

## イベント駆動に基づく情報提示支援系 について

大戸英隆 辻野嘉宏 都倉信樹  
大阪大学基礎工学部情報工学科

マルチメディアデータをユーザからの要求、または必要に応じてリアルタイムに提示するような機能を持つソフトウェアをMMPS (Multi-Media Presentation System) と呼ぶ。近年、MMPSによってプレゼンテーションを行う要求が増しているが、プログラミングの知識のない者には困難である。

本報告では、ノン・プログラマによるMMPSの開発を容易にするため、イベント駆動の概念を基本とするMMPSのモデルであるシナリオ・モデルを提案し、それを用いて、MMPSを作成、編集し、実行することができる環境であるMMPS作成支援系の試作について述べる。

### A Multi-Media Presentation Systems Development Environment based on Scenario-Model

Hidetaka OHTO, Yoshihiro TSUJINO and Nobuki TOKURA

Department of Information and computer sciences, Faculty of Engineering science, Osaka University 1-1, Machikaneyama, Toyonaka, Osaka, 560 Japan

Recent advances in hardware technology have made handling by computers multi-media data such as a text, graphics, audio and video, economically feasible. We call a real time presentation system with multi-media data on such a computer an Multi-Media Presentation System (MMPS).

In order to implement an MMPS easily, we propose a Scenario-Model, which allows a user to represent his image of an MMPS directly.

We also propose an MMPS development environment based on the Scenario-Model.

In this environment, even a non-programmer can easily construct, edit and execute an interactive MMPS such as CAI systems, guide systems, and so on.

## 1. まえがき

最近の計算機のハードウェア技術の進歩は著しく、以前は困難であった画像、音声などのマルチメディア・データをパーソナルコンピュータで扱えるようになってきた。

このようなハードウェア技術の進歩によって、さまざまな分野で計算機によるマルチメディア・データの有効利用が考えられている。

既に、実用化が進んでいる分野としてあげられるのは観光案内などの情報サービスの分野である。また、研究発表などを行うといったプレゼンテーションの分野もマルチメディア・データの有効な利用分野としてあげられる。その他にも、CAIなどの分野での利用が考えられている。

これらの分野で利用されるソフトウェアに共通して要求される機能は「マルチメディア・データをユーザからの要求、または必要に応じてリアルタイムに提示すること」である。このような機能を持つソフトウェアを、ここではマルチメディア・プレゼンテーション・システム (MMPS = Multi-Media Presentation System) と呼ぶ。

将来、パーソナルコンピュータ上でMMPSを実行することが容易になると、パーソナルコンピュータを使用している一般のビジネスマンが営業用の資料として用いたり、計算機以外の分野の研究者が研究発表用の資料として用いたりするなど、プログラミングの知識のないエンド・ユーザ自身が自分でMMPSを作成したいという要求がでてくるものと思われる。

現在、MMPSを作成するためには、プログラムを作らなくてはならない。ところが、現在のプログラミング言語による記述を前提とする開発環境では、エンド・ユーザによるMMPSの開発は非常に困難である。それは、現在の開発環境が次のような問題点を持つためである。

(1) 一般のプログラミング言語には、MMPSの開発に必要な静止画データ、動画データなどのマルチメディア・データに対応するデータ型やこれらのデータに対する操作のプリミティブが用意されていないので、自分で作らなくてはならない。

(2) 一般にMMPSはユーザからの指示や時間の経過に基づいて動作するので、かなり複雑なタイミングの制御が必要になる。この実現には、高度なプログラミングの技術が必要とする。

(3) MMPSの開発では、画面レイアウトなどの実行時の様子を確かめながら作業を進める必要があるが、通常の開発環境ではプログラムの作成中に実行時の様子を確かめることができない。

(4) エンド・ユーザにとっては、プログラミング自体が困難である。

このような問題点の解決手段として、MMPS記述用の言語の開発<sup>(1)(2)</sup>や、プレゼンテーションの分野では、プレゼンテーション資料作成用のシステムの開発などに関する研究が行われている。

本報告では、一般のユーザによるMMPSの作成を容易にするため、プログラミングの知識がない者(ノン・プログラマ)でも理解できるようにMMPS用のモデルを提案している。このモデルを「シナリオ・モデル」と呼ぶ。シナリオ・モデルの概要は以下の通りである。

(1) 複数のマルチメディア・データを一括して扱うために、一括したデータの単位として「シーン」という概念を導入する。

(2) MMPS全体の動作を「シナリオ」というシーンの流れによって表す。

(3) シーンの提示や静止画の拡大、縮小など、MMPSの基本的な動作を定め、これらを「アクション」と呼ぶ。また、ユーザからの指示や時間の経過など、アクションのきっかけになるものを「イベント」と呼び、イベントとアクションの組合せによって「イベント駆動型」の動作記述を可能とした。

次に、シナリオ・モデルに基づいてMMPSを容易に作成、編集、実行できるように、MMPSの作成支援系を試作した。これを用いてMMPSを作成する際には、編集の対象となるMMPSのシナリオ・モデルが視覚的に表示され、画面のレイアウトも容易に確かめることができる。

以下、2節ではMMPSの利用例を挙げてMMPSの有用性について述べ、MMPSの実現に必要なとされる機能について考察する。次に、3節ではユーザから見た演示の流れのモデルとしてのシナリオ・モデルについて述べる。4節ではシナリオ・モデルに

