2004－AVM－44　(17)
2004／3／5

分散処理におけるオブジェクト合成のためのマクロ記述方法およびＧＵＩの検討

金子 豊　　竹内 真也　　山本 真　　藤田 欣裕

概要：IP ネットワーク上に分散して存在する機器やコンテンツをオブジェクトとみなし、これらのオブジェクトを自由に組み合わせて利用することができる分散処理システムの検討を進めている。本報告では、オブジェクトを組み合わせて構築したシステムのシステム構成を保存するためのマクロ記述方法を提案する。また、提案するマクロ記述を簡易に作成するために試作した GUI について述べる。提案するマクロ記述方法および GUI を、放送局内での利用を想定した実験システムに実装し、その有効性を確認した。

A Macro Description and Graphical User Interface for Synthesizing Objects in Distributed System

Yutaka KANEKO, Shinya TAKEUCHI, Makoto YAMAMOTO and Yoshihiro FUJITA

Abstract: We are studying a distributed system that is constructed by synthesizing some objects, which are abstracted equipment and contents on IP network. This paper proposes a novel macro description for storing the system configuration for the distributed system. A graphical user interface is also described for creating easily the proposed macro description. To verify the proposed macro description and GUI, they were included in an experimental system for broadcasting program production. The experimental system is also described.

１．はじめに

放送局内では番組制作を行うために様々な機器が使われている。機器間の接続は、映像、音声、制御等の専用線で行われるため、機器の設置場所が限定されている。そのため、番組制作者は、編集室やスタジオなど、特定の場所に番組素材を持ち込んで作業しなければならない。また、作業に必要な機材を一時的に持ち込み、システムにつなぎ込んで使う場合もある。

ＩＴ技術の発達により、放送局内のネットワーク化が進み始めている。しかし、現状のネットワークを利用したシステムは、ピア・ツー・ピアの形態を基本としており、ファイルサーバを中心としたファイル共有ベースのシステムか、端末間のストリーム伝送システムとなっている。そのため、ネットワーク上の様々な機器を共有して利用することができない。また、コンテンツをサーバからサーバに移動させるなど、コンテンツの移動に対しては、ユーザ自身でその所在を管理しなければならない問題もある。

図１はローカル局からセンターの機器を使って番組制作を行う放送局内の作業イメージを示したものである。このようにコンテンツを共有するだけでなく、機器の共有も可能なシステムを実現するには、装置や方式間の相互運用性や、信頼性、セキュリティなど多くの課題がある[1]。

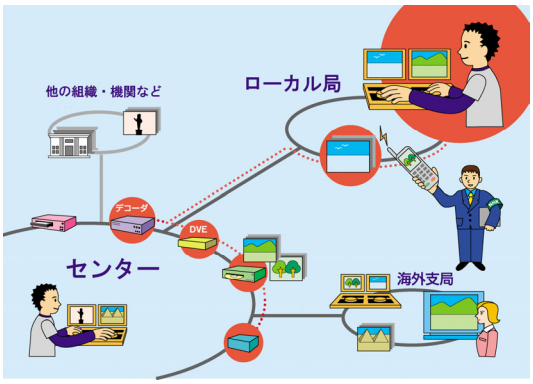


図１ 分散環境による放送局のイメージ

NHK 放送技術研究所　次世代符号化
Advanced Audio and Video Coding, Science & Technical Research Laboratories, NHK

筆者らは、主に番組制作者の利便性を改善することを主眼に、ネットワークを利用した分散処理システムの検討を進めている。主なシステム要件を以下に示す。

- どこからでも機器やコンテンツを扱える
- ネットワークにつなげるだけで使える
- 利用者やコンテンツが移動しても、シームレスに作業が続けられる
- 利用者に技術的な知識がなくても使える

