

## タスク管理システム Tcube の評価実験

松山 知生\* 井上 祐子\* 村上 篤\* 岡田 謙一\*

現在新しい作業環境, オフィス環境が様々な形で提案され, またニーズも高まっていることから近未来的なオフィス空間を実現するシステムの研究が進められてきている. バーチャルオフィスや電子会議システム, ワークフロー管理システムが良く聞くシステムである.

しかし日常業務においては, これらのシステムはあまり浸透しておらず限られた業務を支援するものとして認識されてきた. 組織を運用していく上でもっとも重要と考えるコミュニケーション, 作業の依頼などは電子メールや直接的な対話によりなされている.

我々は, ユーザーの意思で自由に広がる作業連携において, 仕事と仕事という人間を介さないコミュニケーションの概念を導入することで, タスクの追求及びモニタリングを行い, 動的な作業連携をサポートする Tcube(Task Tracking and Transfer) システムを提案し, 実装に伴う評価について述べる.

### Experiment with Task Management System Tcube

Tomoo Matsuyama\* Yuko inoue\* Atsushi Murakami \* Ken-ichi Okada\*

Nowadays, new office environment and new working spaces were proposed with several forms, their needs are going up. Therefore, near future office space researches are in progress. Virtual office, electric conversation system, and workflow management system are famous.

But in the daily works, they are not popular and are used only in particular situations. Communications and requests of the works are done by the e-mail or direct talk.

In the case of free work interaction expanded by the users needs, we propose Tcube(Task Tracking and Transfer) System. Tcube are based on new communication concept that task communicates with another task, and it makes us to be able to track the users which related with the task and to monitor the workflow. After our propose, we talk about the implementation and experiment.

#### 1 はじめに

昨今の BPR(Business Process Re-engineering) による, オフィス内のコンピューターの普及, 通信ネットワーク技術の発展により, オフィスの作業形態, 作業環境が著しく変化しつつある. その一環として, タイムゾーンや場所に制限のない非同期分散環境における連携作業を行うシステムが実現可能なインフラストラクチャーも整備されてきた. これらの一例には, ワークフロー管理システムや, 営業支援システム, 電子メールがあり, 現実には様々な場面で運用されている.

ワークフロー管理システムとはあらかじめ作業順序を定義し, その手順に従いコンピューターにビジネスプロセスを管理させるものであり, 営業支援システムと同様に定型業務を行うような場合に非常に

有効なシステムである.

電子メールでは, 汎用性という意味では非常に有効であるが特別なサポートがなされていないため, 依頼背景や関連情報の提供, 依頼に伴う手続き, 作業を行う際のサポートをいっさい行わない. 逆にワークフロー管理システムなどは, それらのサポートという面では充実しているが反面, 様々なタイプの作業や突発的な連携に対応できない.

我々は新しいオフィスコミュニケーション実現の第一歩として, 今回日常の業務に目を向けた. 連携作業においては電子メールや口頭での依頼がなされているが, 履歴管理が行えず流動的に次々と様々な人に作業がわたるため, 自分の依頼した仕事が「今現在誰の元にあるのか」又は「どの位完成しているのか」などの情報把握が困難であるという問題点がある.

そこで本稿では, 仕事同士が互いに連携を行うタスクコミュニケーションを提案し, 作業手順を定義しな

\* 慶應義塾大学 理工学部 計測 (情報) 工学科  
Department of Instrumentation(Information), Faculty of  
Science and Technology, Keio University

くてもユーザが自由にタスクを派生でき、さらに自分の手から離れたタスクの追跡を行い連携業務の履歴管理を行う Tcube (Task Tracking & Transfer) システムを実装した。2章において Task Communication の概念、3章で Tcube システムの実装、4章で評価実験、5章で Tcube システムの位置づけをそれぞれ述べる。

## 2 Task Communication の概念

我々は、ユーザが自由に連鎖的に依頼を繰り返す非同期共同作業の支援を目的としている。ユーザー同士は簡単な「指示」「要求」「依頼」が成立し得る間柄としグループ内の情報はオープンであると仮定する。