基づくMMPS作成支援のために必要と考えられる機能、枠組みについて考察し、それらを取り入れて、シナリオ・モデルに基づくMMPSの作成、編集の可能なエディタとそれを実際に演示する実行系を統合したMMPS作成支援系の試作について述べる。

## 2. MMPSの概要

### 2.1 MMPSの用途

最初に、MMPSのイメージを明らかにし、その機能を検討するために、「プレゼンテーション」、「情報サービス」、「教育」の各分野でのMMPS応用の例をあげる。

#### (1) プレゼンテーション分野

(例) 研究発表用プレゼンテーション・システム

このシステムは、研究者が研究発表を行う際に図や表を示すために、従来、用いていたOHPに代わるものである。ディスプレイにはOHPに書かれていた図や表が表示され、発表者がすすむにつれて発表者の操作により、OHPを取り替えるのと同じように表示内容が変わっていくようになっている。

さらにコンピュータを用いることにより、動画の表示や音声の出力を組み合わせ、プレゼンテーションの効果をより高めることができる。画面の構成、および動作の流れを図1に示す。

研究発表にこのようなシステムを使用する場合の効果を以下にあげる。

① ワードプロセッサで文書を作成する場合と同じように、コンピュータを用いて図や表を作成することができ、変更、保存が容易になる。

② 提示内容の変更を、コンピュータのキーを押すなどの簡単な操作によって行える。

(画面構成)

画面1～画面N



説明内容に応じて、文字、静止画、動画を組み合わせて表示する。また、必要ならば、音声も出力する。

(動作の流れ) 発表者の指示にしたがって次々と画面が変わる。

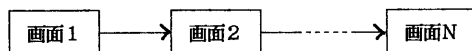


図1. 研究発表用プレゼンテーション・システム

#### (2) 情報サービス分野

(例) 観光案内システム

このシステムは、京都、奈良など、ある地域の観光のガイドを行うシステムである。最初、ディスプレイにはその地域の地図が表示され、名所、旧跡などの場所を地図中で確かめることができる。また、この地図の中から希望の場所を選んでその場所に関するより詳しい情報を得ることもできる。画面の構成と、動作の流れを図2に示す。

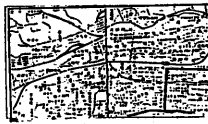
観光案内などにMMPSを用いる場合の効果を以下に示す。

① ガイドブックのように文章や写真を用いるだけでなく、動画、音声も用いることにより案内の対象となっている場所の雰囲気も伝えることができる。

② 場所を指定するだけで、その場所についてユーザが知りたい情報を即座に提示することができる。また、場所を指定する手段として、地名を入力するという手段以外に、マウスなどのポインティング・デバイスを用いてディスプレイに表示された地図上の位置によって指定するという手段を用いることが可能である。

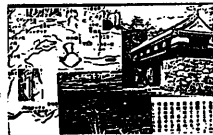
(画面構成)

画面1 (観光地の地図)



観光地の地図を表示する。

画面2～画面N (各地の案内)



地図中の特定の場所に関する案内の文章や写真を表示、さらに場所によっては、動画や音声を用いてその場所の雰囲気伝える。

(動作の流れ) 画面1で任意の場所を選択すると、画面2～画面Nに切り替わる。画面2～画面Nの表示をやめると画面1に戻る。

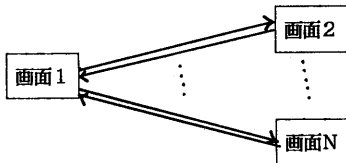


図2. 観光案内システム

(3) 教育分野

(例) 英会話教育システム

このシステムは、様々な場面での会話例を動画、音声を用いて示すものである。また、次の場面をいくつかある選択枝の中から選択することにより、学習者は好きな場面を選んで英会話を学習することができる。画面の構成と動作の流れを図3に示す。

英会話などの教育にMMP Sを用いる場合の効果を以下に示す。

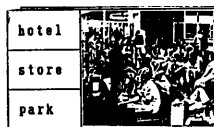
- ① 会話の例を示す際に、音声だけでなく動画を用いることによって臨場感を増すことができる。
- ② テープを用いる場合は、教材があらかじめ順番にテープに録音されていて、その順序を変えることができない。したがって、図3のように学習者が自分の好きな順序で学習するのは困難である。しかし、MMP Sではこのような学習方法が容易に実現できる。
- ③ 学習者が間違ったり、わからなかったりした箇所があったらそこまで戻って繰り返し学習することができる。

以上、MMP Sの具体的な例を3つあげ、各分野でMMP Sを用いた場合の特徴について述べた。

次に、これらの例をもとにして、MMP Sに必要とされると考えられる機能について考察する。

(画面構成)

画面1～画面N



さまざまな場面における会話の様子を動画、音声を用いて示す。また、次に選択可能な場面が画面の左側に表示され、その中から任意の場面を選択することができる。

(動作の流れ)

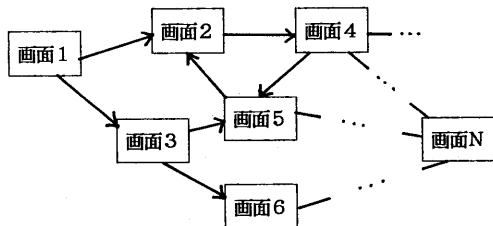


図3. 英会話教育システム

2.2 MMP Sに必要とされる機能

2.1で述べたような様々のMMP Sを作成するために、計算機環境上に必要とされる機能として、以下のようなものが考えられる。

(1) ユーザ入力受付

MMP Sではユーザからの指示によって、提示内容を切り替えるなどの動作を行うことが多い。ユーザがコンピュータに対する動作を指示する入力形式としてキーボードによるキー入力、文字列入力、数値入力とポインティングデバイスによる位置入力が必要とされる。

