

## プログラマブルなメディアとその記述言語

大市 津義<sup>1</sup> 布川 博士<sup>2</sup> 宮崎 正俊<sup>1</sup>

<sup>1</sup>東北大学大学院情報科学研究科 <sup>2</sup>宮城教育大学理科教育研究施設

本稿では、柔軟なコミュニケーションに必要とされるプログラマブルなメディアと、そのメディアを記述するためのメディア記述言語の提案を行う。また、コンピュータネットワークを利用した、プログラマブルなメディアを実現するためのシステムの設計と実装も行う。

本来、コミュニケーションを行う場合、伝える内容に合わせてそれを行う人が自由にコミュニケーションの形態を選択できなければならない。しかし、従来の道具を介したコミュニケーションではその自由度はほとんど与えられなかった。

メディア記述言語は、このような問題を解決するためにメディアを動的に組み合わせて道具の機能をプログラミングするための言語である。この言語によりコミュニケーションを行う人々は自由な形態でのコミュニケーションを得られる。

## Proguammable Media and Media Description Language

Tsuyoshi Ohichi<sup>1</sup> Hiroshi Nunokawa<sup>2</sup> Masatoshi Miyazaki<sup>1</sup>

<sup>1</sup>Graduate School of Information Sciences, Tohoku University

<sup>2</sup>Institute for Science Education, Miyagi University of Education

In this article, we propose Programmable Media required by flexible communications and Media Description Language to describe it. We also design and implement a system to realize Programmable Media over computer networks.

Properly speaking, communicator should freely be able to choice content-related forms of communication. But, there have been few freedom of that kind in usual communication with tools.

In order to solve this problem, functions of tool will be programmed with dynamic media combinations. Media Description Language is the possible language to do it. If communicator use this language, s/he will be able to obtain communications with free forms.

## 1 はじめに

メディアは人と人とのコミュニケーションを媒介する手段を提供するものである。したがって、多様なメディアの利用は人々のコミュニケーションを支援する。また、多彩なコミュニケーションはスムーズな意思の疎通をはかる上で必要不可欠である。

元来、コミュニケーションは人同士が面と向かい合っではじめてできるものであった。しかし、現在では電話などに代表される道具の登場により人々は直接会うことなくコミュニケーションを行うことが可能となった。そして、さらにコンピュータとそれをつなぐネットワークは、これまでにない多様なメディアの発達を助長し、新たなコミュニケーションの形態を提供するであろう。

我々は、このような点を踏まえて、日々着実に進歩するコンピュータとそれを取り巻く技術を如何にコミュニケーションに利用するかについて検討を行った。そして、一つの解答として、本稿ではプログラマブルなメディアを実現するためのメディア記述言語の提案を行う。メディア記述言語はコミュニケーションにより意思の疎通をはかりたいと考える人々が、その場で最良であると考えたメディアを自ら選択し、そのメディアの実現をシステムに要求するための言語である。したがって、コミュニケーションを行う人々、すなわちコミュニケーターたちは絶えず最良のメディアで確実な意思の疎通をはかることができる。また、本稿では同時に技術の発達と共に実用化するメディアの追加を考慮にいれて、プログラマブルなメディアを実現するためのシステムの設計も行う。このシステムはメディア記述言語により駆動され、多様化するコミュニケーターの要求に対応できるコミュニケーションのための道具である。

以下では、まず2章でコミュニケーションに対する考察を行う。次に、3章ではコミュニケーションへのメディアの利用について検討する。4章ではプログラマブルなメディアを実現するシステムの設計を行い、5章ではメディア記述言語の記述例を挙げる。そして、6章で設計に基づいた実装について述べる。最後に、7章でまとめを行い、これからの課題と問題点についてふれる。

## 2 コミュニケーションの形態

コミュニケーションには既存のものでも会話、郵便、出版、テレビ放送など多種多様な形態が存在する。これらのコミュニケーションは人々が意思の疎通をはかるために日常的に利用されるものである。本章ではこのようなコミュニケーションの特徴を抽出することにより、より良いコミュニケーションの形態について検討する。

まず、コミュニケーションをマイクロにとらえた場合について考える。マイクロなコミュニケーションとは、一方のコミュニケーターが何らかの情報を発信し、他方のコミュニケーターがその情報を受信するという極単純なものである。このマイクロなコミュニケーションの特徴としては次のようなものが考えられる。一つ目として、情報が