これまでに、機器とコンテンツを抽象化したオブジェクトでシステムを構成するオブジェクトモデルを提案した[2]。また、扱える信号フォーマットが異なる機器間であっても、ネットワーク上にある変換機を自動的に探し出して接続することで、コンテンツの相互運用性を実現する手法を提案した[3]。

本報告では、ユーザがオブジェクトを組み合わせて構築したシステムのシステム構成を保存するための記述方法としてマクロ記述方法を提案する。提案するマクロ記述は、複数の機能を合成したものを１つの機能として、また、複数の機能を用いて処理したコンテンツを１つのコンテンツとして扱うことができる特徴がある。さらに、マクロ記述を簡易に作成するための GUI を試作し、実験システムに実装したのでその概要を述べる。

以下、２章ではオブジェクトモデルによるシステム構成を説明し、３章では提案するマクロ記述方法、４章ではマクロ記述を作成するための GUI について述べる。５章では実装を行った実験システムについて述べる。６章では本研究に関連する文献をあげ、７章でまとめを行う。

２．システム構成

２.１ オブジェクトモデル

提案しているシステムの階層構造を図２に示す。ここでは、コンテンツを抽象化したものをメディアオブジェクト、コンテンツを処理するデバイスやソフトウェアを抽象化したものを機能モジュールとよぶ。編集システムや映像処理システムなど、個々のシステムは、機能モジュールを組み合わせて構成し、そのシステムでメディアオブジェクトの処理を行う。

本システムでは、全てのオブジェクトは一意な名前を持ち、そのオブジェクト名でオブジェクトを特定する。また、全てのオブジェクトはメタデータを読み出すインタフェースを共通に持つ。オブジェクトはそのインタフェースを通してお互いのメタデータを読み出し合い、種別や、性能、信号フォーマットなどを認識し合う[3]。

メディアオブジェクトは機能モジュールから生成され、全てのメディアオブジェクトは生成元である機能モジュールが存在する。これは、メディアオブジェクトが機器から出力される信号という関係に対応する。

２.２ オブジェクト間の接続

オブジェクト間の接続シーケンスを図３に示す。外部から機能モジュールに対してメディアオブジェクトとの接続を要求すると、メディアオブジェクトの生成元である機能モジュールと、接続先の機能モジュールとの間で接続が行われる。このように機能モジュール間の接続を、機能モジュールへのメディアオブジェクトの接続要求という形態とすることで、

