

エージェントによるワークフローの動的再計画

垂水 浩幸^{†*} 喜田 弘司[†]
柳生 弘之^{††} 石黒 義英[†]

グループウェアの一分野として、ワークフローシステムがある。一般にワークフローシステムは個々の業務ごとに独立した管理を行っているが、現実にはオフィスでは複数の業務が営まれており、リソースを奪い合っている。本論文では、マルチエージェントを用いてワークフローを含むオフィスの複数の業務を管理する WorkWeb System の概念を提唱し、その実現方式について述べる。特に、他の業務の影響によりワークフローの計画をやり直す場合の方式については詳しく述べる。この機能により、たとえばワークフロー開始後に急な出張が入ったような場合であっても柔軟に対処できる。本システムは Windows 上で動作している。

Dynamic Replanning of Workflow by an Agent System

HIROYUKI TARUMI,^{†*} KOJI KIDA,[†] HIROYUKI YAGYU^{††}
and YOSHIHIDE ISHIGURO[†]

The *WorkWeb System*, which is an expanded workflow system implemented with a multi-agent system is proposed. Conventional workflow systems cannot plan time schedule of workflow by confirming and reserving personal schedules. The WorkWeb System can do it and also is able to dynamically re-plan the schedule when higher priority tasks are ordered even after a workflow has started. This paper describes the workflow scheduling and, especially, dynamic re-planning mechanisms of the WorkWeb System. The system is implemented on Windows.

1. はじめに

グループウェアの一形態として、ワークフローシステム^{12),17)}が注目されはじめてから早くも数年が経過した。最近では国内でもワークフローシステムの市場が確立しつつあり、適用事例も発表されてきている^{18),21)}。また、研究事例も多く発表されている。

ところで、従来のワークフローシステムは、個々の業務の設計・運用・管理を業務ごとに独立に行っている。たとえば、出張伝票業務のフロー管理はそれ以外の業務の管理とまったく独立して動作している。すなわち、従来のシステムには複数の業務がリソース（ヒト、モノ、カネ等）を奪い合うことによって生ずる相互干渉が考慮されていないという問題がある。このため、たとえばワークフロー開始後に、ある担当者に緊

急の優先的なワークフロー作業が依頼されたときに、対処できない。

この問題は従来のワークフローシステムで主として取り扱われている定型的オフィス業務（あらかじめ手順が定義された業務）に限ったことではない。アドホックに行われている仕事の依頼等の非定型業務も含めた、オフィス業務システム全般にかかわる問題であり、全体的に解決していく必要がある。

我々は、このような観点から、次世代ワークフローシステムの研究開発を行っている。それが WorkWeb System である。

WorkWeb System では、オフィスに複数のエージェントを配置し、各エージェントにそれぞれのリソースの管理や業務の管理をさせ、互いに干渉しながらそれぞれの目的を達成していく。ここで、エージェントとは、個々のリソースや業務の管理を、ユーザ（作業員、管理者）に代わって行うソフトウェアで、他のエージェントと通信可能なものを意味している。

WorkWeb の Web とはワークフローが業務の流れ（縦糸）を表現しているのに加え、複数の業務を横断的に見渡すビュー（横糸）が存在するイメージ（図1）

[†] NEC ヒューマンメディア研究所
NEC Human Media Research Laboratories

^{*} 現在、京都大学大学院情報学研究科
Presently with Graduate School of Informatics, Kyoto University

^{††} 株式会社 NEC 情報システムズ
NEC Informatec Systems, Ltd.

