マルチメディア通信と分散処理 80－5
グループウェア 21－5
（1997. 1. 30）

映像散策のためのビデオハイパーメディア
－ VHM 統合開発環境の全体構想 －

平野泰宏，坂田哲夫，星隆司，柴垣斉
ＮＴＴ情報通信研究所　データベース研究部

筆者らは，映像を中心としたマルチメディア情報をユーザがインタラクティブな操作で取り出せるビデオハイパーメディア（ＶＨＭ）技術を提案してきた．

従来のＶＨＭシステムはオーサリングツールである構造化支援ツールと実行時ライブラリであるインタラクティブプレーヤで構成されるが，VHM 統合開発環境では，構造化支援ツールに欠けていた画面レイアウトとイベント定義，作成されたシナリオを解釈してインタラクティブプレーヤを制御するシナリオインタプリタを追加し，GUI のみでマルチメディアアプリケーションを開発できるようにした．

Video Hyper-Media for Interactive Video Walk Through
－ A Conception of VHM Integrated Environment －

Yasuhiro Hirano, Tetsuo Sakata, Takashi Hoshi and Hitoshi Shibagaki
Database Systems Laboratory
NTT Information and Communication Systems Laboratories

We have designed and implemented Video Hyper-Media (VHM) which provides interactive multimedia presentation systems.

VHM system consists of two main components, authoring tool and media player libraries. VHM integrated environment provides display layout and event handling capabilities which are not supported by previous VHM authoring systems. New VHM system also has scenario executor which evaluates scenarios written by authoring tools and controls media players. These new features enable up to develop a multimedia presentation systems only by using GUI based tools.

1 はじめに

魅力的なマルチメディアプレゼンテーションを実現するには，単に決められたストーリーで再生するだけではなく，多様なメディアを組み合わせてインタラクティブなユーザーの操作によってストーリー展開させることが重要である．そのためには，ユーザの望む形態でメディア間の関連を記述する技術や，組み立てに沿ってメディア情報を提供する技術が重要である．筆者らは，映像を中心としたマルチメディアプレゼンテーションのためのビデオハイパーメディア (Video Hypermedia: VHM) システムを提案してきた [1] [2].

しかし，現状の VHM システムはレイアウトやイベントを扱う機能を持たないため，アプリケーション作成のためにはプログラムで記述せねばならない部分が多かった．本論文では，オーサリングから実行までの全工程をサポートする VHM 統合開発環境の構想について述べる．

VHM 統合開発環境では，オーサリングツールである構造化支援ツールに欠けていた画面レイアウトとイベント定義を追加し，作成されたシナリオを解釈してインタラクティブプレーヤを制御するシナリオインタプリタを追加する．これによって GUI のみでマルチメディアアプリケーションを開発できる．

また，シナリオの継承機構が追加し，シナリオの記述が容易になるのみならず，仕様等に変更があった際の変更箇所を局所化することができるようにする．

本論文では，2で VHM システムの概要を述べ，3で VHM システムの課題とそれを解決する VHM 統合開発環境について述べる．

2 ビデオハイパーメディア (VHM) システム

2.1 概要

複数のメディアを柔軟に組合せ，ユーザの操作によってストーリー展開を変化させられるアプリケーションを構築できるように，オーサリングから実際のプレゼンテーションまでを支援する環境が VHM システムである．

