

大規模マルチメディアオーサリングのためのシナリオ記述体系

田中 栄市郎 Rodrigo BOTAFOGO 原田 浩明 田口 大悟
NEC 情報メディア研究所

本稿では、大規模マルチメディアアプリケーション開発におけるシナリオテンプレート記述モデルについて述べる。大規模なマルチメディアアプリケーション(以下MMAP)の作成において、工数削減を図るには、テンプレートによるアプリケーションの再利用が有効である。我々は、シナリオ記述を1) ペトリネットをベースとし、状態遷移を論理的に記述する論理構造記述、2) メディアのレイアウト情報を記述するスタイル記述、3) 提示における状態遷移のパラメータなどを記述するメソッド記述、に分割した。これらを組み合わせる事でMMAPのテンプレートを構築することができる。以上のように記述を分割することで、用途に合わせたシナリオのテンプレートが実現できる。

A Scenario Description Architecture for Large Scale Multimedia Applications

Eiichiro TANAKA Rodrigo BOTAFOGO Komei HARADA
Daigo TAGUCHI
Information Technology Research Laboratories, **NEC** Corporation
1 - 1 Miyazaki 4-Chome, Miyamae-Ku, Kawasaki, Kanagawa 216 Japan
E-mail img@joke.cl.nec.co.jp

This paper describes a model for the multimedia scenario description of large scale multimedia applications. In order to cut down the authoring cost of large-scale multimedia applications, it is desirable to use templates which describe the application structure. In this paper, we propose a scenario template description architecture which consists of three specification levels: logical, style, and method. The logical level specification uses a Petri net based model to describes the multimedia application structure. The style level specification defines the layout position of medias. Finally, the method level specification describes parameters of media presentation such as visual effects. We can create large multimedia applications using these scenario template descriptions.

1 まえがき

近年、PCのハードウェア、OSの進歩によりマルチメディアを利用する環境が整いつつあり、マルチメディアを利用したCD-ROMタイトルもこの数年でかなりの量が発売されている。今後、さらに環境整備が進むにつれ、マルチメディア市場は急速に拡大すると考えられる。一方、CD-ROMで提供されるような実用規模のMMAP開発には膨大な時間と労力が必要で、ハードウェアの進歩にソフト開発が追いつけず、ソフト不足が発生するという状況が予想される。一般に実用規模のマルチメディアアプリケーション(以下MMAP)は、データ量や状態遷移数が多く大規模である。しかし、大規模MMAPの開発の工数削減方法は現在模索段階にあり、十分な議論がなされていない[小川92]。その本質的な課題として、以下の2点があげられる。

- (1) ストーリー作成からMMAPの完成までのオーサリング工程を通じて、工数削減を実現する開発手法、品質管理手法がまだ模索段階である。これは、効率的なMMAP開発手法の確立、およびマルチメディア版CASEツールの開発などに通じるが、この議論の詳細は別稿にゆずる。
- (2) 大規模MMAP開発をターゲットとしたオーサリングツールにおけるデータ管理、データ設計手法が未確立である。プログラム開発と比較した場合、開発を容易にするライブラリが存在しないため、ライブラリ利用による開発工数の削減ができないという問題がある。

本稿では、主に(2)の問題について検討する。

2 MMAP 再利用のための要件

2.1 従来の開発方法の課題

従来のMMAP開発は、おおまかに図1に示すような、ストーリー作成、シナリオ化、メディア作成、メディア統合/編集、ディスク作成までの工程からなる。ただし、MMAPでは、仕様と実際に出来上がる物との差異が生じやすく、メディアの品質を向上させたり、途中でシナリオやストーリーに変更を加えるといった作業が発生するため、実際にはもっと複雑になる。

ここで、シナリオ化とは、ストーリーをコンピュータの記述に変換するための工程で、映画でいう脚本(シナリオ)とほぼ同じ意味を持つ。メディア統合/編集は、オーサリングツールなどを用いてメディアのレイアウト、タイミング、ハイパーリンク等を編集する工程である。