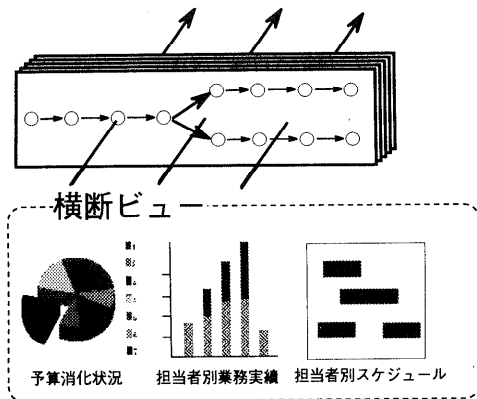


図1 WorkWeb Systemのイメージ
Fig. 1 A WorkWeb System image.

である。ビューとしては、各リソースの配分・予約状況を見通すもの他、複数の業務の達成状況を集約して見るものなどが考えられる。

2. WorkWeb System

2.1 なぜマルチエージェントか

従来のワークフローシステムでは、単一種類の業務を効率的に管理するために、サーバにその業務に関するすべての情報を集中させるクライアント・サーバ型のシステムによって構成されることが多かった。しかし、WorkWeb Systemでは、管理担当部門・管理者の異なる複数の業務を取り扱う必要があり、単一のサーバでこれを実現するのは無理がある。また、ヒト、モノ、カネのリソースについても各々管理主体が異なるため、リソース管理についても集中することは難しい。特に、最近さかんになってきているアウトソーシングや、企業内組織横断プロジェクトなどの業務形態を考慮すると、分散型管理を前提にする必要がある[☆]。

そこで、マルチエージェントシステムを利用して、WorkWeb Systemを構成することを提案する。各々のエージェントは、各リソースあるいは各業務の管理を行う(図2)。

2.2 構成

WorkWeb Systemは、以下のエージェントで構成される。

パーソナルエージェント 各ユーザの秘書的な役割を果たすエージェントであり、そのユーザの時間リソースの管理を行うほか、情報収集や整理などの支援も行う。本システムでは、これらの支援内容

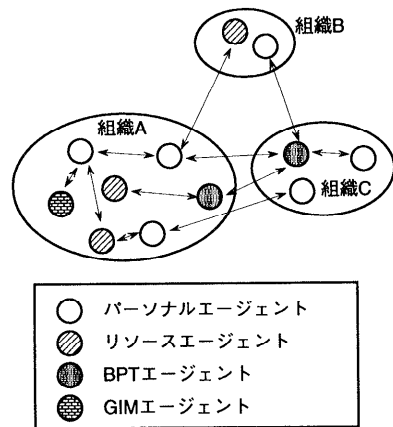


図2 マルチエージェントによるWorkWeb System
Fig. 2 A WorkWeb System image with multi agents.

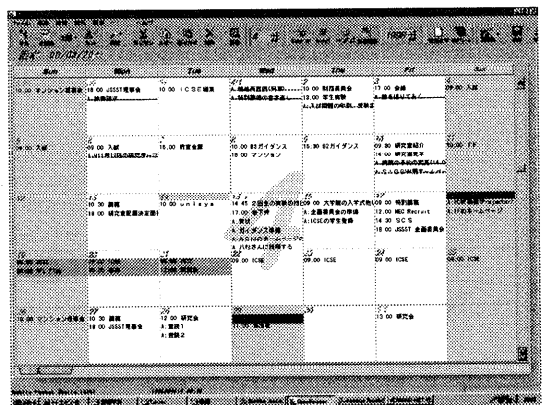


図3 Pochetの画面例
Fig. 3 A window example of Pochet.

ごとに個別のサブエージェントをパーソナルエージェント内に用意しているが、本稿では、時間リソース(スケジュール)を管理するエージェントのみに着目する。スケジュール管理エージェントはPochetという愛称であり、以後Pochetと表記する⁹⁾。Pochetの画面例を図3に示す^{☆☆}。

リソースエージェント モノ、カネなどのリソースを管理するエージェントである。具体的には予算や会議室を管理する。

GIMエージェント 組織の共有データを蓄積・管理するエージェントである。具体的には、営業達成目標、ワークフロー業務の達成実績などを蓄積する(GIMはGroup Information Managementの略)。