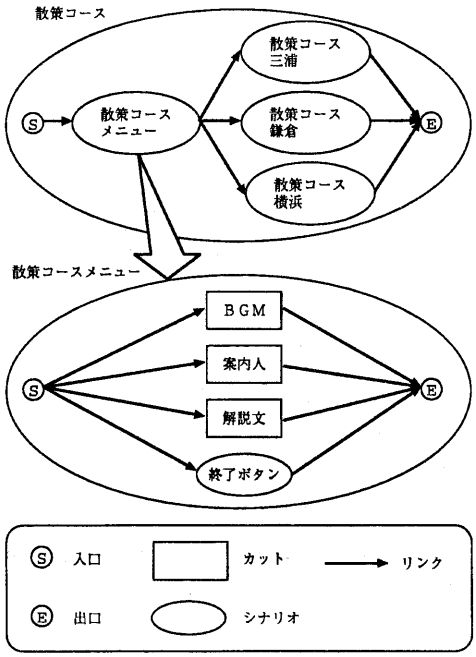

図 1: 構造化シナリオの例

VHM システムは，オーサリングツールである構造化支援ツール，メディア再生系であるインタラクティブプレーヤ，マルチメディアデータベースとで構成される．

2.2 構造化シナリオ

VHM システムでは，ストーリーの展開を有向グラフ構造を基本とした「シナリオ」で表現する．ノードにはプレゼンテーションの対象となる映像・音声・テキストなどのメディア (以後カットと呼ぶ) を配置する．カット間の関連である連続・同期・選択は，ノード間を結ぶリンクで表す．

複雑なストーリーや大きなシステムを 1 つのグラフ構造で記述すると，リンクが交差して視認性が悪く記述が難しい．また，シナリオの再利用やモジュール化が困難になる．そこで，シナリオは階層的な記述を可能とする．すなわち，ノードにはカットのみなら

ず，それらを組み合わせたシナリオを置くこともできる．

これによって，意味的にまとまりを持った単位で階層的にシナリオを記述することで，可読性，再利用性を向上し，複雑なストーリーを簡易に記述できる．構造化シナリオの例を図1に示す．

2.3 被写体定義

構造化シナリオでは，ノード間のリンクの部分で静的にストーリー展開の変化を記述する．一方，ユーザが興味を持った映像中の物体に触れることでその物体の情報を表示したりストーリーを変化させることができると，より動的な展開が可能になる．この機能を実現するのが被写体検索である．

VHM システムでは，映像中の被写体が存在する時刻・位置情報を登録しておき，実行時にユーザがポイントした時刻・位置情報を用いて登録しておいた被写体の情報を検索し，得られた情報によって次のストーリー展開を決定する機能を有する．

3 VHM 統合開発環境

3.1 VHM システムの課題

従来の VHM システムには，以下のような課題があった．

1. 画面レイアウトを設計する機能がなかった．
2. 1つのノードから複数のリンクが出ている場合にそれらが同時に実行されるか選択されるかといったリンクの意味が一意に定まらなかった．
3. VHM のデータモデルがイベントを含んでいないため，例えば，ボタンイベントによって分岐するようなアプリケーションの場合，ボタンを作成してボタンイベントによってリンクを選択するように，データモデル以外の部分で記述していた．

そのため，汎用的なシナリオ処理系を作成することができず，アプリケーション毎にシナリオ処理系を作成する必要があった．

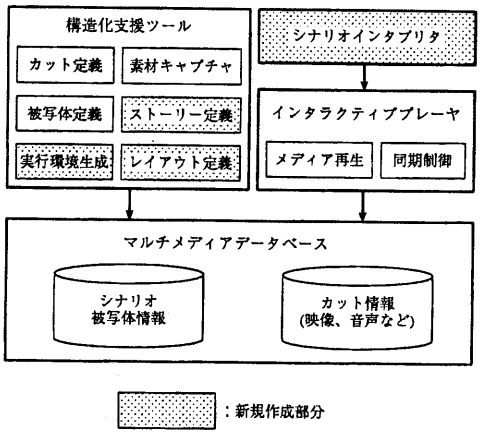

図 2: VHM 統合開発環境の構成

例えば，電子図書館 [2] では，リンクに付与されたラベル情報を用いて必要なボタンを作成するように記述していた．また，観光案内システム「TakeMe」[3] では指定型，推奨型，散策型の3つのシーンを解釈するプログラムを作成した．これらの例では，シナリオの記述でアプリケーションの変更ができるものの，アプリケーション毎にこのような処理を書く必要があることが，開発効率上のネックとなっていた．

