

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

小川 隆一、原田 浩明、田中栄一郎、田口 大吾 (日電)

(口頭発表のみ)

(*原稿は次回研究報告(資料 No. 92-DBS-90) に掲載)

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

小川 隆一 原田 浩明 田中 栄市郎 田口 大悟
NEC C&C情報研究所

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化とデータ管理について述べる。ストーリー主体のマルチメディアアプリケーションを開発する場合、データ実体作成の前にストーリーやメディアの仕様を文書化することが多い。これはデータ管理の視点から見ればスキーマ設計に相当し、ストーリー依存のスキーマを、システム上で管理しやすいデータ構造に変換することがシナリオ化の重要課題になる。本稿では、筆者らが開発中のCD-ROM版英語ヒアリング教材を例として、シナリオ化におけるデータ管理戦略について検討する。実用規模の教材データ(オブジェクト数10,000)をMS-DOS上のハイパーメディアツールで効率よく実装/管理する戦略を具体的に示す。

Data Management Strategies for Designing Interactive Multimedia Applications

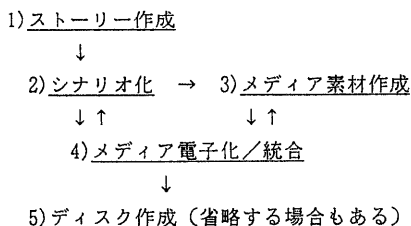
Ryuichi Ogawa, Komei Harada, Eiichiro Tanaka, Daigo Taguchi
C&C Information Technology Research Laboratories
NEC Corporation

This paper describes data management strategies for designing interactive multimedia applications. In order to design a large-sized, story-based application, it is common to prepare a scenario which specifies details of the story and media data contents. From a data management viewpoint, the scenario writing can be regarded as a process of the schema design. In the process, well considered data management strategies are required to translate a story dependent schema into a manageable data structure of a targeted authoring platform. This paper shows the strategies actually employed by the authors to develop a CD-ROM based interactive English listening course for Japanese students.

1. はじめに

教育／娯楽／プレゼンテーション分野において、ストーリー（原作）に基づくマルチメディアアプリケーションを作成する場合、データ実体作成の前にアプリケーション仕様を文書化（シナリオ化）することが一般的である。音声・映像を多用するアプリケーションの場合は、メディア仕様を記述するために、シナリオの重要性はより高まる。

実用規模のアプリケーションのオーサリングは、図1のように、企画からシナリオ化、メディア作成、ディスク作成までのフェーズを含む複雑な工程である。工程全体の中でのシナリオ作成の意義は[原田91][小川91]で指摘されているが、現在我々がコンピュータ上で支援可能なのは、メディア電子化／統合のプラットフォームにとどまっている。しかも現状のプラットフォームはそれを単独で利用することが前提であり、シナリオ化の結果をスムーズにプラットフォーム上へ移す枠組み（あるいはその逆の枠組み）が提供されない。



ハイパーメディアツールは工程4)を支援

図1 オーサリング工程

これを改善するため、筆者らは拡張ハイパーメディアシステム「ビデオブック」を開発した[Ogawa90]。ビデオブックでは、音声・映像を含むマルチメディアの提示スタイルを記述するシナリオ（いつ、画面上のどこにどのメディアを提示する）を複合ノードとして視覚的に編集、提示できる。この視覚シナリオ（シーンと呼ぶ）は、プレゼンテーションなどの比較的小規模のアプリケーション作成（ノード200個程度）には有効であるが、ノードが1,000をこえる大規模のアプリケーション設計のためにどんなシナリオ化を行なえ

ばよいか、については十分議論されていない。その本質的な課題として、以下の2点があげられる。

① 対話性を備えたマルチメディアアプリケーションのストーリー記述方式が未知であり、作者が訴えたいことを表現する方法論がない。これはハイパーメディア設計論の不在にも関連する問題である。最近のハイパーメディア研究では、テンプレートや概念スキーマをトップダウン的に与え、高品質なアプリケーション作成を試みている[Jordan89][Garzot91]。一種のシナリオ化による設計支援とみることもできるが、この議論は別稿にゆずる。

