

## タスク間コミュニケーションの提案

松山 知生 井上 祐子 村上 篤 重野 寛 岡田 謙一 松下 温

慶応義塾大学

近年の通信技術の発展により、時間・空間に左右されない連携業務が可能となり、またそのニーズも高まっていることから非同期共同作業を実現するシステムの研究が進められている。日常業務においては、電子メールや直接的な対話により業務の依頼がなされている。しかしながら、これらの方法では連携作業の履歴管理が行えないために、依頼した作業が「進捗しているのか」「今誰が作業しているのか」の把握が困難であり、依頼したことを忘れてしまうことさえもある。そこで本稿では、ユーザーの意思で自由に広がるタスク連携において、仕事と仕事とのコミュニケーションの概念を導入することによって、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)システムを提案する。

## A proposal of communication system between the tasks

Tomoo Matuyama, Yuko Inoue, Atsushi Murakami, Hiroshi Shigeno, Ken-ichi

Okada, Yutaka Matsushita

Dep. of Science and Technology, Keio University

3-14-1, Hiyoshi, Kouhoku-ku, Yokohama, JAPAN

E-mail: matuyama@mos.ics.keio.ac.jp

By the communication technology development, it is able to realize cooperate works in the environment separated by the time and space, and their needs are going up. Now, asynchronous collaboration support system research is in progress.

At present, daily work requests were done with e-mails and conversations. But in these ways, because of not to be able to manage cooperate work history, it is difficult to research the work, I request, 'Is it in progress?' and 'Who works about it?'. There is a case even it was forgotten.

As for this paper, we propose TTT(Task Tracking and Transfer) system, that support dynamic cooperate with task tracking and monitoring by the management of each task relationship which expand freely by user's intentions.

## 1 はじめに

最近のコンピューターの普及、その通信ネットワーク技術の発展により、これらを用いて作業形態、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されずに非同期に分散環境において連携作業を行うシステムを実現可能なインフラも整備されてきており、実用への期待が高まってきている。これらの一例には、ワークフロー管理システムや、電子メールがあり、現実に様々な場面で運用されている。

この場合電子メールは単に依頼を送るもので、ワークフロー管理システムとはあらかじめの作業連携等の定義にしたがって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る。

しかし日常での業務に目を向けてみると、あらかじめ作業担当者や手続きが定義されていなくても、仕事の一部の依頼など突発的に連携が必要になる場合が多く、電子メールや口頭で「ちょっとこれをやっという」「資料を集めという」「報告書つくっという」のように業務の依頼を行っているのが現状である。

しかしながら、これらの方法では連携作業の履歴管理が行えずに、特にその場で発生した作業等の場合は依頼者、被依頼者ともに何を、いつ、どこで、誰に依頼したのか記憶に残りにくい。そのため、仕事内容を覚えていても誰に依頼したかを忘れてしまう、頼んだことすら忘れてしまう等が起り得る。また、作業は流動的に様々な人の元にわたるため、自分が依頼した仕事、「今現在誰の元にあるのか」又は「どの位完成しているのか」の把握が困難であるという問題点がある。

手順を予め規定しにくいような業務を支援するワークフロー管理システムとして“WorkCoordinator”があるが、ある程度の流れの予測を幾つかのルールに基づいて最初に定義しておく必要のあるトップダウン式である。

また、ワークフローシステム実行中に新たな業務を追加する自由度は低く、さらにその実現は各開発ベンダーによる動的な記述変更技術の

開発に委ねられている。

そこで本稿では、作業手順を定義しな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) システムの提案を行う。

## 2 システムの提案

### 2.1 支援対象とする業務形態

本システムが支援対象とする業務は、ユーザー組織が日常的・非定型的・連鎖的に行われるものであり、簡単な形式の「指示」「要求」「依頼」が成立しうる間柄での、図1のようなユーザが自由に仕事を依頼していくことで形成される連携業務である。

これらの連携手順は始めから決定されているものではなく、作業を担当するユーザが動的に次のタスクの担当者・仕事内容を決め仕事を派生させていくことによって、連携業務が形成されていく。つまり、自分が依頼された仕事に対して、権限を自分自身が持ち、さらに他の人に依頼しようと、分割しようと自由となる。

同一グループ内の情報はオープンであるとし、自分の仕事から派生した連携関係と現在の最新内容が公開されても差し支えないと想定する。自分が更新した作業に対して、及び自分が派生させ他人に依頼したこともオープンであるとする。

例えば図1において、Aは、Bに依頼したタスクがCとDによって2つに分割され、その後EとFも連携プロセスに関わっていることを認識している。また、Cは自分のタスクから派生したプロセスの最新データであるFの作業内容を閲覧することができ、AはDとFが保持するデータを閲覧出来る。つまり、誰が依頼業務を完成させたかが重要なのではなく、業務内容自体が完成したかどうか重点が置かれている業務を対象としている。

