

# 柔軟な組織間連携を実現するインターワークフロー支援機構

勝 間 田 仁<sup>†</sup> 速 水 治 夫<sup>†</sup>

ワークフロー管理システムが組織に導入され、BPR ( Business Process Reengineering ) の効果があげられている。一方、近年のインターネットの普及は、企業間における電子商取引の分野に大きく影響をもたらしている。特に、組織間の業務プロセスを各組織のワークフロープロセスの相互接続により運用し、業務の効率化とコスト削減が実現されている。このような、ワークフロープロセスの相互連携をインターワークフローと呼ぶ。インターワークフローの実現は、すでに各組織に導入されている異種ワークフロー管理システムを連携した上に業務連携を定義して実行することになる。このインターワークフローの実現方式は、企業間電子商取引システムの構築・運用の支援機構として期待できる。さらに、インターネット上の動的でオープンな環境では、企業間電子商取引に参入する組織の枠にとらわれない組織間連携が要求され、迅速に対応できる柔軟性が求められる。本論文では、インターワークフローとして実現された組織間にまたがった業務連携を動的に変更することを目的とし、その運用を実現する支援機構を提案する。さらに、本支援機構の実現性を確認するために試作したシステムについても述べる。本支援機構によって実現されるインターワークフローの動的な変更方式は、組織間連携を部分的に追加、解消、変更するといった特徴を持っている。

## Interworkflow Support System with Realizing Flexible Cooperation among Organizations

MASASHI KATSUMATA<sup>†</sup> and HARUO HAYAMI<sup>†</sup>

Workflow management systems are being introduced in many organizations to automate the business process. The next step for introducing workflow management system into organization is to realize the cooperation among multiple organizations by interconnecting the existing workflow process in organization. We call it interworkflow. On the other hand, business-to-business electronic commerce infrastructure has been prepared with the growth of the Internet. And then, without considering the scale of organization, the realization for interconnection among organizations in any scale is required. In this situation, we think that some mechanisms for realizing the rapid and flexible interconnection among organization are important. In this paper, we propose the support mechanism with realizing dynamic modification for interworkflow and describe the developed prototype system to confirm its realization. This prototype system is modeled as interworkflow process for business to consumer electronic commerce system as an example. The future of dynamic modification for interworkflow has the cooperation among organizations is partially executed by addition, cancellation and change.

### 1. はじめに

複数の担当者によって遂行される業務の自動化と管理を目的として、ワークフロー管理システムが組織に導入され、BPR ( Business Process Reengineering ) の実績があげられている<sup>1)</sup>。一方、近年のインターネットの普及は、企業間 ( business to business : 以後、B2B と記す ) における電子商取引の分野に大きく影響をもたらしている。特に、組織間にまたがった業務プロセス

を各組織のワークフロープロセスの相互接続によって運用し、業務の効率化とコスト削減の達成を実現している<sup>2)</sup>。このようなワークフロープロセスの相互連携をインターワークフロー ( interworkflow ) と呼ぶ<sup>3)</sup>。インターワークフローの実現は、すでに各組織に導入されている異種ワークフロー管理システムを連携した上に業務連携を定義して実行することになる<sup>4)</sup>。この実現の第1の課題は、ワークフロー管理システムの国際的な標準化団体である WfMC ( Workflow Management Coalition ) が標準化を進めている異種ワークフロー管理システム間の相互接続プロトコルによって実現される<sup>5),6)</sup>。その上で、業務連携の取り決めを行い、

<sup>†</sup> 神奈川工科大学情報工学科  
Department of Information and Computer Sciences,  
Kanagawa Institute of Technology

それによって各ワークフローのプロセス定義を行う。そのプロセス定義データに従って、ワークフローサーバが連携し、インターワークフローとして運用されることになる。このインターワークフローを構築するプロセス定義の支援として、複数の組織にまたがる業務連携を一括して記述し、各組織ごとのワークフロー管理システム上のプロセス定義データに展開することを目的とした技術開発がインターワークフローに向けた第2の課題として実現されている<sup>4),7)</sup>。このようなインターワークフローの実現方式は、B2B電子商取引システムの構築・運用の支援機構として期待できる。

さらに、インターネット上の動的でオープンな環境において、B2B電子商取引システムを構築・運用する基盤が整えられてくると、B2B電子商取引に参入する組織の枠がよりいっそう拡大されることが期待される。そうなると、組織の規模にとらわれることなく、あらゆる規模の組織間における業務連携の実現が要求される。特に、電子商取引の分野ではサービスの変化が厳しいので、組織内部の効率化に加え、他の組織との連携に迅速に対応できる柔軟性が求められる。このような状況では、より良い組織間連携を迅速かつ柔軟に行い、組織間の情報共有を効率良く行うための仕組みの実現が重要になると考えられる。

本論文では、インターワークフローとして実現される複数の組織にまたがった業務連携の一部を動的に変更することを目的とし、その運用を実現する支援機構について提案する。さらに、この支援機構の実現性を確認するために試作したシステムについて述べる。本支援機構によって実現されるインターワークフローの動的な変更方式は、組織間の連携関係を部分的に追加、解消、変更するといった特徴を持っている。B2B電子商取引システムの運用基盤が今後ますます発展することを想定し、本試作システムは、B2C (business to consumer) サービスを提供する組織が他の組織と柔軟にB2Bの連携を行うシステムを実現している。この試作システムでは、エンドユーザからの要求に応じた組織間連携を実現することをインターワークフローの動的な変更が生じる一例としている。本試作システムは、提案した支援機構により、インターワークフローを形成する組織間の連携を迅速かつ柔軟に変更することを実現している。

以下、2章では、インターワークフローの要件について説明する。3章では、組織間連携モデルの分類について述べ、4章で、柔軟な組織間連携を実現するための要件について述べる。5章では、柔軟な組織間連携を実現する支援機構と運用イメージについて説明し、

6章で、本試作システムの目的と概要について述べる。7章では、他の関連研究との比較をし、最後に、8章でまとめを行う。

## 2. インターワークフローの要件

### 2.1 異種ワークフロー管理システムの相互接続

インターワークフローの実現方式としては、同種のマルチサーバ対応ワークフロー管理システムによる方式、異種ワークフロー管理システム間の相互接続方式がある<sup>4)</sup>。特に、広範な組織においてインターワークフローを実現するには、異種ワークフロー管理システム間の相互接続方式が有効である。ワークフローの国際的標準化団体であるWFMCでは、異種ワークフロー管理システム間の相互接続 (interoperability) のためのインタフェースを定め、相互運用モデルとして、引継型、請負型、並行同期型の3つのモデル<sup>5)</sup>をあげている (図1参照)。