どのような表現形式で伝達されるかである。例えば、会話では言葉を音声として、郵便では文章や絵を紙の上にかかれた画像として伝達する。二つ目として、情報の発信から受信が可能になるまでに掛かる時間である。例えば、会話ではほとんど時間は掛からない。これに対して、郵便や出版では日の単位で時間が掛かる。当然、即座に受信が可能になることが望ましい。三つ目として、受信のタイミングまたは情報の蓄積に関する特徴である。例えば、会話は受信のタイミングを逃すと再発信されるまでその情報を得ることができない。これに対して、出版では情報は蓄積されて伝達されるので受信のタイミングは受信者が自由に選択できる。次に、四つ目として、発信者と受信者の空間的な位置関係が挙げられる。そして最後に、五つ目として、伝達を行う時に事前に必要となる準備がコミュニケーションを特徴付ける。例えば、電気信号を用いた電話では、コミュニケーションをする以前に電話機や配線を用意しなくてはならない。

次に、コミュニケーションをマクロにとらえた場合について考える。マクロなコミュニケーションの特徴はマイクロなコミュニケーションがどのように反復されるかにある。例えば、会話や郵便では発信者と受信者を入れ替えてマイクロなコミュニケーションが交互に繰り返される。これに対して、出版やテレビ放送では不特定多数の受信者に対して同時に発信される。

以上のような特徴によりコミュニケーションを分類することが可能である。これに基づいて、最良であると思われる性質を抽出することにより、以下に示す性質を持つコミュニケーションが理想的であることがわかる。この性質の内、(1)から(5)までの性質はマイクロなコミュニケーションの特徴から、(6)および(7)の性質はマクロなコミュニケーションの特徴から得られたものである。しかし、このような特徴の中でも(1)の性質についてはさらに検討が必要である。それは自由な表現形式とはどのようなものであるかが特定できないためである。

- (1) 任意の表現形式の選択が可能
- (2) 即座に受信が可能
- (3) 情報の蓄積が可能
- (4) 発信者と受信者の位置関係に非依存
- (5) 事前の準備が不要
- (6) 任意の時間に発信が可能
- (7) 任意の相手に発信が可能

自由な表現形式とは、受信者が受信可能なすべての表現形式を示している。これは発信者が発信可能な表現形式ではなく、受信者を主体として決定する。なぜならば、発信者はコミュニケーションを行うために情報を発信するが、その情報が受信者に受信不可能な表現形式であったならば、そのコミュニケーションは成り立たないからである。したがって、受信者の受信能力に依存した表現形式がコミュニケーションのための自由な表現形式

である。

次に、人である受信者の受信能力について考察する。人はどのような情報でも五感からしか取り込むことができない。したがって、どのような情報でも、結果的に五感で感じ取れる表現形式に変換されなければならない。この表現形式はいわゆる刺激といわれるものである。ここで本題に戻るが、任意の表現形式の選択が可能であるためには、意図する情報を最も効果的であると思われる刺激に変換できなければならない。したがって、理想的なコミュニケーションを行うためには任意の刺激の生成が可能でなければならない。しかし、残念ながら図1に示すように人が道具を利用せずに直接作り出せる刺激は、それを感知取る能力に比べ遥かに劣っているのが現実である。

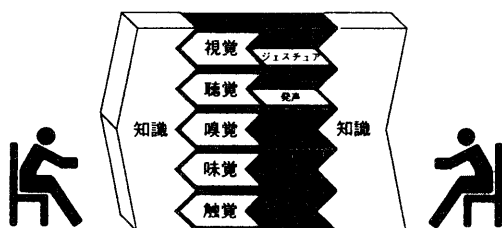


図1 人のコミュニケーション能力

### 3 メディアとその記述言語

ここでは、2章で挙げた理想的なコミュニケーションを実現するために、コンピュータとネットワークを如何に利用するかについて検討する。ただし、コンピュータやネットワークの物理的な配線などを用意しなければならないことから(5)の性質は考慮にいれない。また、ネットワーク内のノードになれるコンピュータ間でしかコミュニケーションできないという制約と、ネットワーク上のコンピュータは空間的な距離に関係なく対等であることから、(4)の性質についても考慮しない。

