順応性を実現している [1]。これはマルチメディアのプレゼンテーションの論理的な側面の記述と言える。

シナリオを再生するには、論理的な側面に加えて、映像カットが出力される際の画面上の位置や重なり、音声カットの出力チャンネルといった、いわば物理的な側面の記述が必要である。シナリオの物理的側面の情

報をレイアウトと呼ぶことにすると、従来のシナリオ構造にはレイアウトの記述機能が不足していた。

2.1 シナリオ拡張の方針

シナリオ構造を拡張してレイアウト情報を記述すれば、画面上での映像の配置などまで含めた、シナリオ再生時の見え方を指定できるので、レイアウトの柔軟性が増す。また、再生環境によってそれに適したレイアウトを指定できれば、シナリオの可用性が高まる。

シナリオの環境に対する可用性を高めるためには、環境に応じて、再生されるべきカット列をシナリオの中から選択して、レイアウト情報に従って再生する必要がある。ここで環境とは、動画プレイヤーや音声プレイヤーの有無や性能など、再生に利用できる資源の総体を指す。

そのためには、シナリオ再生のある時点において、どのカットが並列して再生されているかを陽に記述する手段と、再生される環境に応じてそれらをレイアウトするための情報を記述する手段とが必要となる。前者をシーン、後者をレイアウトと呼ぶ。

更に、環境の持っている限られた資源の中で、なるべくシナリオの内容を豊かに伝えるようにカット列を選択することが、再生時には求められる。そのためには、もっとも良い再生を可能とするレイアウト情報を選択して、それに従って、シーンからカット列を選択する方式が必要となる。

3 シナリオ構造の拡張

従来のシナリオは、カットまたはシナリオをリンクで結んだものであったが、前述のように、シナリオにおける並列再生を陽に記述するため、シナリオ構造にシーンを導入する。シナリオは、シーンまたはシナリオをリンクで結合したものと定義される (図 1)。シーン導入の目的から、一つのシナリオにおいて一度に再生されるのは一つのシーンまたはシナリオに限定する。シーンは並列再生されるカット列を明示する。並列再生されるカット列が決まると、その列が、どのように再生されるのかを定めることが出来る。再生のされ方を決めるのは、シーンに対して付加されるレイアウト情報と再生時の環境とである。

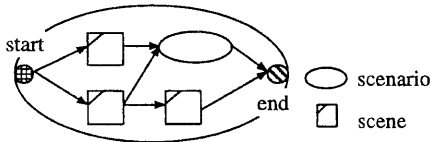


図 1: シナリオの例

3.1 レイアウト情報

レイアウト情報は、あるシーンがある環境下で再生する時の、カット列毎の画面上への割り当てを決める。カット列が再生される時の方法はレイアウトのチャンネルに記述される。一つのカット列に対して、一つのチャンネルを割り当てる(図2)。

チャンネル情報は、レイアウトする位置に加えて、カット列を再生するのに必要な素材情報の型を記述している。すなわち、動画/静止画であれば画面上のサイズ、音声であればサンプリングレート・トラック数である。例えば、図2の Layout No.1 の video.1 というチャンネルは、次のような属性を持つ。

属性	値
チャンネル番号	1
型	video: 640 × 480
位置	(100, 100)

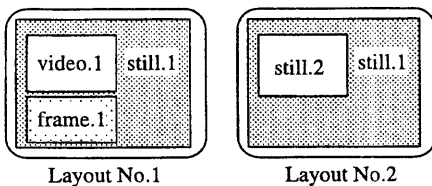


図 2: レイアウト情報の例

シーンの中にシナリオ含まれる場合、そのシナリオのシーンが入り子状に出力される。そこで、他のシーンを出力するためのフレームを設ける(図2の Layout No.1 の frame.1)。フレームは、映像素材を出力するためのウィンドウであり、チャンネル番号・型・サイズを持つ(下表)。

属性	値
チャンネル番号	1
型	frame: 640 × 480
位置	(100, 100)