#### (1) 引継型

一方のワークフロープロセスが他のワークフロープロセスのインスタンスを生成し業務を引き継ぐ。

#### (2) 請負型

一方のワークフロープロセスが他のワークフロープロセスのインスタンスを生成し、その業務の一部を代行させ、その結果報告を受けて最初のプロセスは業務を続行する。

#### (3) 並行同期型

独立に走行しているワークフロープロセスのインスタンスが相互に同期をとりながら、情報を交換することによって、対等に業務を連携する。

### 2.2 連携プロセスのモデル化

各ワークフロー管理システムは、異なるワークフ

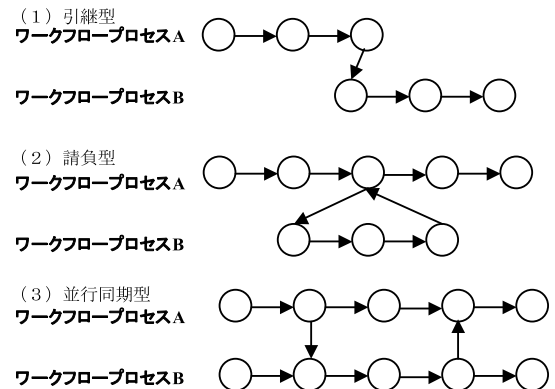


図1 ワークフロープロセス間連携モデル  
Fig.1 Models of workflow cooperation.

ロープロセス定義体系を用いているため、あるワークフロー管理システムで定義したワークフローを他のワークフロー管理システムでは利用できないという問題がある。

これとは別に、個々の組織は、必ずしも組織内部の情報を公開できないという制約があり、各組織は他の組織から必要以上に組織内部の業務プロセスに干渉を受けることなく、独立に物事を決定する権限を持っている。このように独立して物事を決定する権限を持っていることを組織の独立性と呼ぶ<sup>(8)</sup>。

これらの問題から、インターワークフローのモデル化においては、組織間にまたがった業務プロセスの連携部を厳密に規定する必要がある。そこで、各組織が独立ではなく、組織間の協調により1カ所でインターワークフロープロセスを記述する方式が有効であると考えられるが、これは組織の独立性と相反する要求であり、インターワークフローを構築するうえでのジレンマとなる。このような問題を解決するのに、文献<sup>(3)</sup>、<sup>(4)</sup>、<sup>(7)</sup>では、第1段階として、各組織のワークフロープロセスの内部をブラックボックスとして、相互連携のみに着目して記述する。第2段階として、その結果を各組織が持ちかえり、ブラックボックスとしたところに、内部プロセスを追加してホワイトボックス化する。すなわち、インターワークフロープロセス記述の階層化方式によって、組織間にまたがった業務連携プロセスの定義を行っている。これは、クライアント・サーバの形態を有したワークフロー管理システムに適した方式である。

### 2.3 異種ワークフロー管理システム間の相互連携プロトコルの標準化動向

インターネット上でのワークフロー管理システム間の接続プロトコルとして、SWAP (Simple Workflow Access Protocol)<sup>(9)~(11)</sup>が提案され、それを基に、WfMCによって標準化が進められているWf-XMLがある<sup>(6), (12)</sup>。Wf-XMLは、インターネット上で広範に利用されているHTTPプロトコルをベースとし、組織間でやりとりされる命令やデータ形式をXML (eXtensible Markup Language)としている。組織間でやりとりされるデータ形式がXMLによって表現されているために、データを組織内部で蓄積し、他のシステムで再利用することも容易になるとされている<sup>(13)</sup>。

## 3. 組織間連携モデルの分類

### 3.1 定型タイプの組織間連携

組織間における業務連携プロセスをインターワーク

フロープロセスとして運用すると、組織間において繰り返し行われる定型的業務プロセスの効率を向上させることが可能となる。現在、B2B電子商取引サービスにおいて成功している一例として、ORM (Operating Resource Management) の自動化があげられる。ORMの特徴としては、組織における物品の購入や保守サービスなどがあげられる。これらのORM活動は、そのつど、見積書の作成依頼をし、決済時には、電話あるいはファックスによって行うのが通常である。このような組織間で発生する業務の連携は、頻繁に発生し、かつ、それぞれが不定期に要求される。このような業務をインターワークフローとして実現すると、組織における業務の迅速化とそれに関わる費用の削減が実現される。このようなインターワークフローの実現は実際に行われてきている(文献<sup>(2)</sup>、<sup>(14)</sup>、<sup>(15)</sup>)を参照)。

### 3.2 非定型タイプの組織間連携

本論文では、必要に応じてそのつど、組織間の業務連携を行うタイプを非定型タイプの組織間連携とする。非定型タイプの組織間連携が実現されると、定型タイプの業務連携プロセスのように、組織間で仕事の依頼関係が固定されておらず、必要に応じて、他の組織との連携を行い、既存の運用されている業務プロセスに対し、付加的な効果をもたらすことが可能となる。本論文では、組織間の連携関係の追加、解消、変更という概念を導入し、組織間の柔軟な連携を実現するアプローチを採用した。

#### (1) 組織間の連携関係の追加

ワークフロープロセスのアクティビティにおいて、自発的な判断により、部分的に他組織のワークフロープロセスのアクティビティと連携を行う。連携を行った結果は、既存のプロセスに反映される。図2で示される組織Aのワークフロープロセスと組織Bのワークフロープロセスの連携関係がすでにあった場合、組織BのワークフロープロセスにおけるアクティビティB2の段階で、次の処理をアクティビティB3とするのではなく、組織Cのワークフロープロセスと連携を行うということになる。図2の(b)では、

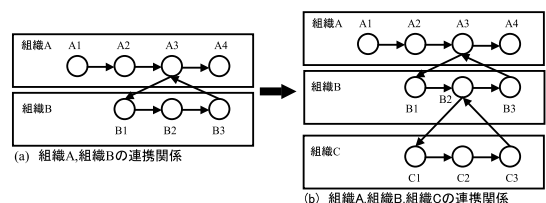


図2 組織間の連携関係の追加

Fig. 2 Addition of cooperation among organizations.

