1 はじめに

ここ数年の間にWWW (World Wide Web) はネットワークの相互接続などにより急速に浸透し、計算機の性能の向上ともあいまってインターネットの利用者は爆発的に増加した。それに比例するように、現在では把握しきれないほど多くの機関・企業・個人がWWW上に情報を発信している。たとえば数年前までは学会・研究機関等を通して配布されていたようなさまざまな学術情報等も、個人ベースで世界に発信することが容易にできるようになっている。

このように、WWW上で提供される情報は多様化・複雑化・大規模化の一途が進んでおり、ややもすれば情報過多という問題が生じつつある。

こうした背景の中、情報の受け手に対しては、「より効率的な情報へのアクセス」「情報の無駄の無い選択」「情報の効果的な活用」が求められるようになり、それに対応する技術研究も数多く行われている。

同様に、情報の送り手側にも、より効果的な情報発信を行うために、「効果的かつアトラクティブな情報」を「誰にでも、わかりやすく視聴できる形で提供する」ことが必要になっている。

本稿では、こうした効果的な情報発信 (プレゼンテーション) を可能にするための技術に関して報告を行う。

まず、エージェントキャラクタを利用したマルチモーダル・プレゼンテーションについて略説を行う。そして、それをより簡単に実現させる目的で開発された言語MPMLを紹介し、さらに実際にMPMLを利用するためのツール、及び今後の研究に関して考察および報告を行う。

2 プレゼンテーションエージェント

2.1 マルチメディア・プレゼンテーション

最近のWWWホームページには、テキストやグラフィクスだけでなく、アニメーションや音楽、さらには音声まで多様に組み合わせたマルチメディア・プレゼンテーションの形態をとっているものが多くみられる。情報提供者 (presenter) はコンテンツを作成し、インターネット上に公開しておけばそのプレゼンテーションは誰にでも視聴可能になる (図2)。

その多くは魅力的ではあるが、ほとんど一方的な情報伝達しか行われず、ユーザが行うことのできる操作は、マウス操作によってページをジャンプしたりウィンドウを閉じるといった「購読の停止」を目的としたものに限られている。

また、現実世界でのプレゼンテーション (図1) と異なり、質問をするなどして情報提供者へ視聴者 (audience) の意志を伝えることはできない。

我々はこうした一方通行のシステムを改良し、より効果的なプレゼンテーションを実現する為に、擬人化エージェントシステムを利用する方法に注目した。

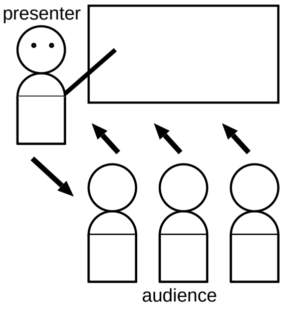

図1: 実世界のプレゼンテーション

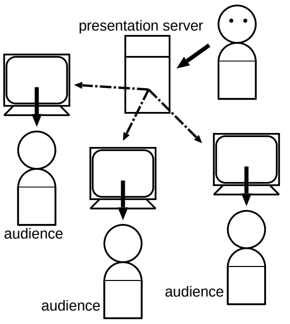

図2: WWW上のプレゼンテーション

2.2 擬人化エージェントシステム

マン–マシンのインタフェースにおいて、直接操作の基本となるのはグラフィクスであり、ユーザはマウスやキーボードを用いてインタラクションを行う。それに加えて、音声やテキストなどで言語情報を伝えるなど、複数のコミュニケーションチャネルを同時に相互補完的に利用し、言葉や行為のモダリティを働かせてインタラクションを行うシステムをマルチモーダル・インタフェースシステムという。

こうしたインタフェースシステムでは、CGキャラクタに表情や動作などの擬人的表現を生成させる、「擬人化エージェント技術」が用いられていることが多い。

たとえば、東大石塚研で開発されたマルチモーダル擬人化インタフェースエージェントVSA[3]や、通産省電子技術総合研究所で開発されたマルチモーダル対話システム[4]をはじめ、音声対話まではサポートしていないが、ペンシルバニア大のVirtual Human Presenter[5]、WWW連携したWebPersona[6]などがある。

2.3 エージェントによるプレゼンテーション

ウェブ上でプレゼンテーションを行うことの最もすぐれたメリットは、情報提供者の都合に関係なく、時、場所を選ばずにプレゼンテーションを視聴可能にするということであるが、一方で情報提供者とのインタラクションが行われないというデメリットもある。