② マルチメディアアプリケーションのシナリオ化は、抽象度の高いストーリーを詳細化し、ターゲットシステムで実装可能なデータ構造を与える過程とみなせる。アプリケーション規模が大きい場合、データをいかに効率よく管理するか、がシナリオ化の要件になり、スキーマ設計／データ構造設計の要素が非常に強くなる。（以下の議論では、「シナリオ化」と「スキーマ設計」をほぼ同義に用いる。）一方で、ストーリーや検索フローの設計が基本となること、ターゲットシステムが必ずしも強力なデータベース機能を備えていないことは、通常データベース設計とは異なる特徴であり、これらを考慮したシナリオ化の手法を構築しなければならない。

本稿では、②の問題を詳細に検討する。まず第2章で、ストーリーを段階的にシナリオ化するため、1)全体構造、2)詳細構造、3)実体仕様、4)スタイルの4つのスキーマ設計レベルを導入する。次に第3章で、ストーリー主体のアプリケーション例として、筆者らが開発中のCD-ROM版英語ヒアリング教材の概要を述べる[小川92-1]。第4章では、教材オーサリングにおいて、各設計レベルで採用された戦略を述べる。第5章で、教材オーサリングの総合評価と、シナリオ化の課題について検討する。

2. シナリオ化とデータ管理

2.1 問題の整理

実用レベルのアプリケーションを想定して、以下の前提をおく。

1) 編集マルチメディアオブジェクト数は10,000、画面数は1,000のオーダーである。格納媒体としてCD-ROM等の光ディスク媒体を用いる。

2) アプリケーションは骨子となるストーリー（また

は検索パス)を持ち、その論理構造は、文書の章/節のような順序関係を内包する階層構造で表わされる。ストーリー性の弱いハイパーメディアアプリケーションについても、ノードの集約関係を同様な階層構造で表現し、スキーマ設計を行なうことが有効である[Garzot91]。

3)ターゲットとなるプラットフォームはPC上のハイパーメディアシステムである。ハイパーメディアのノード/リンク構造とオーサリング機能は、種々のアプリケーション(ハイパーメディアに限らない)作成に最も柔軟に対応し得る。反面、検索機能はブラウザ/ナビゲーション主体であり、10,000個のオブジェクトを効率よく編集/管理するデータベース機能は期待できない。

シナリオ化は、順序関係を内包する階層構造をノード/リンク構造へ、さらにノード中の個々のメディアへブレイクダウンする過程とみなせる。この作業は、例えばHyTime[HyTime91]やHyperODA[ODA91]のようなドキュメント標準化仕様にそって行なうのが一見効果的に見える。しかし、HyTimeやHyperODAはあくまでデータ交換仕様であり、実際の設計戦略は何も与えてくれない。仕様とは独立に、データ管理を十分考慮したスキーマ設計戦略が必要である。

2.2 データ管理面からの要件

データ管理の面からの設計戦略の要件を整理しよう。

2.2.1 ストーリー分割

大規模アプリケーションのストーリーを部分構造(サブストーリー)に分割し、オーサリング対象範囲を適性規模にすることが基本である。ブラウザで処理しきれないような数のオブジェクトを常に編集している場合は、オーサリング効率は落ちる。

2.2.2 モジュール/パターンの導入

ストーリー中の内容的に独立した部分をモジュール化し、よく用いられるモジュールのパターン(タイプ)を定義することもデータ管理上重要である。この場合、ストーリーをオブジェクト指向的にクラス階層で表わせるかどうかは興味ある問題だが、筆者らはクラスは強すぎる制約で、ハイパーメディア的な自由度を損うと考える。ここでいうモジュールは、以下のようなゆるい制約を持った集約ノードである。1)独立した意味単位/機能単位である、2)内部構造(下位のノード/リンク構造)に固有のパターンがある、3)他のモジ

ールとのリンク(検索パス)が少ない。

2.2.3 提示単位の定義

ハイパーメディア用語でいえば、情報提示の基本単位となる複合ノードと、複合ノード間の状態遷移リンクの定義である。アプリケーション中で提示単位が一貫すれば、状態遷移が明確化し、ノード/リンク構造が平易になる。通常「フレーム」「カード」等の画面ベースの提示単位ノードが定義できるが、音声・動画等の時間依存メディアを含むシーケンシャルな提示単位をいかに定義するか、が重要である。

2.2.4 提示単位とデータ実体との分離

同一のデータ実体を複数ノードで共有することは、作成コストがかかる時間依存メディアのデータ量を削減できる点で非常に重要である。このため、提示単位ノードにはスタイルのみを記述し、データ実体仕様は別途記述し、提示単位ノードと対応づけることが望ましい。