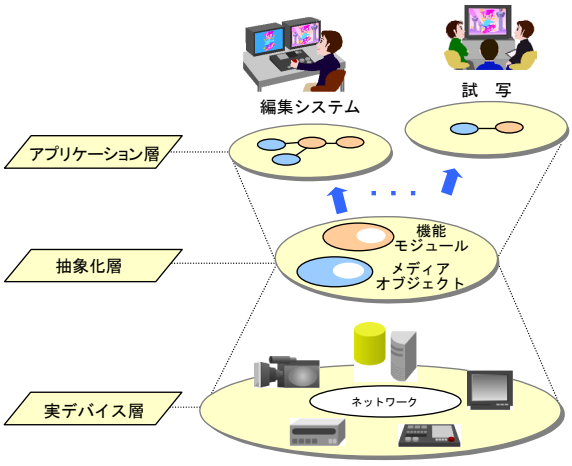


図２ オブジェクトモデルによるシステムの階層構造

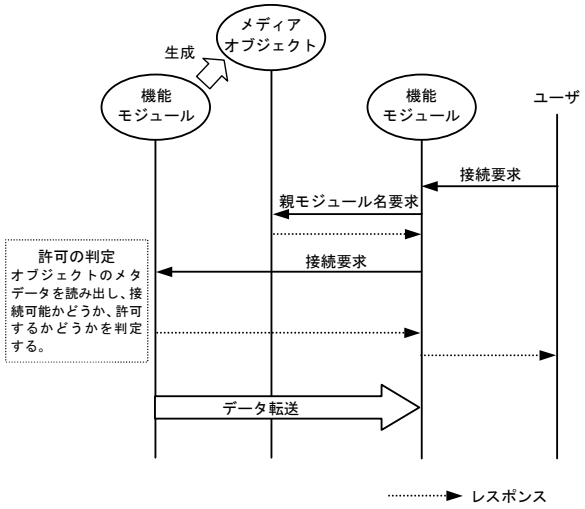


図３ オブジェクト間の接続シーケンス

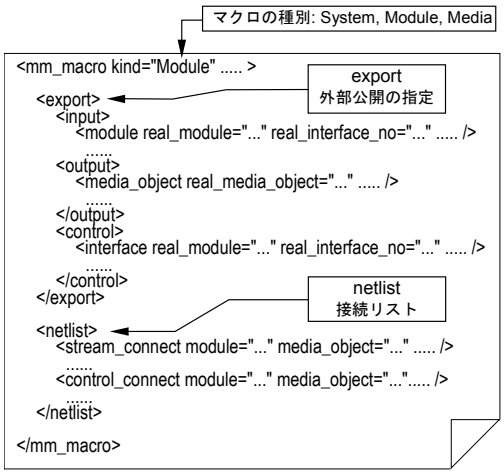


図４ マクロ記述の構造

コンテンツが移動した場合でも、一意なメディアオブジェクト名を使ってアクセスすることが可能となる。

３．マクロ記述

３.１ マクロ記述の概要

オブジェクトを組み合わせて構築したシステムは、使ったメディアオブジェクト名と機能モジュール名を保存しておくことで、同じシステムを再現できる。

ここでは、システム構成を保存するための記述方法としてマクロ記述法を提案する。マクロ記述法は、記述したものが外部からどのように見えるかにより、マクロシステム、マクロモジュール、マクロメディアの３種類に分類される。

マクロシステムは、システム構成を保存するものであり、この記述に従って各オブジェクトに接続要求を出すことで、システムを再構成できる。

マクロモジュールは、複数の機能モジュールを組み合わせたものを単一の機能モジュールと同等に扱えるようにするためのものである。

マクロメディアは、機能モジュールを使ってメディアオブジェクトを処理した結果を単一のメディアオブジェクトとして扱うためのものである。

３.２ マクロ記述の構造

提案するマクロ記述は XML を使って記述する。マクロ記述の構造を図４に示す。

提案するマクロ記述は、<mm_macro>タグで始まり、内部は<netlist>と<export>の２つのエレメントから構成される。マクロの種別は<mm_macro>タグの kind 属性値で指定する。

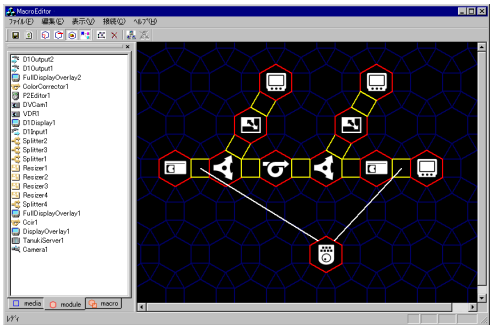


図５ マクロ記述作成用 GUI

<netlist>エレメントは、オブジェクト間の接続リストを示す。<stream_connect>タグおよび<control_connect>タグにより、ストリーム系と制御系それぞれの接続を指定する。接続を指定するオブジェクト名には、他のマクロ記述で作成されたマクロ名を指定することもできる。

マクロ記述の種別がマクロモジュールまたはマクロメディアの場合、<export>エレメントで、そのマクロが外部に公開するオブジェクトを指定する。

４．マクロ記述のためのＧＵＩ

マクロ記述を容易に作成し、接続要求を出すために GUI(Graphical User Interface)を試作した。GUI を作成するにあたり以下の点を考慮した。