日常における人・仕事間の関わり方には3つのコミュニケーション形態が存在していると考えられる。

- 人と仕事のコミュニケーション：人が仕事を行う
- 人と人のコミュニケーション：人間同士が依頼、問い合わせを行う
- 仕事と仕事のコミュニケーション：仕事というオブジェクトが自律的に他のオブジェクトと連携を行う。

「人と仕事」「人と人」のコミュニケーションは、日常良く行われており、「コンピューターを用いて書類を作る」などが「人と仕事」、「電子メールで仕事を他の人に依頼する」などが「人と人」のコミュニケーションとなる。

ここで新たに「仕事と仕事」のコミュニケーションを提案する。これは、仕事というオブジェクトが様々な状況判断を行い、人の要求や必要に応じて自律的に他のオブジェクトと連携を行うことである。我々はこれを **Task Communication** と定義した。

現在、人同士が仕事の依頼を行うには口頭や電子メールを使用している。しかし、実際に会って依頼を行うためにアポイントメントをとらなければならなかったり、メールを用いて依頼するときなどは相手がいづ読むかわからず、いつ返ってくるかわからない返事を待つ必要がある。作業に関わる人が幾人にもなると、上記のような原因により膨大な時間・コストがかかる。

そこで、仕事というオブジェクトに仕事間の派生関係を持たせる。図1において、ユーザーAが自分の仕事A'に問い合わせると自動的に Task Communication(A'-B'-C'...)を行い、派生している仕事

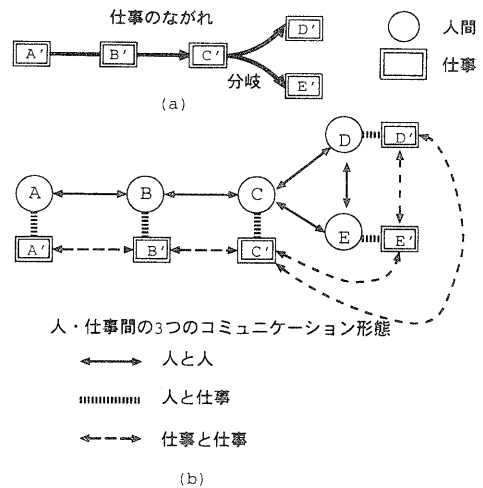


図1: 人と仕事の連携関係

を探し出し、現在の進捗状況や依頼関係をユーザに伝えることが可能となる。つまり人は自分の仕事に問い合わせるだけで、人を介さない仕事間のコミュニケーションによって必要な情報を必要な時に得られる。

## 3 Tcube システムの実装

### 3.1 実装環境

Tcube システムはサーバー、クライアント方式を用いて Java 言語で実装しており、UNIX 及び WindowsNT 上で動作し、約30クラス、157815バイトで構成されている。クライアントは分散環境での利用の考慮からプラットフォーム非依存で動作する Java アプリケーションで作成した。Task Communication では XML(eXtensible Markup Language) 言語で書かれたメタデータを使用し、パースする際には IBM 社製作のフリーソフト XMLParser for Java を利用した。サーバ・クライアント間通信には TCP/IP プロトコルを用いた。

### 3.2 システムの構成

図2の上部がサーバサイド、下部がクライアントサイドを表している。

### 3.3 システムの実行

ユーザが Tcube にログインすると、タスクマネージャがユーザが所持しているタスクを収集し、それらをワークリストに載せる。ワークリストに載る情報は以下の通りである。