まず、(1)の性質の実現について考える。受信者の受信能力を余すところなく利用するためには、任意の刺激の生成が欠かせないことは2章で述べた通りである。そのような刺激の基となる情報をネットワークを介して伝達するためには、発信者の持つ情報を刺激生成のための符号列に変換する機構と、そのような符号列から発信者が意図する刺激を生成する機構が必要である。例えば、知識として記憶している香水の香りの情報を符号化する機構と、その符号列から香りを再現する機構などである。本稿では、これらの対となった機構が行う作業がコミュニケーションの媒体となっていることから、これらの機構の機能をメディアとして考える。特に、ここで考えたメディアは情報の表現に関係するものであるので表現メディアと呼ぶものとする。したがって、(1)の性質を実現するためには、ありとあらゆる情報と刺激に関する表現メディアをコンピュータ上に用意することが必要となる。しかし、現実的には情報の入力や刺激の出力に関す

るデバイスが十分に整っていないため、そのようなデバイスの開発を待たなければならない。代わりに、逐次、メディアの追加が行える機構をコンピュータ上に用意することにより、時間と共に機能が充実するように実現する必要がある。

次に、(2)および(3)の性質の実現について考える。これらの機能を同時に実現するのは本質的に不可能ではないが、実際に必要とされる機能はどちらか一方である。したがって、符号列をネットワークを利用して(2)または(3)の機能で伝達する機構をコンピュータ上に用意すれば良い。これらの機構の機能も、表現メディア同様、伝達メディアと呼ぶものとする。また、表現メディアと同様に、例えば、(2)および(3)の性質を同時に実現する必要が発生した場合などに備えて、伝達メディアを追加するための機構を用意する必要がある。

最後に、(6)および(7)の性質の実現について考える。任意の時間に任意の相手に情報の発信が可能であるためには、前述の表現メディアと伝達メディアをその場その場で組み合わせて、必要なコミュニケーションのためのメディアをすぐに構築できなければならない。そのためにはコンピュータに対して指示を与える機構が必要となる。本稿ではこの機構としてメディア記述言語とそれを処理するコンピュータ上のインタプリタを用いた設計を行う。メディア記述言語は図2および図3に示すように、コミュニケーションの進行と共に変化するコミュニケーションの要求を満足させるメディアの記述ができなければならない。

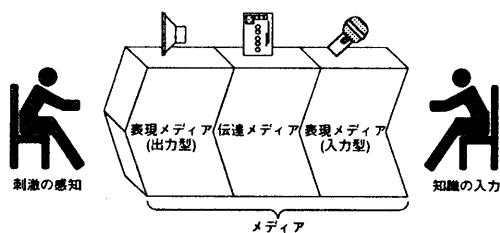


図2 メディアの構築

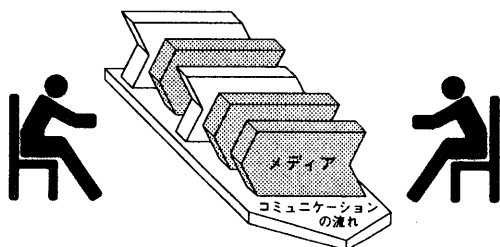


図3 メディアによるコミュニケーション

以上、本章では理想的なコミュニケーションを行うためのプログラマブルなメディアの実現について述べた。実際に稼働するシステムを構築した場合、コミュニケーションを行う以前にコンピュータは他のコンピュータと

協調的に大量の処理を行うことになる。しかし、そのおかげで、コミュニケータはより支援的なコミュニケーションのためのメディアを手に入れることが可能になる。

#### 4 プログラマブルなメディアの実現とシステム設計

本章では、3章で提示したプログラマブルなメディアを実現するためのシステム的设计を行う。このシステムは、ユーザとのインタラクションおよびメディアとそのオブジェクトの管理を行う複数のノードの有機的な結合により構成される。この結合はノード自身が必要に応じて行うため、システム中のノード数は逐次変化する。時には、システム全体が一つのノードのみで構成されることもある。

以下では、システム上でのメディアの扱いと、システムを構成するノードの実現について述べる。

##### 4.1 メディアの扱い

メディアはメディアオブジェクトを生成するための雛形として後述のメディアマネージャ内に存在している。したがって、メディアオブジェクトはメディアを雛形に具現化したものである。