[☆] 社会学の分野で、高木ら¹⁶⁾は中央コントロールのないネットワーク型組織をポリエージェントシステムと呼んでいるが、これも同様の考えに立つものである。

^{☆☆} 本稿ではパーソナルエージェントを以下単にPochetと記すことがある。

BPT エージェント ワークフロー業務の制御、動的最適化を行うエージェントである。(BPTはBusiness Process Tacticsの略)

このうち、パーソナルエージェントとBPTエージェントがそれぞれ個人と組織の都合を代弁する代表的なエージェントであり、ワークフロー業務はこれらのエージェントの間でスケジュールの交渉を行いながら実行される。本稿ではこれらを中心に述べる。

WorkWeb Systemの横糸をなす横断ビューという論点で述べると、パーソナルエージェントは、ある個人に対してその個人の関わっている複数の業務を横断するビューを提供し、リソースエージェントは同一リソースを共有あるいは奪い合う複数の業務や個人についての横断的なビューを提供し、GIMエージェントは、たとえば「担当者別業務実績」のようにワークフロー業務から派生して生じるデータを蓄積したビューを提供する。

2.3 エージェント間通信

エージェント間の通信には、我々の開発したINA/LI環境を利用した⁶⁾。これは、MS-Windows上で提供されている通信ライブラリであり、エージェントどうしがピアツーピアの通信をできる環境を提供している。エージェントのアドレスはinalip:で始まるURL形式で指定する。

エージェント間のリソースの調整では、表1に示すようなプロトコルが使われる。ここで、RRA (Resource Requesting Agent) はリソースの要求者の立場、RMA (Resource Management Agent) はリソースの管理者の立場にあるエージェントを示す。たとえばリソースエージェントは基本的にRMA、BPTエージェントは基本的にRRAであるが、パーソナルエージェントは両方の立場をとりうる。

たとえば、AさんがBさんに仕事を依頼する場合、AさんのパーソナルエージェントがRRAの立場、BさんのパーソナルエージェントがRMAの立場をとり、前者から後者に対して作業依頼のプロトコルによる呼び掛けが行われる。

2.4 BPT エージェントの実現

BPTエージェントは、実行されるワークフローごとに1つ存在する。たとえば、出張伝票処理のワークフ

表1 リソースに関するエージェント間プロトコル
Table 1 Agent protocols for resource management.

| | |
|--------------|--------------------------------------|
| RRA から RMA へ | 空状況の問合せ、仮予約、本予約、利用開始、利用終了、作業依頼、キャンセル |
| RMA から RRA へ | 作業完了、キャンセル |

ローであれば、伝票1枚ごとにBPTエージェントが作られる。各BPTエージェントは自らが管理するワークフローについて、それが締切を守って運用されるよう計画し、各パーソナルエージェント(内のPochet)に個別の作業、すなわちワークフローの各作業ステップを依頼する。

したがって、BPTエージェントはRRAの立場である。パーソナルエージェントはRMAの立場であり、ユーザの時間という資源を管理している。

以下に、BPTエージェントの動作の概要を示す。BPTエージェントから各パーソナルエージェントへは3段階のメッセージ送信(納期間合せ、予約、実際の作業依頼)によって作業が依頼される(図4)。

- (1) ワークフロー起動時に、ワークフローの第1ステップの実行者であるユーザのパーソナルエージェント(PA₁とする)によって起動される。起動されたBPTエージェントを、以下単にBPT(斜体字)と記す。
- (2) PA₁からBPTにNEW-WORKFLOWというメッセージによって、ワークフロー定義が渡される。ワークフロー定義は、ノード定義、フローの経路定義、帳票定義、役割担当者の定義、日程計画などからなっている。ワークフロー定義は、Scheme風の書式で記述されているが、実際にはグラフィカルなワークフロー定義ツールから

