オブジェクトの合成によるメディアの記述

只野 俊介† 布川 博士‡ 佐藤 究† 宮崎 正俊†

†東北大学大学院情報科学研究科 ‡宮城教育大学理科教育研究施設

現在WWWや電子メール等のメディアを用いて、様々なコミュニケーショ ンがコンピュータ上で行なわれている。しかし、それらのメディアは第三者によって与えられた固定的なものであり、その使用法も第三者によって固定されている。本研究ではメディアをオブジェクトとしてモデル化し、そのモデルに従いメディア記述言語を作成する。その結果としてプログラマブルなメディアをユーザに提供し、より柔軟なコミュニケーションを可能とするのが本研究の目的である。

1 はじめに

現在コンピュータおよび、それを取り巻くネットワークが急速に普及している。 それに伴いコンピュータ上で、様々なサービスやアプリケーションを用いたコミュニケーションが行われる様になってきた。またコンピュータの処理能力の増大に伴い、アプリケーションも電子メールの様に単一のプリミティブなメディアを用いるものから、WWWやCU-SeeMe等のテレビ会議システムのような複数のプリミティブなメディアを用いるものへと進化してきている。

しかし現状では、特定のアプリケーションによって特定のコミュニケーションが規定されている。さらに、そこで利用されているメディアの取り扱い方、すなわちメディアの操作、記述、関係、ユーザとのインタラクションまでもがアプリケーションで規定されている。つまり現状ではユーザにとってコミュニケーションのためのメディアは第三者によって与えられる固定的なものであり、その利用法も第三者によって限定されている。そのため、ユーザが望む形でのコミュニケーションを行うために、ユーザが既存のメディアを組み合わせたり、新たなメディアを構築することは非常に困難である。

そこで現在我々は、ユーザが統一的な枠組みで自由なメディアの構築を可能とし、さらにそのメディアを使ってコミュニケーションすることを可能とする環境を実現するための研究を行っている[1]。ここでいうメディアの構築とは単なるプリミティブなメディアの定義、合成ではない。プリミティブなメディアの合成によって作られた、従来の意味でのサービスやアプリケーションといったものをさらにメディアとして取り扱うことを可能とするものである。

そのために本稿では、メディアの構造を分析しメディアをオブジェクトとしてモデル化していく。

本稿は4章からなる。まず2章でメディアの構造の分析を行い、本稿で扱っていくメディアを定義する。次に3章で2章で述べたメディ

Describing Media by Composing Object, Shunsuke TADANO†, Hiroshi NUNOKAWA‡, Kiwamu SATO†, Masatoshi MIYAZAKI†, †Graduate School of Information Siences, Tohoku University ‡Research Institute for Science Education, Miyagi University of Education

アをオブジェクトとしてモデル化したメディアオブジェクトという計算モデルを提案し、その記述法を考察する。最後に4章でまとめとする。

2 我々の扱うメディア

2.1 メディアとメディアデータ

一般にメディアには2種類の意味がある。一つは電話機等のような何らかのデータを転送する媒体としてのメディアであり、もう一つは新聞や動画等の実際に転送されるデータとしてのメディアである。本稿ではこれらを区別して扱い、前者をメディア、後者をメディアデータと呼ぶものとする (図1)。つまりメディアはそれ自体情報を持つものではなく、メディアデータをメディエイトする (伝える) ものである。本研究で扱うメディアデータは音声、動画、テキスト等のマルチメディアデータである。

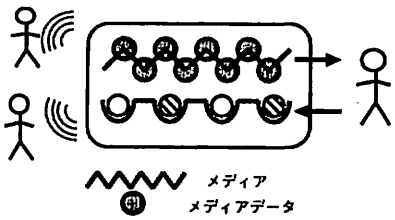

図 1: メディアとメディアデータ

複数の異なるメディアを統一的に扱う環境を構築するために、メディアは以下の条件を満たすものとする。

1. ユーザがメディアを自由に生成できる
2. ユーザがメディアを自由に合成できる
3. ユーザがメディアを自由に転送できる

これらの条件の基にメディア記述言語を作成し、ユーザがプログラム可能なメディアを実現する。メディアを利用する際には、この言語を用いてメディアを記述し、メディア記述をメディアインタプリタによって解釈、実行することによりメディアが出現する。メディアインタプリタは各ユーザにつき一つ存在する。ユーザはメディアインタプリタを通してメディアを生成、操作しコミュニケーションを行う。