(2) 時間の経過に基づいた動作

プレゼンテーションの分野では、「今から10秒後に画面を切り替える」、「5秒ごとに音声を出力する」などのリアルタイムな動作が必要になることが多い。したがって、時間の経過に基づいて動作する機能が必要である。

(3) 個々のデータに対する操作

MMP Sが提示内容の切り替えだけで動作するものと考えると、かなり限られた形式のプレゼンテーションしか行えないことになる。そこで、例えば、静止画がディスプレイより大きくて一度に全部表示できない場合には、その静止画に対してスクロールを実行するなど、個々のデータに対する操作の機能が必要になってくる。また、動画や音声に対してもただ提示するだけでなく、必要に応じて早送り、巻戻し、コマ送りなどの操作が可能になると、聞き取りにくい部分を繰り返し何度も聞いたり、 unnecessaryな部分を飛ばしたりすることができるようになる。

(4) 動画や音声の提示状態の変化に基づいた動作

一般に動画や音声は、映画のようにお互いに組み合わせて提示されることが多い。このような場合、(3)で述べた早送り、巻戻しなどの操作もお互いに同期をとって行う必要がある。例えば、動画の再生をやめて早送りを始めたら、それをきっかけにして音声も早送りを始めなくてはならない。

このように動画や音声の提示状態の変化に基づいて動作するという機能もMMP Sに必要である。

3. シナリオ・モデル

MMP Sの作成が困難である理由として、2.2に挙げたMMP Sに必要とされる機能が、計算機環境上に十分備わっていないという場合、現在まだまだ多いということが挙げられるが、しかし、もし備わっていたとしても、それを用いて実際にMMP Sを作成するには、一般にプログラミング作業が必要であり、手軽な作成は困難である。

プログラミングが困難である理由としては、

- ① プログラミング言語、ライブラリ関数、ハードウェアなどの知識が必要となる、
- ② ノン・プログラマには、プログラミング言語に特有の、変数とループの組み合わせを用いた制御フローによって自分の思い描くMMP Sを表現することが難しい、

などがある。要するにプログラミング言語を用いる場合には、自分の作成したいMMP Sに対するイメージを一旦プログラミング言語の世界の表現に翻訳してやる作業(プログラミング)を経なければ実際にMMP Sを計算機上で動かすことができないことがMMP S作成を困難にしている。

よって、MMP S作成の際に、作成者がMMP Sの動作に対して持つイメージに沿ってMMP Sの作成作業をおこなうことができればその問題は解決される。

そこで、作成者のイメージに基づいたMMP Sの作成、実行を支援するために、まず、本節ではMMP S作成者がMMP Sの動作の流れに対して持つと考えられるイメージのモデル化をおこなった。このモデルを「シナリオ・モデル」と呼ぶ。

3.1 シナリオ・モデルにおける制御構造

MMP Sの動作の流れ(制御構造)を、通常のプログラミング言語を用いて記述するためには繰り返し、分岐などの制御構造を理解し、適切に使用する能力が要求される。

通常のプログラミング言語を用いてMMP Sを作成する場合、

ユーザからの指示や時間の経過などをきっかけとする動作は、次のような逐次的な動作の繰り返し、分岐によって記述されることが多いのではないかと考えられる。

- ① ユーザからの入力受付、時間の経過の測定を行う。
- ② 入力内容、時間を判定し、それに基づいて行うべき動作を決定する。
- ③ ②で決定された動作を行う。

また、この場合、変数を介して制御フローは動的に変化しているものと考えられる。なぜならばMMP Sの場合に限らず、通常のプログラム言語を用いてプログラムを行う場合、入力とその時点での内部状態(変数)との組合せによって次の動作を決定したり、呼び出す手続きに引き数を渡すことにより手続きの動作を変えたりするような方法は有用であり、よく用いられるからである。このように、変数を用いてプログラムの制御フローを動的に変えなければプログラミングの自由度は著しく下がってしまう。

確かにMMP Sにおいても、変数を用いてプログラミングの自由度をあげるということは、プログラミングの熟練者にとっては有難いことである。しかし、このことによりノン・プログラマは、制御フローを正確に把握することが困難となり、混乱を生じてしまう。

そこで、提示内容の切り替えの対応関係の固定化(内部変数によって対応関係を動的に変化させることをなくす)を行なって制御フローを単純化し、MMP Sの動作の流れをノン・プログラマにも理解できるように表すためには、プログラミング言語の制御構造とは異なった概念が必要である。

本来、ユーザから見たMMP Sは、画面が次々と入れ替わっていくように見えるシステムである。MMP Sの個々の画面を1つのシーンとみなすと、MMP Sの動作はシーンが次々と変わっていくことによって行われるものとみなせる。提示内容の切り替えはユーザからの入力、時間の経過などをきっかけにして行われる。ユーザから見た場合、MMP Sが何の理由もなしに突然に動作するということはほとんどない。全ての動作は、ユーザも認識している何らかの事象(イベント)をきっかけにして行われるものと考えてよい。キーボード入力などによって、ユーザ自身がイベントを引き起こすこともある。

ユーザにとってMMP Sは、「自分が何かを指示した」、「時間が過ぎた」、「動画を終わりに表示した」など、何らかのイベントをきっかけにして動作するシステムである。このような動作形態をイベント駆動型の動作形態と呼ぶ。

