

## メディアオブジェクトのフローに基づくサービス連携方式

大嶋 嘉人

NTT 情報通信研究所  
oshima@isl.ntt.co.jp

田中 譲

北海道大学 工学部  
tanaka@meme.hokudai.ac.jp

本稿では、分散環境上のサービス連携モデル、パッドフローモデルを提案し、その実現機構について述べる。パッドフローモデルはシンセティックメディアシステム IntelligentPad をベースとし、計算機画面上のソフトウェア部品のドラッグ & ドロップ操作に基づき、さまざまなサービスの統一的な利用と、同期 / 非同期の連携を可能にする。また、連携したサービス群は、さらに大きなサービスの部品として利用でき、連携の階層化と高い再利用性を提供する。パッドフローモデルは分散環境におけるさまざまな計算資源を利用した柔軟な処理を実現する。

## Service Coordination Model Based on Pad Flow

YOSHIHITO OSHIMA

NTT Information and Communication  
Systems Laboratories

YUZURU TANAKA

Faculty of Engineering, Hokkaido University

This paper proposes a model, called Pad Flow Model (PFM), for service coordination in open distributed environments and describes its architecture. PFM is a drag & drop-based service coordination model constructed on the IntelligentPad system. PFM facilitates a way of using various services uniformly and of processing coordinated services asynchronously. In addition, coordinated services can be treated as components of larger services, enabling systems to be hierarchically organized and highly reusable. PFM realizes the adaptive processing with various resources in open distributed environments.

### 1 はじめに

近年さまざまな情報メディアが利用可能となって来ている。マルチメディアという言葉で代表される多様なメディアは、計算機を用いた人間の表現力を大幅に向上し、理解を助けることで知的活動の可能性を増大させる。また一方で、計算機とネットワークの発達、普及により、情報の流通や計算資源の共有の活性化、効率化が図られてきている。各計算機独自の情報資源やアプリケーション、サービスなどを、ネットワークを介して遠隔から利用可能になってきた。

企業内などの情報システムの領域では古くから、事務処理の流れのモデル化、ネットワークとアプリケーションの統合システムについて、文献[4]を始めとし、多くの研究がなされてきた。しかしながらこれ

らの多くは、行なわれる処理やアプリケーション、伝達する情報の種類などを固定した、閉じた環境での統合を目指しており、それら要因の多様性、異種性を考慮していない。そのため、現在のように異種性に富み、急激に変化する環境への適用は困難である。

近年では、WWWを利用したIntranetによって、企業内における情報共有などの活性化と効率化が進んできている。WWWはネットワークを意識することなく、分散したさまざまな資源へのアクセス方法を提供する。企業内ばかりでなく、Internetにおいても、個人による情報発信と閲覧を可能とし、ネットワークの急激な成長を引き起こした。しかしながら、WWWは情報の閲覧方法を提供するものであり、処理機能の利用方法を提供するものではない。

例えば異なるホームページで提供される CGI プログラムを組み合わせて利用することはできない。また、CORBA などの分散オブジェクト技術は異なる環境間の処理の相互運用を提供するが、専門知識を持たないエンドユーザが、これらの技術を用いてプログラミングすることは困難である。

人間をこのようなメディアやアプリケーションの差異、ネットワークの存在に対する意識から解放し、さまざまな計算資源を自由に組み合わせて利用可能にすることによって、知的活動はより効率的に、柔軟に行なわれるようになり、その可能性は増大する。

本稿ではシンセティックメディア IntelligentPad 上の処理モデル、バッドフローモデルを提案し、その機構の概要を述べる。バッドフローモデルは各々独立したさまざまなサービスを利用した処理の連携モデルである。可視化された計算機上のソフトウェア部品を、つかまえて動かし、離す、というドラッグ&ドロップに基づき、分散環境でのさまざまなサービスの柔軟な連携を実現する。

計算機とネットワーク、および、あらゆるメディア、アプリケーションを用いたサービスの連携は、

- ・さまざまな処理から成る多人数参加の協調作業
- ・分散した多様な計算資源を用いた個人的活動

という二つの領域の知的活動を支援する。

前者はワークフローなど、企業活動を行なう上での業務、アプリケーションのネットワークを介した連携、統合を指す。後者はエンドユーザが広域ネットワーク上で提供されているさまざまなサービスを自由に組み合わせ、それら一連の処理によって目的を達するという個人的活動を指している。これらは、異なる領域として捉えられているが、バッドフローモデルは同じサービス連携処理として一つの枠組で支援する。