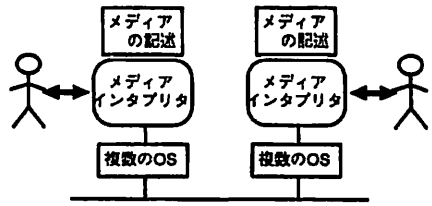

図 2: メディア記述とその実行

2.2 メディアの例

我々が構築したいメディアについて例を用いて説明する。同時にそれら複数の例について従来のメディアと我々のメディアとの比較を行なう。

(1) 一部をパラメタ化したHTML記述

HTML記述aHTML(図3)において、はめ込むメディアデータ (例: 画像) をパラメタ化することによってWWWのページのテンプレートaHTML(X)を作成することができる (図3)。このaHTML(X)はパラメタXによって画像を転送する媒体を記述したことになり我々の扱うメディアであるといえる。つまり、aHTML(X)はメディアデータとして画像が流れ、画像を流すことによってWWWのページが出来上がるメディアである。

これは現在でも、パラメタを与えたら対応するHTML記述を生成するシェルスクリプトを用いて実現できる。しかし、シェルスクリプトによってOSを限定することになり、メディア自身を転送することは極めて困難である。

(2) メディアを合成するWWWページ

我々の考えるメディアは自由にメディアの

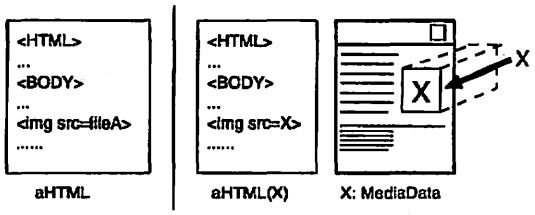

図 3: パラメタ化した HTML 記述

合成をできるものである。そこで aHTML(X) にさらに双方向文字通信メディア phone を合成してみる。その結果、comp(aHTML(X),phone) という新しいメディアができる (図 4)。

これは現在でも Netscape の様な特定のビューワにおいて、特定のメディア (例えば phone) に対して特定の Plug-in を用いることによって実現できるが、一般性に欠ける。

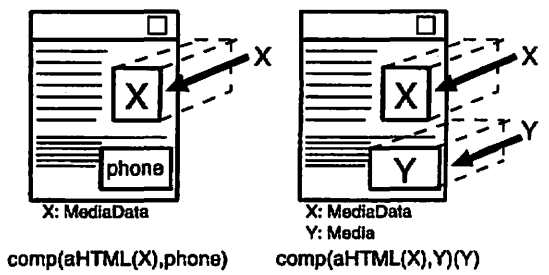

図 4: メディアの合成、転送

さらに我々のメディアはメディア自身を転送することができるので、メディアをパラメタライズし comp(aHTML(X),Y)(Y) を作成することができる (図 4)。これは一般的なメディアの定義を Y で受けとることを意味し、これを Network 上で行なうとすればメディアの転送に相当する。

これは現在の WWW 上では難しい。それは WWW 上で扱えるメディアは既存の限られたメディアであり、それを自由に生成、合成、転送する方法が無いためである。

3 メディア構築の方法

3.1 メディアオブジェクトによるモデル化

ここでモデル化するメディアは、電子メール、双方向音声通信などのプリミティブなメディアと、基本的なメディアを合成してできる合成メディアである。これらのメディアをオブジェクト指向の考え方をもとにメディアオブジェクトとしてモデル化する。メディアオブジェクトには monoMObj と compMObj の 2 種類ある。