そこで考えられたのが、擬人化エージェントにプレゼンテーションを実行させるという方法である。視聴者はエージェントとの対話を通してプレゼンテーションの制御に注文をつけたり、(遠慮することなく) 質問をはさんだりすることも可能になる (図3)。

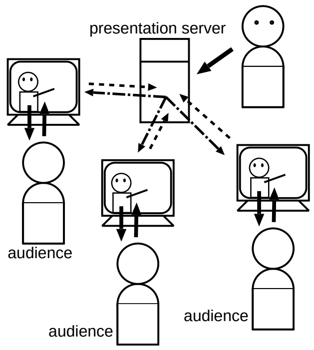

図 3: マルチモーダル ・エージェントによるプレゼンテーション

また、視聴者とエージェントのインタラクションの情報をプレゼンテーションサーバに返すようにしておけば、情報提供者は、視聴者の興味、リクエストを反映させてプレゼンテーションを改善していくことができる。

2.1章で延べたように、WWW/インターネット情報空間においてはすでにマルチメディアコンテンツを視聴するためのブラウザ、ツールが充実している。したがって、マルチモーダルエージェントを受け入れるための土壌はすでに揃っていると言え、またその利用価値も高いと考えられる。

擬人化エージェントをプレゼンテーションに用いることの利点はまだある。CG キャラクタを用いてのプレゼンテーションは、従来のマルチメディアプレゼンテーションに比べてもより魅力的であり、動き回るキャラクタを目で追わせるなどの、視聴者の注意を引き付ける効果も期待できる。

また、エージェントキャラクタの表情、動作等の付加的な情報によって、システムの状態を視聴者に理解させることができ、認知的負担を軽減させることができる。

問題は、こうしたエージェントを用いたプレゼンテーションを実行するには、特定のシステムが必要であったり、スクリプトが複雑であったりするために、誰にでも作成、視聴ができるわけではないということである。これを解決するために、システムに依存せず、簡潔な記述によって、プレゼンテーションをコーディングすることを可能にする言語を開発した。

3 マルチモーダル ・プレゼンテーション記述言語 MPML

3.1 言語概要

MPML (Multimodal Presentation Markup Language) は擬人化キャラクタを用いたインタラクティブなプレゼンテーションを容易に記述できるように設計、開発された言語である [1][2]。

情報提供者がプレゼンテーションを配信する際、通常は OS、ブラウザ、リソース等の視聴者の環境を考慮しなくてはならない。MPML は特定のブラウザやシステムに依存する言語ではなく、MPML コーディングされたスクリプトを様々なツールによって再生、視聴することを目的として設計されている。したがって、プレゼンテーションの作成者は視聴者の環境を考慮することなくコード作成する事ができる。

また MPML は XML (Extensible Markup Language) [7] の仕様に準拠している。MPML で用いられるタグの種類は高々19 種類であるため、HTML スクリプトを理解できる程度の知識がある人なら、誰でも簡単にエージェントを用いたプレゼンテーションを記述できる。また熟練者は様々な属性値を変化させることによって、より詳細にわたってプレゼンテーションを設定できるようにもなっている。

3.2 言語の特徴

マルチメディア ・プレゼンテーションにおいては、様々なメディアを効果的に再生することが必要である。これに関して、WWW 上で複数のメディアデータの表示を制御するための記述言語である、SMIL (Syncronized Multimedia Integration language) [8] の勧告が W3C によって 1998 年に公開された。MPML はこの仕様を参考に、最低限のメディア同期を図ることができる機能を有している。

MPML の最も特徴的な部分はエージェントの動作の記述である。ユーザは簡単なタグ付けで、エージェントに挨拶や指さしなどの様々な動作をはじめ、自由な移動、音声合成による発話などをさせることができる。各動作はメディア再生と同期をとることができる。

さらに、MPML ではハイパーリンク機能もサポートしており、音声認識システムと組み合わせることで、エージェントとの音声対話などによるインタラクティブな操作によって、視聴者を効果的にナビゲートすることが可能になる。

4 MPML の言語仕様

4.1 MPML ドキュメント

本章では MPML の言語仕様について簡単な解説を行う。各要素に関するより詳しい定義については付録を参考にしていただきたい。また、MPML1.0 の仕様書は MPML ホームページ :`http://www.miv.t.u-tokyo.ac.jp/mpml.html` において公開されている。