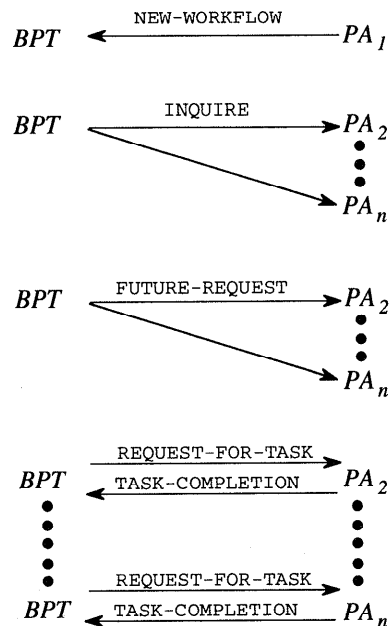


図4 BPT エージェントの動作概要
Fig. 4 Workflow management process by a BPT agent.

生成される¹⁸⁾。

- (3) *BPT* はワークフロー定義を解釈し、第2ステップ以降の作業実行者と、各ステップの締切を得る。ここで、第2ステップ以降の作業実行者のパーソナルエージェントを PA_2, PA_3, \dots, PA_n とする。
- (4) *BPT* は PA_2 に対して *INQUIRE* というメッセージを送り、作業を依頼した場合の終了時刻を尋ねる。 PA_2 はそれを予測し、回答する。当該ステップの締切までに終了しない場合には PA_2 は NG という回答をする。この場合、*BPT* はワークフロー運用の再計画を行う（再計画については後述）。
- (5) *BPT* は、 PA_2 から得られた回答を、 PA_3 に作業を渡す時刻とし、その時刻に作業を渡した場合に PA_3 が作業を終了する時刻を *INQUIRE* を用いて PA_3 に尋ねる。これを最後 (PA_n) まで繰り返し、ワークフローの全ステップが終了する時刻を予測する。
- (6) 予測されたワークフロー終了時刻が、そのワークフロー自身の締切や、組織の目標（これは GIM エージェントから得られる）に対して不都合（たとえば、達成目標に対して一定以上の不足がある）であれば、ワークフロー運用の再計画を行う。
- (7) *BPT* は *INQUIRE* の代わりに *FUTURE-REQUEST* というメッセージを用いて再び PA_2, \dots, PA_n に順に作業終了時刻を尋ねる。*FUTURE-REQUEST* と *INQUIRE* の違いは、*FUTURE-REQUEST* の場合パーソナルエージェント側で作業時間を予約しなければならない点である。
- (8) *FUTURE-REQUEST* によるワークフロー終了予測時刻が不都合であれば、管理者は再計画を行うことができる。
- (9) *BPT* は PA_2 に対して *REQUEST-FOR-TASK* というメッセージを送り、実際に作業を依頼する。 PA_2 は作業が終了すると *TASK-COMPLETION* というメッセージを *BPT* に送る。
- (10) *BPT* は、 PA_2 が実際に作業を終えた時刻が当初の予測と大きく異なる場合、 PA_3 以降の再計画を行う。
- (11) PA_3 以降についても同様に、*REQUEST-FOR-TASK* による作業依頼を行っていく。

上記手順は、ワークフローの計画と運用において特に問題が発生しない場合を基本に示した。複数のワークフローの計画が衝突した場合等については、「再計

画」の手続きが起動される。この場合の動作については、3章で詳しく説明する。

2.5 終了時刻の予測

上記手順中、パーソナルエージェントによる作業終了時刻の予測について補足説明する。Pochet は会議等の拘束時間（アポイントメント）と、課題とされている作業（To-Do）との2種類のデータを持っている。To-Do については締切時刻、必要工数のデータがある。必要工数のデータは、手入力と過去の経験に基づく自動予測の2通りの方法で得ている。またアポイントメントと To-Do の両方について、各項目ごとに優先度値のデータを有している。Pochet は与えられたワークフロー作業についてこれらの拘束時間や必要工数以外の時間を割り振って、終了時刻を予測する。場合によっては優先度の低いものを自動的にキャンセルすることがある。