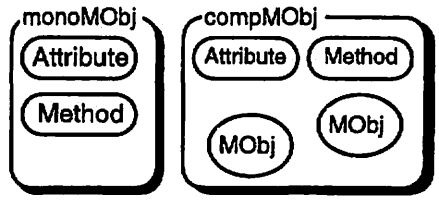

図 5: メディアオブジェクト

monoMObj はプリミティブなメディアをモデル化したメディアオブジェクトで、状態を保持する属性と、属性を操作するメソッドで構成される (図 5)。

compMObj は合成メディアをモデル化したメディアオブジェクトである。これは属性とメソッドの他に、合成するメディアオブジェクトで構成される (図 5)。compMObj は中に含まれているメディアオブジェクトにとっては存在する場であるが、外のメディアオブジェクトに対しては 1 つのメディアオブジェクトとして振舞う。また compMObj の中に別の compMObj を含むこともできるので、compMObj は階層的に構成される。

3.2 メディアの合成と転送

メディアオブジェクトを用いてメディアを合成する場合には、まずその基となるメディアをモデル化したメディアオブジェクトを用意す

る。次にそれらのメディアオブジェクトを含むcompMObjを作成する。compMObjに含まれる各メディアオブジェクトは、それぞれ自律的に他のメディアオブジェクトと同期しながら動作し、compMObj全体で1つの合成メディアとして機能する。その際の同期して動作するための材料としてcompMObjの属性値が使われる。つまりcompMObjの属性値は協調計算モデルで例題にされる黒板モデルの黒板に相当し、compMObjは中に含まれるメディアオブジェクトにとって存在する場であるといえる。

例として電子メールとFaxを統合したFaxメールメディアを考える[3]。これは電子メールの文章をFaxで送信できるメディアである。このメディアをメディアオブジェクトを用いてモデル化したのが図6である。FaxMObjとMailMObjは、テキストをFaxMailMobjの属性値を用いてやりとりする。

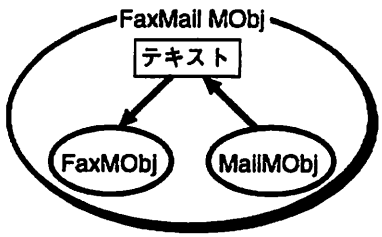

図6: FaxMailメディアオブジェクト

メディアオブジェクトへの操作はメッセージパッシングによって行われる。しかしcompMObj内のメディアオブジェクトは外部には隠蔽されており、直接内部のメディアオブジェクトを操作することはできない。compMObjは外部のメディアオブジェクトに対しては1つのメディアオブジェクトとして振舞う。そこで内部のメディアオブジェクトは、compMObjのメソッドを通じて間接的に操作する。

compMObjの内部のメディアオブジェクトを操作するメソッドは、中に含まれるメディアオブジェクトへのメッセージの集合で記述される。これは、メディアオブジェクトの操作の言語体系と、生成の言語体系が同一であるということを示している。そのためユーザが容易にメディアを合成することができる。またメッセージとしてメディアの記述を転送することができる。

例としてテレビ会議メディアオブジェクトTVmeetingMObjを考える。これはその要素として、双方向画像通信メディアのVideoMObjと双方向音声通信メディアのVoiceMObjを含んでいる。この場合TVmeetingMObjの通信を開始するstartメソッドは次の様に書かれる。

```
(start ((send VideoMObj (set_size (100,80))),
        (send VideoMObj (set_at (10,10))),
        (send all start)))
```

TVmeetingメディアがstartメソッドを受け取ると、図7の様にメッセージが送信される。

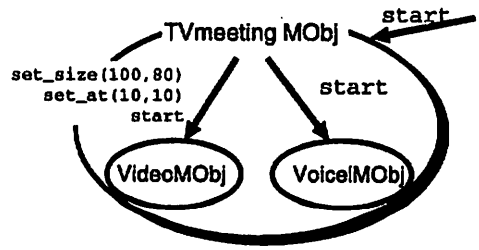

図7: TVmeetingメディアオブジェクト

3.3 メディア記述言語による記述例

メディアオブジェクトを用いて、双方向音声通信メディア、双方向画像通信メディアを記述し、さらにそれらを組み合わせてテレビ会議メディアを構築する例を記述してみる(図8)。