MPML の定義は XML で行われている。XML の詳しい言語構造に関しては文献 [7] 等を参考にしていただきた

く、本稿では最低限の解説を行う。

各要素（タグ）は要素属性と要素内容によって意味付けがなされる。要素属性は要素の識別子 (ID) や特性、性質などを指定し、要素内容は入れ子構造で、子要素、子孫要素を付け加えていくものである。また、子孫要素を持たない要素は空要素と呼ばれ`<element... />`のように記述する。たとえば書式は以下のようになる。

```
<element attribute=value>
   <child_element1>
      ...
   </child_element1>
   <child_element2 />
   ...
</element>
```

MPML の構造木は図 4のように表すことができる。図中の"?" はその要素が 0 個もしくは 1 個出現するという意味で、"*" はその要素が 0 個以上任意個出現するという意味である。また、`#PCDATA` はテキストデータを表している。

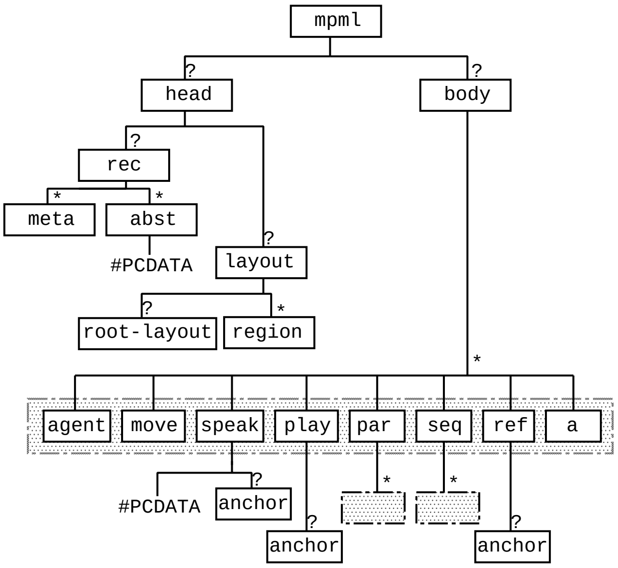

図 4: MPML の構造木

全ての要素の祖先要素になるのが要素 `<mpml>` である。要素 `<mpml>` はその属性として `id` を持つ。`id` は要素を識別するために用いられる識別値であり、ほとんどの要素に記載することができる。また、子要素としては `<head>` と `<body>` を持つことができる。以下では各要素に関して順次解説を行っていく。

4.2 ドキュメントのヘッダ

`<head>...</head>`で囲まれたドキュメントのヘッダには、プレゼンテーションに関するメタデータ及び、レイアウトに関する情報を記述することができる。メタデータは要素`<rec>`に、レイアウトは要素`<layout>`に記述される。

メタデータ記述

スクリプト作成者は、要素`<rec>`のコンテンツに、作成したプレゼンテーション全般に関するプロパティを記載しておくことができる。

要素 `<meta>` は空要素であるが、属性を用いて、そのタグ一つに一組ずつメタデータを記載しておくことができる。

要素`<abst>`はテキストデータを要素内容に持ち、プレゼンテーションの概要が記載できる。

レイアウト記述

要素`<layout>`のコンテンツにはプレゼンテーションのレイアウト情報が置かれる。要素内容は任意であり、任意の書式でレイアウト記述ができるが、デフォルトでは `MPML` 特有の書式になり、`<root-layout>`及び`<region>`を子要素に持つ。

要素`<root-layout>`では、プレゼンテーションを行うルートウィンドウに関する性質が設定される。

また要素 `<region>` の属性において「点」、もしくは長方形型の「範囲」としてリージョン情報を記述できる。一つのタグで一つのリージョンを設定できる。

4.3 ドキュメントのボディ

要素`<body>...</body>`で囲まれた要素内容には、プレゼンテーションの実行に関わるコンテンツが置かれる。要素`<body>`は暗黙に要素`<seq>`の定義を含むため、その子要素の実行はシーケンシャルに行われる。

エージェント定義

要素`<agent>`で、プレゼンテーションを行うエージェントキャラクタの設定を行うことができる。要素`<agent>`で定義したエージェントを要素`<move>`、要素`<speak>`、要素`<play>` などで参照する。