- オブジェクトとマクロを区別なく扱える
- 機能モジュールおよびメディアオブジェクトが対等に表示される
- オブジェクト間の配置配線を容易に行える
- システムの構造や接続状態を視覚的に分かりやすく表示する
- ネットワーク上に存在するオブジェクトの数が増えても判別しやすくする

作成した GUI の全体図を図５に示す。左側のウインドウには、ネットワーク上に存在するすべてのオブジェクトと、保存されているマクロ記述ファイルが表示される。右側のウインドウはオブジェクトを配置配線するウインドウである。左側のウインドウで使いたいオブジェクトまたはマクロを選択し、このウインドウ内でシステム構成を指示する。

本 GUI では、機能モジュールとマクロモジュールを６角形、メディアオブジェクトとマクロメディアを４角形として表示し、オブジェクトとマクロは区別なしに扱うことができる。オブジェクトを多角形のブロック状として表示することで、ストリーム系の接続指示はブロックを並べるようにして行うこ

とができる。コントロール系の接続指示は、制御する端点を指示することで行う。２入力、１出力、１制御を持つ機器の表示例を図６に示す。

機器の種別を視覚的に判別できるようにするため、図７に示すピクトグラムを使って表示できるようにした。また、接続中のオブジェクトを反転表示させることで接続状態であることが判別できるようにした。

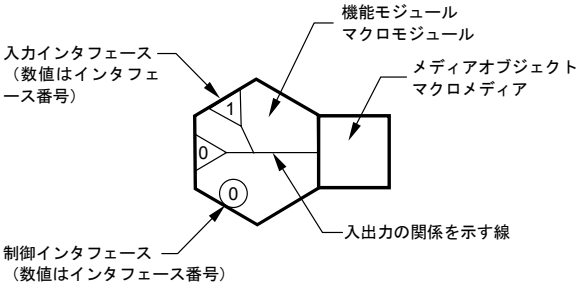


図６ ２入力、１出力、１制御の機器の表示例

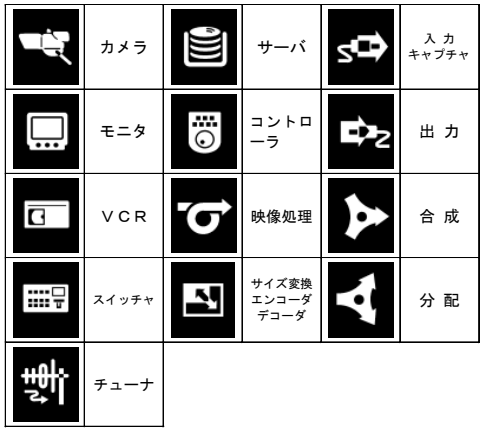


図７ ピクトグラム

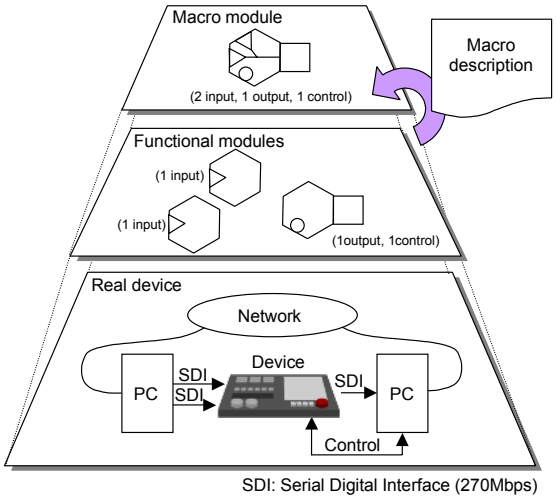


図８ 放送機材のキャプチャ構造

５．試作システム

５.１ 試作システムの概要

提案するマクロ記述および GUI の有効性を確認するため、実験用に試作しているネットワーク利用映像編集システム[1]に実装した。

試作システムはギガイーサネット上に構築されている。オブジェクト間通信および機器の制御系には CORBA/IIOP、ストリーム系には RTP over TCP(または UDP)を実装している。これらのコア部分は Windows、Linux、T-Kernel の各 OS 上で動作する。ストリーム系は、SDTV 非圧縮の映像信号を中心に, HD-Cam, H.263, MPEG-2 TS の信号をリアルタイムで扱うことができる。