## 2 分散サービスの連携

本稿では計算機上で行なわれる処理を、その実行者以外の外部から何らかの方法で利用できる場合に、それがどのように実現されているかとは無関係にサービスと呼ぶ。例えば、大型 DB の検索も、人間による帳票処理といった処理も、同じようにサービスとして扱う。また、単純なデータのやりとりだけでなく、処理で発生するイベントやその他副作用も含め、あらゆる形で処理はサービスとして提供、利用される。

サービスの連携とは、いくつかのサービスの組合せによる一連の処理に対して、その利用者が何らかの意味を見出したものである。言い換えると、サービ

スの集合全体により、ある目的が達成されるものである。あるサービスの出力データを次のサービスの入力にするといった連携の方法もあれば、あるサービスで発生するイベントを他のサービスの処理開始のきっかけにする、といった制御関係による連携も含まれる。

計算機とそのネットワークから成る分散環境、および、さまざまなメディア、アプリケーションを活用した柔軟なサービスの連携では、以下のような要求を満たす必要がある。

- ・あらゆるサービスを統一的に利用できること。アプリケーションやメディアの差異、あるいはサービスの存在する場所の違いに依らず統一的に利用し、自在に連携できなければいけない。
- ・利用者にとって容易で自然な形で連携を定義できること。高度な専門知識や、スクリプト言語などを用いた記述を要求してはいけない。
- ・連携の内容と構造の再利用が容易であること。効率的な連携利用のためには、サービス連携を一つのサービスとして利用し、階層化させることが可能でなければいけない。
- ・サービスの性格や手順などに応じて非同期に連携を実行できること。処理開始のタイミング、要する時間など、不確定な要素が多いため。

### 2.1 従来技術

#### 分散オブジェクト技術

CORBA[3]、DCOM[2] などの分散オブジェクト技術は、ネットワークの存在を隠蔽し、プラットフォームに依存しないサービスの相互運用性を提供する。しかしながら、これらの技術を用いて分散したサービスを自由に組み合わせたアプリケーションを作成するには、高いプログラミング能力を必要とする。

また、オブジェクト間の通信が手続き呼び出しに基づいた低レベルなものであり、そこで授受されるデータがプリミティブなレベルで厳密に型づけされている。そのため、新しいデータ型や複雑な構造を持った複合型データに対する柔軟性を欠いており、多様なメディアを統一的に扱うことは困難である。

#### Active X

DCOM 上のプログラム連携機構である ActiveX[1]では、HTML 文書中にさまざまなプログラムを埋め込むことができる。ActiveX プログラムはサーバからクライアントへ文書とともにダウンロードされた後、ローカルに実行され、DCOM を介してネットワーク上の他のサービスを呼び出すことができる。ク

クライアントは、サーバの提供するサービスを Web を利用して簡単に入手し、利用することができる。

しかしながら、それらサービスをシステムティックに連携させるには高いプログラミング能力を必要とする。また、このようなクライアント / サーバ型のアーキテクチャでは、クライアントで定義したサービスの連携を一つのサービスとして扱えないため、その再利用や他への提供ができない。

### IntelligentPad

IntelligentPad (以下 IP) は北海道大学工学部田中研究室で研究、開発されているシンセティックメディアシステムである [6]。IP は計算機上で実現されるあらゆるアプリケーション、ツールや、扱われるデータを、メディアオブジェクトパッドとして統一的に扱う。パッドは紙のイメージを持った可視オブジェクトである。一枚のパッドは単純な機能を持つ汎用的な部品として設計、実現される。パッドは貼り合わせることでその機能を合成することができる。

パッドが他のパッドに公開している内部状態やメソッドは、統一的なインターフェイスであるスロットとして表現されている。機能合成は貼り合わせ関係にあるパッド間でメッセージを授受し、スロットのアクセス手続きを起動することで実現される。このパッド間メッセージは三種類のメッセージに標準化されており、あらゆるパッドの組合せを可能としている。