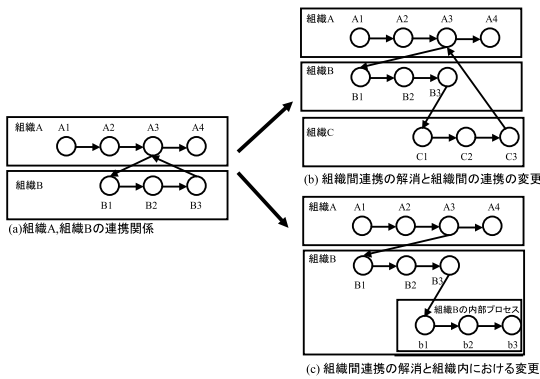


図3 組織間の連携関係の解消と変更

Fig. 3 Cancellation and modification of cooperation among organizations.

組織 B と組織 C との新たな連携関係が、請負型の連携関係として追加される。その結果、組織 B のワークフロープロセスと組織 C のワークフロープロセスとの連携が終了したあとに、組織 B におけるワークフロープロセスのアクティビティ B2 から次のアクティビティ B3 へと処理が進められる。

#### (2) 組織間の連携関係の解消

ワークフロープロセスのアクティビティの自発的な判断により、既存の業務フローを部分的に解消する。

連携関係の解消は、図 3 の (a) で示される組織 B のアクティビティ B3 において、組織 A のワークフロープロセスのアクティビティ A3 との連携関係を行わないような場合を示す。

#### (3) 組織間の連携関係の変更

すでに運用されているタスクに影響を与えないで、業務フローの変更を行う。連携関係の変更は、変更が組織間で行われるか、組織内で行われるかによって分けられる。

##### (a) 組織間における変更

組織間における変更は、図 3 の (b) で示される組織 B のワークフロープロセスにおけるアクティビティ B3 と組織 A のワークフロープロセスにおけるアクティビティ A3 の連携が解消され、組織 B のワークフロープロセスのアクティビティ B3 におけるタスクの遂行が組織 C のワークフロープロセスのアクティビティ C1 との連携により継続されるような場合である。

##### (b) 組織内における変更

変更が組織内部で行われる場合は、図 3 の (C) で示されるように、組織 B のワークフロープロセスにおけるアクティビティ B3 と組織 A のワークフロープロセスにおけるアクティビティ A3 の組織間の連携が部分的に解消され、組織 B の内部にあるサブプロセスとの連携により、タスクの処理が継続されるような場合である。

組織間連携の基盤が整備され、B2B 電子商取引へ参入する組織の枠が拡大されると、より良い連携関係の価値を導くために、多くの組織の中から適切な組織を選択することが望まれる。その結果、既存の処理をその状況にあった適切な外部の組織へ委託することで、組織としての処理の効率化をあげることが可能となる。また、既存の外部との取引を一時的にとりやめ、内部処理として済ますことも可能となる。このように、組織間における連携が迅速かつ柔軟に行われると、電子商取引において、柔軟なサービスを提供することが可能になると期待できる。

## 4. 柔軟な組織間連携を実現する支援機構

### 4.1 柔軟な組織間連携を実現する要件

現在、WfMC が標準化を進めている異種ワークフロー管理システム間の相互接続プロトコル Wf-XML は、ワークフロー管理システムの機構として、ワークフローの単位となる仕事の実行者と依頼者がネットワーク上に散在しており、これらの間を連携オペレーションによって結ぶというモデルをとっている。このようなモデルでは、ワークフローの実行者が提供している見積、注文などの Wf-XML の連携インタフェースをワークフローの依頼者側で利用することになる。これまでのワークフロー管理システムに Wf-XML のインタフェースを利用できるように拡張すれば、インターネット上で組織間の連携を行うことが可能である。これは、ワークフローの実行者が提供しているインタフェースを連携業務における取り決めとし、組織内部のワークフローの仕事の単位として定義を行うことになる。しかし、ワークフローの仕事の単位レベルでは、その処理、すなわち、連携先の組織を変更することは不可能であり、ワークフローの定義を変更するところから始めなければならない。あらかじめ、連携する組織が複数と定められていれば、ワークフロー定義を行うには可能であるが、不特定多数の組織の中から連携先を決定する場合を想定すると容易ではない。また、新たに連携する組織との取り決めを行う必要性が生じる。さらに、インターネット上の広域な分散環境で展

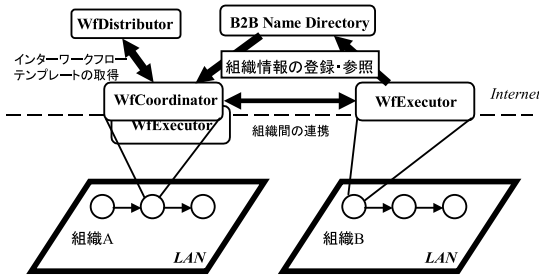


図 4 支援機構の構成

Fig. 4 Configuration of support system.

開される業務連携を実現する際に、組織内部あるいは組織間における業務の再編成や新たな業務を他の組織に対し引継型や請負型の連携業務として運用する要求も考えられる。このような要求に迅速かつ柔軟に対応するためには、組織独自の判断のもとで組織間連携を行うことを支援する機構が必要となる。

本論文では、Wf-XML を利用するワークフローの実行者とワークフローの依頼者からなる機構モデルと区別するために、組織間連携のプロセスの定義、構築、実行する主体を WfCoordinator と呼び、それと連携を行い、タスクを実行する主体を WfExecutor と呼ぶ。WfCoordinator は、B2B Name Directory にネットワーク上の WfExecutor の物理的なアドレスを問い合わせ、インターネット上で相互に連携を行うことで、サーバを介さずにピア・ツー・ピア (peer to peer) 通信を行うことが可能となる。WfCoordinator は、運用する組織間連携定義データを WfDistributor からダウンロードする (図 4 参照)。

#### 4.2 組織間の連携関係を記述するモデリング支援

組織間の連携を行うためには、その組織間における連携をどのような手順で行い、どのようなメッセージを交換するかといった連携関係のモデリングを行うための支援が必要である。この連携のための手順やメッセージのフォーマットを各組織が連携先ごとに定義するとすると、連携先ごとの定義データが必要になり、その数にともなって、連携方法を構築する必要がある。そこで、ある同業種ごとに連携のための手順や連携に要するメッセージのフォーマットを統一する。同業種ごとに統一された組織間連携定義データを利用することにより、連携先組織ごとに対応した個別の連携方法をとることがなくなる。また、連携の際に、必要となるメッセージのフォーマットが統一されているので、要求される項目に、各組織の組織情報やメッセージデータを割り当てるだけで、組織間の連携のために必要となる連携の手順の構築とメッセージデータの入力が必要

