

VHMにおけるシナリオ柔構造化実現方式の検討

湯口 徹¹ 平野 泰宏¹ 池田 哲夫¹ 星 隆司¹

¹NTT情報通信研究所

柔構造化シナリオによるマルチメディアアプリケーションのオーサリング環境構築の実現方式について報告する。

ノードリンク型シナリオ構造の柔構造化と、その応用例として、アプリケーション開発における素材の再利用化による開発およびメンテナンスコストの低減や、より柔軟性の高いアプリケーション開発への応用が可能である点についてはすでに報告した。

本論文では、柔構造化シナリオを利用した統合的なオーサリング環境の構築にあたり、オーサリングツール上での柔構造化シナリオの実現方式およびルールを利用したシナリオ制御方式の検討について述べる。

Implementation of Flexible Scenario on Video Hyper Media

Toru YUGUCHI¹, Yasuhiro HIRANO¹, Tetsuo IKEDA¹, Takashi HOSHI¹

¹NTT Information and Communication Systems Laboratories

We design and implement Video Hyper Media(VHM) Authoring environment which using flexible scenario structure.

We have proposed flexible scenario in VHM and its effectiveness. In this model, we can reduce a cost of its development and maintenance for re-use multimedia contents. And we can design more interactive applications.

In this paper, we describe a way of implement of flexible scenario in VHM authoring tool and control of scenario for rules

1 はじめに

筆者らは、ビデオハイパーメディア（以下VHM）システムに関する研究開発の中で、マルチメディアアプリケーション開発のための、メディア間の関連定義モデルについての提案[1]や、効率的な開発に向けたGUIによる統合的な開発環境の構築と検証[2][3]を行ってきた。

さらに、コンテンツの作成やメンテナンスにかかる稼働やコストを低減するためのシナリオ再利用を容易にするシナリオ柔構造化方式について提案した[4]。

本稿では、柔構造化シナリオを利用した統合的なマルチメディアアプリケーションのオーサリング環境を構築するにあたり、オーサリングツール上での柔構造化シナリオの実現方式およびシナリオ制御方式に関する検討について報告する。

以下、2章で柔構造化シナリオの概要と具体例、3章で柔構造化シナリオを利用したシナリオ動的制御のための検討、4章でオーサリングツール上への実装方針について述べ、5章でまとめとして今後の展開について述べる。

2 柔構造シナリオ

2.1 概要

VHMシステムにおけるシナリオや、その構成要素の再利用促進のため、従来のVHMシステムにおけるシナリオのデータモデルを拡張して、これまではオーサリング時に1対1の関係として固定的に定義していたオブジェクト間の相互関係や実体である素材データとの関連づけについて、論理的に複数のデータを管理できるようなモデルとした。

これによって、従来までは固定的な定義であったオブジェクト間の関係を柔軟に表現することが可能となり、ストーリーやレイアウト単位での再利用によってアプリケーション作成コストの低減とともに視認性の高いアプリケーションの作成ができる。以下に、シナリオ柔構造化の例を示す。

2.2 ノードへの複数カット定義

従来のシナリオでは、ファイル名やサブシナリオ名などの素材を示す情報は、ストーリー上の1つのノード

に対して一意に定義されており、一つのカットノードは一つの素材データに対応していた。

このノードのモデルを拡張して、複数の素材データを管理できるようにした。このことによって、形式の違う全く別の素材ファイルであっても、意味的に同じことを表現しているようなものをノードオブジェクトの中にバッキングすることが可能となり、素材の多様性を隠蔽したアプリケーションの意味的な構造レベルでのオーサリングが可能となり、アプリケーションの視認性が向上する。

図1は1ノードへの複数カット定義の例である。商品紹介のようなアプリケーションにおいて、解説のための音声日本語もしくは英語を使用して行う際に、従来ストーリー中にそれぞれ素材データとリンクさせた日本語解説と英語解説のノードを作成する必要があったが、一つのノードに対して複数素材の定義を可能とすることによって、ストーリー編集はアプリケーションの構造のみを意識すればよく、実際の素材との関連はストーリーの編集とは切り離して行うことができる。