#### 3.3.1 仕事の状態と可能なイベント

タスクはそれぞれ、仕事のどの段階にあるのか、何のイベントを待っているのかを判断するためにタ

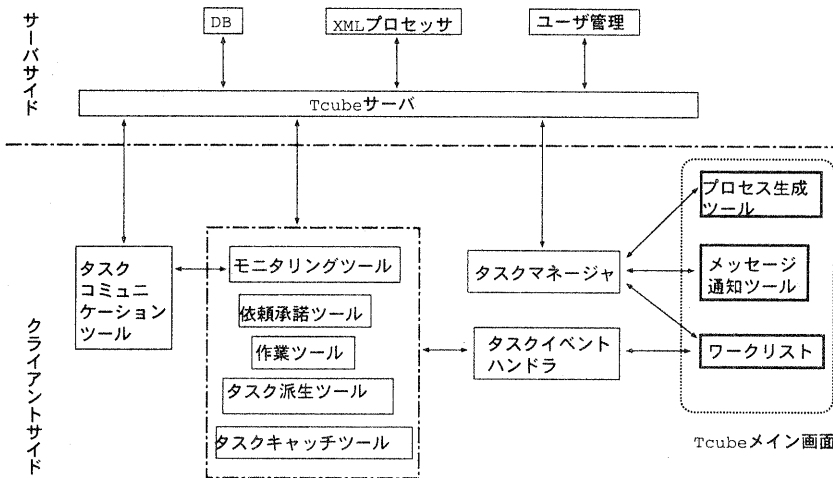


図 2: システム構成

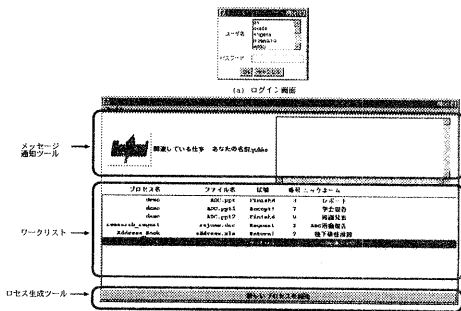


図 3: ログイン画面とメイン画面

タスクの状態を保持している。リスト上のタスクをクリックすることによって、タスクイベントハンドラが動作し、タスクの状態に応じたイベントを呼び出す。

人の行動パターンから連携作業においては一定の規則が存在することが様々な研究で述べられてきた。[9] [7] [8] 私達のモデルにおいてタスクの内部構造は(1)仕事の承諾、(2)実際に作業、(3)タスクの移動先の決定、の3フェーズから構成される。同じフェーズ中であっても、待っているイベントが異なる状態があることから、実装ではさらに、タスクが可能としているイベント毎に状態を分けて仕事を管理するようにした。タスクの状態には、以下の7つがある。これを3つのフェーズと対応付けて説明する。図4にタスクの状態遷移の様子を示す。

- Accept?

仕事の承諾を待っている状態。フェーズ1にあたる。承諾されると Working の状態に移行する。

- Working  
タスクがフェーズ2にあり、作業を行える状態。
- Request  
タスクがフェーズ3にあり、誰かに仕事の依頼をし承諾結果を待っている状態。依頼相手が承諾すると Finishd の状態に移行する。相手から拒絶された場合は Working の状態に戻る。

- Finishd  
タスクのフェーズサイクルが全て終了した状態。フェーズ2の状態から一時的に派生をさせ、タスクが帰るのを待っている状態。フェーズ2の状態にあるので、作業を続行させることが可能。

- Return!  
フェーズ2の状態から一時的に派生をさせたタスクが帰ってきた状態。フェーズ2の状態にあるので、作業を続行させることが可能。

- Cancel!  
タスクの消滅をユーザに通知する状態。

全タスクに共通のイベントとして、ワークリストに表示されたタスクを右クリックをすることによりタスクの依頼内容の確認とニックネームの変更を行うことが可能である。ニックネームから仕事内容の確認ができ、仕事を自分独自の識別子をつけて分かりやすく管理できる。