複数の`<agent>`を定義して、それぞれの `id` 値を逐次参照することで、複数のキャラクタを用いたマルチエージェントによるプレゼンテーションが可能になる。

エージェントの動作記述

要素`<move>`でエージェントキャラクタを移動させることができる。すでに設定したリージョンへの移動のほか、移動位置を座標指定しての移動も可能である。

要素<speak>のコンテンツにはテキストが置かれる。プレイヤは音声合成エンジンなどを用いて、その内容をエージェントキャラクタに発話させる。

また、要素<play>によって、エージェントキャラクタに様々な動作をさせることができる。MPMLでは挨拶などの基本動作のほか、指定したリージョンを参照しての指さしをさせたり、発話しながら、まばたきしながら等のながら動作も記述できるようになっている。

メディア同期

要素<par>の子要素としておかれたコンテンツは、そのドキュメント中の順序に関係なく、並列に実行される。

また、要素<seq>の子要素としておかれたコンテンツは、そのドキュメント中の順序にしたがって、シーケンシャルに実行される。

メディアオブジェクトの再生は要素<ref>を用いてロケーションを指定することによって行われる。

ハイパーリンク/プレゼンテーション制御

要素<a>を用いて、クリック、もしくは音声入力によるプレゼンテーション制御のためのキーを設定する。プレゼンテーションのジャンプ、中断、再開等の基本的な制御をサポートしている。

また、要素 <anchor> は以下の二通りに用いられる。

- メディアオブジェクトの実行やエージェント動作の間の時間的なポイントにリンクの終点となるアンカーを設定する
- メディアオブジェクトの空間的なポイントにエージェントの動作対象などになるアンカーを設定する。

アンカーとして指定された識別子を要素 <a> などで指定するジャンプ先のURIの後に付与することで、メディアオブジェクトやエージェントの発話などの途中へジャンプすることも可能になる。例えば下の例のように使用することができる。

```
<speak>
  This part will not be read if jumped to "anc1".
  <anchor id="anc1" />
  Read from here!
</speak>
```

代替コンテンツ

要素<switch>には、プレイヤがコンテンツの再生に適さない場合に備えて、代替コンテンツを複数置いておくことができる。ユーザは最も視聴してもらいたいコンテンツから順番に指定し、最も安全なコンテンツ（テキスト表示など）を最後に指定することが望ましい。

4.4 サンプルスクリプト

MPMLコーディングされたプレゼンテーションのサンプルスクリプトを示す。

```
<mpml>
 <head>
  <layout>
   <root-layout id="root" width="800" height="600" />
   <region id="spot1" location="500, 300" />
   <region id="spot2" location="100, 300" />
  </layout>
 </head>
 <body>
  <ref region="root"
   src="http://www.miv.t.u-tokyo.ac.jp/mpml.html" />
  <agent region="spot1" />
  <par>
   <play act="point" region="spot2" />
   <speak>
    This is MPML Home Page!
   </speak>
  </par>
 </body>
</mpml>
```

このスクリプトによって、以下のプレゼンテーションが行われる。

<ref>で指定されたURLのページを背景に、エージェントキャラクタがspot1として指定された場所に出現する。
エージェントはspot2として指定された位置を指さしながら、<speak>のコンテンツを発話する。

5 MPMLの利用

5.1 MPMLの周辺ツール

本研究の目的は、MPMLを利用することによって、誰でも簡単にマルチモーダル・エージェントを用いたプレゼンテーションを作成、視聴することを可能にすることである。図5に、MPMLを用いたプレゼンテーションを実現するために開発が計画されているツール群を示す。

まず、ユーザにとって、より簡単なMPMLスクリプト作成を可能にするための、MPMLオーサリングツールがあげられる。これは、HTMLのスクリプトビルダのように、タグ挿入、リージョン指定などが、マウス操作などで簡単に行うことができる編集ツールである。

作成されたMPMLスクリプトを実行するためのMPMLプレイヤも必要である。マルチエージェント、音声認識、音声合成、Web連携など必要とされる技術は多いが、最終的にプレイヤも含めたプレゼンテーションのWeb配信も視野に入れている。