従来のMMAP開発では、メディア作成の後、そのメディアをオーサリングツールで編集するというオーサリング方法が一般的である。しかし、以下のような問題がある。

従来の方法では、メディア統合/編集工程でメディアが必要になるため、メディアが完成しないと直接メディア統合/編集工程に取りかかれない。

上述したように、MMAPのオーサリングでは、途中で仕

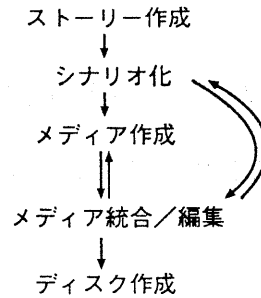


図1: オーサリング工程

様の変更が加えられることが多く、画質向上等に伴うメディアの差し替え作業が発生しやすい。この場合、メディア提示時間の変化やメディアサイズの変更などによる細かい修正が生じ、結局細かい編集をやり直す必要がある。

作者、アーティスト、編集者などのイメージを確認し合うため、MMAP開発初期ではプロトタイプを作成するケースが多いが、プロトタイプで作成したデータはMMAPに再利用されることは少ない。

2.2 再利用を実現するための開発工程

我々は上記の問題に対し、ダミーデータを用いて、メディアの空間的な位置や提示時刻といったメディアの提示情報の枠組を作成し、ダミーデータの代わりにメディアを後から割り当てるといった編集方式[原田91]や、画面の枠組によるレイアウト情報の再利用方式[田口93]などを提案している。以上のような枠組を総じてテンプレートと呼ぶことにする。

これらテンプレートを用いた編集方法により、1)プロトタイプで作成したデータの利用やメディアの入れ換えによる編集工数の削減が図れる、2)MMAPで使用するメディア作成とメディア統合/編集の工程が並行して行えるためオーサリング工程が大幅に短縮できる。3)共通の状態遷移や画面レイアウト構造が多いCAI教材では、設計時に共通の状態遷移パターンや画面パターンを作成することで、MMAPの各状態遷移や画面を共有でき、開発工数を大幅に減らすことができる。

このようにテンプレートをあらかじめ作成しておき、後からメディアをテンプレートに割り当てるといった編集方法は、大規模MMAPの工数削減には有効であると考えられる。

そこで、状態遷移、画面テンプレートの提供だけでなく、それらを組み合わせMMAP全体をテンプレート化できるようにシナリオ記述が必要となる。

2.3 シナリオのテンプレート記述の要件

MMAPのテンプレートを実現するシナリオ記述の要件を以下に列挙する。

- (1) レイアウト、状態遷移を分離した形でテンプレートが提供でき、かつこれらを有機的に組み合わせさせたMMAPのテンプレートも提供できる必要がある。
- (2) 異なるサイズのメディアや異なる提示時間のメディアをテンプレートに組み込んだ時でも、画面上的位置や同期タイミングを最適な場所や時刻に自動的に配置するように、位置の制約情報や、意味的關係を記述できる。
- (3) 開発環境や実行環境が異なってもシナリオテンプレートの利用が可能になるように、プラットフォームに依存しないで記述できる。

(1)を実現するため、シナリオ記述をメディアの提示などの状態遷移を論理的に記述する論理構造記述、メディアのレイアウト情報を記述するスタイル記述、提示における状態遷移のパラメータなどを記述するメソッド記述に分割し、これらを組み合わせる事で、MMAPのテンプレートを構築する。

以上のように記述を分割することで、テンプレートの用途が広がり、用途に合わせたテンプレートライブラリも実現できる。

これら各3つの記述に対して(2)、(3)の要件を満たすように記述モデルを提案する。以下、各記述について説明する。

3 論理構造記述

3.1 従来例

論理構造記述では、メディアの提示順序や提示の状態遷移をメディア間の時間的な関係を用いて記述する方法。時間的な提示関係を記述する方法には、現在、以下のようなものがある。