映像スイッチャなどの放送用機材は入出力にキャプチャボードを搭載した PC を接続してネットワーク対応機器としている。SDTV 非圧縮映像は約 180Mbps、HD-Cam では約 230Mbps の伝送容量が必要である。リアルタイムでキャプチャおよびネットワーク伝送を行うには、現状の PC では１台で同時に２チャンネル程度までしか処理できない。そのため、入出力が３チャンネル以上ある映像スイッチャやクロマキー装置などの場合には、複数の PC に分けてキャプチャを行う必要がある。本システムでは、図２で示したシステムの階層構造に従い、図８に示すように、入出力の各チャンネルを全て別々の機能モジュールとし、それらをマクロモジュールとして合成することで、GUI 上で１つの機能モジュールと同等に扱う。

５.２ システム検証

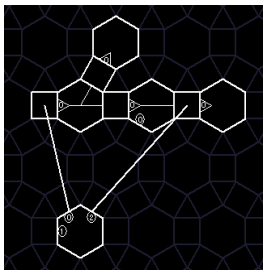
図９に接続例および動作例を示す。(a)は編集システムを構成した例であり、(b)、(c)は GUI の表示画面、(d)はマクロ記述である。 (e)は同じ編集をノート PC で行う場合の例であり、ノート PC 上のモニタおよびコントローラを使って編集作業ができる。ノート PC 上のモニタの画像サイズにするため、画像縮小モジュールをモニタの前に挿入している。(f)は映像スイッチャ、スーパー装置を使って３つのコンテンツを合成した例である。(g)はその合成出力までを１つのマクロメディアとし、H263 エンコーダを通して簡易モニタ上に表示した例である。このときのマクロメディアの記述を(h)に示す。

試作システム上で、以上のようなマクロ記述および GUI を動作させることで、以前に利用したシステムをネットワーク上で即座に構成できることを確

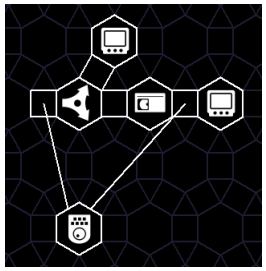
認できた。また、マクロモジュールを用いることで、合成した機器を、単体機器と区別無く利用できることを確認した。さらに、マクロメディアを使うことで、複数のコンテンツの合成結果を１つのコンテンツとして利用できることを確認した。

実装した GUI では、ストリーム系の接続はブロ

(a) 編集システム



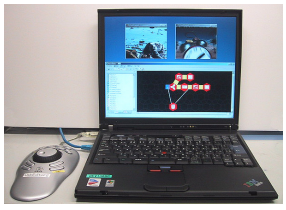
(b) GUI 表示(系統表示)

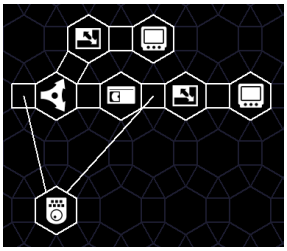


(c) GUI 表示(ピクトグラム表示)

```
<mm_macro name="editor_normal" kind="System" class="System">
  <export>
    .....
  </export>
  <netlist stream_connect_number="4" control_connect_number="2">
    <stream_connect no="0" module="Splitter1" interface_no="0" media_object="racco1" />
    <stream_connect no="1" module="D1Display1" interface_no="0" media_object="m_splitter1" />
    <stream_connect no="2" module="D1VTR" interface_no="0" media_object="m_splitter1" />
    <stream_connect no="3" module="D1Display2" interface_no="0" media_object="m_d1vtr" />
    <control_connect no="0" module="P2Editor1" interface_no="0" media_object="racco1" />
    <control_connect no="1" module="P2Editor1" interface_no="2" media_object="m_d1vtr" />
  </netlist>
</mm_macro>
```