汎用的なシナリオ処理系の研究のために試作したリゾルバ [4] では，シナリオの意味に制限を加え，リンクの選択条件の構文を限定し，選択条件が不定の場合にはユーザに問い合わせるようにすることで，無限に待ち合わせが起こったりすることのないシナリオ解釈を可能とした．しかし，画面レイアウトやイベントを扱う機能がなく，ユーザとのインタラクションが制限されているという問題点が残っていた．

そこで，VHM 統合開発環境では，これらの問題点を解決できるようにデータモデルを変更し，実行時環境を整備する．

3.2 VHM 統合開発環境の構成

VHM 統合開発環境の構成を図2に示す．

オーサリングツールである構造化支援ツールには，ストーリーの記述にイベント定義を追加し，画面レイアウトの設定と実行時環境の生成の機能を追加する．

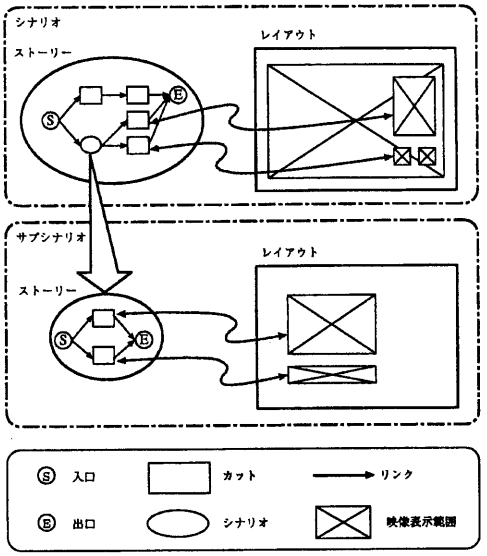

図 3: ストーリーとレイアウトの例

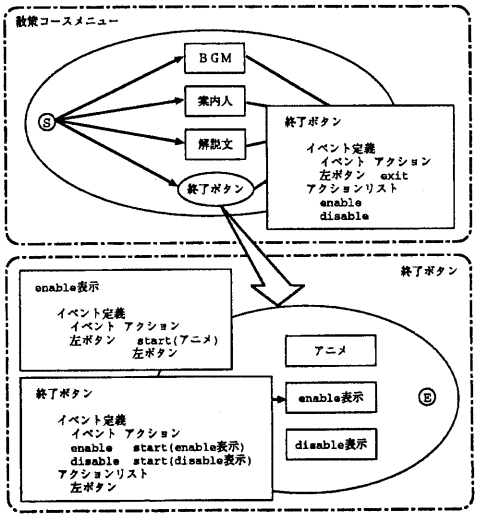

図 4: イベントの例

実行時環境生成機能によりシナリオインタプリタが生成されるため，ユーザーはアプリケーションプログラムを記述する必要がなくなる．

3.3 ストーリーとレイアウト

VHM 統合開発環境では，シナリオを，カットの再生順序やイベント発生時の振舞いを規定するストーリーと，カットや部品の配置情報を格納するレイアウトで構成する．ストーリーとレイアウトの例を図 3に示す．

ストーリーは 2.2で述べた構造化シナリオに対応し，構造化シナリオにイベントの記述等を追加したものである．また，ノードとして映像・音声などのカットとサブシナリオに加えて，Tcl/Tk のスクリプトを追加する．ファイル入出力や他のプログラムとの通信など，VHM でサポートしていない機能をスクリプトで組み込める．

レイアウトはストーリーと 1 対 1 に対応し，ストーリーに現れるカットなどのノードの画面上での位置や大きさ，ノード間の上下関係を定義する．

3.4 イベントとアクション

ユーザからの操作や，タイマー，映像や音声の特定の位置を再生したときなどに何らかの動作を起こすために，イベントとそれに対応するアクションを設定できるようにする．

イベントには，マウスやキーボードによる操作，タイマー，映像や音声の特定の位置に設定したブレークポインタなどがあり，その結果として起こるアクションには分岐の設定，ノードの開始および終了，カットの再生オプション (再生方向，速度，表示ページ等) の変更などがある．