2.3 スキーマ設計のための4レベルモデル

上記要件を考慮して、スキーマブレイクダウンのための4つの設計レベルを定義する。以下では、記述の抽象度の高い順にレベルを説明するが、実際の設計プロセスは必ずしもこの順序のとおりではない。

2.3.1 全体構造(Global structure)

ストーリー全体を順序関係を内包する階層構造として記述する。階層の各ノードは2.2.2章で定義したモジュールである。下位ノードがその内部構造になる。後述する英語ヒアリング教材では、例えばオープニング/設問/解説/練習、等のモジュールが定義される。全体構造が大きい場合、2.2.1章に従い、全体構造を複数の部分構造に分割する。この分割のしかたがここでの設計ポイントである。

2.3.2 詳細構造(Detailed structure)

部分構造の各モジュールの内部構造を、下位ノードとそれらの間の状態遷移リンクで詳細に記述する。すべての上位ノードの基本となる提示単位ノード(2.2.3章)をいかに定義するかが、ここでの設計ポイントである。上記例では、例えば「設問」モジュールが、問題/解答/ヒント/会話聞き取り/辞書、等の提示単位ノードとその間の遷移リンクで表現される。

2.3.3 実体仕様(Content specification)

個々の提示単位ノードをどんなメディアで表現するか、データ実体仕様を記述する。(2.2.4章に従い、提示単位ノードとは独立に記述しておく。) 上記例では、英語音声/日本語ナレーションスクリプトや、英会話場面の静止画・動画の仕様がこれにあたる。どのデータがどのノードで共有できるかを明確にすることが重要な設計ポイントである。

2.3.4 スタイル(Presentation style)

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(画面上の位置とタイミング)を記述する。時間依存メディアを含み、シーケンシャルに提示されるノードのスタイル記述には、ビデオブックの「シーン」のような時間概念が必要となる。

以上は非常に概念的な設計モデルであるが、筆者らがCD-ROM版英語ヒアリング教材設計を行なった際に、実際に指針としたものである。以下では、本教材の概要を紹介し、設計の詳細を述べる。

3. 英語ヒアリング教材の概要

3.1 背景

語学教材は、1)よい語学学習ソフトへのニーズが非常に高い、2)種々の教育法(コースウェア)や音声・映像素材の蓄積がある、3)他のアプリケーションに比べ評価が容易にできる、等の理由により、マルチメディアアプリケーションの有望な応用分野と考えられている。

筆者らは、千葉大学教育学部との共同研究により、日本人学習者の英語ヒアリング力を効率よく向上させるマルチメディア学習教材を開発している。CD-ROM版英語ヒアリング教材は、この一貫として当社社内研修用に開発した教材で、ビデオブックをオーサリングプラットフォームとして用いている。本教材のコースウェア設計からCD-ROM化までの全オーサリング工程に筆者らが関与し、教材の学習効果測定、ビデオブックの評価/改善、シナリオ作成支援方式の開発などを行っている[小川92-2]。

3.2 基本仕様

本教材は、PC環境における自習用として設計された。対象ユーザはPC利用/英会話ともに初級の日本人学習

者である。学習素材として、彼らがアメリカ旅行中に経験する日常的な5個のトピック、18個の会話場面(パッセージ)を選定し、これを音声/静止画(写真、イラスト)により提供する。アプリケーション規模を指標で示すと以下ようになる。音声データ(英語/ナレーション)の量が多いことが特徴である。

CD-ROM枚数	5
予定学習時間(1回)	30分
予定学習時間(全体)	15時間
音声ファイル	2,300
テキストファイル	600
イメージファイル	500
画面(提示単位ノード)数	1,800

教材の基本構造(学習フロー)は、3ラウンド制と呼ぶタスク指向学習方式に基づく[竹蓋91]。本方式では、学習者は関連する複数のパッセージについて、3段階にわけて聞き取り学習を行なう。各段階(ラウンドと呼ぶ)では、それぞれ1)キーワード、2)会話の詳細、3)全体の文脈や話者の心理に関する設問が課される。設問に答えられない場合には、聞き取りのポイントや会話の背景を説明するヒントが参照できる。図2に本方式の学習フローを矢印で示す。図中の「ユニット1、2・・・」は同じトピックに関するパッセージをまとめた単位であり、CD-ROM一枚に対応する。前のラウンドでの学習が次のラウンドのヒントとなるように学習フロー/設問が設計され、学習者が難しすぎる、