3. ワークフローの再計画

3.1 再計画とは

再計画とは、ワークフロー定義の一部または全部について、臨時的定義変更を行い、作業者のエージェント（Pochet）にそれを通知し、定義変更の承諾を得ることであり、主として日程変更、経路変更、優先度変更が対象である。再計画が必要になるケースの一部を前章で紹介した。改めて、再計画が必要な場合を分類すると以下のとおりである。各場合とも再計画処理の仕方は基本的に同じである。

- a) *BPT* からの問合せ、予約、依頼に対して Pochet が NG (No Good) の返答をした。
- b) Pochet が予約を取り消した。
- c) ワークフロー開始後、ある作業者の業務遂行が不可能になったため、Pochet がキャンセルした。
- d) ワークフロー開始後、ある作業者の業務遂行が不可能になったため、締切の延期を依頼した。あるいは、ある作業者からの業務終了報告が締切時刻になっても得られなかった。
- e) 組織目標が変更された。
- f) その他管理者の判断による。

本研究では、再計画の戦略を再計画アルゴリズムとして部品化し、ワークフロー定義者が業務の性格によって再計画戦略を任意に指定する。これにより、再計画の方式を業務の性格に応じて柔軟に使い分けることが可能になる。以下に、アルゴリズムを紹介し、さらにそれらの指定方法について述べる。

3.2 再計画部品

これから説明する例題では、N01, N02, N03, N04,

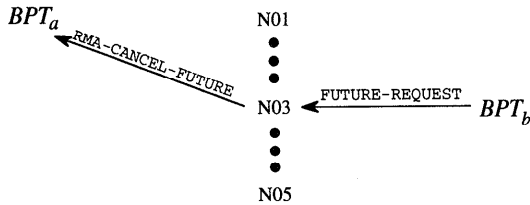


図5 複数のワークフローの干渉例

Fig. 5 A case of inter-workflow interference.

N05の5つのノードが順に業務を行うワークフローを仮定する。たとえば上記b)のケースとしては、すでにN01からN05までの各ノードのパーソナルエージェントには BPT_a による予約が入っていたが、別の BPT_b からN03のパーソナルエージェントに非常に重要な仕事が予約され、 BPT_a からの予約がキャンセルされた場合が想定される。本システムでは、RMA-CANCEL-FUTUREと名付けられたプロトコルでキャンセルが実現されている(図5)。

ここで、 BPT_a は再計画を行うことになるが、以後の動作は選択されるアルゴリズムにより異なる。本システムでは、以下の再計画アルゴリズム部品を提供した。

3.2.1 alt

altは当初予定の作業者に代えて、代理人に業務を依頼する。代理人はワークフロー定義中にあらかじめ定義されているか、altの引数として代理人名を与える。これはたとえば、(alt :to (URL URL ...)) のように書く。ここでURLは代理人のPochetのエージェントアドレスであり、複数指定も可である。

代理人が定義されていない場合と、代理人がNGを回答した場合はaltは失敗である。なお、代理人に依頼する際、締切と優先度は変更しない。

3.2.2 delay

delayアルゴリズムは、後のノードに日程を繰り越せる戦略である。delayは:stepと:maxの2つの引数を取り、:step刻みで:maxまでN03での締切を順に延長して、N03がOKを返せば成功。:maxまで延長してもN03がNGを返しつづければdelayは失敗である。または、N03の締切を延長した結果がN04の締切を越えた場合は失敗とする*。