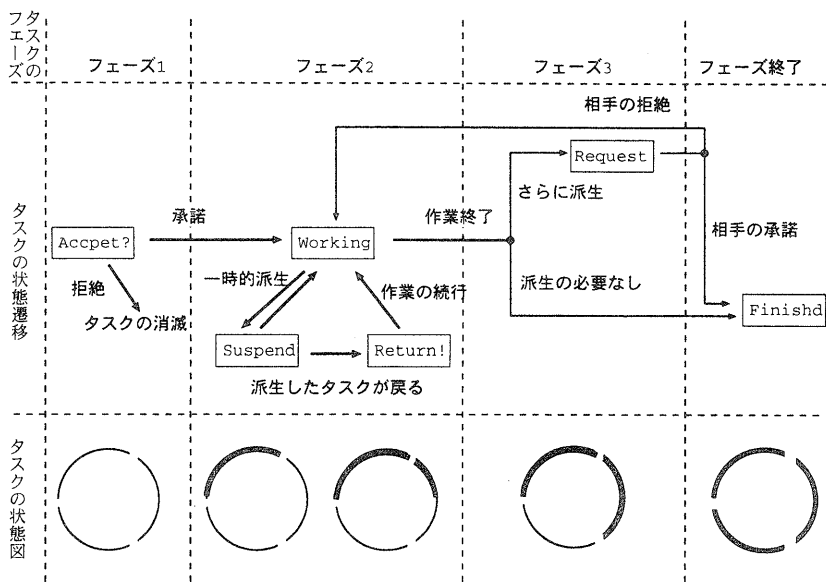


図 4: タスクの状態遷移

### 3.3.2 連携関係の把握

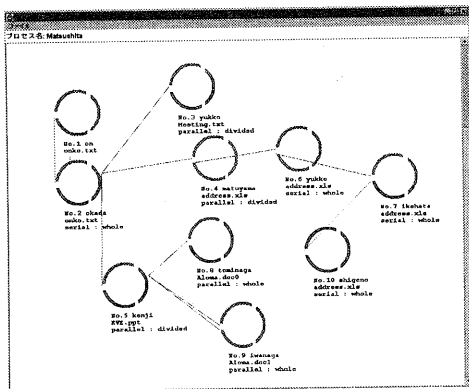


図 5: モニタリングツール

モニタリングツールにより、自分から派生した仕事の展開の様子を視覚的に把握することが可能とした。図5から、誰がどのような順序で仕事に関わっているのか、あるいは、仕事の依頼を一時的に受けているのかどうか、またそのときの依頼では仕事分割されているのか複数人に仕事依頼されているのか等の情報を一度に把握できる。

タスクの派生がフェーズ2から行われていれば一時的な派生を意味し、いずれタスクが戻ることが期

待されていることが分かる。また、タスクの派生がタスクのフェーズが全て終了したフェーズ3からのものであれば、派生タスクは派生元ユーザの管理下には無く、その後の行き先は派生元ユーザの権限の及ばない範囲にあることが把握できる。

## 4 評価実験

Tcube システムの作業連携の効率向上の評価を行うために、電子メールシステムとの比較実験を行った。我々が実装した Tcube システムと電子メールシステムの両方を利用してもらい、進捗管理に要する時間の定量的評価と、「仕事の管理」「進捗状況の把握」「派生関係の把握」「利用性（使いやすさ、気軽に派生できる）」に関するアンケートを用いた定性的評価を行った。

### 4.1 実験方法

A グループ：情報工学科の学生（計18名）

B グループ：システム設計に携わった学生（2名）を実験参加者として、Aグループのユーザ3人とBグループのユーザ2人の5人1組で、Aグループユーザーを入れ替え計6回実験を行った。

5人は同部屋の異なる端末（WindowsNT）に座り、被験者同士の会話は禁止した。電子メールシステムを利用した実験では、メールソフトとしてユーザの使いたれたもの（Netscape Messenger か Outlook