速に行えることになる。この統一的な組織間連携定義データの記述では、連携を行う組織名を記述し、それに対応する URI (Universal Resource Identifier) で示すリソース情報を連携の実行時に与える。これにより、連携時におけるリソース名の変更が影響で、連携ができなくなることが回避でき、インターネット上の自律した組織が分散する環境下で、動的な変化に柔軟に対応することが可能となる。また、各組織は連携部のみを統一的な組織間連携定義データに対応できる仕様とし、組織内部の処理を独自の業務プロセスとして構築することで組織の独立性を維持することが可能となる。

#### 4.3 組織間連携定義データの流用支援

統一的に定義された組織間連携定義データの記述を基に組織間での連携を迅速かつ柔軟に行う必要がある。そのためには、統一的に記述された組織間連携定義データがある組織のユーザが独自の判断で利用できる支援機構が提供される必要がある。これは、ある組織間で決定された組織間連携定義データの登録、管理、配布する支援機構である。ある組織間連携サービスを利用するユーザは、組織間連携定義データを取得し、その統一的な連携記述に必要な項目を割り当てることによって、組織間の連携関係のデータを具体化することになる。

#### 4.4 組織間連携の運用支援

連携関係の手順やメッセージの具体化が行われたら、WfCoordinator と WfExecutor の間での相互運用を実現するための支援機構が必要となる。WfCoordinator と WfExecutor が連携を行う際、統一的に記述された組織間連携定義データの記述にある論理的な組織情報を基に、物理的な組織情報の参照を行う。組織情報を管理する機構は、論理的な組織の名称とそれに対応する物理的な名称を管理し、組織間連携定義データを管理する機構とは独立である。したがって、組織間連携定義データの記述が汎用化され、他の組織間連携の場合にも利用されることになる。

### 5. 支援機構と運用イメージ

組織間の柔軟な連携を実現する支援機構を組織間連携定義データの構築フェーズと組織間連携の運用フェーズの 2 つに分けて説明する。

#### 5.1 組織間連携定義データの構築フェーズ

図 5 では、組織間連携のための手続きとそのやりとりで扱われるメッセージデータを構築するまでの流れを示している。図 5 で示した各段階では、次の支援機能により実現される。

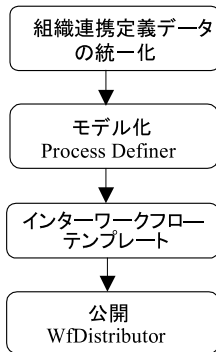


図 5 組織間連携定義データの構築フェーズ

Fig. 5 Phase of construction for definition data of cooperation among organizations.

- (1) 組織間連携定義データの統一化  
見積書の提供，発注の受け付け，請求書の発行などのサービスを提供する組織が連携手続きやメッセージ構造を同業種間で統一化する．その同業種の組織と連携する組織は複数の異なる組織が想定される．したがって，サービスを利用する組織は，組織独自の利用方法で提供されるサービスを利用することが可能になる．
- (2) Process Definer  
組織間連携定義データを GUI 操作によって定義するツールである．定義された組織間連携定義データは，インターワークフローテンプレートとして保存される．
- (3) インターワークフローテンプレート  
インターワークフローテンプレートは，図 6 に示すように，組織間の連携に必要なメッセージ構造をヘッダ部，ボディ部，メッセージ部の 3 つにより構成し，XML によって記述している．このインターワークフローテンプレートの各項目に実際の組織名などを割り当てることによって，動的な組織間連携を実現するインターワークフロー定義データとして具体化することが可能になる．
- (4) WfDistributor  
WfDistributor は，インターワークフローテンプレートを公開するために，登録，参照，配布の機能を持ち，HTTP プロトコルを利用し，Web ブラウザを通して，それぞれの操作を行うことが可能である．

## 5.2 組織間連携の運用フェーズ

組織間連携の運用は，次の支援機構により実現される．

- (1) WfCoordinator

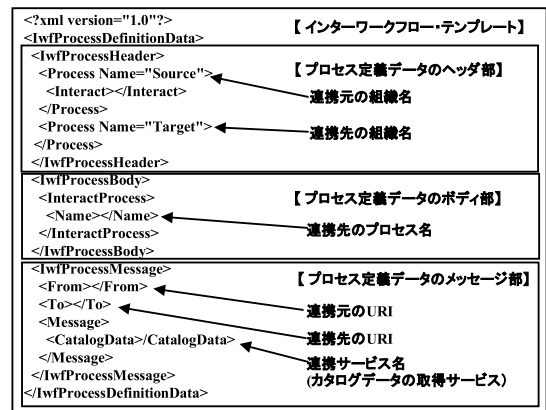


図 6 インターワークフローテンプレート

Fig. 6 Interworkflow template.

- (2) WfExecutor  
WfExecutor は，WfCoordinator から依頼されたタスクの要求に応じた実行を行う．WfCoordinator から依頼されたタスクの要求に応じることができず，他の組織と引継型，請負型の連携を行う必要がある場合，WfExecutor は，必要に応じて組織内の WfCoordinator に対して，新たなタスクの生成を依頼し，他の組織の WfExecutor と連携を行うことも可能である．ただし，WfCoordinator を実装していない実行者としての組織も存在する．
- (3) B2B Name Directory  
インターネット上では，組織の物理的なリソース名が変更されることが想定される．B2B Name Directory では，組織に対する論理名と，それに対応する物理的なリソース名を対に管理し，参照できる機能を提供する．WfCoordinator が WfExecutor と連携を行う際に，B2B Name Directory に問合せをし，組織名に対応する URI を参照可能とする．

