データベースシステム 90－5
（1992. 9. 11）

マルチメディアオーサリングのためのシナリオデータモデル

田中 栄市郎　小川隆一
NEC C&C 情報研究所

マルチメディアアプリケーションのオーサリングにおけるシナリオ化工程を支援するためのデータモデルについて述べる。シナリオ化工程はハイパーメディア構造設計の視点からみれば、ハイパーメディア構造のテンプレートを作成することに相当する。本稿では、ハイパーメディア構造設計におけるテンプレートの有効性について検討し、構造設計支援をするためのテンプレートモデルを提案する。本モデルにより、(1) 開発工数の削減、(2) トップダウン設計による無駄のない、わかりやすい構造設計 (3) 分散オーサリングの実現、が可能となる。また、筆者らが開発した英語ヒアリング教材のハイパーメディア構造の設計工数が大幅に削減されることを示す。

A Scenario Writing Data Model for Designing Multimedia Applications

Eiichiro TANAKA　Ryuichi OGAWA
C&C Information Technology Research Laboratories,
NEC Corporation
1 - 1 Miyazaki 4-Chome, Miyamae-Ku, Kawasaki, Kanagawa 216 Japan

This paper describes a scenario writing data model for designing multimedia applications. From a hypermedia designing view point, the scenario writing can be regard as process of the template designing about hypermedia structure. This paper proposes a scenario writing data model named “template model” to design hypermedia structure. The proposal model provides (1)reducing authoring cost,(2)simple hypermedia structure by a top down designing strategy,(3)sharing authoring.

1 はじめに

近年、ハードウェアの進歩にともないコンピュータによるマルチメディア情報の取扱いが容易になってきた。特に、メディア編集を主にするオーサリングツールの普及により簡単なマルチメディアアプリケーションの作成が身近なものになってきたが、時間軸に基づくシーケンシャルなマルチメディア提示を簡易な記述で実現するハイパーメディアツールが提供されていない。

筆者らはこれに対し、ハイパーメディア概念を拡張したオーサリングシステム「ビデオブック」の開発を行ってきた [1][2]。ビデオブックでは、音声／映像を含むマルチメディアの画面レイアウト／提示タイミング (スタイルと呼ぶ) が記述されているシナリオを、複合ノードとして視覚的に編集できる。

しかし、大規模なストーリー主体のアプリケーションを作成する場合、スタイルだけでなく抽象度のことなるいくつかのシナリオ記述レベルが必要となる。例えば小川は、このシナリオ化の工程を次の4つのレベルに分類している [3]。

(1) 全体構造
アプリケーション構造を順序関係を内包する階層構造として記述する。各層のノードは、ストーリー中で内容の独立した部分をモジュール化したものである。

(2) 詳細構造
全体構造で記述された各モジュールの内部構造を、下位ノードとそれらを結ぶ状態遷移リンクで記述する。また、全ての上位ノードの基本となる提示単位 (例えばカードやフレーム) を定義する。

(3) 実体仕様
全ての上位ノードの基本となる提示単位ノードで用いられるメディアの仕様 (音声スクリプトなどのデータ実体仕様) を決める。

(4) スタイル
提示単位ノード中のメディアやメニューボタンの提示スタイル (画面上の位置と提示タイミング、メニューボタンの機能) を記述する。

これらのシナリオ化工程をトータルに支援するアーキテクチャについて、検討すべきことは非常に多い [4]。本稿では特に、上記レベル (2) のためのシナリオ作成支援方法を考える。(2) の詳細構造の設計は、ハイパーメディア構造の設計に相当するが、これに対する適切な設計方法論やモデルは確立しておらず、様々な模索が行われている。本稿ではまず、上記シナリオ化の立場から、ハイパーメディア構造設計における要件を示す。次に、これに基づいて筆者らが提案するテンプレートモデルについて紹介し、最後に提案したモデルの有効性について検討する。

2 シナリオ作成とハイパーメディア

2.1 シナリオ化の意義

まず最初に、ハイパーメディア構造の設計におけるシナリオ化工程の意義を明確にしておく。シナリオ化はトップダウン的なハイパーメディア構造設計の一種であり、タイプ化ノード、タイプ化リンクによるネットワーク構造として定義することに相当する。言い替えればシナリオ化工程はハイパーメディア構造のテンプレートを作成することである。特に、共通のハイパーメディア構造を持つアプリケーションのオーサリングでは、テンプレートの再利用により開発工数を大幅に削減できる。