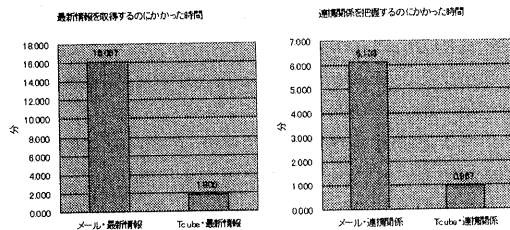


図 6: 作業時間の計測結果

Express) を使用した。

Aグループの3人のユーザには「1人には被験者5人分の”電話番号リスト”を、1人には”生年月日リスト”もう1人には”住所リスト”をそれぞれMicrosoft Excel文書で作成する」という課題を与えた。実験の参加者は仕事が回ってきたらファイルの中の自分の名前の欄に情報(電話番号, 生年月日, 住所いずれか適当なもの)を記入し, まだ記入していない人に仕事を回す回覧型の作業を行った。

次の作業者はまだ作業をしていない, 自分と異なるグループのユーザに依頼するものとし, メールでのCC機能は使用せず一人づつ順番に作業が流れるようにした。

Bグループユーザは4人目の作業者となった時点で, Aグループユーザに知らせずに作業をとめた。

Bグループのユーザによってすべての作業が止められたところで, 全体の作業を一旦中止した。その後, Aグループのユーザに対して, 各々自分が担当したリストの最新ファイルを取得させ, その取得までにかかる時間を測定し, その後作業に関わったユーザの順番を確認するまでの時間を測定した。

以上のような実験をまず電子メールシステムを用いて行い, その後同様の実験をTcubeを用いて行った。

Tcubeの使い方を実験前10分程度行い, また簡単なマニュアルを渡した。電子メールでの依頼や問い合わせ方法は, ユーザの自由とした。作業終了後すぐに, アンケートに答えてもらった。

## 4.2 実験結果

### 4.2.1 作業時間の計測結果

最新ファイルを取得するのにかかった時間と連携関係を把握するのにかかった時間を計測した結果を図6に示す。

最新ファイルを取得するためにかかった時間の平均値は, Tcubeシステムを利用した場合1.8分で

表 1: 着目点 (関連質問数) 毎の7段階評価平均値

着目点	メールシステム平均値	Tcube 平均値
連携関係の把握 (5)	3.94	6.58
進捗状況の把握 (9)	2.89	6.51
仕事の管理 (5)	2.58	6.20
利用性 (7)	4.62	6.27

あり, 電子メールシステムを利用した場合16.1分と比較して, 約1/9の時間しかかかっていないことが分かる。

連携関係の取得では, Tcubeを利用した場合0.99分で, 電子メールを用いた場合6.13分と比較して約1/6となった。

### 4.2.2 アンケートの検定

アンケート項目は, 電子メールシステムを利用した場合とTcubeシステムを利用した場合で共通で, 着目点4点をそれぞれ, 又は複数問う内容の19項目を用意した。各項目について, 最も悪い評価を1, 最も良い評価を7とし, 1~7の7段階で評価してもらった。これら7段階に1点から7点をそれぞれ与え, システム別に各質問に対する平均得点を求めた。

## 4.3 実験結果の検討

### 4.3.1 作業時間測定時間の検討

本実験では電子メールでも頻繁に受信要求を行ない, また短時間の実験のため依頼の相手を確実に覚えていたことから, 電子メールにとってより有利な実験だった。また被験者は, Tcubeシステムについて簡単な使用講習を受け, 実験中はマニュアルを読みながら作業を行った。そのため, 実験を長期的に非同期で利用する形態で行った場合は, 差がさらに広がるものと考ええる。

### 4.3.2 アンケート結果の分析

電子メールシステムとTcubeシステムの各データ郡に対してウィルコクソンの符号付順位検定を行った。ウィルコクソンの符号付順位検定は2者の比較に用い, 今回はTcubeの電子メールシステムに対する優位性を示す。P値は, 5.00e-2以下で有意水準5%をクリアする。

アンケートより4つの着目点すべてにおいて有意

表 2: 着目点 (関連質問数) 毎の P 値の平均値

着目点	P 値の平均値
連携関係の把握が行えていたか (5)	1.24e-3
進捗状況の把握が行えていたか (9)	4.14e-3
仕事の管理が行えていたか (5)	1.47e-3
利用性 (7)	2.60e-2