ユニット1 アメリカ入国

パッセージ	内 容	ラウンド1	ラウンド2	ラウンド3
1	入国手続き(1) 入国手続き(2) 税関での荷物検査	↓	↓	↓
2		↓	↓	↓
3		↓	↓	↓

ユニット2 ホテルにて

パッセージ	内 容	ラウンド1	ラウンド2	ラウンド3
4	チェックイン ベルボーイとの会話 ルームサービスを頼む チェックアウト	↓	↓	↓
5		↓	↓	↓
6		↓	↓	↓
7	↓	↓	↓	

ユニット3 公共交通機関に乗る

パッセージ	内 容	ラウンド1	ラウンド2	ラウンド3
8	乗るバスをたずねる 旅行代理店での会話 バス運転手との会話 列車の時刻をたずねる	↓	↓	↓
9		↓	↓	↓
10		↓	↓	↓
11	↓	↓	↓	

図2 3ラウンド制による学習フロー

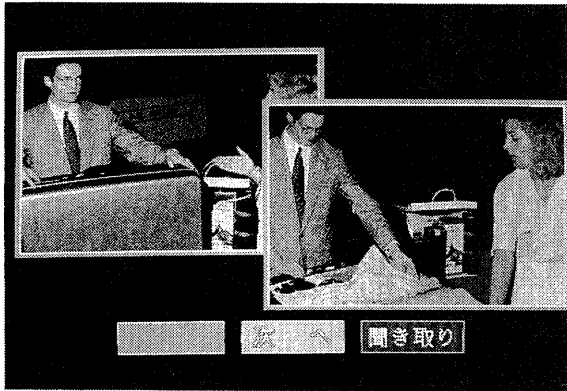


図3 英語ヒアリング教材画面例

または簡単すぎてつまらない、と感じないように配慮している。図3にビデオブック上に実装された教材の画面例を示す。

3.3 オーサリングプロセス

オーサリングの概要は以下のようである。

- 1) 企画：3.2章で述べた教材の基本仕様を決定する。
- 2) 基本コースウエア作成：ストーリー（会話場面や設問など）とその分割戦略を決定し、コースウエア仕様書を作成する。
- 3) 詳細化：提示単位ノードを決定し、コースウエア仕様書をビデオブックのノード／リンク構造で詳細化する（ビデオブックでは提示単位ノードをシーン、リンクをトリガと呼ぶ）。
- 4) メディア素材作成：音声素材（英語スクリプト／日本語ナレーション）、画像素材（写真／イラスト）を仕様書に基づき作成する。
- 5) ビデオブック上での電子化／統合：ビデオブックのツール群を用いてメディア素材を電子化、編集する。同時に、シーン中のメディア提示スタイルを編集する。
- 6) CD-ROM化：編集されたデータをCD-ROM化する。

上記のうち、シナリオ化に関わる工程は2)3)5)である。これらのプロセスは必ずしもシーケンシャルに進まず、途中多くのフィードバックループが発生した。また、2.2、2.3章で述べた設計戦略について、いくつかの経験則が得られた。次章で、これらのうちで重要なものについて詳しく述べる。

4. 各レベルでの設計戦略

4.1 全体構造

ストーリーの全体構造は3ラウンド制に基づき、ユニット／ラウンド／パッセージの階層構造で表わされる（図4参照）。最下層（図で最も右）の隣接モジュール間にはリンクが定義され、学習者はこれに従い、ラウンド1/パッセージ1→ラウンド1/パッセージ2→ラウンド1/パッセージ3→ラウンド2/パッセージ1→・・・の順に学習を進める。