さらに、ハイパーメディア構造のテンプレート作成により、構造のモジュール化が容易にできる。そのため、モジュール化による構造のトップダウン的な設計が可能となり、無駄のない、わかりやすい構造のアプリケーションが作成できる。また、モジュール単位でハイパーメディア構造の設計ができるため、複数人による分散オーサリングが実現できる。

そこで、筆者らは、ハイパーメディア構造を、テンプレートを用いてトップダウン的に設計し、実体のノードをテンプレートにいれることでアプリケー

ションを作成するオーサリングスタイルを考え、これを実現するためのモデルを提案する。

2.2 ハイパーメディア構造設計における要件

テンプレートを用いてハイパーメディア構造の設計を支援しようとする研究はいくつかなされている。例えば、IDE[5]では、ハイパーテキストのノード、リンクをタイプ化ノード、タイプ化リンクによる構造化テンプレートで設計するという試みがなされているが、ノードはテキストのみでマルチメディアに対応しておらず、また、プラットフォーム（Notecards）に依存した記述である。また、Intermedia[6]では、ハイパーメディアデータのコピーによりテンプレートを実現しているが、これもプラットフォームに依存し、ノードの実体がテンプレートの作成に必要である。一方、Trellis[7]は、プラットフォーム独立の形式で、ハイパーメディアの状態遷移を記述できるが、記述が難解でコンピュータの専門家でないと扱えない。SEPIA[8]は、デザインオブジェクトクラスによるノードのクラスが規定されているが、階層構造中心である。

以上のことをふまえ、テンプレートを用いたハイパーメディア構造の設計支援における要件をまとめると、以下のようになる。

(1) オーサリングが簡単にできる。

(2) 状態遷移を自由にタイプ化できる。

(3) プラットフォーム独立である。

以下にこのような機能に基づいたテンプレートモデルについて詳しく述べる。

3 テンプレートモデル

本モデルは、4種類のオブジェクト、すなわちインスタンスノード／フレームノード／ポート／モジュールノードで構成される。

3.1 インスタンスノード

インスタンスノードは、提示単位ノードの実体である。メディアやメニューボタンの提示スタイルの情報、メニューボタンを選択された場合のリンク先の情報を持つ。リンク先は、3.1.3節で述べる出力ポートを指定する。インスタンスノード内の設計や詳細な構造については、別稿にゆずる。

3.2 フレームノード (図1)

フレームノードはタイプ化ノードで、内部にインスタンスノードを記述するテンプレートである。フレームノード名、フレームノードタイプ、インスタンスノード名、ポート情報を属性として持つ。フレームノードタイプは、内部に入るインスタンスノードのタイプを規定するもので、指定されたタイプ以外のインスタンスノードは入らない。

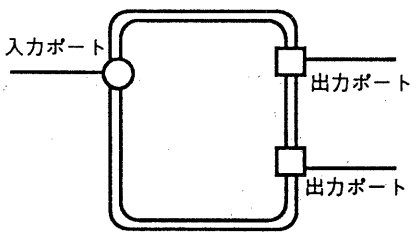

図1: フレームノード

3.3 ポート

ポートは、タイプ化リンクを記述するテンプレートである。ポート名、ポートタイプ、ターゲット名、ターゲットポート名を属性として持つ。

ポートタイプはポートが、リンク元を表す出力ポートか、リンク先を表す入力ポートのどちらかを指定する。ターゲット名はリンク先のフレームノード名を示し、ターゲットポート名はリンク先のフレームノードのポート名を表す。ポートはフレームノード中に幾つでも設定できる。また、リンクは、出力ポートが入力ポート名を記述することで表現する。図2は、内部にそれぞれインスタンスノードa、インスタンスノードbを持つフレームノード

A、フレームノードBを示したものである。Pa1、Pa2、Pa3はAのポート、Pb1、Pb2はBのポートを表す。

図2のAにおけるポートの記述は以下のとおりである。

ポート名	ポートタイプ	ターゲット名	ターゲットポート名
Pa1	IN		
Pa2	OUT	B	Pb1
Pa3	OUT	—	—

例えば、一行目のPa1は入力ポートで、二行目のPa2は出力ポートで、Bのポート Pb1にリンクが張られていることを示す。

aにはAのポート(Pa1,Pa2,Pa3)を指定する記述(La1,La2,La3)がある。また、bにはBのポート(Pb1,Pb2)を指定する記述(Lb1,Lb2)があるため、図2において、aとbをフレームノードに組み込めば、La2-Pa2-Pb1-Lb1というリンクのパスが設定される。