独自のMPMLプレイヤが利用できない場合は、すでに流通しているエージェントシステムを使用してプレゼン

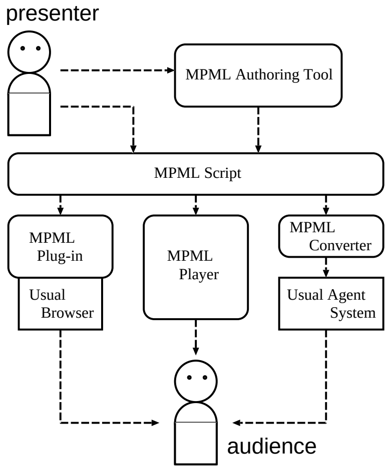

図 5: MPML の周辺ツール

テーションを行う事を考える。既存のブラウザに MPML スクリプトを認識させるプラグインや、MPML スクリプトを既存のエージェントシステムに対応したスクリプトへコンバートする MPML コンバータを介することによって、MPML スクリプトによるプレゼンテーションを実現することができる。

5.2 MPML コンバータ

本稿執筆時点では、MPML スクリプトを利用してプレゼンテーションを行うツールとして、MPML コンバータが二種類開発されている。

一つは Microsoft Agent システム [9] を利用して、Web と連携したプレゼンテーションスクリプトを作成するもので、MPML スクリプトを VBScript にコンバートして HTML ファイル中に埋め込むものである。システムの都合上、MPML の機能の全てをサポートしてはいないが、エージェント動作、音声合成、音声入力、リンク等の基本的な機能を使用することができる。図 6は、4.4章のサンプルスクリプトをこのコンバータを介して実行したものの実行例である。

もう一つは MPML スクリプトによって、CG ツール Ashow[4] を動作させるマスタ / スレーブシステムである。マスタにおいて MPML スクリプトを解析し、ソケット通信を行ってスレーブである Ashow に様々な動作をさせるものである。

これらのツールを用いる事で、複雑なスクリプトを書かねばならないエージェントシステム、メディアプレゼンテーションにおいても、MPML の簡略なタグ記述によって簡単にプレゼンテーションを実現することが可能になった。

MPML HomePage

This is MPML Home Page!

MPML(ver. 1.0)の仕様は こちらです。(last update Jan./5/1999)
前バージョンからの変更点

MPMLサンプルスクリプトは こちらです。
ソースはここです。

石塚研究室トップへ戻る
Last update is at 01/13/99 11:27:26

筒井貴之 (tsutsui@miv.t.u-tokyo.ac.jp) / 石塚研究室

図 6: MPML コンバータを用いて作成したプレゼンテーション例

6 考察

マルチモーダル ·プレゼンテーションの最も効果的な使用方法の一つは、インターネットを利用したプレゼンテーション配信である。プレゼンテーションを再生するリソースやシステムは視聴者に依存するため、情報の送り手は通常それらを考慮して様々な方法を提供する必要があるが、MPML スクリプトを用いてプレゼンテーションをコーディングしておけば、一種類の方法で (ただし代替コンテンツの参照ロケーション等を示す必要はあるが) 最も効果的なプレゼンテーションが実現する。

たとえばユーザが学会の発表論文に興味を持った場合、資料だけでは伝わりにくいニュアンス等も、発信者がプレゼンテーションを MPML で記述しておくことで、発表資料のテキストと共に発表時の音声、OHP の画像等が視聴でき、エージェントキャラクタがそれらを用いて発表者と同様の発表をすることで臨場感を味わうことができる。

今後は上述したような MPML を利用するためのツールを順次開発する必要がある。また、視聴者からサーバへのレスポンスを何らかの形で保存しておく機能を持たせることによって、視聴者からのインタラクションを統計的に処理し、最も効果的なプレゼンテーションを実現するために利用するなどの応用が考えられる。

7 おわりに

本稿では、効果的な情報授受が必要になっている最近の情報社会を背景に、擬人化エージェントを用いたマルチモーダル ·プレゼンテーションの有効性に着目し、それを実現する為の技術として、独自開発した言語 MPML の紹介を行った。また、MPML を利用するためのツールとして開発した MPML コンバータについてその報告を行い、

今後の展望を述べた。

参考文献

[1] 筒井、李、土肥、石塚: マルチモーダルプレゼンテーションエージェントの作成. 情処全大 (57), No.1N-9, 1998.9.