このようにプレゼンテーションの構成と実際の素材情報とを分離したオーサリングによって、素材レベルでの追加・変更・削除などを、シナリオ構造の変更を伴うことなく行うことが可能となり、アプリケーションのメンテナンス性の向上となる。

2.3 レイアウトの変更

これまでのVHMでは、シナリオを一つずつのストーリーとレイアウトの組として定義しており、同じ場面でレイアウトを変更して表示するようなケースでは、全ての表示パターンをそれぞれ別々のシナリオとして定義する必要があった。

この点については、これまでの統合的なオーサリング環境構築の中でも階層化したシナリオの継承機能によって、テンプレートシナリオに対して差分を記述することで効率的なオーサリングの実現を図ってきたが、ストーリーやレイアウトレベルでの再利用への拡張が不十分であった。

そこで、ストーリーとレイアウトの関係についても多対多関係の表現を可能にするモデルへの拡張を行うことによって、ストーリーやレイアウト単位でのオブジェクトの再利用を促進することができ、アプリケーション

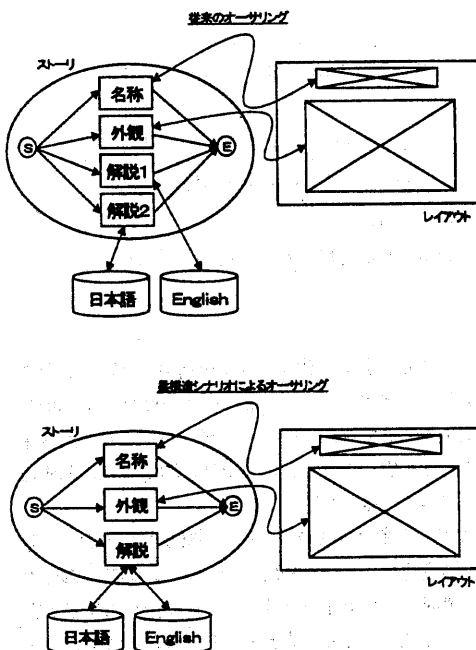


図1 ノードへの複数カット定義

ンの新規作成や変更などのメンテナンスのコストを削減することが可能となる。

図2に複数レイアウト定義の例を示す。観光地の施設案内などにおいて、ユーザの趣向や動作環境などに応じて画面上のオブジェクトの配置を変更して表示させるような場合には、これまでのシナリオではそれぞれの表示パターンごとに同一のストーリーでレイアウトのみが異なるシナリオを作成して、使用するシナリオを変更することによって実現しなければならなかった。

柔軟なシナリオによって、一つのストーリーに対して複数のレイアウト定義が可能となり、ストーリーやレイアウト単位での再利用により、アプリケーションのメンテナンスにかかるコストを削減することができる。

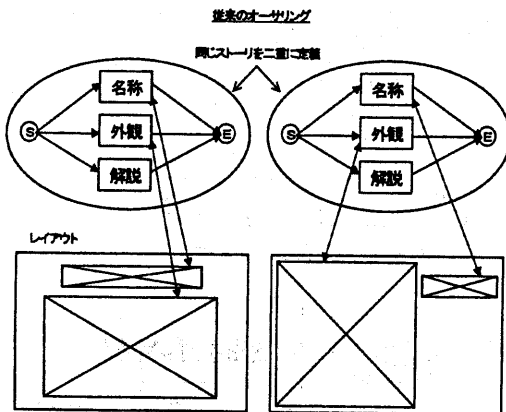


図2 ストーリーとレイアウトの関係

2. 4 シーンの導入

観光地案内やカタログによる商品の紹介など、マルチメディアアプリケーションにおいては、ストーリーやレイアウトが全く同じで、表示や再生する素材データのみが異なるようなケースが多く見られる。