図 4 は，上記に述べた各支援機構間の関連と構成を示している．各支援機構の連携により，組織間連携の運用が次のように行われる．

- (1) ある組織ユーザが連携先の組織として必要となるインターワークフローテンプレートを WfDistributor にて検索し，それをダウンロー

ドする。

- (2) インターワークフロー・テンプレートを取得した組織ユーザは、WfCoordinatorへそれを組み込む。WfCoordinatorは、組織自身の情報と連携先の組織情報をインターワークフロー・テンプレートへ埋め込み、インターワークフロー定義データとして具体化する。
- (3) 具体化されたインターワークフロー定義データは、WfCoordinatorの実行機能によりインスタンス化される(タスクの実行が開始される)。
- (4) 連携先となる WfExecutor の URI は、B2B Name Directory に問い合わせることにより、インターワークフロー定義データの連携先組織の論理名に対応した物理的なリソース名として割り当てられる。この結果、連携先との相互運用が実現される。WfCoordinatorからのタスク依頼のリクエストメッセージは、WfExecutorにより解釈され実行される。WfExecutorがタスクの実行処理機能を持つ場合もあるが、組織内部の処理として他の実行処理機能を持ったアプリケーションプログラムへ引き継がれる場合もある。

## 6. 支援機構の試作システム

### 6.1 試作システムの目的と概要

本論文で提案している柔軟な組織間連携を実現する支援機構の試作システムを構築した。本試作システムの構築の目的は、4章で述べた3つの支援機構からなるシステムの実現性の確認である。

柔軟な組織間連携が要求される背景として、組織あるいは組織間で生じる業務プロセスの再編成により、これまで運用されている業務プロセスを変更することがあげられる。さらに、連携先組織のサーバのダウンなどによる連携業務の一時的な停止なども考えられる。また、発注業務のように、発注品の量や納入時期によって、連携先組織とは折合いがつかないということも考えられる。B2B電子商取引システム構築基盤が今後ますます発展することを想定し、本試作システムでは、B2C(business to consumer)サービスを提供する組織が他の組織と柔軟にB2Bの連携を迅速かつ柔軟に行うシステムを実現している。この試作システムでは、エンドユーザからの要求に応じた組織間連携を実現することをインターワークフローの動的な変更が生じる一例としている。本試作システムでは、エンドユーザにサービスを提供する組織として、PCの構築と販売を行うことを目的とした電子商取引システ

ムを例としている。この試作システムは、エンドユーザがPCのある組立部品を複数の企業から選択でき、その要求に応じて、PCの組立てを行い、PCの搬送へと至るプロセスを運用する。エンドユーザはWebブラウザでPCの組立てに必要な部品構成を選択し、その結果を受注部門へ送信することになる。受注部門では、エンドユーザからの要求に応じる企業との業務取引が既存の業務フローとして運用されていないと判断すると、要求のあった部品を提供する企業と取引を行うために連携業務を開始する。その結果、PCの組立部門へ要求のあった部品が搬送され、組立工程へと移ることになる。

試作システムの構築において、次の工程を実現する。

- (1) エンドユーザがPCの組立部品を組立工程上で定められた部品メーカーではなく、他の部品メーカーを選択する。その結果、エンドユーザからの要求を受け付けた受注部門は、その要求に応じた組織との連携を行う。
- (2) エンドユーザの要求に応じて組織間の連携関係を新たに追加し、その結果を既存のプロセスに反映させる。

### 6.2 試作システムの実装モデル

本論文で提案する柔軟な組織間連携を実現するインターワークフローの支援機構を具体的に実現するために、PCの受注部門、組立部門、部品のサプライヤ、搬送部門からなる連携プロセスをインターワークフローとして実装した。受注部門は、不特定多数のエンドユーザにPCのカスタマイズと販売のサービスを提供する。エンドユーザからの注文要求を受け取った受注部門は、組立部門へ注文要求を送り、そこで、製品が組み立てられる。組立てが完了したら、搬送部門へと搬送依頼が送られる。これらの4つの部門により運用されるインターワークフロープロセスを図7に示す。さらに、エンドユーザ個別の要求に応じる組織間連携を実現するため受注部門にWfCoordinatorを実装し、PC組立部品を提供するサプライヤにWfExecutorを実装した試作システムの実装モデルを図8に示す。

### 6.3 連携プロトコル

本試作システムで実装した図8におけるWfCoordinatorとWfExecutorの間の連携プロトコルについて述べる。WfCoordinatorとWfExecutorの間の通信は、HTTP上でXMLを利用したメッセージを交換することで行われる(図9参照)。本論文では、WfCoordinatorとWfExecutorの連携プロトコルをIWf-XMLと呼ぶ。IWf-XMLは、IwfMessageをルートタグとして持つXMLメッセージで、IwfMessage-

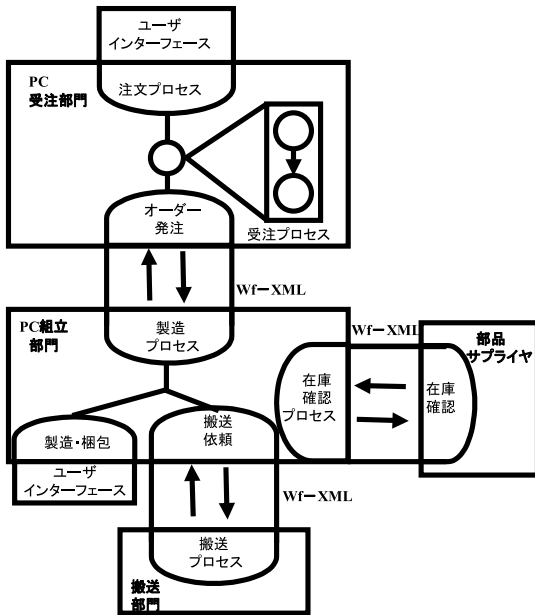


図 7 インターワークフロープロセス  
Fig. 7 Process of interworkflow.

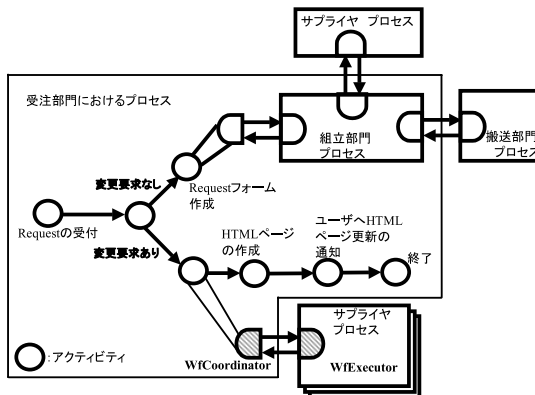


図 8 試作システムの実装モデル  
Fig. 8 Implementation model of prototype system.