[2] 筒井、土肥、石塚: マルチモーダル・プレゼンテーション生成言語ＭＰＭＬ. 情処全大 (58), No.5E-1, 1999.3.

[3] H. Dohi and M. Ishizuka: A Visual Software Agent: An Internet-Based Interface Agent with Rocking Realistic Face and Speech Dialog Function. *AAAI technical report of Internet-Based Information Systems.* WS-96-06, pp.35-40, 1996.

[4] Ashow: Hasegawa, O., Sakaue, K.: A CG Tool for Constructing Anthropomorphic Interface Agents. *IJCAI-97, W.S. Animated Interface Agents: Making Them Intelligent.* pages 23–26, Nagoya, Japan, 1997.

[5] Tsukasa Noma and Norman I. Badler: A Virtual Human Presenter. *In Work note of IJCAI-97, W.S. Animated Interface Agents: Making Them Intelligent.* pages 45–51, Nagoya, Japan, 1997.

[6] E. André, T. Rist and J. Müller: WebPersona: A Life-Like Presentation Agent for the World-Wide Web. *In Work note of IJCAI-97, W.S. Animated Interface Agents: Making Them Intelligent.* pages 53–60, Nagoya, Japan, 1997.

[7] XML: http://www.w3.org/TR/REC-xml/

[8] SMIL: http://www.w3.org/AudioVideo/

[9] Microsoft Agent: Developing for Microsoft Agent, Microsoft Press, 1998.

付録– MPML 要素の言語仕様–

要素名 head

要素属性 id

要素内容 <rec>?, <layout>|<switch>

要素名 rec

要素属性 id,

要素内容 <meta>*, <abst>*

要素名 meta

要素属性 id, name, content

要素内容 無し

name にデータ名を、content にその値を記載する。たとえば以下のような書式になる。
<meta name="author" content="author_name" />

要素名 abst

要素属性 id, lang

要素内容 テキストデータ

要素内容の言語を lang で指定できる。

要素名 layout

要素属性 id, type

要素内容 任意

type でレイアウトの書式を指定できる。

要素名 root-layout

要素属性 id, title, width, height, background-color

要素内容 無し

title でブラウザのタイトル、background-color で背景の色、width, height でルートウィンドウの幅、高さを指定できる。

要素名 region

要素属性 id, location, box, left, top, width, height, z-index, fit, background-color, title

要素内容 無し

ブラウザ上の「点」は location="50, 30" のような座標表記、もしくは left="50", top="30" といった指定をすることができる。

ブラウザ中の「範囲」は box="50, 30, 150, 110" という表記、または左上の点からの相対距離による box="50, 30, +100, +80" といった指定方法も可能である。これらの指定方法は先程の left, top に width="100", height="80" を加えて指定した場合と同じである。また、属性 location、box、left、top、width、height はルートウィンドウのサイズに対するパーセント表記で指定することも可能である（図７）。

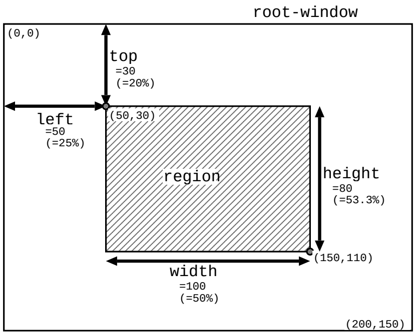

図 7: リージョン指定

z-index は占有範囲が重なるリージョン等のレベルを指定し、fit は指定したリージョン範囲において、固有の大きさを持つメディアオブジェクトを実行する場合の処理方法を指定することができる。

属性 title では そのリージョンに関する情報を記載することができる。たとえばポインタがそのリージョンにある場合にこの属性値をツールバーで表示する等の使用方法が考えられる。

要素名 body

要素属性 id

要素内容 <agent>*, <move>*, <speak>*, <play>*, <par>*, <seq>*, <ref>*, <switch>*, <a>*, <switch>*