このように、ユーザにとってMMP Sはイベント駆動型の動作形態をとるものであるから、MMP Sの作成者(以下シナリオ作成者と呼ぶ)による動作の記述も以下のようなイベント駆動型の記述形式で行うのが自然である。

(イベント駆動型の記述形式)

シーン1では、

- イベントAが発生すると、動作aを行う。
- イベントBが発生すると、動作bを行う。

シーン2では、

- イベントCが発生すると、動作cを行う。
- イベントDが発生すると、動作dを行う。

つまり、このような制御構造を用いたMMP Sの作成が可能になれば、シナリオ作成者はユーザの立場からみたMMP Sの動作をイメージしながら動作記述を行うことができる。また、このような記述により、同じシーンで同じイベントが発生した場合、同じ動作を行うことが保証されるので、実行時の制御フローを頭の中でトレースすることが非常に容易となり、コーディング、デバッグ時間の減少が期待できる。

このような、提示内容の切り替え関係を固定化した、イベント駆動型のMMP Sの動作記述をシナリオと呼ぶ。シナリオの概念を導入することによって、様々なMMP Sの動作の流れを単純な、理解しやすいモデルとして表現することができる。

### 3.2 シナリオ・モデルにおける階層

一般に、MMP Sでは一度に1つのデータだけを提示すること

より、複数の様々なタイプのデータを組み合わせ一度に提示することの方が多い。このような提示形式によって効果的なプレゼンテーションを可能にしている。さらに、提示内容を切り替える際も、提示されているデータを1つずつ消してから、次に提示されるデータを1つずつ出していくような場合は少ない。ほとんどの場合、組み合わせ提示されているデータを一度に消して、次に提示されるデータの組が一挙にまとめて提示される。

したがって、ユーザにとってのMMP Sの動作は、マルチメディア・データが組み合わせて提示されている個々の画面が、次々と入れ替わっていくことによって進行しているように見えるものと考えられる。つまりMMP Sは複数のマルチメディア・データの組み合わせを1つの単位として提示したり消去したりする。この個々の場面に相当する単位をシーンと呼ぶ。

しかし、シーンを、次々切り替わっていく場面のように、同時に画面上、または、音声出力装置を占有するまとまりを指す概念とすると、例えば画面の一部に常にメニューが表示されていて、ユーザがそのメニューを選択して、いつでも自分の希望する情報を見ることができるとようなシステムを記述する場合には不都合が生じる。よって生成した側のシーンがなお存在する形式で記述できるようなモデルの方が都合がよい。すなわち、シーンとは、単に同時に生成消滅するマルチメディア・データのまとまりであって、同一画面上に複数存在してもよいものとする。

また、シーンという概念だけでは個々のデータに対する操作(例えば、ピクチャー・データのスクロール)を記述する際に、個々のデータを特定することができない。そこで、個々のデータに対応する概念を定め、これをサブシーンと呼ぶ。シナリオ・モデルにおいてはシナリオはいくつかのシーンからなりシナリオはいくつかのサブシーンからなるという階層構造を含んで記述されるものとする(図4)。

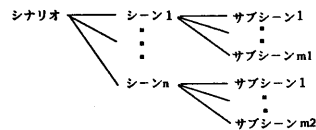


図4. シナリオ・モデルにおける階層構造

サブシーンには以下の3つの種類があると考えられる。

- (1) マルチメディアデータに対するサブシーン

サブシーンの主な機能は、マルチメディア・データの一つだけ提示することで、シナリオ・モデルで提示できると考えられるテキスト、画像、音声などデータのタイプに対応したサブシーンが用意されていると考える。

- (2) ユーザ入力に対するサブシーン

MMP Sに必要な機能としてユーザからの文字列、数値などの入力をうけつけるという機能があつた。したがって、シーンの中に文字列入力、数値入力を受け付けるための場所が必要である。このような場所を(1)のデータ提示のためのサブシーンの中に取るのは、各サブシーンの本来の機能であるデータの提示の機能を妨害することになり望ましくない。そこで、ユーザからの文字列入力、数値入力を受け付けるため、文字列入力、数値入力のためのサブシーンを用意する。

- (3) 時間の経過に対するサブシーン

時間の経過を測る機能を記述するために、時計に相当するサブシーンも導入すると都合がよいと考えられる。

シーンは、(1)から(3)のサブシーンを任意に組み合わせることによって定義される。

### 3.3 シナリオ・モデルにおける動作記述

MMP Sの動作を記述するためには、動作記述用のプリミティブが必要である。シナリオ・モデルでは、MMP Sの2.2の(2)から(4)で述べたような機能を実現するために必要なプリミティブをアクションと呼ぶ。また、他のシーンの生成、消滅もアクションの1つである。

3.1で述べたように、MMP Sはイベント駆動型の動作を行うものとしてモデル化される。基本的にシナリオ・モデルでは、

シーンごとにどんなイベントを受け付けるかを記述される。

MMP Sの実行中の様子を、シナリオ・モデルに基づいて次のように表すことができる。

- ① シーンが現れたり、消えたりしながらMMP Sの動作が進み、イベントを受け付けると、そのイベントに応じた動作を行う。
- ② シーンは、自分が現れている間、マルチメディア・データを提示している。
- ③ 同時に、モデル化の時に記述されたイベントの発生を待っている。