### 2.2 人・仕事間の3つのコミュニケーション

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

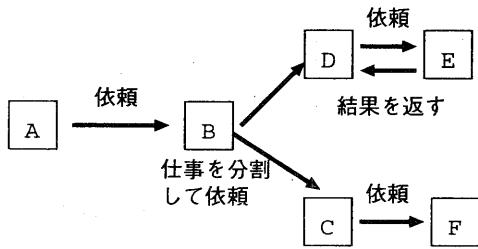


図 1: 支援対象とする業務形態

えられる。

- 人と仕事：人が仕事を行う
- 人と人：人同士のインタラクション
- 仕事と仕事：仕事の派生関係

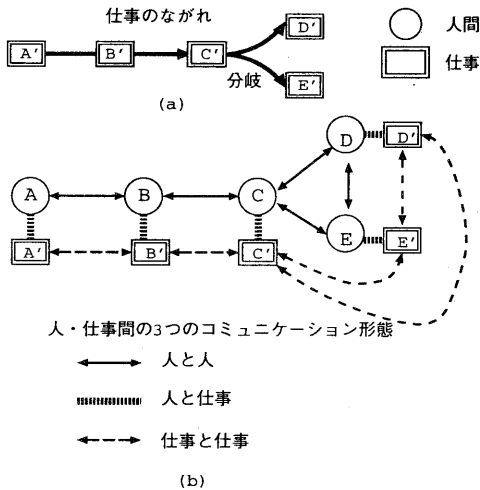


図 2: 人・仕事間の3つのコミュニケーション

「人と人」「人と仕事」の関係は既に存在している関係である。

図2において、Aが現在の仕事のできあがり具合を把握したい場合を考える。通常、まず直接の会話や電子メール等の手段によって自分が直接仕事を依頼したBに問い合わせる。Bがさらに他の人に仕事を割り振っている場合は連携相手（この場合はC）に問い合わせを行う、というように、人と人のインタラクションを繰り返す。そして最後のユーザ（DとE）までたどり着き、人と仕事の連携関係を用いること、つまり現在の作業者が実際の作業物を参照するこ

とによって初めて、最新の作業内容を手にいれることが可能となる。それを問い合わせた人に教えるという段取りである。

しかしこの現在の方法では、人の記憶が曖昧であるため誰に頼んだのか忘れてしまったり、人と人のコミュニケーションではアポイントメントをとったり、メールで問い合わせをいつ戻ってくるか分からないメールを待つなどの不都合が生じる。さらにこれが作業に関わる人が幾人にもなると膨大な時間・コストがかかってしまうという問題がある。

そこで、今回私たちは新たに「仕事と仕事」のコミュニケーションの概念を導入し、仕事間の派生関係を仕事に持たせるようにする。これにより、Aが自分の仕事A'に問い合わせると、自動的に仕事間のコミュニケーション(A'-B'-C'...)を行うことにより、自分から派生している仕事を探し出して最新内容と仕事の派生関係をユーザに伝えることが可能となる。つまり、人を介さずに仕事間での連携を用いることにより、必要な情報を必要なときに得られるようにするのである。

## 2.3 仕事の派生のモデル化

### 2.3.1 仕事の派生形態

Tcubeシステムではユーザが自由に仕事の派生を行うが、複雑に展開する仕事の派生を管理するために、「分割方法」と「依頼方法」の2つの基準を導入する。

仕事の分割方法。

- Divided型：仕事をいくつかに分けて受け渡す
- Whole型：仕事を分割せずに、一括して受け渡す

仕事の依頼方法。

- Branch型：仕事を複数の人に依頼する
- Straight型：仕事を単一人の人に依頼する

この「分割方法」と「依頼方法」を組み合わせることにより、全ての仕事の派生関係をモデル化して管理することが可能となる。

### 2.3.2 派生した仕事の権限

このように派生させた仕事の権限は、基本的には現在の作業者がもつが、分割した場合などはその派生主が自由に子供を破棄できるようにする。これは、要らなくなった仕事をいち早く消去することで無駄な作業をなくすのに役立つ。

### 2.3.3 仕事の派生の例

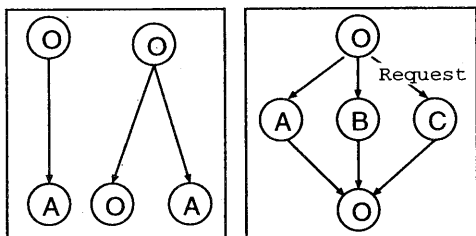


図 3: 派生の例

図 3 左のような場合、左側は単に O から A に依頼するものだが、ここでも W-S 型 D-S 型が考えられる。前者は完全にすべての作業を依頼する場合で、後者は部分的に A に依頼するものである。この後者は図左、右側の W-B 型と同じであり、O が A に手伝わってもらうイメージとして考えられる。

図 3 右のような場合 2 つの場合が仕事の依頼方法により考えられる。

- W-B 型

これは、W = すべてを、B = 複数の人に、依頼する方法である。つまりもっとも早いもの、又はもっともできの良いものを採用する公募型と考えられる。

- D-B 型

これは、D = いくつかに分けて、B = 複数の人に、依頼する方法である。これは、幾人かで分担して作業を行う型である。

これらはそれぞれ、派生主である O が権限を持つため、例えば前者の場合など出来の良いのが A から帰ってきたので B と C に頼んだことは無しにすることもできる。