3.2 シーン

シーンは、並列して再生されるカットの列と、それらの列がどのチャンネルで出力されるかを決定する情報とから成る。これを図3のように記す。

シーンは頭部(head)を本体(body)からなる。頭部の構成要素は次の通り；

条件式 条件式が満たされる場合にだけ、そのカット列が再生の対象になる(図中の cond)。

レイアウトタグ カット列をレイアウトするための情報を与える以下の属性からなる(図中の L-tag)。

チャンネル番号 そのシーンを再生する時に、レイアウト情報の中のどの要素と対応付けるかを指示するための識別子である。

型 カット列は全て同一種類の素材データからなるものとし、その属する型を記述する。画像ならば画面サイズ、音声ならばトラック数などを指定する。

演出度 そのカット列を出力することで、どれくらいそのシーンの与える情報が増えるかを記述する。(図中の v)。

本体の構成要素は、カット列(sequence)かまたはシナリオである。個々のカット列の要素は、全て等しい型を持ち、同じ位置にレイアウトされるものとする。シナリオの場合も、同一型のカット列を持つシーンからなるものとする。

本体の要素がシナリオである場合、そのシナリオは他のカット列と並列に再帰的に再生される。この場合も、そのシナリオ内で再生されるシーンは高々1つである。他のシナリオを含むシーンを複合シーン、そうでないものを基本シーンと呼ぶ。

4 レイアウト情報の選択

先に述べたように、シナリオの再生は再生環境の許す範囲でなるべく豊かな情報を提示すべきである。それには、環境に応じてシーンの演出度をもっとも高めるレイアウト情報を選択する必要がある。

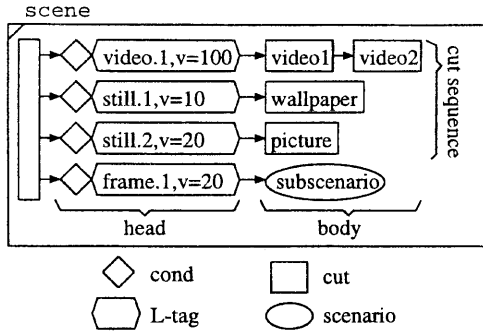


図 3: シーンの例

本節では、再生時のレイアウト選択を議論するために、シーン・レイアウト・環境などをモデル化した後、最適なレイアウトの選択について議論する。

4.1 レイアウト選択のモデル化

ここではシーン・レイアウト・環境をモデル化し、その上で、与えられた環境下での最適な再生を定義する。これらを、まず基本シーンについてモデル化し、複合シーンの場合に拡張する。

4.1.1 基本シーンの定式化

定義 1 (シーン) D : 素材データの種類。

O : サイズ。

$$T = \{(d, o)\}, d \in D, o \in O.$$

$$E = \{(t, n, v)\}, t \in T, n \in \mathbb{Z}, v \in \mathbb{R}, v > 0.$$

$$S_0 = \{(e_1, e_2, \dots, e_n)\}, e_i \in E (i = 1 \dots n).$$

素材データの種類 D はこのシーンが扱う素材の種類 (動画・静止画・音声・テキスト) を表す。要素 e は、全て同一の型 $t \in T$ をもつカットからなる。また、カット列の型 t は素材の種類 d とそのサイズ o によって表される。カット列 e には、それを出力するチャンネルの番号 n と演出度 v とが付与される。なお、 \mathbb{R}, \mathbb{Z} は実数、整数の集合を表す。

定義 2 (レイアウト) レイアウト情報 $l \in L_0$ はチャンネル c の集まりである。

P : 出力位置。

$$C = \{(n, t, p)\}, n \in \mathbb{Z}, t \in T, p \in P.$$

$$L_0 = \{(c_1, \dots, c_n)\}, c_i \in C (i = 1 \dots n).$$

チャンネル c は、チャンネル番号 n と素材の型 t の組であ

る。出力位置 p は、素材情報が出力される位置、映像ならば画面上の座標、音声ならば音声デバイスのチャンネルを表す。

定義 3 (環境) 環境 U は、素材種類 d とその再生に使用できる資源の量 h の組の集合である。

$$U = \{(d, h) \mid d \in D, h \in \mathbb{R}\}.$$

定義 4 (再生) シーン s をレイアウト l で再生すると、レイアウト l 中のチャンネル $c = (t_c, n_c, p_c)$ にシーンのカット列 $e = (t_e, n_e, v_e)$ を対応させることである。この時、両者のチャンネル番号・カット型は等しいものとする。この対応関係 $A_0(s, l)$ を次のように定式化する。

$$A_0(s, l) \equiv \{(e, c) \mid c \in l \wedge e \in s \wedge t_e = t_c \wedge n_e = n_c\}.$$