①のイベントに対する動作を記述するためのプリミティブがアクションである。1つのイベントに対して複数のアクションが実行されてもよい。

アクションには、システム全体に対するアクション、シーンに対するアクション、サブシーンに対するアクションの3つのレベルが存在する。

#### (1) システムに対するアクション

シナリオに対するアクションとしては、MMP Sの実行を終了するというアクションがある。

#### (2) シーンに対するアクション

①で述べたシーンが現れたり、消えたりするという動作も何等かのイベントをきっかけとして起こる動作であり、シーンに対するアクションであると考えられる。よって、シーンに対するアクションとしてはシーンの生成、シーンの消去という2つのアクションがある。

#### (3) サブシーンに対するアクション

テキスト、静止画のサブシーンに対するスクロール、拡大、縮小や、動画、音声のサブシーンに対する再生、早送り、巻戻し、タイマーのサブシーンに対するリセット、ストップ、スタートなどがある。

また、イベントもアクションと同様に、「シーンが生成された」などのシーンを認識の対象とするイベント(シーン・イベント)と、「静止画の上にある位置にカーソルがある時にマウスのボタンが押された」などのサブシーンを認識の対象とするイベント(サブシーン・イベント)の2つのレベルに分けられる。

シーンごとに、そのシーンに対するシーンイベントと、そのシーン中に含まれるサブシーンに対するサブシーン・イベントをすべて記述するだけでは、レベルの異なるイベントが混在して、構築したモデルの動作に関する記述の部分が読みにくいものになってしまう可能性が高い。

そこで、シーンに対しては、シーンが認識するイベントのみを定義し、サブシーンに対しては、サブシーンが認識するイベントのみを定義することにした。つまり、どんなイベントを定義するかは、シーン、サブシーンごとに分けて記述されることになる。したがって、サブシーンごとに独立にイベントの記述ができるので、モジュール性の高いイベントの記述ができる(図5)。

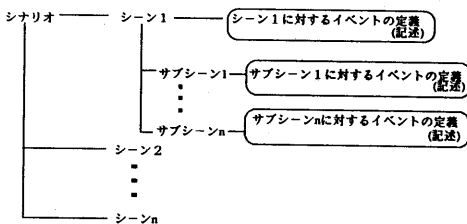


図5. シナリオ・モデルにおけるイベントの記述

また、あるサブシーンが認識できるイベントの種類もサブシーンのタイプによって限定されるので、全体としてのイベントの種類が多さに比して、MMP Sのモデルを構築する者が、個々のサブシーンにおけるイベントの記述の際に、複雑さを感じることはないものと思われる。

以下に、シーン・イベントとサブシーン・イベントの2つのレベルについて2.1で述べた機能を実現するのに必要と考えられるイベントを挙げる。

#### (1) シーン・イベント

複数のシーンが並行して提示されるようなシステムでは、シー

ンどうしの同期をとるために、他のシーンの生成、消滅をきっかけにしてアクションを実行する必要が生じることがある。よって、シーンの生成、シーンの消滅をイベントとして認識する必要がある。この、シーンに対して定義される2つのイベントをシーン・イベントと呼ぶ。

#### (2) サブシーン・イベント

3.2で述べたようにサブシーンには3つにタイプが存在しているので、それぞれについて必要と考えられるイベントを挙げてみる。

①マルチメディアデータに対するサブシーンに関するイベント  
動画サブシーンに対しては、その動画に同期して何らかのアクションを起こしたいという場合があると考えられる。このような場合には、そのサブシーンに対して現在のコマが表示されているのかをイベントとして検知したい。

このように、マルチメディアサブシーンについては、個々のサブシーンに対するアクションをイベントとする必要がある。

さらに、静止画サブシーンが例えば地図だったとき、その地図のある場所をポインティング・デバイスを用いて指定するとその場所に対する案内が表示される様にしたい場合が考えられる。

このように、②のユーザ入力に対して定義されたサブシーンでなくてもユーザ入力を受け付けられる必要がある。

マルチメディアデータに対するサブシーンについて考えられるイベントを表に挙げる。

#### ②ユーザ入力に対するサブシーンに関するイベント

ユーザ入力に対するサブシーンはユーザからの文字列、数値の入力を受け付け、入力された文字列、数値によって異なった動作(分岐)をするようなMMP S記述用のサブシーンである。よってこのようなサブシーンに対しては入力された文字列、数値が予め設定されていたものと一致した(または一致しなかった、大きかった、小さかったなど)ということを入イベントとする必要がある。

#### ③時間の経過に対するサブシーンに関するイベント

時間の経過に対するサブシーンは時間を測定して指定された時間になったら何等かの動作を起こすようなMMP Sを記述するためのサブシーンである。

よって、このサブシーンにおいては、測っている時間が予め指定されていた時間に到達したことをイベントとする必要がある。

## 4. MMP S作成支援系の試作

シナリオ・モデルを用いることによって、MMP Sをノン・プログラマでも容易に理解できるようにモデル化することができるようになる。しかし、モデル化ができて、そのモデルに基づいて実行可能なMMP Sを作成できるのであれば、シナリオ・モデルが真にノン・プログラマによるMMP S開発の助けになっているとは言えない。そこで、シナリオ・モデルに基づいてMMP Sの作成、編集、実行ができる環境としてMMP S作成支援系を試作した。

### 4.1 設計方針

MMP S作成支援系の設計方針は以下の通りである。

#### (1) シナリオ・モデル指向

MMP S作成支援系はシナリオ・モデルをイメージしながら作成が行える環境を目標としている。そのため、エディタ上ではシナリオ、シーン、サブシーンを単位としてそれらの間のイベント、アクションを定めていくことで編集作業を行えるようにする。