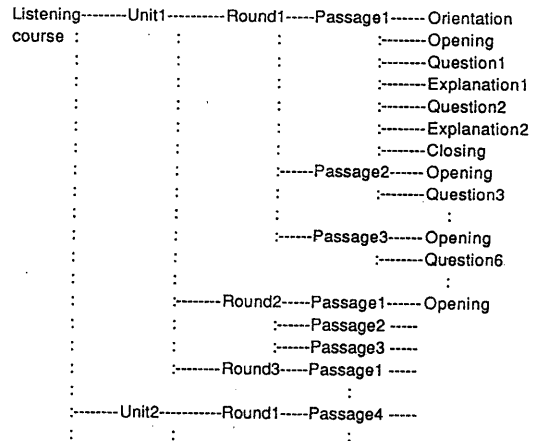


図4 全体構造

全体構造を分割する最も素直な方法として、図4の構造をそのまま分割してみよう。ハイパーメディアシステム（ビデオブック）で一括編集可能な範囲を考えると、図5のようなユニット／ラウンド単位の部分構造ができる（編集オブジェクト数は600のオーダー）。モジュールの順序がフローを反映するという意味で、これをフロー指向分割と呼ぶ。この場合、同一パッセージのモジュールが3個の部分構造に分散するため、パッセージの更新が各構造に波及しうる。一度に編集する範囲を部分構造とするのだから、これは明らかに不便である。実際問題として、パッセージ単位の更新が頻繁に生じるため、この分割は効率的でない。

これに対し、同一パッセージのモジュールをすべて集めた部分構造を考える（図6参照）。パッセージ固有のデータ（例えば会話場面の音声・画像）が局所化されるという意味で、データ指向分割と呼ぶ。図6では、学習フローは、部分構造間を飛び移る複雑なリンク表現になってしまうが、データ管理の視点からは、パッセージ単位の更新を局所化できる点で有利である。

興味深いことに、英語教師は、フローどおりラウンド1/パッセージ1→ラウンド1/パッセージ2・・・とコースウェア仕様を記述するより、パッセージ単位にまとめて記述することを好んだ。すなわち彼らにとって、フローよりもデータを中心に設計を行なうことが自然であり、データ指向分割はこの点で理にかなっている。以上から、今回の設計では、図6のパッセージ単位の部分構造をMS-DOS上のディレクトリとして作成した。

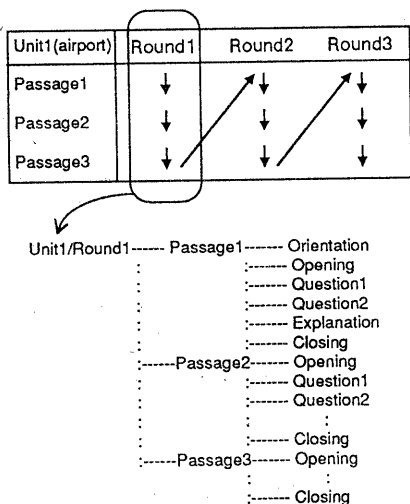


図5 フロー指向分割

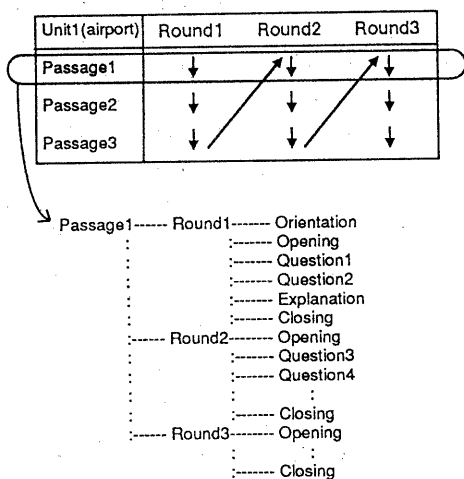


図6 データ指向分割

より一般的な指針としてまとめると、ストーリーの分割戦略として、フローに沿った分割は必ずしも設計上効率的でない。むしろ設計の中心となるメディア素材に着目し、それを共有する部分構造を切り出す戦略が有効である。

4.2 詳細構造

4.2.1 提示単位ノードの定義

基本となる提示単位ノードは、完結したひとつのメッセージを学習者に提示すべきである、というのが英語教師と筆者らの共通認識である。マルチウインドウなどの断片的な情報提示は避け、メッセージを構成するメディアの組み合わせを以下のような形式で統一した。1)英語/指示/解説など、主たるメッセージを音声で提示する、2)それを補助する1~2個のウインドウ(テキスト/静止画)、3)3個のメニューボタンを同期して提示する、3)画面レイアウトを統一し、提示情報量を学習者の処理可能な範囲に絞る。以上より、提示単位ノードを「1つの音声+同期提示する4~5個のウインドウ/ボタン」で構成した。以下、このノードをビデオブックの呼称にならない「シーン」と呼ぶ。

シーン間の状態遷移リンクとして、以下の3種類を定義した。

1) ユーザリンク:メニューボタンへのユーザ入力により起動する通常リンク。

2) タイマーリンク:時間によって自動的に駆動されるリンク。イントロダクションの後、自動的に設問を提示する、などに用いた。

3) 条件リンク:あらかじめ起動条件を文字列として設定しておき、条件文字列に一致するメッセージをうけた場合のみ駆動されるリンク。設問に対する解答をキー入力させ、それによって次の画面を適応的に提示するために用いた。