この時、シーンとしての演出度 $V(s, l)$ は、カット列 e の演出度 v_e の合計に等しいと定義する。

$$V(s, l) = \sum_{(e, c) \in A_0(s, l)} v_e.$$

また、この再生に要する資源 d の量 $F_d(s, l)$ は、それぞれのカット列の消費する資源 d の量の合計に等しい。

$$F_d(s, l) = \sum_{(e, c) \in A_0(s, l)} f_d(t_e).$$

ただし $f_d(t_e)$ は型 t_e の素材を再生するのに必要な資源 d の量とする。

定義 5 (再生可能性) ある環境 U の下でシーン s がレイアウト l で再生できる時、シーン s は環境 U でレイアウト l で再生可能であるという。再生可能性 $\rho(s, l, U)$ を次のように定義する。

$$\rho(s, l, U) \equiv \forall d \in D \exists (d, h) \in U [F_d(s, l) \leq h].$$

定義 6 (最適レイアウト) ある環境 U の下でシーン s の演出度 $V(s, l)$ を最大にするレイアウト l を最適レイアウト \bar{l} という。この時、次の式が成り立つ。

$$V(s, \bar{l}) = \max\{V(s, l) \mid \rho(s, l, U)\}.$$

以上で基本シーンとその再生が定式化された。

4.1.2 複合シーンの定式化

複合シーンは、シーンにシナリオを含む。従って、それを再生するためのレイアウト情報にはフレームが必要である。シナリオにおいては、一度に一つだけのシーンが再生されるので、再生時のある時点での複合シーンは結局、シーンを含むシーンとして定式化される。再生は、複数のシーンをそれぞれに適したレイアウトに対応付けることによって、定式化される。

定義 7 (複合シーン) 複合シーンの集合 S を定義する。

S_i : 第 i 階 ($i \geq 1$) の複合シーンの集合。

$S_i = \{(e_{i_1}, \dots, e_{i_j}, \dots, e_{i_n})\}, e_j \in E \cup S_k (k < i)$.

$S = \bigcup_{i \geq 0} S_i$.

定義 8 (複合レイアウト) 複合シーンに対する複合レイアウト L は、基本レイアウト L_0 にフレームの型 $w \in W$ を追加したものである。ただし $T \cap W = \emptyset$ とする。

$C = \{(n, t, p)\}, n \in Z, t \in T \cup W, p \in P$.

$L = \{(c_1, c_2, \dots, c_n)\}, c_i \in C (i = 1 \dots n)$.

チャンネル C はカットの列の型を持つか、シナリオを出力するためのフレーム型 w を持つ。

定義 9 (複合シーンの再生) まず、 s 中のカット列だけからなる部分シーン s' を定義する。

$s' = (e_{i_1}, \dots, e_{i_j}, \dots, e_{i_k}), e_{i_j} \in E$.

l の部分レイアウト l' は s 中の s' にあるカット列だけをレイアウトする。

$l' = (c_{i_1}, \dots, c_{i_j}, \dots, c_{i_k}), c_{i_j} = (n, t, p) \wedge t \in T$.