メディアは論理的な構造によりアトミックメディアとコンポジットメディアの二つに分類される。アトミックメディアは事前に用意された動的には追加不可能な組み込みのメディアである。これに対して、コンポジットメディアは他の複数のメディアを組み合わせて動的に構築可能なメディアである。また、後述のメッセージマネージャの機能により、一つのコンポジットメディアより生成された複数のメディアオブジェクトは、統一的に一つのメディアオブジェクトとして扱うことができる。なお、ここでいう動的とはシステムの稼働中を示す。

メディアオブジェクトは概念的に図4に示すような構造を持った自立的なオブジェクトである。メディアオブジェクトは実装においても完全に独立した実行単位として実現される。また、図に示したようにメディアオブジェクトは多数のインタフェースを備えている。そのインタフェースの一つであるストリームは、メディアオブジェクト間の表現情報の交換に利用される。したがって、標準となる交換手法が必要である。その他のインタフェースとしてメッセージインタプリタがある。これは後述のメッセージマネージャからのメッセージによってメディアオブジェクト自身の挙動を決定する制御機構である。したがって、これについても標準となるメッセージ授受の手法が必要である。そして、もう一つのインタフェースとしてヒューマンメディアインタラクションがある。このインタフェースはメディアオブジェクトと人が相互に情報を入出力するための機構である。例えば、メディアオブジェクトが持つ音声取り込み機構などがこれに相当する。

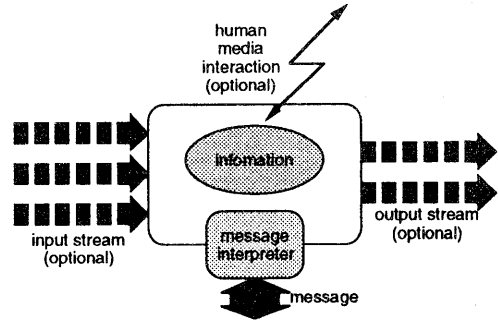


図4 メディア・オブジェクトの構造

3章で述べたようにシステムにはメディアを追加するための機構が必要である。先にも述べたように、アトミックメディアは動的にメディアマネージャに追加することはできない。その代わりに、追加したいアトミックメディアを雛形とするメディアオブジェクトを実現するための実行形式の作成を事前に行うことにより、追加することができる。ただし、このような実行形式の作成にはいくつかの制約がある。その一つとして、メッセージインタプリタとストリームについては、決められた標準仕様に従って実装しなければならない。これにより、他のメディアオブジェクトとの情報交換やシステムとメッセージを授受することができる。ただし、ストリームにより交換される情報の構造については特に規定しないので、入力型の表現メディアと出力型の表現メディアの間で折衝が必要である。また、ヒューマンメディアインタラクションに関しては自由に実装することができる。これにより、新しいデバイスが開発されたとき、それを利用した表現メディアを実現することができる。この点が、既存のメディアを結合して構築したコンポジットメディアと最も異なっている点である。以上のことに従えば、例えば、マイクからの音のサンプリング情報をメッセージに従って出力ストリームから送り出すようなメディアオブジェクトのための実行形式を実装することができる。このようにして実装された実行形式はアトミックメディアとしてメディアマネージャに登録することでシステムから利用可能となる。以上により新しいデバイスの出現などに対処できることを保証する。

##### 4.2 ノードの実現

ノードはシステムを構成する最も重要な部分である。ノードは図5に示すいくつかのレイヤにより実現される。それぞれのレイヤは果たすべき役割と隣り合うレイヤへのインタフェースが規定されるが、実現は一通りではない。また、ノードは各レイヤの他にメディアスペースを構成する。メディアスペースはシステム内のノード全体にわたって広がる一つの仮想的な空間である。また、メディアオブジェクトはあたかもこのメディアスペース内に存在しているかのように振る舞う。そして、メディアスペース内のメディアオブジェクトは、協調的に動作してコミュニケーションのための機能を提供す