Header, IwfMessageBody の 2 つの部分により構成されている。IwfMessageHeader は, MessageID と Key の 2 つの要素を含んでいる。MessageID は, メッセージのやりとりに一意に付けられる。メッセージの混合を防ぐためのものである。Key は, 連携関係の開始時に一意に付けられる連携先または連携元のリソース名を識別する ID である。

IwfMessageBody は, 要求時には, “オペレーション名・Request”, 応答時には, “オペレーション名・Response” を要素として持つ。それぞれの要素は, メッセージの内容を示す ContextData を要素として持つ。オペレーションは, InteractProcessInstance, GetPro-

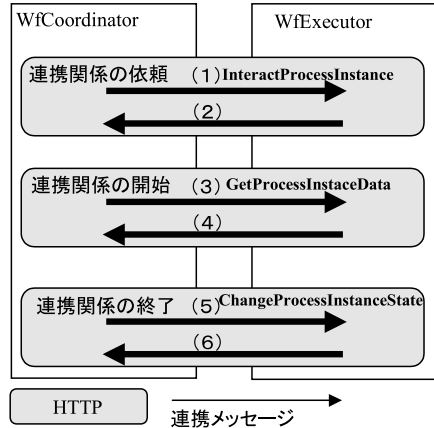


図 9 WfCoordinator と WfExecutor の連携  
Fig. 9 Operations among WfCoordinator and WfExecutor.

cessInstanceData, ChangeProcessInstanceState の 3 つからなる。実際に, 試作システムで利用されている連携手続きの例を次に示す。

- (1) 連携関係の依頼要求  
WfCoordinator から WfExecutor へ連携関係の生成依頼を送信する。
- (2) 連携関係の識別 ID の発行  
WfCoordinator と WfExecutor の連携関係を識別するために, WfExecutor は, ProcessInstanceKey を発行する。この ProcessInstanceKey によって, 連携関係の状態について問合せを行う。
- (3) タスクの開始依頼  
ProcessInstanceKey を受信した WfCoordinator は, その ProcessInstanceKey を基に, タスクの開始依頼を WfExecutor へ送信する。
- (4) 依頼要求の結果の返信  
WfCoordinator からタスクの依頼を受けた WfExecutor は, タスクの処理結果を WfCoordinator へ送信する。
- (5) 連携関係の終了  
タスクの処理結果を受信した WfCoordinator は, WfExecutor へタスク完了のメッセージを送信する。
- (6) 連携関係の終了確認  
WfExecutor から連携関係の終了確認が通知される。

上記の連携手順は, 連携関係を生成するとき, 例外的な問題が発生しない場合である。また, 試作システムにおける他の部門間の連携は, 文献 (6), (12) を参考に Wf-XML に基づいて行うように実装した。



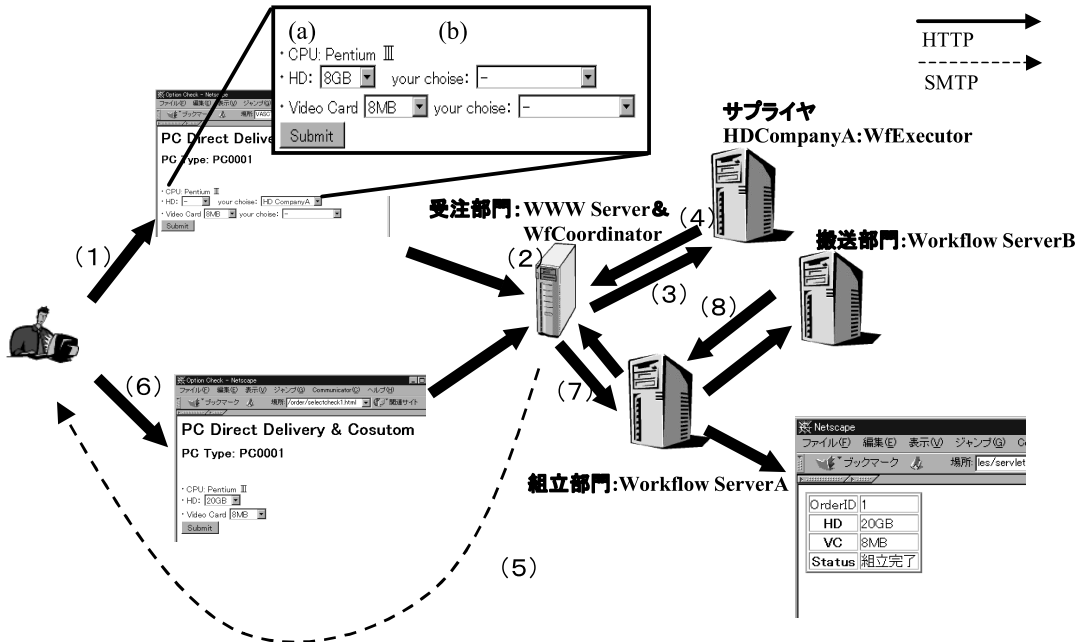


図 10 試作システムの運用の流れ  
Fig. 10 Flow of execution for prototype system.

#### 6.4 試作システムの運用の流れ

ここでは、本試作システムの利用と運用の流れについて説明する(図 10 参照)。

- (1) エンドユーザは、Web ブラウザを利用して、受注部門で提供されるフォームを好みによって選択する。各項目は、プルダウンメニューになっている。(a)のプルダウンメニューで提供されている部品の仕様やメーカーを変更したい場合は、(b)のプルダウンメニューにより、ユーザの好みの組織を選択する。すべての項目を選択したら送信する。
- (2) ユーザから送信されたパラメータは、受注部門側でオーダーフォームに変換され、組織変更がある場合は、WfCoordinator にそのオーダーフォームが渡される。WfCoordinator では、オーダーフォームから変更組織に該当する情報を抽出し、変更部品に該当する組織で定めているインターネットワークフロー・テンプレートからインターネットワークフロー定義データへと変換する。複数の組織変更がある場合は、変更要求に応じたインターネットワークフロー・テンプレートからインターネットワークフロー定義データへと変換される。受注部門では、あらかじめ各部品を提供する組織と連携を行うインターネットワークフロー・テンプレートを WfCoordinator へ導入済みと