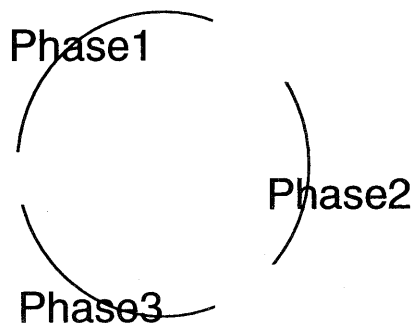


図 4: 3 つの Phase

## 2.4 Tcube システムのタスク管理

「タスク」とは、ユーザが作業を行うオブジェクトであり、作業内容が同じであってもユーザが異なれば違うタスクとして認識される。タスクには、次の 3 つの特徴がある。

- 移動する

仕事の連携があると、ユーザからユーザの元へタスクが移動する。

タスクが派生する場合、新たにタスクを作成して移動するため、今まで作業されていたタスクは終了したタスクとして派生元にも残る。後に、タスクのトラッキングを行う際には、この手元に残ったタスクに問い合わせる。

タスクが移動する際には、ユーザが行き先を決定する場合もあるが、タスクに自律的に行動させることも出来る。

- 3 つの Phase による内部構造

タスクの内部構造は次の 3 つの Phase から構成される。

- Phase 1 : 依頼承諾

タスクの依頼と承諾は、Tcube システム上で行われる。仕事の依頼がある場合、依頼要件と共に、タスクが移動してくる。このタスクを承諾するかどうか決定するのが Phase 1 である。

- Phase 2 : 実際の作業

作業を行いデータに変更を加える。また、作業途中に一時的にタスクの派生

を行うことも可能である。この場合、派生したタスクが必ず戻ってくるため、その後作業を続行させることができる。つまり、この段階で複数に分割派生させたタスクは、それぞれの作業者の元で実行され派生主のところに戻ってくる。その際に派生主はそのタスクの結合という作業を行う。

そのため、作業を行う Phase 2 を suspend した状態と見なさなければならない。

- Phase 3 : タスクの移動先の決定  
自分の作業を全て終了させ、次に派生させるタスクを作成して移動させる。派生タスクは完全に自分の手元から離れるため、その後自分の元に戻ってくるとは限らない。

この段階で分割派生させた場合、Phase 2 で派生させた場合とは異なり派生させた Task の結合は行えない。分割された Task は、以降の作業者の権限でそれぞれ扱われる。

- タスク間の派生関係の保持  
各タスクは、派生のモデル化に基づいたタスク間の派生関係 (例: タスク A から Divided-Branch 型で派生した) を保持している。

自分が関わったタスクが、その後どのように展開し、現在はどこまでできあがっているかを知りたい場合、ユーザは自分の手元にあるタスクに問い合わせを行う。タスクは、派生情報をもとに自分から派生したタスクを次々に探し出すことによってトラッキングを行い、タスクの連携関係と現在の作業内容をユーザに伝える。これにより、自分から離れた後の仕事の連携関係が明確化されると共に、従来問い合わせにかかっていたコスト・時間が大幅に短縮されると思われる。

トラッキングによる連携関係の把握の一例を図 5 に示す。タスク内部処理構造を 3 つ

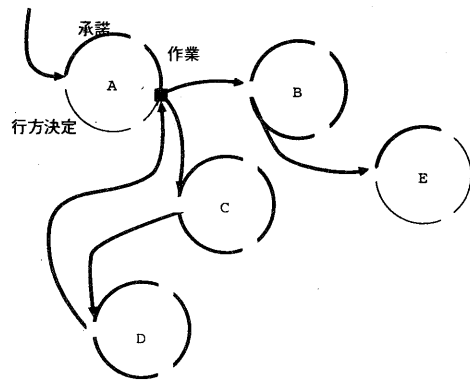


図 5: トラッキング結果の一例

の Phase に分けたことにより、作業状態と連携関係を把握しやすいモニタリングが可能となると考える。

### 3 おわりに

本稿では、日常に頻繁に起こるであろう簡単な連携「軽いワークフロー」をより効率的に確実に行うことを目的としている。そこで、ユーザが自由に・動的に仕事を依頼することによって連携関係を形成でき、さらに自分から派生したタスクの追跡・管理が行える Tcube システムを提案した。今後は、提案に基づいて実際にシステムを実装し、システムの有効性や課題を検証していく予定である。

また、他のシステムとの連携も考えている。特にワークフローシステムとの連携を実現する予定である。ワークフローシステムによりおおまかな流れを定義しておきその各部分での作業は Tcube システムにより実行する事で、より自由度の高いビジネス連携を実現することができる。このように Tcube システムの有効性は単一のシステムとしてだけでなく、他のソフトウェアのポテンシャルを上げるのにも有効であると考えられる。

### 4 参考文献

横山 隆幸, “ワークフロー管理システム「WorkCoordinator」-定義モデル-, 第 58 回情処全大 (4)4G-4, 1999.