イベントとアクションは，サブシナリオやスクリプトノードと上位のシナリオの間のやりとりにも用いられる．すなわち，サブシナリオを外部から操作できるように，サブシナリオに対するアクションを定義できる．サブシナリオからは，このアクションはイベントのように見える．また，逆にサブシナリオやスクリプトを参照する上位のシナリオで何らかのアクションを起こさせるアクションを定義できる．上位のシナリオからは，シナリオまたはスクリプトノードで起こるイベントのように見える．

サブシナリオと上位のシナリオ間のイベントと

アクションのやりとりの例を図4に示す．この例では，「終了ボタン」サブシナリオは，マウスクリックを受け付ける「enable」と受け付けない「disable」の2つの状態をとり，それぞれの状態で「enable表示」カットまたは「disable表示」カットを表示している．「enable」状態でマウスクリックされると，「アニメ」カットを再生した後に，シナリオを終了させる．

図4では，「終了ボタン」サブシナリオに対してアクション「enable」や「disable」を実行することで，「enable表示」カットや「disable表示」カットに対して「start」のアクションが送られ，「終了ボタン」シナリオの状態を変更できる．また，「enable表示」カットで「左ボタン」イベントが起こると，「アニメ」カットをスタートさせてから，「終了ボタン」シナリオの「左ボタン」アクションを実行する．これによって，上位の「散策コースメニュー」シナリオの「終了ボタン」サブシナリオノードで「左ボタン」イベントが起こり，「exit」アクションが実行される．

3.5 分岐の記述

1つのノードから出る複数のリンクから，選択されたものだけが起動する場合の記述を考える．

リンクを選択するためのイベントがリンクと同じシナリオの中で起こる場合と，サブシナリオやスクリプトで起こる場合がある．そこで，図5の記述例のように2種類の記述ができるようにする．

リンクと同じシナリオの中でイベントが起こる場合，そのイベントに対するアクションとして分岐先を設定するアクションを発行するようにする．図5の終了ボタンがこの形式の例である．

サブシナリオやスクリプトで起こる場合，サブシナリオやスクリプトの戻り値を設定するアクションを発行し，戻り値と同じ値を持つリンクが起動するようにする．図5の右折・左折ボタンがこの形式の例である．

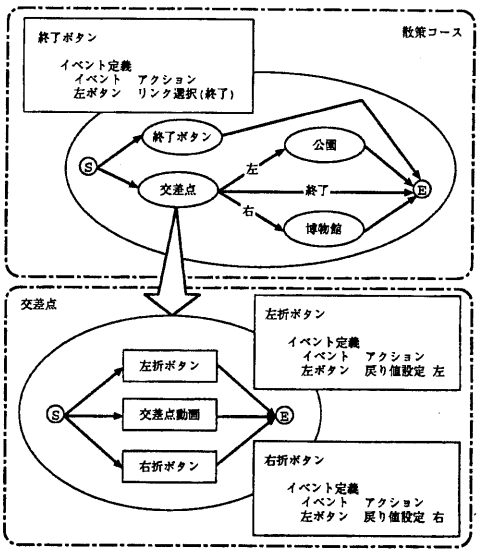

図 5: 分岐の例

3.6 シナリオの継承

情報提供システムのようなアプリケーションの場合，同じフォーマットの多数のシーンから，ユーザーの望みのシーンが選択されるような構造になることが予想される．例えば，観光案内システム「TakeMe」[3]の場合，紹介映像，地図，案内文等の配置は，全ての観光地で同じであった．

ここで，元になるシナリオをコピーして変更していく方法を用いると，画面デザイン等の軽微な変更ですら，多くの(「TakeMe」の場合，シーン数は約300であった)シーンの変更を引き起こすおそれがある．