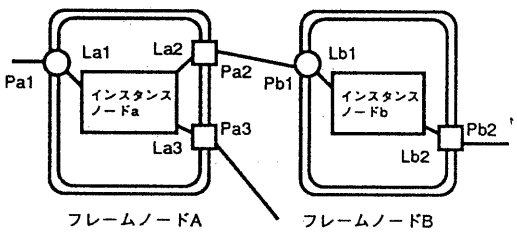

図 2: フレームノード間のリンク

3.4 モジュールノード

モジュールノードは集約ノードのテンプレートである。モジュールノードは、図3で示すように、フレームノードのハイパーメディア構造をモジュール化したもので、フレームノードと同様にポートを持つ。そのため、モジュールノードはフレームノードと同様に扱うことができる。モジュールノード名、モジュールノードタイプ、フレームノード情報、ポート情報を属性として持つ。フレームノード情報は、モジュールノードを構成しているフレームノードの数と、フレームノード名を属性として持つ。

3.5 モジュールノードにおけるリンクの制約

モジュールノードの内部情報(フレームノードの数や構造)は、外部から隠蔽される。外部のフレームノードからは、直接内部のフレームノードにリンクを張ることができず、必ずモジュールノードのポートを介してリンクを張らなくてはならない。また、内部のフレームノードも直接外部のフレームノードにリンクを張ることができず、必ず、モジュールノードのポートを介してリンクを張らなくてはならない。

モジュールノードのポートについて、例をあげて説明する。図3のモジュールノードModuleAにおけるポート情報は、以下のとおりである。

ポート名	ポートタイプ	ターゲット名	ターゲットポート名
Mp1	IN	frame1	P1
Mp2	OUT	ModuleB	Mpb

Mp1,Mp2はそれぞれModuleAの入力ポート、出力ポート、frame1はModuleA内部にあるフレームノード、P1はframe1の入力ポート、ModuleBはモジュールノード、MpbはModuleBの入力ポートを表す。

Mp1は、ターゲットとしてframe1のP1を指定しているので、例えば、外部フレームノードframeEの出力ポートPEがターゲットにModuleAのMp1を指定した場合、PEのターゲットはframe1のP1である。

Mp2は、ターゲットとしてmoduleBのMpaを指定しているので、例えば、フレームノードframe2の出力ポートP2がターゲットにMp2を指定した場合、P2のターゲットはModuleBのMpbである。

3.6 モジュールノードの階層化

モジュールノードは内部に持つフレームノードの数やポートの数を制限しない。ただし、長大なモジュールノードを構造化/モジュール化できるように、モジュールノードの階層的記述を許す。これにより、モジュールノードの内部を階層的に詳細化し

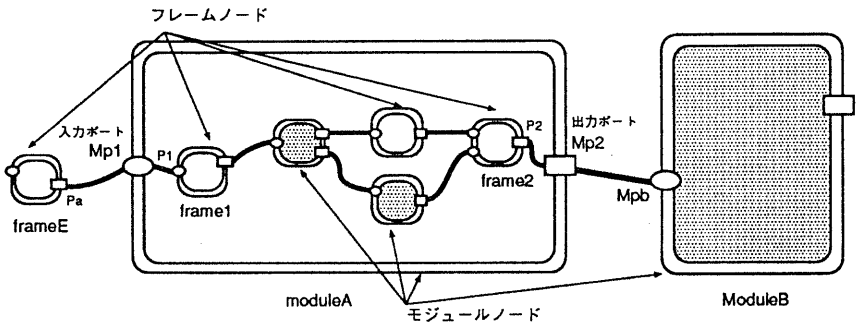

図 3: モジュールノード

ていくトップダウン的なハイパーメディア構造の設計が可能となる。

3.7 テンプレート化と再利用

モジュールノードは内部にあるフレームノードのノード名を属性に持つ。再利用する場合は、フレームノードが指定するインスタンスノード名を指定し直せばよい。

4 テンプレートモデルによる工数削減

ここでは、筆者らが開発したビデオブック上のアプリケーションである英語ヒアリング教材 [9] を例にして、テンプレートの有効性を検討してみる。

図4は英語ヒアリング教材における「設問」のモジュール内部の構造を表している。「設問」のモジュールは、質問 (Question)、解答 (Answer)、聞き取り (Passage)、ヒント (Hint)、確認 (Confirm) のフレームノードから構成される。図4においてQuestion1、Answer1 は、それぞれフレームノード Question、Answer が指定するインスタンスノードを表している。このモジュールノードをコピーし、インスタンスノード名を変えれば、共通なハイパーメディアの状態遷移を簡単に作成できる。