- (1) メディアの提示を順提示(sequential)、並列提示(parallel)といったメディアの提示関係を用いて記述する方法。この方法はHardmanらのCMIF[Hardman93]や、HyperODA[藤村93]で採用されている。
- (2) メディア間の提示の開始/終了などのイベントにおいて、イベント間の制約をもうけることで提示関係を記述する方法。この方法は、MHEG[亀山93]などで採用されている。
- (3) プログラムのように提示開始/終了を表すオブジェクトをフローチャート上に配置する方法。この方法は、マクロメディア社のAuthorWare[AuthorWare]で採用されている。

(1)、(2)の方法では、ユーザのインタラクションによってメディアの提示状態を変化させる場合、別のシナリオに状態遷移するしかなく、例えば、提示中のメディアを

同期を取りながら消去したりするといった、現在、提示中のシナリオを修正して提示することができない。

(3)の方法では、シーケンシャルなフローチャート上にメディアの提示、終了を配置しているため、例えば、図2のように音声Aの終了と同時に音声Bを提示し、また、動画Cの終了と同時に動画Dを提示し、音声Aと動画Cを同時に提示開始する場合、メディアの提示時間が $A > C$ の時と $A < C$ の時では、図のように記述の順序が異なってしまうため、テンプレートとして使用できない。

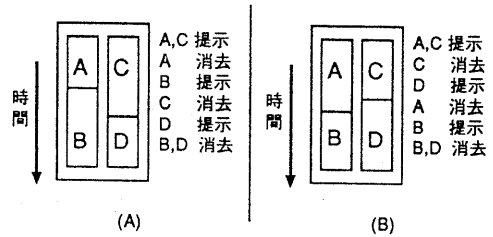


図2: メディアの提示時間による記述の変化例

3.2 拡張ベトリネットモデルによる状態遷移記述

筆者らは、前述の問題点を解決するため、論理構造記述においてベトリネット利用したマルチメディアデータの状態遷移記述モデル「拡張ベトリネットモデル」を提案する。

3.2.1 ベトリネット

拡張ベトリネットモデルの説明の前に、ベトリネットの概要を述べる [Reisig85]。ベトリネットは、図3のようにブレース、トランジション、および、ブレースとトランジションを結ぶアークから構成されるグラフである。ここで、ブレースは状態を表し、トランジションは状態遷移を表す。

ブレースにはトークンとよばれるマークを置くことができ、トークンとブレースにより現在の状態を表す。ここで便宜上、図3(b)のトランジションT1に向けてアークが張られているブレースP1,P2を入力ブレース、T1からアークが張られているブレースP3を出力ブレースと呼ぶ。

ベトリネットは、次のルールに従い、状態を記述する。

- トランジションは、全ての入力ブレースにトークンが存在する時、発火可能となる。
- トランジションが発火すると入力ブレースからトークンを取り去り、全部の出力ブレースにトークンを発生させる。

例えば、図3でP1とP2の両方にトークンが存在すれば、T1のトランジションが発火し、P1、P2からトークンが消去され、P3にトークンが発生する。

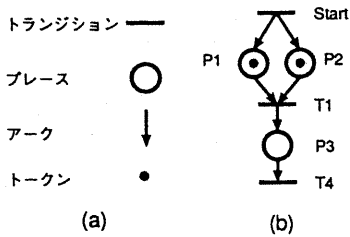


図3: ペトリネット記述

3.2.2 拡張ペトリネットモデル

ペトリネットをマルチメディアの状態遷移記述に対応させるため、基本的なものとして次の4つの拡張を行う。

- (1) プレースはトークンが発生すると、指定されたメディアを提示する。
- (2) メッセージという概念を新たに導入する。メッセージは、プレースがトークンが発生してから一定の処理が終了すると、プレースからトランジションに送られる。
- (3) トランジションにスタンバイ状態を導入する。スタンバイ状態は、発火可能な状態のことで、トランジションが接続する全ての入力プレースにトークンが発生している場合、発火するのではなくスタンバイ状態となる。スタンバイ状態のトランジションはプレースからメッセージを1つでも受け取ったら発火する。
- (4) ユーザ入力用トランジションを導入する。ユーザ入力用トランジションはスタンバイ状態になるとメニューボタンを提示し、ユーザからの入力があると無条件に発火する。