ユーザは必要な機能を持ったパッドを貼り合わせて機能合成することによって、新たなアプリケーションやツール、情報メディアを定義することができる。また、IP は計算機や OS に依存しない環境を提供しており、全てのパッドは異なる計算機間で交換し、利用することが可能である。

しかしながら IP システムでは、共有コピーと呼ばれるパッドの状態の分散共有の機構はあるものの、分散環境上のさまざまな処理を統一的に扱って連携させるサービス連携の機構がない。

## 3 Pad Flow Model

IP システム上のメディアオブジェクト、パッドは、機能や情報を外在化し、その内容への統一的なアクセス方法を提供する。異種の情報からなる複合情報は、単純な情報を持つパッドの合成で表現され、それは一枚のパッドと同様に扱える。サービス間のデータ授受はパッドを媒体とすることで、柔軟なものとなる。またパッドは直接操作可能な可視オブジェクトであるから、情報の移動をパッドの移動で実現することは、優れたエンドユーザインターフェイスを提供する。

さらに IP システムは、OS に依存しない環境を提供すると同時に、外部のプロセスを IP システム内に取り込む機構を用意している。サービスをパッドによって実現することで、あらゆる処理を統一的に扱い、また異環境間で相互運用することが可能になる。

以上の点から本稿では、IP システムをベースとして、パッドのドラッグ & ドロップに基づく分散環境におけるサービス連携モデル、パッドフローモデルを提案する。先述のサービス連携の要件を満たすべく、以下の要素技術を加え、これを実現する。

- ・柔軟で、容易に利用可能なインターフェイスを持ったサービスの統一モデルの規定。
- ・ネットワークを意識せずに、あらゆるパッド上にパッドを貼り付ける機構。つまり、あるパッドの上から別のパッドの上へとパッドを移動させるための機構。
- ・それらパッドの移動をパッドの組合せによってシステムティックに容易に記述、実行する機構。
- ・移動するパッドと、それを受け入れるサービスやホスト、環境の双方向の保護を実現する部品

これらの技術、機構により、パッドフローモデルは分散環境上のさまざまなサービスを統一的に扱い、同期 / 非同期に連携させた柔軟な処理を実現する。また、サービスの部品化と階層化を可能とし、高い再利用性を提供する。

### 3.1 サービスのモデル化

サービスは計算機上で行なわれる処理であり、その結果生じるデータだけでなく、発生するイベント、副作用を含め、何らかの形で外部から利用できるものである。また、サービスは実際にどのように実現されているかとは無関係に提供利用される。ここでサービスを統一的に扱うために、その処理内容とは独立に、次のようにインターフェイスを定めた。

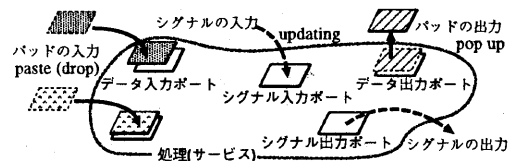


図 1: サービスのモデル

サービスとして利用できるパッドをサービスパッドと呼ぶ。サービスパッドはドロップされたパッドを入力データとして受けとり何らかの処理を行なった後、その結果としてパッドを生成し、ポップアップし

て出力するものである。サービスパッドを構成するパッドの中で、入力データパッドのドロップ先となるパッドを入力ポートと呼ぶ。また出力データのパッドがポップアップされるパッドを出力ポートと呼ぶ。

動的な連携や同期制御のために、サービスは状態の変化やイベントの発生を伝えるシグナルを送受するものとする。シグナル入力ポートはシグナルの入力部で、処理の開始、停止などの制御の一つに対応する。このポートを状態更新させることでその制御が実行される。シグナル出力ポートはシグナルの出力部で、処理終了やエラー発生など一つのイベントに対応し、そのイベントの発生時に状態更新して外部に通知する。

サービスパッドをこれら四種類のポートを任意の数だけ持ち、それらだけを通じて外部から処理機能を利用できる任意の処理と定義する。入力データを受けとったサービスパッドは他とは独立に、非同期に処理を実行できる。同期の必要がある場合にはシグナルを利用すればよい。また IP システムでは複合パッドも一枚のパッドと区別せずに扱えるため、サービスパッドは任意の種類、数、構造のパッドを入出力データとする柔軟なインターフェイスを持つものとなる。