英語ヒアリング教材は、この「設問」のモジュール構造が 70 近くあり、設問の後に補助的な学習を行わせる「学習作業」のモジュール構造においては 100 近くある。ノード数でいえば、「設問」「学習作業」は、ともに全体の構造の約 1/5 を占める。これら共通な構造をモジュールノードとして作成し、再利用することにより、構造設計の工数を大幅に削減することができる。

このように、テンプレートモデルは、英語ヒアリング教材のような共通の状態遷移構造を多くもつアプリケーション設計に有効であると考えられる。

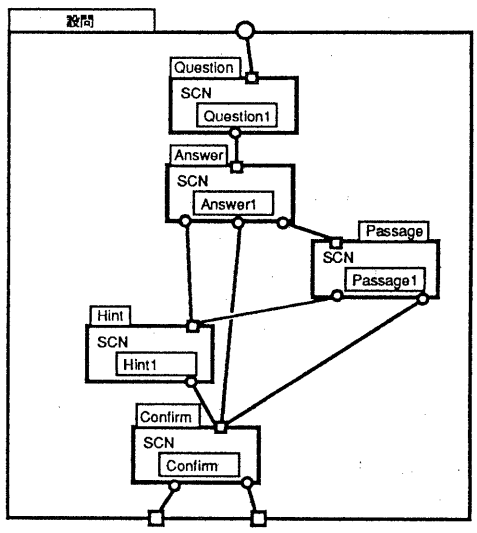

図 4: 「設問」のモジュールノード

5 おわりに

ハイパーメディア構造設計におけるテンプレートの有効性について検討し、(1) 開発工数の削減、(2) トップダウン設計による無駄のない、わかりやすい構造設計 (3) 分散オーサリングの実現、が可能であることを確認した。そこで、ハイパーメディア構造をテンプレートを用いて設計し、実体のノードをテンプレートに入れることでアプリケーションを作成するオーサリングスタイルを考え、構造設計支援をするために、インスタンスノード／フレームノード／ポート／モジュールノードの4種類のオブジェクトを基本とするテンプレートモデルを提案した。本モデルを用いた構造設計をすることにより、筆者らが開発した英語ヒアリング教材のハイパーメディア構造の設計工数が大幅に削減されることを示した。今後は、本モデルに基づくハイパーメディア構造作成支援システムを開発していく予定である。

謝辞

最後に本稿を作成するに当たり、多大なる助言をしていただいたNEC パーソナル C&C 開発研究所の松本英博氏に感謝いたします。

参考文献

[1] R.Ogawa,H.Harada,A.Kaneko:Scenario-based Hypermedia: A Model and a System,Proc. of ECHT'90,pp.38–52(1990).

[2] 小川、原田：マルチメディアシナリオ記述のためのデータモデルとオーサリング環境について、信学技法、DE91–3、pp.17–24(1991).

[3] 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).

[4] 原田、小川：シナリオプロセッサー概念レベルのシナリオ作成支援環境ー、信学技法、DE91–21、pp.67–82(1991).

[5] D.Jordan, D.Russell, A.Jensen, R.Rogers:Facilitating the Development of Representations in Hypertext with IDE, Proc. of Hypertext'89,pp.93–104.(1989)

[6] K.Catlin, L.Nancy:Hypermedia Templates:An Author's Tool, Proc. of Hypertext'91,pp.147–160.(1991).

[7] D.Stotts:Dynamic Adaptation of Hypertext Structure, Proc. of Hypertext'91,pp.219–231.(1991).

[8] M.Thuring,J.Haake and J.Hannemann:What's Eliza doing in the Chinese Room? Incoherent hyperdocuments – and how to avoid them,Proc. of Hypertext'91,pp.161–177.(1991).

[9] 小川、田中、田口他：英語ヒアリング学習におけるマルチメディア提示：CD-ROM 教材の開発と試行、第6回人工知能学会全国大会、S–2–4、pp.81–86、(1992).