#### (2) ユーザインタフェースの考慮

本来、MMP S作成支援系はノン・プログラマの使用を前提としているので、良好なユーザインタフェースは不可欠である。

ユーザインタフェースを向上させるために以下のような機能を備える。

#### ①制御フローの可視化

各シーンをノードと考え、その各ノードがイベントとアクションの対応関係を表す有向辺でつながれているとすると、シナリオを有向グラフの形で図式化することができる(図6)。

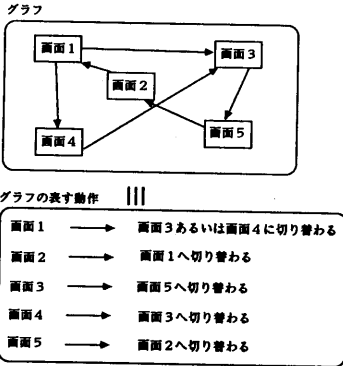


図6. MMPSのシナリオの図式化

このようにしてMMPSの制御フローが簡単に静的なグラフとして表現できるのは、制御フローが動的に変化しないイベント駆動型の制御構造をシナリオ・モデルの制御構造としたからである。このグラフをシナリオ作成者に常に提示することにより、シナリオ作成者は作成するシナリオのイメージを視覚的により明確にすることができる。

②ポインティングデバイスの使用

シナリオ作成作業はできる限りポインティングデバイス（マウスなど）を用いて行えるようにする。

(3) システムの2部構成

MMPS作成支援系はMMPSの作成、編集を行うエディタとMMPSの解釈実行を行う実行系の2つから構成する。このような構成をとったのはエディタと実行系を独立、並列に開発するためである。この理由からエディタと実行系とのデータ交換に明確な仕様を定める必要があったので、その仕様を中間言語を用いることにより定義し、エディタから実行系へは中間言語を用いてデータを送るという方針をとった。よって、実行系は中間言語の一種のインタプリタであると考えてよい。

(4) 限定的な試作

今回の試作環境においては音声、動画に関して、その入出力を行うハードウェアがなかったためその部分について実現は行わないこととした。

4.2 MMPS作成支援系の動作

上記の方針に従って設計したMMPS作成支援系は、以下に示すウインドウから構成されている。

(1) シナリオ・ウインドウ

シナリオ・ウインドウはフルスクリーンウインドウで、コマンド・バーとシナリオ・マップの2つの部分から構成される(図7)。

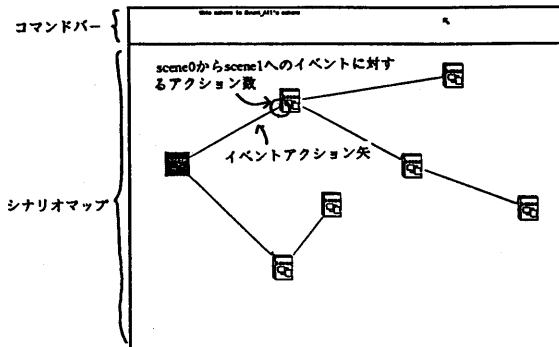


図7. シナリオ・ウインドウ

コマンド・バーは主に編集結果のセーブ、ロード、実行などの操作を行うためのものである(表1)。

表1. コマンドバーのメニュー

load	エディットファイルのロード
exec	実行
save	エディットファイルへのセーブ
delete	エディットファイルの消去
close	シナリオウインドウのクローズ
end	現在実行中のファイルをセーブして終了
quit	強制終了
scheme	表示形式の変更

メニューをポップアップし、セーブ時にはファイル名の入力を受け付ける。シナリオの表示形式の変更のためのメニューも含まれている。また、現在の表示形式などのメッセージの表示も行われる。

シナリオ・マップは、シナリオ・グラフを表示するウインドウである。シナリオ・グラフは各シーンを表すシーンアイコンと、あるシーンで起こったイベントが他のシーンにアクションを起こすという対応関係を表す有向辺(矢印)とで構成される。矢印の先にはイベントに対して実行されるアクションの個数が表示されている。コマンド・バーのメニューによってイベントのタイプ毎に有向辺を表示することができる。このようにシナリオマップを見ることにより、MMPS作成支援系のユーザはMMPS全体の動作の流れを容易に理解することができる。シナリオ・マップでは主にシーンの生成、削除、移動、コピー、初期表示シーンの設定などシーンに対する操作を行う。また、シーンに対するイベント、アクションの編集はこのウインドウ中で行う。

(2) シーン・ウインドウ

シナリオ中の個々のシーンを構成するサブシーンをMMPSの実行時と同じ状態で表示したものをサブシーン・ウインドウと呼ぶ。シーン・ウインドウはサブシーン・ウインドウの編集を行うためのウインドウで、シナリオ・ウインドウ同様フルスクリーンのウインドウであり、画面全体の背景とシーンアイコンからなる(図8)。