仮定している。

- (3) (2)で生成されたインターネットワークフロー定義データは WfCoordinator により、インスタンス化され、部品のサプライヤである組織の WfExecutor と連携を開始する。
- (4) HDCCompanyA の WfExecutor は、受注部門からのリクエストに応じて、HD の価格情報と仕様をリクエスト結果として返信する。本実装では、HDCCompanyA がリクエスト結果を XML データとして用意している。
- (5) リクエスト結果を受け取った受注部門では、リクエスト結果のデータを抽出し、エンドユーザへリクエスト結果を反映した情報を提供するために、HTML へ変換し、電子メールでページの更新を促す内容の通知を行う。
- (6) リクエスト結果の通知を受けたエンドユーザは、指定された URI を更新し、再度、フォームを完成させて送信する。
- (7) 再度、エンドユーザからの要求を受け取った受注部門では、そのオーダーフォームに従って組立部門へと要求を出す。
- (8) 組立てが完了すると、搬送部門へ通知がいき、エンドユーザのもとへ製品が納品される。

#### 6.5 試作システムの実現性と課題

インターネットワークフローの動的な変更が生じる例とし

て、B2C 電子商取引システムを例に本試作システムを実現した。本試作システムは、WindowsNT4.0の環境で開発した。WfCoordinator, WfExecutor, WfDistributor, B2B Name Directory は、Java Servlet ( JSDK2.1 ), Java( JDK1.2 )により開発を行った。運用環境としては、WindowsNT4.0 Server 環境に PC の受注部門、組立部門、部品のサプライヤ部門、搬送部門を想定したプロセスを Java Servlet により実装しそれぞれの連携を行った。組織内部のプロセスは、ユーザインタフェースとして Web ベースで運用できるものとして実装を行った。エンドユーザへ提供するフォーム項目のチェック機能は Java Script により実装した。組織内部のプロセスで扱うデータ形式は、本支援機構との連携を容易に行うため XML とした。具体的には、組織内部のプロセスの進捗状況を管理するために、各組織ごとにオーダ ID や連携に必要な MessageID, Key, ProcessInstanceKey を XML データとして管理した。

本試作システムの利用は ( 1 ) エンドユーザからの変更要求が生じない場合 ( 2 ) エンドユーザからの変更要求が生じる場合の 2 通りである。この 2 つのケースを交互に繰り返して行い運用した。その結果、変更要求が生じた場合に、変更先の組織との連携がスムーズに行えることを確認できた。この変更要求は、受注部門の WfCoordinator と PC 部品のサプライヤの WfExecutor との連携により、エンドユーザの要求に応じた組織間連携を実現している。その結果、サプライヤから PC の組立部門へ部品が配送される。これは、物流工程になるので、その組織間の連携の実装を行っていない。したがって、2 組織間での連携にとどまっているが、変更後の組織が独自の判断によって他の組織と連携を行うということも実現可能である。また、注文が発生した場合、実際には、インターワークフローを形成する組織ごとに、別の組織との連携を行っている場合も考えられる。その結果、組織の規模によっては、業務連携の進捗度合いに影響が出ることも考えられる。しかし、組織の独立性を維持することから、組織内部のプロセスの特性を把握することが十分にできないと思われるので、組織間連携を行う前に、組織間のスケジュール交渉を行う仕組みの導入が今後の課題になるとと思われる。

## 7. 議 論

近年では、RosettaNet<sup>16)</sup>に見られるように、B2B 電子商取引を行うためのフレームワークを標準化し、企業間におけるアプリケーションの統合を迅速に行う

ための方法論や実現がいくつか報告されている<sup>17)</sup>。

本論文では、組織間における業務連携を迅速かつ柔軟に行い、動的なインターワークフローの形成と運用の支援機構を提案している。ここでの動的なインターワークフローの形成とは、すでに運用されているインターワークフロープロセスのアクティビティのレベルにおいて、必要に応じて他の組織におけるワークフロープロセスのアクティビティとの連携を行い、その結果としてインターワークフローが形成されることを示す。

インターネットのような広域の分散環境では、異なる組織によって管理されるシステムが連携し業務を行うことが要求される。このような分散環境下では、1 つの管理システムが業務プロセスを統合的に管理することは困難である。なぜならば、各組織には内部プロセスのすべてを他の組織に公開できないという制約があり、組織の独立性を維持する必要があるからである。

ここでは、組織間にまたがった業務連携を 1 つのサーバで管理するのではなく、分散されたサーバを連携することで業務を管理・運用することを目的とした他の学術論文との比較を行う。

文献 18) では、電子メール基盤である“め組”とワークフロー定義支援機能をもったグループウェア開発支援環境“育組”の連携により帳票ワークフローの運用を実現している。このシステムの特徴は、電子メール接続のみを前提とし、ワークフローを分散制御することにより、異種組織間連携を実現可能としている。

ワークフローの運用は、“育組”で定義したワークフローが、電子メールを制御するルールに変換される。次に、このルールを実行することで帳票電子メールが回覧される。ワークフロー定義の変更が生じた場合には、ワークフローの実行をルールの事前配布方式とメール内蔵方式の 2 つの方式により行うことができる。この事前配布方式は、ワークフローの定義が変更されたときに、そのルールをワークフローの実行単位にあたるノードに配布することで実現している。メール内蔵方式は、ワークフロー実行時に電子メールにルールを内蔵することで実現している。ただし、この論文では、組織間連携の動的な変更については考慮されていない。

文献 19) は、インターネット上で運用される分散ワークフローシステム RainMan について報告している。RainMan は RainMaker というワークフローコンポーネントを構築するフレームワークに基づいて実装されている。RainMaker は、タスクの実行、制御、問合せを行う“Performer”と、ワークフローのサービス

をリクエストする“Source”からなる構成を前提としている。ワークフローの運用は，“Source”がワークフローサービスのトレーディングサービスの役目をする“RainMan Directory”に問合せを行うことで、ワークフローのタスクを実行する“Performer”と連携し実現される。実際には，“Source”の概念を実装したJava Appletである“Builder”が“Performer”へのタスク要求を出すことになる。“Builder”は、ワークフロープロセスの開始後に、ワークフローグラフにおける“Performer”で定義されている利用するアプリケーションなどの詳細事項を変更することができる。ただし、この研究でも組織間連携の動的な変更については考慮されていない。

本研究で提案している支援機構は、複数の組織にまたがる業務連携を統合的に管理する方式ではなく、タスクの管理や制御を行う主体とタスクの実行を行う主体をネットワーク上に分散させ、相互に連携を行うことでワークフローを実現する方式である。これは、文献19)で報告されている機構と類似している。これに対して、本支援機構の特徴は、組織間連携を部分的に変更するために、同じサービスを提供する組織が他の組織との連携に必要なとする組織間連携定義データを統一化し、それを利用する組織が独自の判断で利用できるように、インターワークフロー・テンプレートとして配布するWfDistributorをネットワーク上に分散させていることである。