IP システム外部のプログラム、サービスは、そのインタフェイスとなるパッド、ブロッカーパッドを定義することで、容易に IP システム内部に取り入れることができる。例えば、外部の大規模データベースや、CGI プログラムなどのプロキシーが考えられる。また、ユーザによる処理もサービスの一つとして同様に扱うことができる。例えば、処理依頼パッドを受けとったユーザが、表計算ツールパッドを用いて処理を行ない、結果の書類パッドを返すようなものである。サービスパッドのインターフェイスを持たせることにより、ソフトウェア、ユーザによるあらゆる処理をサービスとして統一的に扱うことが可能となる。

### 3.2 フローの実現機構

サービスの利用はサービスパッドのポートへのパッドの貼り付けとはぎとりに基づいて行なわれる。このようなパッドの移動をパッドフローと呼ぶ。それに対し、シグナルの伝達をシグナルフローと呼ぶ。ネットワーク上に分散したサービスの連携を行なうためには、このフローを自動的に、しかもネットワークを越えて実行する機構が必要である。

メタパッドとは、あらゆるパッドの対の間でネットワークを越えてパッドやシグナルを転送、伝搬する機構をパッドとして部品化したものである。メタパッドは既に存在している任意のパッドを元に生成され、

そのパッドと特別な関係を持つパッドである。生成元のパッドを meta に対する referent と呼ぶ。meta、referent はどちらかに特定のイベントが生じた際に通信して、パッドやシグナルを転送する。

meta-referent 間で送受するパッド、シグナルの別と、その転送の方向の別によって、次の四種類のメタパッドを定義する。

**InputMetaPad** meta にパッドが貼り付けられると、それを転送して referent の上に貼り付ける。

**OutputMetaPad** referent にパッドが貼り付けられると、それを転送して meta の上に貼り付ける。

**SignalInMetaPad** meta が状態更新すると、referent も状態更新する。

**SignalOutMetaPad** referent が状態更新すると、meta も状態更新する。

生成された meta とその referent は、お互いに相手を指し示すポインタとして、PadAddress を内部に保持している。これはそのホスト内でのパッドの ID と、ホストのアドレスを組み合わせたもので、ネットワーク上でグローバルなパッドの識別子である。meta と referent のどちらか、もしくは両方が異なるホストに送られても、PadAddress を更新して参照関係を保ち、転送機能を実現する。メタパッドにより分散環境を意識しないあらゆるパッド間でのパッドやシグナルのフローを実現できる。

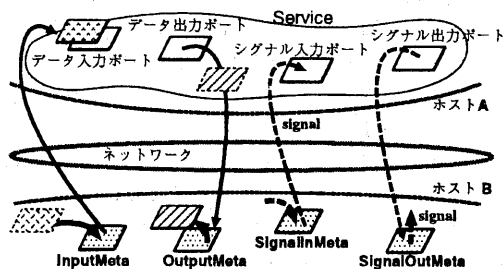


図 2: メタパッドによるサービス利用

サービスパッドのポートはその種類に応じてアクセス方法が決まっており、それは、メタパッドの種類に一对一に対応している。サービスパッドのポートから生成した適切な種類のメタパッドは、実際のポートとして全く同様に利用できる。全てのポートのメタパッドを手元に用意しておくことで、遠隔にあるサービスパッドの処理機能を手元にあるのと同じに利用できる。メタパッドの配布と入手は、WWW を利用することで容易に実現できる。

### 3.3 サービス連携の記述、実行機構

サービスの独立性を保つためには、各サービスを直接ではなく、間接的に結び付ける機構が必要である。また、あらゆるサービスの連携を可能とするためには、そのような機構はサービスの処理内容には立ち入らずにフローの構造のみを記述する必要がある。フロー記述パッドはサービスパッドの処理内容や入出力データとは独立にポートを結び付け、サービス連携のフローを記述する機能を部品化したパッドである。