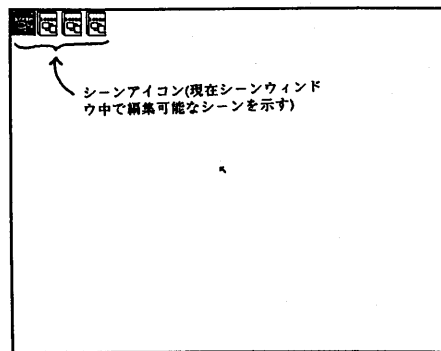


図8. シーン・ウインドウ

また、サブシーンに関する属性、イベント、アクションの編集を行うための属性ウインドウ、イベント・ウインドウ、アクション・ウインドウもシーン・ウインドウ上で開く。

シナリオマップ内でシーンに対応するアイコンを指定して編集コマンドを実行することによりシーン・ウインドウがオープンされる(図9)。

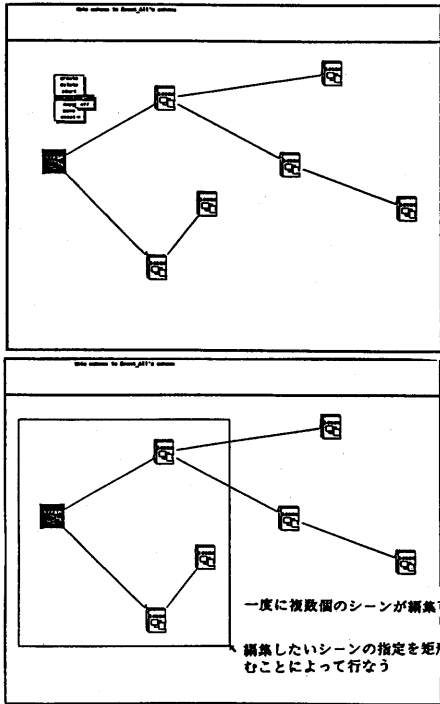


図9. 編集モードへの移行

シーン・ウィンドウ中には、現在エディット中のシーンに対応するシーンアイコンが表示されている。そのシーンアイコンをクリックして得られるメニューを選択することによりそのシーンに属するサブシーン・ウィンドウの生成、削除、表示などの編集作業が行える(図10)。

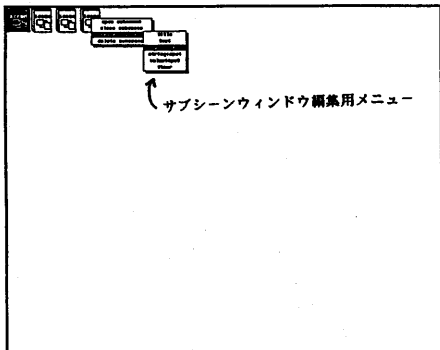
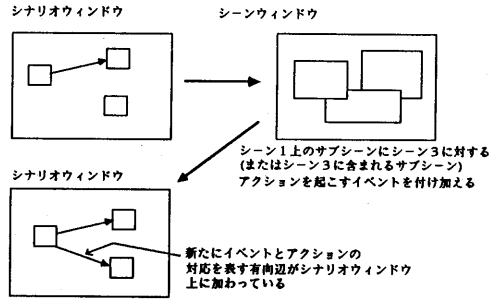


図10. サブシーン・ウィンドウの編集

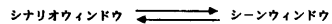
シーン・ウィンドウは背景の部分をクリックし、メニューを選択することによって、シナリオ・ウィンドウと行き来できる。シーン・ウィンドウでイベント、アクションの編集を行う際にはシーン・ウィンドウ上での編集結果をシナリオ・ウィンドウ上で確かめる必要があり、この行き来は頻繁に行われる(図11)。

### (3) 属性ウィンドウ

シーン中の個々のサブシーンの属性に関する情報、および属性の設定、変更用のコマンドを表示したものを属性ウィンドウと呼ぶ(図12)。



このような編集結果の戻りのため



は頻繁に行なわれる

図11. シナリオ・ウィンドウとシーン・ウィンドウ間の移動

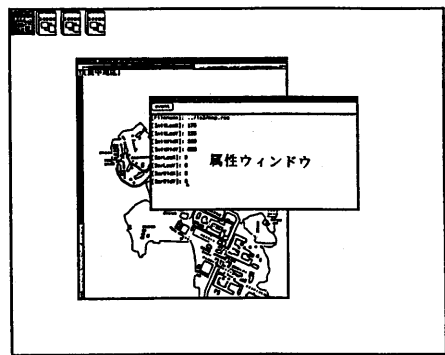


図12. 属性ウィンドウ

属性ウィンドウは各サブシーンに対応する属性を設定するためのウィンドウである。また、サブシーン・ウィンドウの位置、大きさなど、サブシーン・ウィンドウの生成時に決まり、入力のないものについては単に表示のみが行われる。

### (4) イベント・ウィンドウ

サブシーンごとに、もしそのサブシーンがイベントを持つタイプのサブシーンであれば、このウィンドウをオープンすることができる。ウィンドウ内には、そのサブシーンに設定されているイベントのリストが表示される。このリスト内の個々のイベントを表す部分には、そのイベントの引数、または引数を設定、変更するためのコマンドが表示される。イベントの設定、変更はこのウィンドウで行われる(図13)。