タイマーリンク、条件リンクはビデオブックでサポートされている[田中91]。図7に、「設問」モジュールのノード/リンク構造のパターン例を模式的に示す。各ノードの画面イメージがアイコン化して表現されている。

4.2.2 ノードの文脈独立性の保証

提示単位の定義において問題となったノードの文脈独立性について述べる。今、シーン(ノード)AとシーンBが同じ画像Pを同じスタイルで提示するとしよう。AからBにリンクがはられた時、Bで画像Pを重

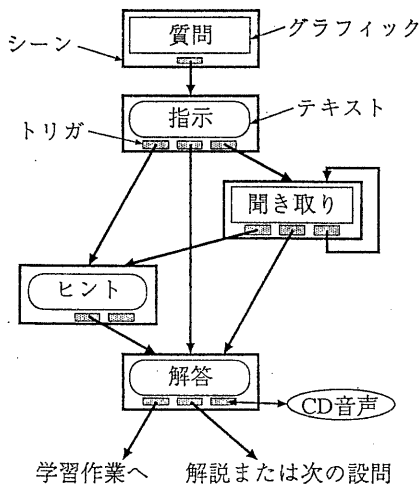


図7 ノード/リンク構造の例

ねがきする必要はない。そこで冗長性をなくすため、BをAのいわばサブクラスのように考え、差分情報のみ記述することが考えられる。しかし、筆者らの経験によれば、これはかえってオーサリングを困難にする。

例えば、Bが全体として何を表現するのか、はAを見なければわからない。画像Pを含まないシーンCからBに新たにリンクをはった場合、CからBを検索しても作者の意図したメッセージは表現されない。Bが「文脈依存」になってしまったからである。文脈依存ノードで構成されるハイパーメディア構造はリンクの変更が困難であり、検索パスによってノードの見え方が異なるという意味でふるまいがわかりにくい。また、設計者以外には理解不能な構造になりやすい。

個々のノードは文脈に依存せず、単独で作者が意図したメッセージを表現できるべきである。文脈独立なノードによるハイパーメディア構造は、冗長な記述の一貫性維持機構を必要とするが、構造とふるまいが簡明であり、変更が柔軟にできる。一貫性と簡明さ/柔軟性のトレードオフについては興味深い議論ができるが、オーサリングを共同で行なう場合、少なくとも開発の最終段階までは、筆者らは後者を重視する。すなわち、一人のエキスパートが設計できる複雑でスマートな構造より、みんなが設計できる簡易で柔軟な構造のほうがオーサリングをスピードアップする。本教材設計では、英語音声の繰り返し参照など特殊な場合を除き、文脈独立な形式でシーンを記述した。もはや構

造の変更が必要ないという段階にまで開発がすすめば、なんらかの変換メカニズムで文脈独立な構造の冗長性を取除くことも考えられよう。

4.2.3 ノード名による構造表現

シーン(ノード)やメディアデータを編集中に、それが部分階層構造のどこに対応しているか、を知る必要が頻繁に生じる。また、シーン/メディアを変更するとき、それがどのメディア/シーンに波及するか、はそれらが部分階層構造のどこに位置しているかを知れば、ある程度予測できる。(システムがその波及先を勝手にわかっているだけではだめで、設計者が、構造のどこに波及するかを自分で把握できることがオーサリングでは重要である。)

そこで今回の設計では、シーン名/メディア名に階層構造を反映させ、個々のオブジェクトが階層のどこに対応するかを明示した。例を図8に示す。この記法は慣れると大変便利であり、名前→階層構造、階層構造→名前の変換が頭の中でできる。実際のインプリメントでは、ブラウザやデータベース検索の助けを借りずにノードをダイレクトに指定することもできた。またノード名の規則性により、例えばパッセージ1のあるノード群を名前により指定して、パッセージ2のノード群としてリネーム/再利用する、などの演算操作の指定が簡単になった。(ビデオブックでは、ハイパーメディア構造をdBase互換形式の独自データベースで管理しており、データベース直接編集/インデックス作成/エクストラクト(部分抽出)/マージ等のユーティリティツールを用意してメンテナンスを行なっている。)