4 おわりに

本稿ではメディアを分析し、その分析に基づいてメディアをメディアオブジェクトとしてモデル化し、メディア記述言語を設計した。メディア記述言語は、メディアオブジェクトの定義と操作を同じ体系で行うものである。その為、ユーザがメディアの生成、合成、転送を容易に行うことができる。

よって、このメディア記述言語とその実行環境であるメディアインタプリタによってユーザはコンピュータ上に自由にメディアを構築し、自分で構築したメディアを用いてコミュニケーションを行うことが可能となる。

現在、メディアオブジェクトは分散型言語DeLis[5]を用いて開発中である。

参考文献

[1] 布川 博士, 只野 俊介, 菊池 一彦, 宮崎 正俊 : "オブジェクト指向に基づくメディアの構造記述実験", 高度データベース松江ワークショップ講演論文集, pp.158-162 (1996).

[2] 只野 俊介, 布川 博士, 宮崎 正俊 : "計算モデルに基づくマルチメディアデータの記述", 情報処理学会研究報告 96-DPS-76, pp.61-66 (1996).

[3] 菊地 一彦, 布川 博士, 宮崎 正俊 : "メディア統合アーキテクチャの提案", 情報処理学会研究報告 96-DPS-76, pp.67-72 (1996).

[4] 武宮 博, 布川 博士, 野口 正一 : "協調型計算モデルにおける field への自律性の導入", 日本ソフトウェア科学会第 8 回大会論文集, pp.61-64 (1991).

[5] 三石 大, 布川 博士, 野口 正一 : "分散環境のための言語系 DeLis", 情報処理学会研究報告 93-PRG-10, pp.57-64 (1993).

```
(define_compMObj TVmeetingMObj
  (Attribute
   (mobj_list  (VideoMObj, VoiceMObj))
            #このMObjはmobj_listのMObjを含む
   (destination hoge@aaa.tohoku.ac.jp) #通信先
   (size .....)  #TVmeetingMObjの表示サイズ
   (at   .....)  #TVmeetingMObjの表示位置
   ( .....))

  (Method
   (start ((send VideoMObj (set_size (100,80))),
           (send VideoMObj (set_at (10,10))),
           (send all start)))
      #VideoMObjの表示サイズと位置を決定する。
      #(この場合位置とはTVmeetingMObj内での相対位置)
      #VideoMObj, VoiceMObjの通信を開始する。
   (stop (send all stop))
      #VideoMObj, VoiceMObjの通信を停止する。
   (set_destination <destination_str>
         ((set destination <destination_str>),
          (send all (set_destination <destination_str>))))
      #TVmeetingMObj, VoiceMObj, VideoMObjに対して通信先を指定する。
   (set_size <(int,int)> (set size <(int,int)>))
      #TVmeetingMObjの表示サイズを指定する。
   (set_at <(int,int)> (set at <(int,int)>))))
      #TVmeetingMObjの表示位置を指定する。

(define_monoMObj VideoMObj
  (Attribute
   (destination hoge@aaa.tohoku.ac.jp) #通信先
   (size .....)   #VideoMObjの表示サイズ
   (at   .....)   #VideoMObjの表示位置
   ( .....))

  (Method
   (start) #VideoMObjの通信を開始する。
   (stop)  #VideoMObjの通信を開始する。
   (set_destination <destination_str> (set destination <destination_str>))
           #VideoMObjの通信先を指定する。
   (set_size <(int,int)> (set size <(int,int)>))
           #VideoMObjの表示サイズを指定する。
   (set_at <(int,int)> (set at <(int,int)>))))
           #VideoMObjの表示位置を指定する。

(define_monoMObj VoiceMObj
  .....
  .....)
```

図 8: 記述例