図4(1)は、あるMMAの画面の状態遷移を、(2)は(1)の状態遷移を示したペトリネットである。

図4(2)のA、B、C、D、Eはメディアを提示するプレースで、Fはメディアを提示せず一定時間経過したらメッセージを送るプレース、X、Y、Zはユーザ入力用トランジション、Tはボタンを提示せず、入力プレースからのメッセージによって発火するトランジションである。また、図4(1)では、プレース、トランジションと同じ名前のメディアやボタンを画面上に配置している。ここで、プレースAにトークンが発生した場合、メディアAが提示される。

画面の状態遷移の説明をすると、まず、画面(a)においてボタンXを選択すると画面(c)へ、ボタンYを選択すると画面(b)へ遷移する。また、画面(b)は一定時間が経過すると自動的に画面(c)へ遷移する。

図4(2)のペトリネットを実行したときの振舞を(1)の画面状態と照らし合わせながら説明する。

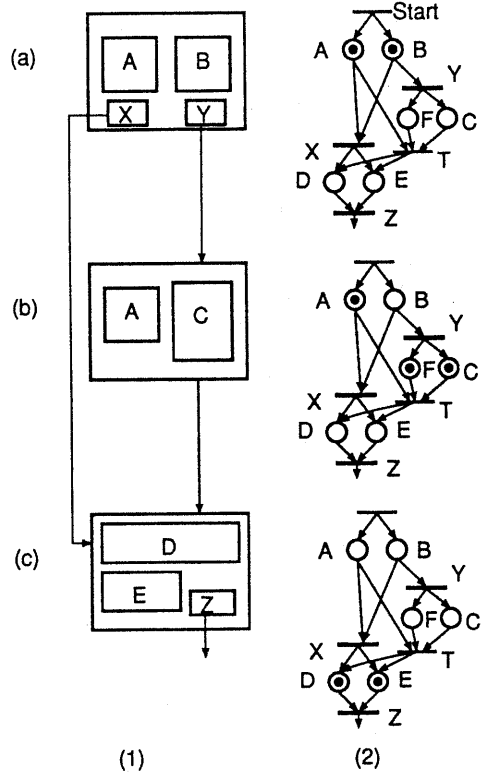


図4: 拡張ペトリネットモデルの記述例

(a) まず、A、Bのプレースにトークンが発生し、また、それと同時にX、Yのトランジションがスタンバイ可能になるため、メディアA、B、及び、メニューボタンX、Yを提示される。

(b) (a)の状態ではボタンYが選択されると、プレースBからトークンが取られ、プレースC、Fにトークンが発生する。これによりメディアBは消去され、代わりにメディアCが提示される。また、ここで、AはYの入力プレースではないので引き続き提示を続ける。

さらに、プレースA、C、F全てにトークンが存在するのでトランジションTはスタンバイ状態になる。ここでトランジションは、Fからメッセージを受けると発火する。

(c) (b)において一定時間を経過するとプレースFからメッセージがトランジションTに送られ、トランジションTが発火する。これにより、プレースA、B、Cからトークンが消去され、プレースD、Eにトークンが発生する。また、(a)でXを選択すると、トークンはプレースA、Bから消去され、プレースD、Eで発生する。

3.2.3 モジュール化

大規模 MMAP 作成には、状態遷移記述を複数のモジュールとして作成する機会が多い。例えば CAI 教材などでは、画面中のメディアの提示方法等、状態遷移が共通であることが多いため、共通の状態遷移をモジュール化し、簡単に利用できることが望ましい。また、MMAP が大規模化するに伴い、不必要な情報の提示により全体構造が把握しにくくなり、編集が煩雑になるという問題が発生する。視覚的な観点からもモジュール化は必要である。