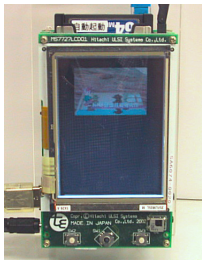
(d) (a)の編集システムのマクロ記述例





(e) 簡易編集システム；ノート PC のモニタとコントローラを使い、(a)と同じ編集システムを構成。モニタの前段に画像縮小モジュールを挿入。

(f) 映像処理の例；映像スイッチャにより２つの映像を合成し、その上にスーパー装置による文字スーパー画像を合成。



(g) H.263 簡易モニタ；(f)の映像をマクロメディアとし、H263 エンコーダを通して T-Engine のモニタに出力。

```
<mm_macro name="tanuki_mnt_sup" kind="Media" class="UnCompVideo">
  <export>
    <input interface_number="0"></input>
    <output interface_number="1" media_object_number="1">
      <media_object no="0" name="tanuki_mnt_sup"
                    real_media_object="file://soft_chromakey1.mmm/m_softChromakey1" />
    </output>
    <control interface_number="0" media_object_number="0"></control>
  </export>
  <netlist stream_connect_number="4" control_connect_number="0">
    <stream_connect no="0" module="file://Swicher.mmm" interface_no="0"
                    media_object="file://tanuki_walking.mmm/m_soft_chromakey2" />
    <stream_connect no="1" module="file://Swicher.mmm" interface_no="1" media_object="mountain2" />
    <stream_connect no="2" module="file://soft_chromakey1.mmm" interface_no="0"
                    media_object="file://Swicher.mmm/m_swer1" />
    <stream_connect no="3" module="file://soft_chromakey1.mmm" interface_no="1" media_object="superA1" />
  </netlist>
</mm_macro>
```

(h) (f)の映像をマクロメディアにした記述例（(g)の入力）

図９ ネットワーク利用映像編集システムの動作例

ックを組み合わせる指示方式、制御系の接続は端点を指示して線で結ぶ方式を用いたが、ブロックによる配線は、線で結ぶ方式に比較して迅速に行えることがわかった。また、ピクトグラムによる表示はあまりメリットが感じられなかった。これは、試作システム上では機能モジュールの数が少なく、このような抽象化した表示を行う必要性がないことが理由の１として考えられる。今後自分の好みのユーザインタフェースを使って、同じ機能の機器であれば統一して制御できる環境が整えば、同種の機器を種類別に表示することのメリットが生かされる可能性もある。今後は、現在、番組制作に使われているユーザインタフェースの使い勝手を考慮しながら、GUIだけでなく、SUI(Solid User Interface)や、認知科学的な観点[4]を含めて使い易いユーザインタフェースを検討していく必要がある。

６．関連研究

以下に、分散処理における機能合成に関連する文献を挙げる。

ソフトウェア開発としての機能合成手法としては、標準出力を標準入力にリダイレクトすることで複数のプロセスを組み合わせる UNIX のパイプライン[5]がある。また、コンポーネント技術の発達により、様々な開発手法が使われているが、その１つとしてDirectShow[6]がある。ここでは、ソース、変換、レンダラの３種類のフィルタを組み合わせることでマルチメディアデータの処理ソフトを開発する。

ネットワーク上の複数の Web サービスを合成する試みとしては、WSFL[7], XLANG[8]などが使われ始めている。また、コンテンツを主体とした機能合成手法として、メディア合成記述[9]なども提案されている。ネットワーク上の機能合成としては、MTRON[10]、Ubiquitous Computing [11]などの発想を根源とするユビキタスコンピューティング、ユビキタスネットワーキングの研究が実装面も含めて活発になっている。また、“ネットワークアーキテクチャを根本から考え直し、新たなフレームワークの構築が必要である” [12]との考えから、STONE[13], Jack-in-the-net[14], Dragon[15]などの提案がされ始めている。