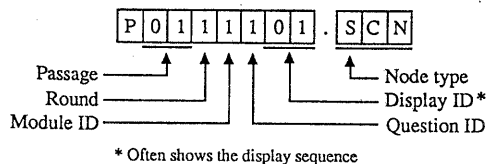


図8 名前による構造表現

4.3 実体仕様

英語音声／ナレーションのスク립ト作成、および写真／イラストレーションの仕様記述が設計の中心である。データ構造／スキーマ設計者の視点からは、実体の仕様記述は副次的に見えるが、一般の作者（ここでは英語教師）は、この仕様を原点としてストーリーを練るのであり、全体構造／詳細構造の設計と密接に関連している。実際に、英語教師は音声スク립トをベースにコースウェア仕様書を作成した。

そこで設計のポイントとなるのが、音声・画像等の大容量メディアの削減である。ひとつのまとまった教材をひとつの媒体（この場合CD-ROM）に納めることを制約条件として、大容量メディアの総量を決めなければならないが、オリジナルストーリー作成時点ではこの見積もりが非常に難しい。結果として、ストーリーの要求するメディア量は制約をオーバーしがちであり、共有化／削減が必要になる。本教材では、音声スク립トが2,300個に達したため、その共有化を図った。コースウェア仕様書を調べた結果、900個を共通ナレーションで代替できることがわかり、音声データ作成コストを約40%削減できた。

データ管理面からの他の重要なポイントは、共通ナレーションがどのノードで使われるか、を明確に対応づけることである。

4.4 スタイル

各ノード（シーン）のスタイル編集は、メディア実体の編集とともに、ビデオブック上で行なった。ここでの設計ポイントを2点あげる。

1) スタイルの大まかな設計は初期段階に行なう：概略でよいから、どんなメディアをどう提示するかを視覚化することにより、提示単位ノードの定義が容易になる。また、図7のように、ノード間の状態遷移を視覚化することが、詳細構造設計を助ける。実際に英語教師は、コースウェア仕様書と別個に画面イメージの状態遷移図を描き、学習者が実際どのように学習をすすめるのか確認に努めた。

2) スタイルの詳細設計はメディア実体作成後に決定する：メディア作成以前にウインドウサイズやタイミングなどの値を詳細に定めても、実際のデータに対して種々の可能性を試すために、パラメタはさかんに修正される。これが今回の設計の実際である。従って、詳細なスタイル設計は実際のデータ編集に即して行なうのが望ましい。いうまでもなくこれは、ノードの提示実行／スタイル編集／メディア編集が同時にできる

オーサリング環境で行なう、ということが前提である。ビデオブックはこの統合編集機能を備えていたが、スタイルの一貫性維持メカニズム（4.2.2章）が十分でなく、今後課題を残した（統合編集環境とは別にユーティリティツールで行なう必要があった）。

5. 評価と課題

英語教材オーサリング全般の評価と課題について、要点を整理する。

5.1 英語教材自体の評価

本教材の第一バージョンは、本年2月以来当社社員研修に利用されており、TOEICテスト／独自テスト等により、効果測定が続いている[小川92-2]。これまで約50人の研修生に利用してもらったが、「TOEIC200-400の範囲の学習者の得点を15時間の学習で100点向上させる」という所期の目的はほぼ達成でき、教材の教育的な品質は十分と思われる。研修生23人に対して行なったアンケート結果では、彼らの教材に対する満足度はかなり高い（5段階評価で4.4）。使い勝手の不満はほとんどなく、簡明なデータ構造／インタフェース設計が有効であったと考えられる。ただし、画質／ナレーションの改善など、メディアの品質向上についての要望が多い。

5.2 オーサリングの評価

オブジェクト数10,000、CD-ROM5枚に及ぶ実用レベルの教材を、MS-DOS環境の比較的簡易なハイパーメディアシステムで実装することができた。効果的だった戦略を整理する。

1) データ指向分割により、オーサリング範囲をビデオブックで編集可能なサイズにした（編集オブジェクト数600程度）。共有データを個々のディレクトリに局所化できたため、更新／メンテナンスが容易であった。