拡張ベトリネットモデルでは、モジュール化する一連のベトリネットをサブネットとして作成し、モジュールを表すプレースで表現する(図5)。ここで、図5のトランジション T1 が発火すると(a)ではモジュール化したプレース M1 にトークンが発生する。この時、(b)ではサブネットのトランジション Start にメッセージが送られて発火し、サブネットのプレース Mp1、Mp2 にトークンが発生する。また、サブネットのトランジション MT1 が発火すると同時にトランジションに対しメッセージが送られ T2 が発火され、モジュール化したプレース M1 からトークンが消去される。

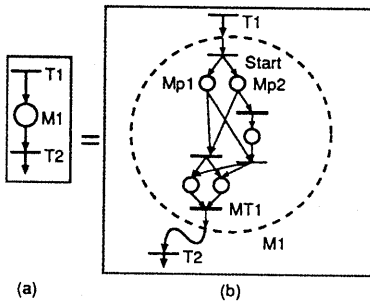


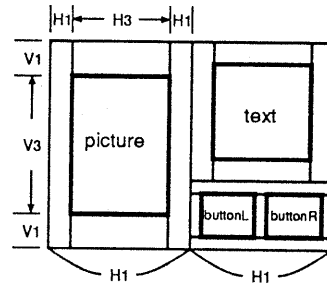
図5: モジュール化

4 スタイル記述

スタイル記述は、メディアの空間的な位置関係を記述する。TeX や InterViews [Linton89] などでは、空間的な位置関係記述を Box や Glue を用いて規定している。スタイル記述では、基本的にはこれらに準ずることにする。

図6のように大きさの比率によるボックスで画面を分割し、最終的にメディアを提示するボックスに領域名をつける。例えば、図6で picture の領域は、(1)から(4)のように定義できる。

- (1) 画面を横に 1 対 1 に分割する。
- (2) 分割した左側をさらに横に 1 対 3 対 1 に分割する。
- (3) (2) で分割した真中のボックスを今度は縦に 1 対 3 対 1 に分割する。
- (4) (3) で分割した真中の領域を picture の領域とする。



- (1) H1:H1
- (2) H1(H1:H3:H1) : H1
- (3) H1(H1:H3(V1:V3:V1) : H1):H1
- (4) H1(H1:H3(V1:V3(picture):V1):H1):H1

図6: 画面分割例

5 メソッド記述

メソッド記述は、論理構造記述、スタイル記述で記述できない視覚効果などの変化率や、メディアの移動時の軌跡の中間点などのパラメータなどを記述する。視覚効果を例にあげると、フェードインの場合、輝度の変化を図7のように複数用意することで、いろいろなバリエーションの視覚効果テンプレートを提供できる。ここで、座標値は規格化された値で記述しておく。

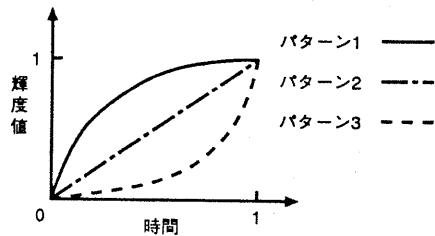


図7: フェードインにおける変化方法

6 各記述の結合

論理構造記述、スタイル記述、メソッド記述を組み合わせることにより、MMAP のテンプレートを作成する。

まず、論理構造記述とスタイル記述を結合するため、プレースで指定するメディアとスタイル記述で指定する提示位置との関係を記述したレイアウトプレースを導入する。レイアウトプレースにトークンが入ると、メディアは、レイアウトプレースで指定した画面テンプレートの特定領域に提示される。

図8は、図4(a)、(b)、(c)の画面状態遷移が記述できるように図4にレイアウトプレース P、Q、R を配置し

たベトリネットである。図8(a)は画面テンプレートで、S1、S2、S3、S4の領域が定義されている。また、レイアウトブレースには図8のような関係記述が含まれる。

ここで、レイアウトブレースで指定する画面テンプレート、または、領域を変更すれば、状態遷移は同じで、レイアウトだけが変更したMMAPを作成することが容易に実現できる。