水準 3% をクリアしている。値が低いほどメールシステムと Tcube の差が有意である。特に連携関係の把握、仕事の管理で有効性が証明された。これはワークリスト、モニタリングツールという特別なアプリケーションを用意したためと思われる。両者は共に特別な知識を必要としなくても視覚的に現状が把握できるツールであり、ワークリストによる仕事管理、モニタリングツールによる連携関係の把握の有効性が証明された。

## 5 Tcube の位置づけ

現在市販されているワークフロー管理システム Staffware[3], InConcert, Groupmax Workflow などは、様々な業務に柔軟に対応できるものとして浸透してきた。しかし、これらはあらかじめの作業手順にしたがって定型業務を進めるシステムであり、フローの動的な変更、例外への対応、他のシステムとの連携が求められた。そこで、StaffWare では異常時の処理、Regatta [4] ではデータの動的な変更を行える機能を提供している。日立の WorkCoordinator では、あらかじめ異常事態を予想し定義しておくことで作業手順の追加が行える。InterWorkflow System[1][2] では、異なるワークフローシステム同士を連携させながら、組織の独立性を保ち、異常事態にも対応している。

しかし、以上のようなシステムはそれぞれあらかじめの定義が必要であり、突発的に発生する連携には対応していない。

またアクションワークフロー [9] は、作業オブジェクトを「提案」「合意」「実行」「満足」という 4 行程を、高度に再帰化させることによりすべての連携を表現できるとしたが、モデリングに高度な知識や経験を要する。また、タスクコミュニケーションのような概念をもたない。

Tcube では、「承諾」「作業」「移動」と 3 行程で

表現したが、ユーザーはあくまでモニタリングの際の進捗状況の参考としてのみ意識すれば良く、システム側で各段階の処理は行われる。タスクコミュニケーションにより進捗状況の把握や連携関係の把握が瞬時にできる。

我々は大きな定義など必要とせず、今仕事を頼みたいという突発的な作業連携を、簡単に誰でも使える形でサポートすべく Tcube システムを提案、実装した。

## 参考文献

- [1] Y. Inoue and K. Hiramatsu and T. Matuyama and H. Hayami and K. Okada and Y. Matsushita.: "Inworkflow management system on cross-organizational workflow processes", Adjunct Conference Proceedings HCI International'99, 219-220, 1999.
- [2] 平松恵子, 速水治夫, 岡田謙一, 松下温.: "3次元ユーザインターフェースを備えたインターネットフローシステムの提案", 情報処理学会論文誌, volume39number10, 2846-2854, 1998.
- [3] "http://www.staffware.com"
- [4] K. D. Swenson, "Visual Support for Reengineering Work Processes", Proc. of Conference on Organizational Computing Systems, 130-141, 1993
- [5] Hiroshi Ishii: "TeamWorkStation: Towards a Seamless Shared Workspace", Proc. ACM CSCW'90, 13-26, 1990.
- [6] 垂水 浩幸: "CS と CW の統合に関する考察 ~ CS から CW へ, そしてまた CS へ~", 情報 GW 研究会 GW-27, 1998, 37-42.
- [7] 市村, 松下: "発言と行動の管理に基づいた協同作業支援電子メール PilotMail", 情報処理学会論文誌, volume33, number7, 1992, 955-963.
- [8] 山上俊彦: "メールリストにおけるメール情報自動抽出: そのグループウェアの考察", 情報 GW 研究会 GW-23, 1-6, 1997.
- [9] R. Medina-Mora, T. Winograd, F. Flores: "The Action Workflow Approach to Workflow Management Technology", Proc. ACM CSCW'92, 281-288, 1992.
- [10] http://matrix.actiontech.com.
- [11] 松尾, 服部, 橋本, 貫井: "エージェントメールシステムのワークフロー制御への適応", 情報 GW 研究会, volume94, number60, 59-66, 1994.