る。

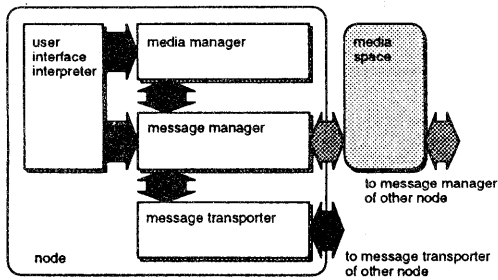


図5 ノードの構造

以下では各レイヤの持つ機構について詳しく述べる。

#### 4.2.1 メッセージトランスポータ

メッセージトランスポータは複数のノードに依存するメディアオブジェクトを生成するときに、メッセージの伝達経路を提供する機構である。この機構により、メディアスペースが複数のノードにまたがって存在しているかのように見せることができる。

メッセージトランスポータによるノードの接続はチャンネルにより行われる。このチャンネルはそれに依存するメディアオブジェクトの寿命とほぼ同じ寿命を持っている。チャンネルの確立は、接続したいと考えるノードのメッセージトランスポータの接続要求待ち状態にあるチャンネルに対して接続要求を出すことで行われる。この接続要求待ち状態にあるチャンネルは、どのメッセージトランスポータにも常時一つは存在する。したがって、どのノードも別のシステムに組み込まれるための切り口を持っていることになる。また、確立されたチャンネルは、リアルタイムでのメッセージ伝達に必要な全二重同期通信を行うことができる。

チャンネルにより相互に接続されたメッセージトランスポータの存在により、メディアオブジェクトに対して統一的なメッセージ授受の機構を得ることができる。ただし、このチャンネルは、入力型の表現メディアのメディアオブジェクトが生成する符号情報を伝達するための経路ではなく、メディアオブジェクトの制御を行うためのメッセージを伝達するためのものである。情報の伝達は伝達メディアのメディアオブジェクトが行わなければならない。

#### 4.2.2 メッセージマネージャ

メッセージマネージャは、メッセージの発行依頼に基づき、メッセージを受理すべきメディアオブジェクトの特定と、実際のメッセージの発行を行うための機構である。メディアスペースはメッセージトランスポータ間のチャンネル接続により仮想的に存在するものであるので、時にはメッセージトランスポータ経由で別のノード上のメッセージマネージャからメッセージが発行されることもある。

メッセージマネージャは実質的にメディアオブジェク

トを管理している。名前から推測した場合、後述のメディアマネージャが管理しているように感じるが、そうではない。

#### 4.2.3 メディアマネージャ

メディアマネージャは、組み込みのアトミックメディアおよび新しくコミュニケータにより定義されたコンポジットメディアの管理と、それらのメディアを雛形としたメディアオブジェクトの生成をするための機構である。

前述のメッセージマネージャは、メディアオブジェクトの生成を行うためのメッセージのみ、メディアマネージャに対して発行する。これは存在しないメディアオブジェクトに対するメッセージの発行を回避するための必然的なものである。したがって、これを受けたメディアマネージャはメディアオブジェクトの生成を行う。

#### 4.2.4 ユーザインタフェースインタプリタ

ユーザインタフェースインタプリタは、コミュニケータから逐次メディア記述言語による記述を受け取り、解析を行い、その結果として他のレイヤを駆動する。受け取った記述がコンポジットメディアの定義であったならば、メディアマネージャを駆動し、メッセージの発行要求であったならばメッセージマネージャを駆動する。

メディア記述言語については次章で例を挙げる。

## 5 メディア記述言語の記述例

メディア記述言語による記述の例をリスト1に示す。この記述はテレビ電話と同等の機能を実現するためのメディアを提供する簡単なものである。記述の内容としては、四種類のコンポジットメディアを入手状に定義し、最後に定義した最終的なコンポジットメディアを雛形にしてメディアオブジェクトの生成を行っている。さらにそのメディアオブジェクトに対して使用開始、使用停止、破壊のメッセージを送っている。なお、図6に生成されたメディアオブジェクトの一部の概念図を示す。

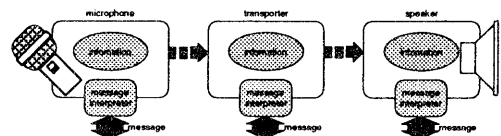


図6 コンポジット・メディアのメディア・オブジェクト

## 6 システムの実装

プログラマブルなメディアがどの程度コミュニケーションを支援するのか検証するために4章で設計したシステムの実装が必要である。しかし、現在のところシステム全体として稼働するには至っていない。

これまでに、Internetで接続されたUNIXが稼働するワークステーション上にアトミックメディアの基となるいくつかの実行形式の実装を行った。そして、システムの代わりにそれらを静的に手動で結合する実験を行っ