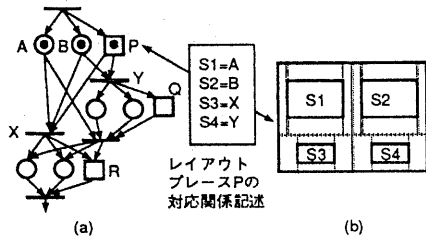


図8: レイアウトブレースを追加したベトリネット

図9のようにレイアウトブレース間だけの状態遷移記述によりメディアの移動を表現できる。

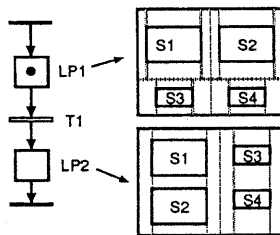


図9: メディアの移動

メソッド記述は、ブレースの提示属性およびトランジションの状態遷移属性として指定する。例えば、図9におけるメディア移動の軌跡はメソッド記述で座標値の組として記述しておき、その記述をトランジションの属性で指定することで実現する。

実メディアの指定は、ブレース、トランジション名にメディア名を対応させる。

7 まとめ

MMAP開発の工数削減のためにテンプレートを用いたオーサリングの重要性、および、テンプレートを実現するためのシナリオ記述体系について述べた。

本稿で提案した記述モデルは、以下のことが実現できる。

(1) シナリオを論理構造記述とスタイル記述、メディア記述に分離することで、状態遷移テンプレート、画面テンプレート、視覚効果のパラメータ用テンプレートがそれぞれ実現できる。

(2) モジュール化したブレースの採用により、AP共通の状態遷移を共通化することができ、さらにプレゼンテーション用テンプレートやCAI教材用テンプレートといったAP特有の状態遷移をライブラリとして提供することができる。

(3) 拡張ベトリネットによる論理構造記述やスタイル記述による画面位置指定により提示位置や提示時間を直接指定しない記述が実現し、メディアのサイズや提示時間による変更があっても、修正の必要が無い。

(4) 規格化した値で指定する等、プラットフォームに依存しない記述を実現できる。

今後の課題として考えられるものとして以下のものがある。

(1) ベトリネットを直接編集するには、それなりの訓練が必要であるため、編集を容易にするための編集モデル、ツールの開発が必要である。

(2) シナリオはマルチプラットフォーム対応であるが、細かい同期タイミングや高速性が要求されるMMAPでは、プラットフォームに依存したランタイム記述が必要である。これに伴い、ランタイム記述に変換するコンパイラの開発も必要となる。

(3) シナリオがデータベース化された時のシナリオの検索方法の開発。

(1)の問題を解決するため、現在、提示される画面上のメディアを編集し、画面の提示/消去毎に状態遷移を作成することで、ベトリネットを作成するという編集ツールの試作を行っている。さらに、(2)についても同時に開発中である。これらに関する詳細はまた別稿で紹介する。

参考文献

- [原田91] 原田、小川: シナリオプロセッサ概念レベルのシナリオ作成支援環境一、信学技法、DE91-21、pp.67-82、1991
- [小川92] 小川、原田他: マルチメディアオーサリングにおけるデータ管理について、情報処理学会研究報告、92-DBS-90、1992
- [田口93] 田口、原田、小川: マルチメディア・オーサリングにおけるインタフェース改善の一考察、情報処理学会第46回全国大会、3-303、1993
- [藤村93] 藤村: HyprODAの最新動向、アドバンスドデータベースシンポジウム'93講習会資料、pp.21-30、1993
- [亀山93] 亀山: MHEGの最新動向、アドバンスドデータベースシンポジウム'93講習会資料、pp.31-70、1993
- [Hardman93] Hardman, Rossum, Bulterman: Structured Multimedia Authoring, ACM Multimedia 93, pp.283-289
- [Linton89] M.Linton, J.Vlissides, P.Calder: Composing user interface with InterViews, Computer, 1989
- [Reisig85] W.Reisig: Petri Net: An Introduction. Springer, New York, 1985
- [ACTION] MacroMind ACTION ユーザガイド、MacroMedia社、1991
- [AuthorWare] AuthorwareStar リファレンスブック、MacroMedia社、1991