そのような場合にも、これまでのオーサリングでは、それぞれの素材データごとにノードに対して関連定義し、同一のシナリオを表示データ数分作成しなければならなかった。

図3に示すように、例えば動画とテキストそして説明用の音声など複数の素材データをまとめて一つのシーンとして定義し、これを利用してアプリケーションを構築することが効率的であると考えられる。これによって、特にデータベースからの検索結果を表示するようなアプリケーションにおいて、データのメンテナンスが容易になる。

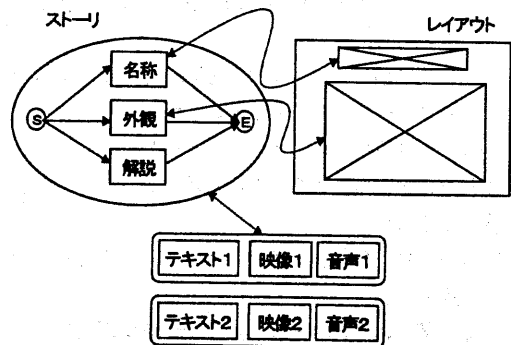


図3 シーンの定義

3 シナリオの動的制御

3. 1 ルールの導入

前章で述べたシナリオの柔軟な構築は、再利用性の向上というオーサリングへのインパクトのみでなく、動的なシナリオ制御というサービスのインタラクティブ性の向上に対してもインパクトを持つ。

柔軟な構築したシナリオにおいて、オーサリング時には不確定な形で定義した要素を実行時に動的に変更・決定させることによって、より柔軟なサービスの実現

が可能となる。

例えば、前章で示したような使用する素材や画面上の表示位置の変更などを実行時の環境や履歴情報などによってアプリケーション実行時にダイナミックな変更を可能とすることで、適応型サービスのようなサービスの実現も可能となる。そのためには、シナリオの柔構造化において、シナリオを動的に制御するための情報とその機構が必要である。

今回、柔構造化シナリオの1要素として、このシナリオ動的制御のための情報を「ルール」と定義した。ルールでは主にシナリオ実行時に外部要因や履歴情報などの各種情報を利用して、柔構造化したシナリオの不確定な要素を確定する。

ルールの検討にあたり、その要求条件を整理すると以下ようになる。

- ・ルールではカットやシーン、サブシナリオやレイアウトの選択を可能とする。
- ・ルールでは、そのパラメータとして環境変数のほか、システムの動作履歴情報などが利用できることとする。
- ・ルールは他のオブジェクトと独立し、ルールの変更がストーリー等へ与える影響を最小限にする。
- ・ルールは再利用ができることとする。

これらの要求条件から、今回実現するルールの仕様を次のようにした。

- ・ルールは、条件部とアクション部の組み合わせで定義する。
- ・条件部にはそのルールの適用条件を論理式により記述し、その条件式が満たされた場合に実行されるアクションの指定を可能とする。
- ・アクション部では、アクション文を一つ以上定義できる。アクション文では、ノードなどの適用対象要素に対して、選択対象からの結果を割り当てる選択文や、スクリプト実行文、他のルールを起動するルール起動文などの記述を可能とする。
- ・ルールへの入出力として、システム時刻、動作環境、乱数のほか、外部データベースなどとのインタフェースも可能とする。
- ・ルールの起動については複数の箇所を設定可能とし、ノードやカットの属性として定義できるほか、イベントに対するアクションとしても定義可能とする。

3. 2 シーンを選択

ルール定義において、アクション部を適切に記述することによって、シーンの選択が実現できる。以下に例を示す。

個々のシーン（カット、サブシナリオ等の組）が格納された表（シーン表）をあらかじめ定義しておく。

ルールのアクション部では、上記のシーン表を検索して条件に当てはまるシーン（シーン表の組）を一つ取り出す。その上で、選択文を使用して適用対象となっているノード等に対して各組の要素を割り当てる。