複合レイアウト l による複合シーン $s \in S_k$ の再生は、次のように定式化される。

$$A(s, l) \equiv A_0(s', l') \cup \bigcup_{s'' \in (s \cap S_{k-1})} A(s'', l'').$$

ただし、 s'' は s 中のシナリオが生成するシーンであり、 l'' はシーン s'' を再生する適切なレイアウトであるとする。上の式を再帰的に展開すると、複合シーンの再生は、次のように基本シーンの再生の集まりになる。

$$A(s, l) = A_0(s', l') \cup A_0(s'', l'') \cup \dots \cup A_0(s^{(n)}, l^{(n)}).$$

この複合シーン s のレイアウト l による再生の演出度 $V(s, l)$ は部分的な再生 $A_0(s^{(i)}, l^{(i)})$ ごとの演出度

$V(s^{(i)}, l^{(i)})$ の総和と定義する。

$$V(s, l) = \sum_{s^{(i)} \in s} V(s^{(i)}, l^{(i)})$$

また、この再生に要する資源 d の量 $F_d(s, l)$ は、その部分的な再生 $A_0(s, l)$ の消費する資源 d の量の合計に等しい。

$$F_d(s, l) = \sum_{s^{(i)} \in s} F_d(s^{(i)}, l^{(i)}).$$

定義 10 (複合シーンの再生可能性) ある環境 U の下でレイアウトの集合 $L = \{l', \dots, l^{(n)}\}$ を用いて複合シーン s が再生できる時、シーン s は環境 U で再生可能であるという。再生可能性 $\varrho(s, U, L)$ を次のように定義する。

$$\varrho(s, L, U) \equiv \forall d \in D \exists (d, h) \in U [F_d(s, L) \leq h].$$

定義 11 (最適レイアウト) ある環境 U の下で複合シーン s の演出度 $V(s, L)$ を最大にするレイアウトの集合 $L = \{l', \dots, l^{(n)}\}$ を最適レイアウト \bar{L} という。この時、次の式が成り立つ。

$$V(s, \bar{L}) = \max\{V(s, L) \mid \varrho(s, L, U)\}$$

4.2 最適なレイアウト選択

4.2.1 基本シーンのレイアウト選択

基本シーン s の最適レイアウトの選択方法は、全てのレイアウト候補 l に対して演出度 $V(s, l)$ を計算して最適なものを選択すればよい。

4.2.2 複合シーンのレイアウト選択

複合シーン s のレイアウト選択では、それを構成する部分シーンは、再生開始の時点で全て識別できるので、 s のレイアウト選択は、それぞれの部分シーン $s^{(i)}$ に対するレイアウト $l^{(i)}$ の組合せ L で表現される。この時、部分シーン間の資源の競合を調整するため、組合せ最適化の手法を適用して、演出度 $V(s, L)$ を最大にするレイアウトの組合せ L を選択する。

この問題を次のように、逐次決定過程として定式化する。まず、有限オートマトン $M = (Q, \Sigma, q_0, \lambda, Q_F)$ を次のように定義する。

$\Sigma = \{(s^{(i)}, l^{(i)}) \mid i = 1 \dots n\}$: レイアウト候補の集合。

$Q = 2^\Sigma$: レイアウト選択状態を表す集合。

$\lambda: Q \times \Sigma \rightarrow Q$, 選択状態間の遷移関数。

$Q_F \subset Q$: 最終状態の集合.

$\phi = q_0 \in Q$: 初期状態.

ただし, $\lambda(q, \sigma)$ は次の式を満たす.

$\forall q \in Q [\lambda(q, \epsilon) = q]$, ϵ は何もしない選択を表す.

$\forall q \in Q \forall \sigma \in \Sigma \forall \sigma' \in \Sigma [\lambda(q, x\sigma) = \lambda(\lambda(q, x), \sigma)]$.

最終状態 Q_F は次の式で与えられる.

$Q_F = \{q \in Q \mid \varrho(s, L, U) \wedge (s, l) \in q \wedge l \in L\}$.

ついで, 逐次決定過程 (M, h, ξ_0) を次のように定式化する.

M : 先に定義した有限オートマトン.

$h: R \times Q \times \Sigma \rightarrow R$, 演出度関数.

$\xi_0 = 0$, 初期状態 q_0 の演出度.

ここで h は次の式を満たす.

$h(\xi, q, \epsilon) = \xi_0 = 0$,

$h(\xi, q, x(s^{(i)}, l^{(i)})) = h(h(\xi, q, x), \lambda(q, x), (s^{(i)}, l^{(i)}))$
 $= h(\xi, q, x) + V(s^{(i)}, l^{(i)})$.