放送分野に関しては、EBU/SMPTE のタスクフォースのレポート[16]が出された後、DS-CC[17], ASCA[18]などネットワーク上で機器を扱う提案も出されたが、標準化という点では現在のところあまり活発ではない。

７．まとめ

ＩＰネットワーク上に分散して存在するオブジェクトを組み合わせて構築したシステムのシステム構成を保存するために、マクロ記述方法の提案をした。また、マクロ記述を容易に制作するための GUI を試作し、ネットワーク利用映像編集システム上に実装し有効性の確認をした。提案したマクロ記述を使うことで、システムをいつでも再構成でき、急なシステム変更や一時的な機器の変更であっても容易に行うことができることを確認した。

参考文献

[1] 山本真, 竹内真也, 金子豊, "ネットワーク利用制作・送出システムに向けて", NHK 技研 R&D, No.80, pp.30-37, 2001

[2] 金子豊, 中川俊夫, 伊藤泰雅, 藤澤俊之, 山田一郎, 南浩樹, 金次保明, 田中豊, "放送局内の分散処理環境におけるコンテンツの相互運用に関する検討," 信学ソ大, D-9-2, 1999

[3] 金子豊, 中川俊夫, 伊藤泰雅, 藤澤俊之, 山田一郎, 南浩樹, 鹿喰善明, 田中豊, "分散処理環境下におけるコンテンツの相互運用に関する検討", 信学技報, OSF2000-75, pp.35-42, 2001

[4] D.A. Norman, “The Psychology of Everyday Things,” 1988 (和訳, “誰のためのデザイン？”, 新曜社, 1990)

[5] M. Ritchie, K. Thompson, "The UNIX Time-sharing System, " Communication of the ACM, Vol. 17, pp.365-375, 1974

[6] Microsoft Corp., "Introduction to DirectShow," 1999

[7] IBM Software Group, "Web Service Flow Language (WSFL 1.0)," 2001

[8] Microsoft Corp., "XLANG Web Services for Business Process Design," 2001

[9] 菊池一彦, 佐藤究, 布川博士, 宮崎正俊, "分散環境におけるメディアの合成記述に関する研究", 情処技報, マルチメディア通信と分散処 80-3, pp.13-18, 1997

[10] 坂村健, "超機能分散システムとしての TRON", システム/制御/情報, Vol.33, NO.1, pp.8-14, 1989

[11] M. Weiser, "Some Computer science issues in Ubiquitous Computing,” Communications of the ACM, Vol.36, No.7, pp.75-84, 1993

[12] 青山友紀, "新世代ネット研究会のねらいと今後の活動", 第二種信学技報, NGN2001-1, pp.1, 2001

[13] 杉田馨, 南正輝, 森川博之, 青山友紀, "ネットワークサービスシンセサイザアーキテクチャの設計と実装", DICOMO2001, pp.109-114, 2001

[14] 藤井慶太, 今田美幸, 松尾真人, 須田達也, "Ja-Net におけるサービス記述方式", 情処全大, 1J-04, 2002

[15] 岩井将行, 中澤仁, 西尾信彦, 徳田英幸, "分散コンポーネントによる即興的アプリケーション構成機構の実現", 情処学論, Vol.43, No.6, pp.1664-1676, 2002

[16] EBU/SMPTE, "Task Force for Harmonized Standards for the Exchange of Programme Material as Bitstreams, Final Report," SMPTE J, Vol.107, No.9, pp.605-815, 1998

[17] NIST ATP, "Digital Studio Command and Control (DS-CC) Ver. 2.1," 1998

[18] "Advanced System Control Architecture, S22.02, System Overview," vol.109, No.5, pp363-440, SMPTE J, 2000