シーン表をメンテナンスすることによって、関連する一群の要素を単位としたオーサリングが可能となる。

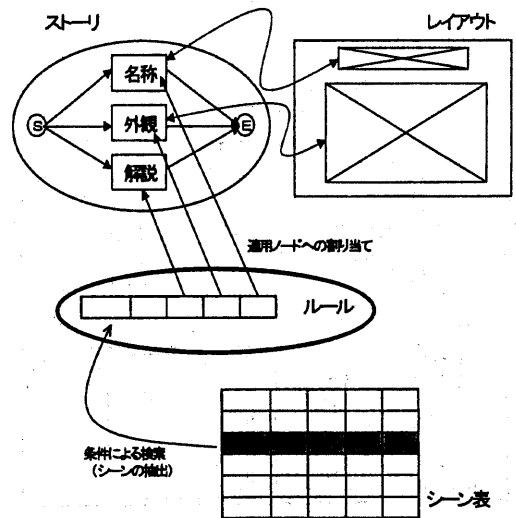


図4 シーンを選択

4 オーサリング環境への組み込み

4. 1 要求条件と方針

前章までに示したシナリオの柔構造化による、シナリオ構成要素間の関連を図5に示す。

柔構造化シナリオでは、その要素としてルールとシーンを新たに追加した。そして、これらの関係を多対多に定義することを可能とするために、シナリオのなかにバインドと呼ぶ要素を設け、このバインドが要素間の関連を管理することとした。

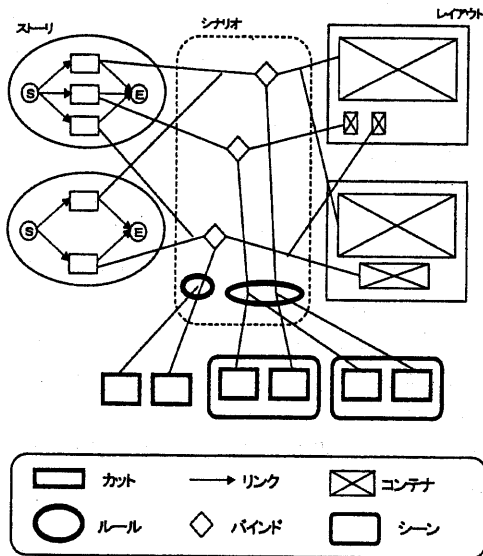


図5 柔軟化シナリオの例

オーサリング環境の構築については、これまでの統合開発環境をベースとして機能追加する方針とした。シナリオの柔軟化にもなったオーサリング機能に対する要求条件を以下に示す。

- ・ノード、コンテナ、カット（またはシーン）の関連付けによりシナリオを定義する。
 - ・ストーリー、レイアウト、ルールなどの要素をシナリオ間で共有可能とする。
 - ・実行時にはそれぞれ一つのストーリーやレイアウトなどの要素をルールによって決定可能とする。
 - ・複数のカットからなるシーンが定義可能で、実行時にルールによって一つのシーンの選択ができる。
- これらの要求を従来のVHM統合開発環境上に拡張し、柔軟化シナリオを利用した統合的なオーサリング環境の構築について検討した。

実装にあたり、その利用シーンの想定から必要かつ利用価値の高いと思われる機能を実現するために、次に示すようなシナリオの自由度にある程度の制限を設けて実装することとした。

- ・ストーリーが決まると、ノード対バインドが1対1に定まる。
- ・レイアウトが決まると、コンテナ対バインドが1対多に定まる。
- ・バインドはカットに対して少なくとも一つのカッ

トと直接結びつくか、少なくとも一つのカット選択ルールを介してカットと結びつく。

上記の制限に違反するシナリオの例を図6に示す。

- 複数のバインドが一つのノードと関連することはない。（ただし、コンテナに対しては許容する）
- バインドとノードやコンテナとの関連をルールで選択することは不可とする。（図中の b1, b2 など）

シナリオの自由度（バインドの定義）に対して上記の制限を設けることによって、ノードやコンテナとバインドの関係が緊密になる。このことは、次項で述べるバインドの編集において、直接バインドを意識させないオーサリングを可能とする。