問題の性質より, $|\Sigma^*| < \infty$ 及び,

$\xi_1 < \xi_2 \Rightarrow h(\xi_1, q, x) < h(\xi_2, q, x)$

が成り立つ. すなわち, 逐次決定過程 (M, h, ξ_0) はループフリーかつ強単調であり, 例えば動的計画法を適用することで, この問題に対する最適解を求めることが可能である [3].

4.3 レイアウト選択の例

本節では, 複合シーンの例と, 基本シーンのレイアウトの選択の例とを示す.

図3に示したシーン s を次のように記述する. 素材データの型を v :video, s :still, t :text, f :frame のように略記する. s 中のカット列を $e_{\tau n}$ と略記する. τ は素材データの型, n はチャンネル番号である. s'' はこの部分にある図中の“subscenario”が生成するシーンを表す.

$s = \{(v, 1, 100), (s, 1, 10), (s, 2, 20), s''\}$
 $= \{e_{v1}, e_{s1}, e_{s2}, s''\}$.

s の部分シーン s' は $s' = \{e_{v1}, e_{s1}, e_{s2}\}$ と表される.

これを再生する二つの環境 U_1, U_2 を次のように定義する.

$U_1 = \{(v, 100), (s, 100), (t, 100)\}$.

$U_2 = \{(s, 100), (t, 100)\}$.

図2にある二つのレイアウトを l_1, l_2 と表す. チャンネルをそれぞれ, $c_{m\tau n}$ (m : レイアウト番号, τ : 素材データの型, n : 素材毎のチャンネル番号) と略記する.

$l_1 = \{(1, v, p_1), (1, s, p_2), (1, t, p_2), (1, f, p_2)\}$
 $= \{c_{1v1}, c_{1s1}, c_{1t1}, w_{1f1}\}$.

$l_2 = \{(1, s, p_2), (2, s, p_2), (1, t, p_2)\}$
 $= \{c_{2s1}, c_{2s2}, c_{2t1}\}$.

基本シーンの再生 先に定義した s の部分シーン s' は基本シーンと見なすことができる. 基本シーン s' の l_1, l_2 による再生は次の通り.

$A_0(s', l_1) = \{(e_{v1}, c_{1v1}), (e_{s1}, c_{1s1}), (e_{t1}, c_{1t1})\}$.

$A_0(s', l_2) = \{(e_{s1}, c_{2s1}), (e_{s2}, c_{2s2}), (e_{t1}, c_{2t1})\}$.

それぞれの演出度は $V(s', l_1) = 120$, $V(s', l_2) = 30$ となるので, 動画の資源が許す限り l_1 が選択され, 動画を再生できない場合には l_2 が使われ, video.1 に替えて still.2 が出力される.

5 まとめ

本稿では, 従来提案していたシナリオに, 並列再生を明示するシーンと, シーンを再生する際の配置情報であるレイアウトを追加した. これによって, シーンが再生される条件が固定でない時にも, シーンの見え方を指定することが出来るため,

- 被写体検索など, 起動されるごとに異なった再生条件を持ち得る場合でも, 一つのシーン記述で対応できる
- 同様の理由から, シーンの再利用性が高まる

などの利点がある. また, 複合シーンの場合のレイアウト選択は, 再生資源の競合が生じるが, 動的計画法を用いれば最適解が求まることを示した.

参考文献

- [1] Sakata, T., Kojima, A., and Satoh, T. : “Hyper Video: Video Based Hypermedia for Walk-Through Applications”, Proc. Multimedia Japan 96, pp. 148–154 (Mar. 1996).
- [2] 小島 明, 花籠 靖, 坂田 哲夫, 佐藤 哲司 : “実写ウォークスルー型電子図書館の構築”, 電子情報通信学会 技術研究報告 [オフィスシステム], Vol.95-50, pp.31–36 (Mar.14, 1996).
- [3] 茨木 俊秀 : “組合せ最適化の理論”, 電子情報通信学会, 第1, 5章, 1979.