フロー記述パッドは任意の種類のパッドを入出力し、サービスパッド間のデータの入出力関係や、シグナルフローによる制御関係を記述する。フローの形態によって数種類のパッドを定義した。最も基本となる部品はゲート機能を持つキューとして働く PadQueue である。これは内部に FIFO のキューを持ち、入力されたパッドをキューに付け加え、取り出しのトリガーシグナルの入力により、先頭のパッドを出力するものである。この他、フローの合流、複製や、分岐などを記述するパッドを定義した。

また、分岐フローにおける分岐先がセレクトシグナルで決定されるように、フローとフローは互いに作用しあう。このようなフローの関係に注目し、「データフローからのシグナルフローの派生」などのフロー定義をパッドとして部品化した。

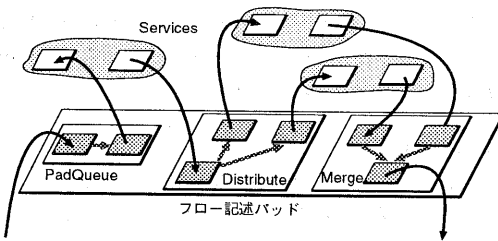


図 3: サービス連携

サービス間のフロー、つまり、サービス連携は、これらフロー記述パッドとサービスのポートのメタパッドを貼り合わせ、機能合成することで容易に定義、実行することができる。その際にはスクリプティングなどの必要はない。サービスパッドをフロー記述パッドで間接的に結び付けることにより、各々の処理の独立性を保ちながら連携させることが可能となる。フロー記述パッドは任意の種類のパッドを入出力するため、特定のサービスに依存することなく、あらゆるサービスを連携させることができる。

フロー記述パッドの入出力部はサービスパッドと同様に四種類のポートである。そのため、フロー記述

パッドで記述されたサービス連携は一つのサービスパッドとして扱うことができる。これにより、連携の部品化と階層化とが可能となり、連携の内容、構造はともに高い再利用性を持ち、より大きなサービス連携のボトムアップな構築も可能となる。

### 3.4 フローパッドとサービスパッドの保護機構

フローするパッドは実際に処理されるサービスに入力される前に他のホストに転送されるかもしれない。しかしその内容は目的のサービス以外では隠蔽できる必要がある。また、パッドは単なるデータではなく機能部品であるから、フローするパッドは何らかの処理を行なうプログラムである可能性もある。そのためフローしてくるパッドから、それを受け入れるサービスやホストも保護する必要がある。

EnvelopePad は封筒のメタファを用いたパッドであり、上に貼られているパッド全てを外部から隠すカバーを持つ。カバーを閉じた状態では中のパッドは外部から見えないだけでなく、一切の機能を停止している。EnvelopePad は開封条件を保持しており、これが満たされた時にカバーを開いて、隠していたパッドを公開し、さらにそれらを活性化させる。条件にはホスト名の一致などを用いることができる。この他 EnvelopePad はエラー発生時に自動的に元のホストに戻るといった機能を持つ。

ホストはまず EnvelopePad を受け取り、その差し出し人のホストなどの情報を元に封筒を開いて処理するかどうかを決定することができる。また、ユーザはパッドを EnvelopePad に入れて送ることによって、内容の機密を図ることができる。

## 4 適用例

### フォームフロー・システム

企業内の部署内 / 間、また企業間で、書類はその形を変化させながら流れる。フォームフロー・システムは書類 (フォーム) の流れに着目し、仕事の進行をモデル化し、自動的な実行を可能とする [5]。

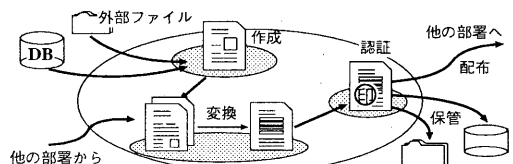


図 4: フォームフロー・システム

複合文書であるフォームはパッドの合成で表される。あらゆる機能、情報は統一的にパッドとして実現

されるため、あらゆるメディアを含んだフォームを実現できる。また、データベースなどの外部オブジェクトにアクセスするパッドを含ませることにより、静的な情報だけでなく動的に変化する情報をフォーム中に含ませることが可能である。

フォームの変換、認証などの処理はサービスパッドとして実現される。これにより、用いられるアプリケーションやツールを限定することなく、さまざまな処理を統一的に扱うことができる。これらのサービスをパッドフローにより連携させることで、一連の業務の定義、実行できる。