要素名 agent

要素属性 id, character, appear, region, lang, voice-type

要素内容 無し

character で使用するキャラクタ名を指定し、lang、voice-type ではそれぞれ発話時の言語と声質を指定する。

region ではキャラクタが現れる位置のリージョンを参照し、また appear では出現のタイミングを設定する。

要素名 move

要素属性 id, agent, region, location, stand, speed

要素内容 無し

agent で移動対象である要素<agent>の id 値を参照し、region では移動させる位置のリージョンを参照する。また、要素<region>で定義していない位置へも、location を用いて座標指定し、移動を行うことができる。

stand は指定した移動位置に対するキャラクタの立ち位置を指定し、speed で移動速度の速さを指定することができる。

要素名 speak

要素属性 id, agent, lang, voice-type, speed, begin, end, dur, alt

要素内容 テキスト，<anchor>*

属性 agent、lang、voice-type、speed ではそれぞれ、発話するキャラクタ、言語、声質、声のはやさを指定する。

また、属性 begin、end、dur では発話している時間を指定することができる。

さらに、視聴者のシステムが音声合成エンジンをサポートしていない場合にそなえて、属性 alt に音声の代替として表示させるテキストを記載しておくことができる。

要素名 play

要素属性 id, agent, act, parts, object, object-loc, degree, speed, begin, end, dur, track, point-gesture

要素内容 <anchor>*

挨拶動作をさせたい場合には <play act="greet" /> と記述する。act 属性の属性値は任意であるが、プレイヤに認識させる為に、仕様書で推奨される値を使用するべきである。

object、もしくは object-loc では、動作の対象オブジェクトを id 参照、もしくは位置指定する。<play act="point" object="image1" /> と記述すれば、"image1" で参照されるオブジェクトを指し示す動作を行わせることができる。また、track は対象オブジェクトが移動した場合に追跡するかどうかを指定する。エージェントが指さし動作を行う場合には、point-gesture を用いてキャラクタの手の形を変化させることもできる。

また、属性 degree、speed、parts ではそれぞれ、動作の強さの程度、素早さの程度、動作させるキャラクタの部位を指定することができる。

要素名 par

要素属性 id, begin, end, dur, endsync, region, repeat, title, abstract, author, copyright

要素内容 <agent>*, <move>*, <speak>*, <play>*, <par>*, <seq>*, <ref>*, <switch>*, <a>*, <switch>*

begin、end、dur、endsync で、コンテンツの実行時間と終了のタイミングを指できる。また、repeat ではコンテンツ実行の繰り返し回数を指定する。

属性 title、abstract、author、copyright はコンテンツに関するメタ情報を記載しておく用途で用いられる。

要素名 seq

要素属性 id, begin, end, dur, region, repeat, title, abstract, author, copyright

要素内容 <agent>*, <move>*, <speak>*, <play>*, <par>*, <seq>*, <ref>*, <switch>*, <a>*, <switch>*

要素名 ref

要素属性 id, src, region, media, type, alt, begin, end, dur, longdesc, repeat, title, abstract, author, copyright

要素内容 <anchor>*

src で参照するメディアオブジェクトの URI を指定する。また属性 media、type ではそれぞれメディアの種類、MIME タイプを指定することができる。

alt では何らかの原因でメディア再生ができない場合に表示する短い代替テキストを記載する。また、代替テキストが長い場合には属性 longdesc において、そのテキストへのリンクを指定することができる。

要素名 a

要素属性 id, title, href, show, mode, begin, end, dur, region, listen, lang, key, confidence

要素内容 無し

mode において、なんらかのインタラクションが起こった際のプレゼンテーションの挙動を設定する。例えば mode="jump" とした場合は、プレゼンテーションが指定されたリンクへジャンプする。リンクの指定は属性 href によって行う。他リンクへのジャンプが行われた場合は、属性 show を用いて現在のプレゼンテーションの挙動などを制御することができる。

受け付ける音声は key で指定する。その際、選択記号として "|"、省略可能記号として"[]"、省略記号として"..."、解釈優先記号として"()"などを使用することができる。例えば key="...[say](hello|hi)" という指定では、"say hello"、"please hello"、"Hi" などにマッチする。

また、confidence でマッチする音声の信頼性を指定することができる。入力音声と key とのマッチがこの値より低い場合には処理は行われない。

listen で音声入力をうけつけるかどうか、また begin、end、dur で音声入力を受け付ける時間を指定することができる。

region が指定された場合には、指定リージョンでクリックが行われた場合にも何らかの制御を行うことになる。

要素名 anchor

要素属性 id, title, begin, region, location

要素内容 無し

要素名 switch

要素属性 id, title

要素内容 <agent>*, <move>*, <speak>*, <play>*, <par>*, <seq>*, <ref>*, <switch>*, <a>*