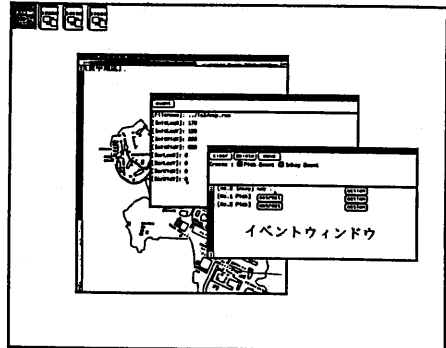


図13. イベント・ウィンドウ

静止画データに対するピックイベントの設定操作を例に示す(図14)。

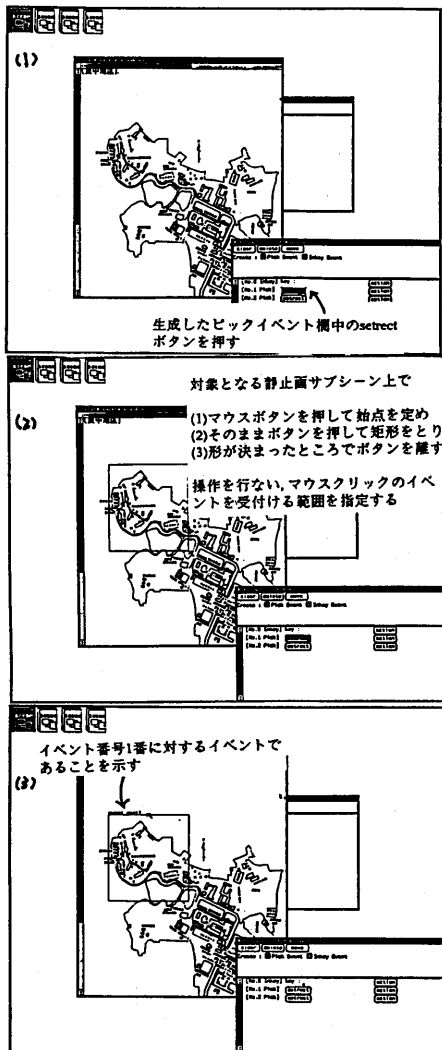


図14. 静止画サブシーンに対するピックイベントの設定操作

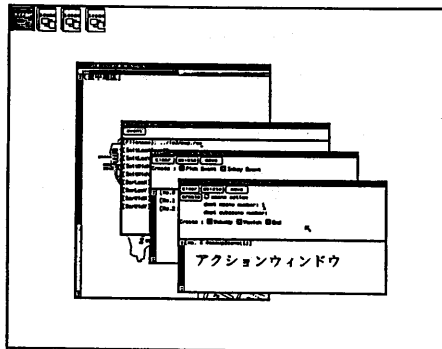


図15. アクション・ウィンドウ

また、イベントは、その並びによって優先順位を表すので並び替えを行うためのコマンドも用意されている。

#### (5) アクション・ウィンドウ

このウィンドウは、イベントが発生した時に実行するアクションを設定、変更するためである(図15)。

個々のイベントに対応するアクション・ウィンドウがあり、イベントごとにオープンすることが可能である。ウィンドウ内にはイベント・ウィンドウと同様に、実行されるアクションのリスト、および編集用のコマンドが表示されている。リスト中の個々のアクションに対応する部分には、各アクションの引数に関する情報が表示される。またアクションもイベント同様、その並びによって優先順位を表すので、並び替えを行うためのコマンドが用意されている。

## 5. あとがき

ノン・プログラマによるMMP Sの開発を容易にするため、イベント駆動の概念を基本とするMMP Sのモデルであるシナリオ・モデルを提案し、それを用いて、MMP Sを作成、編集し、実行することができる環境であるMMP S作成支援系の試作について述べた。

今回提案したモデルは、以下の特徴を持つ。

(1) MMP Sの動作をユーザから見たときのイメージに基づいたモデルである。このことにより、シナリオ作成者にとってシナリオ・モデルはMMP Sの実行時の様子をそのままモデル化したものとなり、シナリオ作成の際のイメージ作りが容易になる。

(2) シーンの切り替えという概念によって、MMP Sの動作の流れが、わかりやすく表現できる。

(3) イベント駆動型の動作の概念によって、MMP Sのユーザの指示、または必要に応じたリアルタイムな動作を表現することができる。

さらに、このモデルを用いてMMP Sを開発するためのMMP S作成支援系は、

(1) モデルを視覚的に表現している。

(2) MMP Sの編集と実行をインタラクティブに行うことができる。

(3) ほとんどの編集コマンドが、マウスを用いた簡単な操作で実行可能である。

などの特徴を持っており、モデルに基づいたMMP Sの作成、編集、実行が、MMP Sエディタの使い方を短期間学習するだけでできるようになるものと思われる。

また、このMMP S作成支援系を用いて、簡単なMMP Sの作成を行った。

これは、筆者の所属する研究室の紹介システムで、廊下の画面の正面をクリックすると前に進んでゆき、廊下の画面上の左右のドアの部分をクリックするとその部屋に入って中を覗けるというように、建物の中を歩いている感じで研究室の様子がわかるようになっている。部屋のなかの画面上でマウスクリックを行うとその部屋の人間の写真の画面がポップアップしてくる。本システムは、シーン数26、サブシーン数57、イベント数52、アクション数96で構成されているが、この程度の大きさであれば全体の構成をはじめにしっかり作っておいた場合、数時間で作成可能である。

今後の課題としては、

(1) プロトタイプの使用感の評価、検討を行った上で、さらに細かな使い勝手の向上を目指す、

(2) 音声、動画データなどを扱えるシステムにする、  
などがある。

#### 参考文献

[1] 小山博生: "マルチメディア演示のためのシナリオ記述言語に関する研究", 大阪大学大学院基礎工学研究科修士学位論文(1985-02)。

[2] 田中, 辻野, 都倉: "マルチメディア表示システム記述用言語の設計とその処理系の実現", ソフトウェア工学研究会資料86-47, (May 1986)。