### フロー・プログラムの遠隔実行

ユーザによって定義されたサービスの連携を記述したフロー記述パッドは、一種のプログラムと見なすことができる。これをフロー・プログラムと呼ぶことにする。フロー・プログラムをサービスパッドの存在するホストに転送してそこで実行させ、結果のパッドだけを手元へとフローさせて入手することをフロー・プログラムの遠隔実行と呼ぶ。これはフローを記述したパッドをフローさせるという、パッドフローの枠組を二重に適用した処理モデルである。

フローするフロー・プログラムにはフロー記述パッドだけではなく、任意の種類のパッドを含ませることができる。ユーザは自身が持っているサービスパッドをフロー・プログラムとともにフローさせ、遠隔のホストのサービスパッドと連携させて利用することができる。フロー・プログラム、それを受けとるホスト、双方の保護は EnvelopePad によって実現できる。

例えば、興味深い研究を行なっているいくつかの大学のホストの膨大な量のテクニカルレポート集の中から興味深いものだけを選別し、手元にかき集めるといった情報検索を考える。この場合、ユーザが定義した、フィルタリングを行なうパッドを併せて大学のホストに転送し、実行させることによって、ユーザの嗜好を反映したレポートのみを集めるといった処理が可能となる。複数のサーバにまたがったフロー・プログラムは、各々のサーバ用のフロー・プログラムを順次フローさせ、実行させることによって実現できる。

遠隔実行はホスト間の断続的な接続と効率的な通信を可能とする。通常のクライアント / サーバ型の実行では、それらの間の通信がフロー・プログラムの終了まで非同期に発生するため、常にネットワークに接続していなければならない。遠隔実行ではサービスパッドとフロー・プログラムの間のフローはサーバ内でローカルに行なわれる。そのため、ホスト間の通信

はフロー・プログラムと結果の転送時のみ必要であり、それ以外の間は接続を断つことができる。また、処理の実行と管理はサービスの存在するホストが行なうため、ユーザ側の負担が軽減される。

### 5 おわりに

本稿ではサービス連携モデルであるパッドフローモデルを提案し、その実現機構について述べた。サービスのモデルは処理を統一的に扱うことを可能とし、アプリケーションやメディアの異種性、多様性といった要求に応える。また、サービスの遠隔利用の機構、連携を部品の組合せにより記述 / 実行する機構は、分散環境上のさまざまなサービスを連携させた、柔軟で、高い再利用性を持つ処理を実現する。

本モデルは複数の参加者による協調作業、さまざまな資源を活用した個人的活動、という分散環境における二つの領域の知的活動を一つの枠組で支援する。分散したさまざまな資源を利用した処理を、より多様で柔軟なものとするだけでなく、その可能性を広げ、新たなパラダイムを提供し得るものであると考える。

今後の課題としては、ホスト間でのパッドのライブラリの同一性がある。その解決方法としては、内部の機能がスクリプト言語で定義される手続きパッドを用いる方法が考えられる。また、必要なサービスをどのように効率的に見つけ、入手するか、という問題がある。これは、同じパッドフローの枠組を用い、サービス検索をやはりサービスパッドとして実現し、利用することで解決できると考える。

### 参考文献

- [1] ActiveX Controls Overview, 1996. <http://www.microsoft.com/intdev/controls/controls-f.htm>.
- [2] Component Object Model Specification, 1996. <http://www.microsoft.com/INTDEV/SDK/DOCS/COM/COM1598C.ZIP>.
- [3] Siegel, J. *CORBA Fundamentals and Programming*. Wiley, 1996.
- [4] Zloof, M. M. "Office-by-Example: A business language that unifies data and word processing and electronic mail". *IBM System Journal*, Vol. 21, No. 3, pp. 272-305, 1982.
- [5] 中川光紀, 牧野学, 田中謙. 「フォーム・フロー・アーキテクチャ」. 1994年 情報学シンポジウム講演論文集, pp. 209-218. 情報処理学会, 1994.
- [6] 長崎祥, 田中謙. 「シンセティック・メディアシステム: IntelligentPad」. コンピュータソフトウェア, Vol. 11, No. 1, pp. 36-48, 1994.