た。その結果、一つのメディアとして協調的に動作することが確認できた。今後は、手動で行った作業をメディア記述言語で動的に行えるようにシステム全体の実装を行わなければならない。また、そのシステムの使用によってコミュニケーションがどのように変化するか検証しなければならない。

## 7 まとめ

コンピュータの処理速度の高速化やネットワークの大容量化は、今後ますます加速するであろう。このような環境において、融通の効かないコミュニケーションのための道具は、もはや相手にされなくなることは避けられない。したがって、コミュニケータたちが要求する自由な形態でのコミュニケーションを提供することは必然的に必要である。

本稿ではどのようなコミュニケーションでも実現できるプログラマブルなメディアの可能性について論じた。また、そのプログラマブルなメディアを実現するためのシステムについても検討を行った。そして、そのシステムで必要不可欠なメディア記述言語の提案も行った。

このメディア記述言語はコミュニケータが意図する内容を的確に相手に伝えるためのメディアをシステムに要求するための言語である。したがって、この言語による記述を処理できて、要求されたメディアを提供できる4章で設計したシステムは、先に挙げたコミュニケータの要求を満足するものである。

今後、このシステムの利用によって、プログラマブルなメディアでしか実現できない新しいコミュニケーションの形態について実験的に調査したいと考えている。また、どのような形態のコミュニケーションが可能であるかは、用意したアトミックメディアの量や質にも依存すると思われるので、その点についても検討する予定である。

## 参考文献

- [1] Ralf Steinmetz, J Christian Fritzsche: Abstractions for continuous media programming, computer communications, Vol.15, No.6, July/August, pp.396-402, 1992.
- [2] 大市 津義, 布川 博士, 宮崎 正俊: コミュニケーションのためのメディア記述言語の提案, 第50回全国大会 講演論文集 (6) ハードウェアシステム, Vol.6, pp.165-166, 1995.

```
# .....
media_av_camera<com>
{
  element:
  video_camera<com> obj1;
  microphone<com> obj2;
  mixer<com> obj3;
  stream:
  output out;
  message:
  create()
  {
    obj1.create();
    obj2.create();
    obj3.create();
    obj1.out>>obj3.in1;
    obj2.out>>obj3.in2;
    out:=obj3.out;
  }
  destroy()
  {
    obj1.destroy();
    obj2.destroy();
    obj3.destroy();
  }
  start(){obj1.start();obj2.start();obj3.start();}
  stop(){obj1.stop();obj2.stop();obj3.stop();}
}

media_av_monitor<com>
{
  element:
  distributor<com> obj1;
  display<com> obj2;
  speaker<com> obj3;
  stream:
  input in;
  message:
  create()
  {
    obj1.create();
    obj2.create();
    obj3.create();
    obj1.out1>>obj2.in;
    obj1.out2>>obj3.in;
    in:=obj1.in;
  }
  destroy()
  {
    obj1.destroy();
    obj2.destroy();
    obj3.destroy();
  }
  start(){obj1.start();obj2.start();obj3.start();}
  stop(){obj1.stop();obj2.stop();obj3.stop();}
}

media_av_system<com1,com2>
{
  element:
  av_camera<com1> obj1;
  transporter<com1,com2> obj2;
  av_monitor<com2> obj3;
  message:
  create()
  {
    obj1.create();
    obj2.create();
    obj3.create();
    obj1.out>>obj2.in;
    obj2.out>>obj3.in;
  }
  destroy()
  {
    obj1.destroy();
    obj2.destroy();
    obj3.destroy();
  }
  start(){obj1.start();obj2.start();obj3.start();}
  stop(){obj1.stop();obj2.stop();obj3.stop();}
}

media_tv_telephone<com1,com2>
{
  element:
  av_system<com1,com2> obj1;
  av_system<com2,com1> obj2;
  message:
  create(){obj1.create();obj2.create();}
  destroy(){obj1.destroy();obj2.destroy();}
  start(){obj1.start();obj2.start();}
  stop(){obj1.stop();obj2.stop();}
}

tv_telephone<own,"aaa@xxx.yyy.zzz.jp"> obj;
obj.create();
obj.start();
obj.stop();
obj.destroy();
# .....

```

リスト1 メディア記述言語による記述例