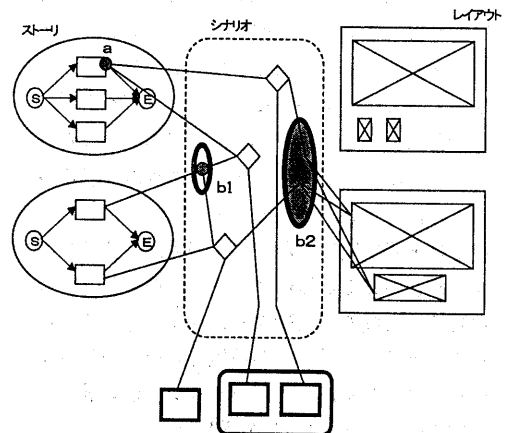


図6 制限違反のシナリオ

4.2 オーサリングイメージ

上記のような要求条件を満たすため、従来のオーサリング環境への機能追加や変更を行う。主な変更点を以下に列挙する。

4.2.1 シーン・ルールの編集

シーンやルールについては、他のオブジェクトと同様、部品化して再利用が可能なこととする。したがって、これらを個別に編集するエディタが必要となる。

これら新規エディタは従来のエディタ類と同様、エディタ間のシームレスな連携を可能とする。

4.2.2 ストーリー・レイアウトエディタ

(1) 単独編集

従来のオーサリングにおいては、シナリオを定義する際にストーリーとレイアウトの編集が必須であったため、それぞれ個別の編集作業は不可能であった。

今回の柔構造化により、ストーリー、レイアウトはそれぞれオブジェクトとして独立することとなるため、個別に編集することを可能とする。

(2) バインドの編集

今回の柔構造化によって、シナリオの要素として新たにバインドが追加された。しかし、バインドを実体として意識したオーサリングはユーザにとって複雑であると考えられ、バインドを意識しないオーサリングが必要である。この点については、バインドの生成や消滅の契機について、他のオブジェクト（ノードやコンテナ）などの編集契機にあわせて、内部構造的にバインドが必要となる契機で生成し、不必要となった時点で自動的に削除するようにした。このことによって、バインドの編集をユーザからは隠蔽し、ユーザに対してオーサリングが必要以上に複雑にならないように配慮した。

4.2.3 コピー機能

従来のコピー機能は、シナリオごと一括コピーのみの機能であり、部分的なコピーを行うことはできなかった。

既存のシナリオの一部分を変更して使用する際に、シナリオの一部分を指定してコピーすることを可能にすることによって、作業を効率化することができる。今回のオーサリング環境では、このコピー機能について、シナリオの部分的なコピーができるようにする。

5 まとめ

VHMにおけるシナリオ柔構造化の実現方式について検討した。

柔構造化したシナリオを利用したオーサリング環境の使用によって、オブジェクトの再利用による開発コストの低減が図られるとともに、これまでは記述が難しかった柔軟でより高度なインタラクティブ性をもったマルチメディアアプリケーションを比較的容易に記述することが可能になると考える。

今後は、個人適応型サービスなどの複雑なアプリケーション構築を通して、柔構造シナリオの有効性について評価するとともに、さらに多様化するサービスに向けたオーサリング環境の構築について検討する予定である。

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] 平野泰宏、坂田哲夫、星隆司、柴垣斉：“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”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）97-DPS-80-5,Jan 1997
- [3] 平野泰宏、尾美誠司、山口直人、星隆司：“映像散策のためのビデオハイパーメディア-VHM 統合開発環境の評価”，マルチメディア，分散，協調とモバイル（DiCoMo）ワークショップ,pp.353-358,July 1997
- [4] 湯口徹、坂田哲夫、平野泰宏、池田哲夫、星隆司：“ビデオハイパーメディアにおけるシナリオ柔構造化”，マルチメディア，分散，協調とモバイル（DiCoMo）ワークショップ,pp.281-286,July 1997