そこで，シナリオの記述量を削減し，仕様が変更された際などの変更範囲を局所化するために，シナリオの継承を導入する．

すなわち，テンプレートになるシナリオを作成し，そのシナリオとの差分を記述していくことによって，多数のシーンを効率的に作成できる．また，仕様の変更があった場合でも，テンプレートの変更のみで済む場合がある．

3.7 リンクの解釈

VHM 統合開発環境のストーリーでは，リンクの意味を一意に解釈できるように，リンクの種類を定義する．

1つのノードに入るリンクが複数ある場合に1つのリンクの起動でそのノードが起動されるか全てのリンクが起動するまで待つかを，リンクが入るノードのプロパティとして指定する．また，1つのノードから複数のリンクが出る場合に，全てのリンクが起動するか，条件を満たすものが選択的に起動されるかを，リンクが出るノードのプロパティとして指定する．

3.8 ノードの有効期間

シナリオの解釈をシステムが自動的に行なうためには，各ノードが存在する期間を決定できなければならない．VHM 統合開発環境では，ノードの有効期間を以下のように定義する．

ノードは，原則として，入口のリンクまたは後述のアクションによって起動され，終了アクションを受けとるか，シナリオが終了するまで存在する．

ただし，あるシナリオから別のシナリオに遷移する際に，両者が同じシナリオから継承されている場合，親シナリオで定義されたノードは削除されずに継続して存在する．例えば，多くのシーンで共通のBGMや壁紙などは親シナリオで定義しておけばよい．

3.9 シナリオの実行

VHM 統合開発環境では，オーサリングツールで作成したシナリオをもとに，実行時プログラムであるシナリオインタプリタを生成する．シナリオインタプリタは，データベースアクセスやインタラクティブプレーヤー制御を拡張した Tcl/Tk のスクリプトである．

3.10 シナリオの正当性チェック

リンクの張り忘れなどによってシナリオが思わぬ動作をすることを避けるために，ノードに出入りするリンクを以下のようにチェックする．スタート以外のノードは，入ってくるリンクまたは，起動アクションによって起動されるようになっていなければならない．また，出口以外のノードからは必ずリンクが出ていなければならない．

ここで，例えばボタンのシナリオのように，上位のシナリオが終了するまでは存在し続けなければならないため，ボタンの絵を描くノードから出口ノードにリンクを張れない場合がある．そこで，リンクの張り忘れではないことを示すために，終端ノードを設定できるようにする．

4 終りに

VHM システムに欠けていた画面レイアウト・イベントの定義機能およびシナリオインタプリタを追加した VHM 統合開発環境について述べた．VHM 統合開発環境では，GUI のみでマルチメディアアプリケーションを開発できるようになる．

今後は，VHM 統合開発環境を実装し，実際にアプリケーションを構築してその有効性を確認する．

参考文献

[1] 坂田哲夫, 木原民雄, 小島明, 佐藤哲司, “映像散策のためのビデオハイパーモデルの提案”, 信学技報 (データ工学), DE95-35, pp. 65-72, July 1995.

[2] 花篭靖, 小島明, 佐藤哲司, “ビデオハイパーメディアによるビジュアル電子図書館の構築”, 情処研報 (データベースシステム), 96-DBS-107, pp. 17-24, Mar 1996.

[3] 岸田義勝, 木原民雄, 平野泰宏, 岩渕明, 寺中勝美, “ビデオハイパーメディアの観光案内システム「TakeMe」への応用”, 情処研報 (マルチメディア通信と分散処理), 96-DPS-76, pp. 55-60, May 1996.

[4] 坂田哲夫, 佐藤哲司, 柴垣斉, “映像散策のためのビデオハイパーメディア-クライアントサーバ型 VHM のアーキテクチャ”, 情処研報 (マルチメディア通信と分散処理, グループウェア合同研究会) 掲載予定, Jan 1997.