2) 文脈独立性を備えた提示単位を定義し、ノード／リンク構造を平易にした。このため、複数の編集者が編集結果を交換し合うことが容易であった。

3) モジュールの詳細構造（ノード／リンク構造）をある程度パターン化し、類似パターンの再利用を図った。

4) 階層構造をノード／メディア名に反映させ、各オブジェクトと構造を容易に関係づけられるようにした。これにより、データベースのメンテナンスも簡易化で

きた。

5) 共通ナレーションを導入し、音声データ量を40%削減した。

6) 導入した4レベルの設計段階は英語教師にとって自然であり、違和感がなかった。また、提示単位ノードの定義やインタフェース設計思想について、相互理解が十分できていた。

7) ビデオブックがシーン(スタイル記述)とメディア実体を別個に管理するアーキテクチャであるため、データ共有化等がスムーズに実装できた。

5.3 課題

データ管理の面でのシナリオ化の課題は、上記の評価点以上に多い。ここでは、以下の点のみをあげておく。

1) シナリオ再利用: ストーリー構造、ノード/リンク構造、提示スタイルには、あるカテゴリのアプリケーションに共通して使えるパターンが少なくないであろう。これらをシナリオテンプレートとしてデータベース化し、再利用することが設計効率化のために必要である。現状では、先に述べた設計レベルのうち、スタイル編集の他はワープロ以上に電子化できておらず、支援環境の整備が急務である。この支援環境では、本稿で示したようなデータ管理戦略が有効に用いられねばならない。

2) シナリオとアプリケーションの統合管理: 見過ごされがちであるが、オーサリングの過程で、シナリオ(仕様書)から実際のデータを参照する、あるいは実データ編集中に関連する仕様書を参照する、ということが頻繁におこる。文書によるシナリオでは、この相互参照の手間がかかり、またいずれかの更新による一貫性維持も困難である。シナリオ/アプリケーションの統合管理/編集環境も重要な課題である。

6. むすび

ストーリー主体のマルチメディアアプリケーションのシナリオ化について、データ管理の視点から検討し、設計戦略を具体的に示した。記述の抽象度の異なる4レベルのシナリオ設計モデルを定義し、CD-ROM版英語ヒアリング教材設計に実際に適用した。1) ストーリーのデータ指向分割、2) モジュールの導入、3) 文脈独立性を備えた提示単位定義、4) 提示単位とデータ実体の分離によるデータ共有化、等の戦略を採用した結果、大規模(オブジェクト数10,000)の教材をPC上の簡易

なハイパーメディアシステムで実装できた。現状では、シナリオ設計をほとんど机上で行なわねばならないため、シナリオ再利用やアプリケーションとの一貫性維持が難しい。今回のオーサリング経験をもとに、シナリオ化の手法を詳細化するとともに、シナリオ/アプリケーションの統合作成支援環境を構築したい。

謝辞

英語教材の開発にあたり、多大なご協力をいただいた千葉大学教育学部竹蓋幸生教授、大西知佳子氏、千葉大学教養部高橋秀夫講師、当社吉田正勝氏に深謝いたします。

文献

[小川91] 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、信学技報、DE91-3, pp.17-24, May 1991.

[小川92-1] 小川、吉田、金子他 「ビデオブックシステムによるアプリケーション開発(1) 新しい英語ヒアリング学習システムの構想」、第44回情処全大、20-5, March 1992.

[小川92-2] 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、S2-4, pp.81-86, June 1992.

[竹蓋91] 竹蓋他 「言語行動の研究」、第2号、千葉大学 英語学・言語行動研究会、May 1991.

[田中91] 田中、原田、小川 「マルチメディアオーサリングシステム『ビデオブック』における対話制御方式」、第43回情処全大、4F-3, Oct. 1991.

[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 -」、情処学会技術研究報告、91-DBS-84, pp.135-140, July 1991.

[Garzot91] F. Garzot, P. Paolini, and D. Schwabe, "HDM - A Model for the Design of Hypertext Applications," Proc. of Hypertext'91, pp.313-328, Nov. 1991.

[HyTime91] ISO/IEC DIS 10744.1.1, "Information Technology - Hypermedia/Time-based Structuring Language (HyTime)," April 1991.

[Jordan89] S. Jordan, D. Russel, A. Jensen, and R. Rogers, "Facilitating the Development of

Representations in Hypertext with IDE," Proc.
of Hypertext'89, pp.93-104, Nov. 1989.

[ODA91] ISO/IEC JTC1/SC18/WG3 N2084, "HyperODA
2nd WD," 1991.

[Ogawa90] R. Ogawa, H. Harada, and A. Kaneko,
"Scenario-based Hypermedia: A Model and a
System," Proc. of ECHT'90, pp.38-52, Nov.1990.