## 8. ま と め

本論文では、組織間における業務連携を迅速かつ柔軟に行い、動的なインターワークフローを実現する支援機構について提案し、その実現性を確認するために試作したシステムの特徴について述べた。現在、B2B電子商取引サービスの普及ともなって、組織の規模にとらわれず、迅速かつ柔軟にB2B電子商取引サービスを運用するための技術や基盤が整えられてきている。このようなB2B電子商取引サービスの構築・運用基盤が整えられてくると、B2C電子商取引サービスとして、ユーザからの多様な要求に応えるニーズが拡大することが期待される。そのため、柔軟な組織間連携を実現することは、B2C電子商取引サービスの拡大へつなげると期待できる。本試作システムの実装では、組織内部の業務プロセスとの連携を実現するために、独自仕様のアプリケーションを構築して確認した。今後の課題では、商用のワークフロー管理システムやグループウェア製品との連携をスムーズに行うための統合的な方法について検討する必要がある。

謝辞 本研究は、平成11年度、12年度文部省科学研究費補助金基盤研究(C)(2)の援助を受けた。ここに記して謝意を表す。

## 参 考 文 献

- 1) 速水治夫ほか：ここまで来たワークフロー管理システム(3)ワークフロー製品の実際、情報処理, Vol.40, No.5, pp.507-513 (1999).
- 2) 速水治夫：ここまで来たワークフロー管理システム(4)現実となったインターワークフロー、情報処理, Vol.42, No.1, pp.44-50 (2001).
- 3) 森田昌宏, 向垣内岳弥, 山下武史, 速水治夫：インターワークフロー支援：組織間連携ワークフロープロセスの構築と分散型運用管理の支援機構、情報処理学会論文誌, Vol.38, No.11, pp.2298-2308 (1997).
- 4) 速水治夫, 勝間田仁, 世古将洋, 提箸公代, 渋谷亮一, 石丸知之, 大南正人, 岡田謙一：商用ワークフロー管理システムと連動するインターワークフロー支援システム、情報処理学会論文誌, Vol.41, No.10, pp.2708-2718 (2000).
- 5) The Workflow Management Coalition: Workflow Management Coalition Workflow Standard-Interoperability Abstract Specification, Document Number WfMC-TC-1012 (2000).  
<http://www.aiim.org/wfmc/mainframe.htm>
- 6) The Workflow Management Coalition: Workflow Management Coalition Workflow Standard-Interoperability Wf-XML Binding, Document Number WfMC-TC-1023 (2000).  
<http://www.aiim.org/wfmc/mainframe.html>
- 7) 岡田謙一, 速水治夫, 大南正人, 提箸公代：異種ワークフロー管理システムを用いたワークフロー連携実験、情報処理学会論文誌, Vol.41, No.10, pp.2719-2730 (2000).
- 8) Veijalainen, J.: Issues in Open EDI, *Proc. Systems Integration'92*, pp.401-412 (1992).
- 9) 速水治夫, 坂口俊昭, 渋谷亮一：ここまで来たワークフロー管理システム(2)ワークフロー製品の標準化、情報処理, Vol.39, No.12, pp.1258-1263 (1998).
- 10) Schmidt, M.-T.: The Evolution of Workflow Standards, *IEEE Concurrency*, Vol.7, No.3, pp.44-52 (1999).
- 11) Swenson, K.: Simple Workflow Access Protocol (SWAP), Internet draft (7, Aug. 2000).  
<http://www.ics.uci.edu/ietfswap>
- 12) Hayes, J.G., Peyrovian, E., Sarin, S., Schmidt, M.-T., Swendon, K.D. and Weber, R.: Workflow Interoperability Standards for the Internet, *IEEE Internet Computing*, Vol.4, No.3, pp.37-45 (2000).
- 13) XMLで企業間プロセスを自動化する, 日経コン

ピュータ, No.471, pp.106-109 (1999).

- 14) orderit: <http://www.orderit.ne.jp/index.asp> (2000).
- 15) Enterprise Connector: [http://www.nttdata.co.jp/service/enterprise\\_connector](http://www.nttdata.co.jp/service/enterprise_connector) (2000).
- 16) RosettaNet: <http://www.rosettanel.org> (2000).
- 17) BizTalk: <http://www.jp.biztalk.org/BizTalk/default.htm> (2000).
- 18) 垂水浩幸, 田淵 篤, 吉府研治: ルールベースの電子メールによるワークフローの実現, 情報処理学会論文誌, Vol.36, No.6, pp.1322-1331 (1995).
- 19) Paul, S., Park, E. and Chaar, J.K.: RainMan: A Workflow System for the Internet, *1st USENIX Symposium on Internet Technologies and Systems* (2000).  
<http://www.usenix.org/publications/library/proceedings/usits97/paul.html>

(平成 13 年 3 月 14 日受付)

(平成 13 年 9 月 12 日採録)



勝間田 仁 (正会員)

1969 年生. 1992 年日本大学生産工学部数理工学科卒業. 1994 年同大学大学院生産工学研究科博士前期課程修了. 1997 年北海道大学大学院工学研究科博士後期課程修了. 博士 (工学). 同年, 神奈川工科大学情報工学科助手, 現在に至る. 分散ワークフロー, CSCW, モバイル・グループウェア, エージェント技術に興味を持っている. 電子情報通信学会, 計測自動制御学会, 人工知能学会, 日本ソフトウェア科学会, IEEE-CS, ACM 各会員.



速水 治夫 (正会員)

神奈川工科大学情報工学科教授, WfMC 日本支部委員長. 1970 年名古屋大学工学部卒業. 1972 年同大学大学院工学研究科修士課程修了. 1993 年工学博士. 1972 年~1998 年 NTT にて, メインフレーム, データベースプロセッサ, グループウェアの研究開発に従事. 1994 年~1998 年電気通信大学大学院客員教授. 1998 年より現職. 主な研究分野はグループウェア, データベース. 著書「ワークフロー」(日科技連出版)等. 1992 年~1996 年本学会学会誌編集委員. 1996 年~2000 年本学会論文誌編集委員. 本学会創立 40 周年記念論文賞受賞. 電子情報通信学会会員.