3.2.3 hurry

hurryは前のノードに日程を繰り越せることでワークフロー業務を予定どおり終わらせようという戦略であり、N03のスケジュールを優先したい場合に用いるこ

とができる。ただしすでに前のノードまでの業務が終了している場合には必ず失敗する。

hurryは:step、:newpriの2つの引数をとる、次のようなアルゴリズムである。

- (1) N03の業務開始時刻を:step時間(たとえば1日)刻みで早くしながら問合せを行い、N03がOKを返すのを待つ。OKが返されれば次に進む。OKが返される前に、業務依頼時刻が現在時刻より早くなったらhurryは失敗である。
- (2) N03に、1で決めた業務依頼時刻までに業務を依頼できるよう、N02までのワークフローを計画し直す。ここで、N02以前のノードについては、業務の優先度を:newpriに変更し、再計画しやすくする。N02以前のノードに対する時間配分比は、元の定義と同じ比率とする。
- (3) N02までの再計画に成功すればhurryは成功、そうでなければ失敗である。

3.2.4 pri

priは、N03における業務優先度を局所的に上げることによって計画どおりに業務を進めようとする戦略である。この戦略は他の作業者には日程の繰り越せを与えず、当該作業者の他の業務を犠牲にすることになるので、N03の担当者の地位が比較的下位で、重要な業務のワークフローの場合に用いることできる。

priは、引数:newpriをとり、N03に限り優先度を:newpriに変更する。優先度を変更後、N03がOKを返せばpriは成功、そうでなければ失敗である。

3.2.5 force-end

force-endは、ワークフローの最終ノードの締切遵守を最優先に考え、ワークフロー内の各作業者についてすべて業務優先度を上げるという戦略である。

force-endは、2つの引数:deadline、:priをとる。最終ノードの締切を:deadlineに変更し、再計画後の優先度を:priに変更する。

たとえば、(force-end :pri 20 :deadline "1997/8/1 17:00:00")は、優先度を20に変更し、最終ノードの締切を97年8月1日の17時にする。

3.2.6 report

reportは再計画せず、状況を管理者に報告する。管理者がだれであるかは、ワークフロー定義中に示されている。reportはつねに成功する再計画アルゴリズムという位置づけになっている。

3.3 再計画アルゴリズムの指定方法

以上述べた再計画アルゴリズムの適用順、適用方法について、各ワークフロー業務ごとに指定する必要がある。

* これを失敗としない方法も考えられる。

現在、BPT エージェントは、起動時にワークフロー定義を参照して動作しているので、このワークフロー定義の中に、これらを指定する書式を導入した。最初の2つ (define-plan, define-gim-plan) がこの範疇である。

また、それ以外に実行時に再計画アルゴリズムをアドホックに起動するためのプロトコルも提供する。

3.3.1 define-plan

define-plan はノードごとにアルゴリズムの適用順を指定する。この場合、ワークフロー定義者がアルゴリズム適用順や各種パラメータを指定することになる。

たとえば、次のように利用する。

```
(define-plan 'N03
  '((alt) (delay :step "1d" :max "5d")))
```

この例は、'N03 から NG が返された場合、まずアルゴリズム alt を適用し、それが失敗したら delay を :step が 1d :max が 5d というパラメータで適用することを示している (d は日を表す単位)。

これらのアルゴリズムの適用は、再計画が発生するたびに、また各ノードごとに最初のアルゴリズムから順に行う。なお、仮に report が明記されていなくても、すべてのアルゴリズムが失敗すれば、最終的には report が自動的に起動される。これは、定義ミスにより再計画失敗がだれにも報告されないうままワークフローがペンディングをすることを避けるためである。また、再計画がループに陥った場合 (たとえば、複数のワークフローが優先度を上げる競争を始めた場合) にも report が起動されるようになっている。本システムでは再計画の完全な自動化を目指しているわけではなく、このようにして問題点を早期に発見して管理者による対応をとってもらうことも重要な効果と考えている。

すべてのノードについてアルゴリズムリストを指定するのは煩雑なので、ノード ID を 'default とすることにより、デフォルトのアルゴリズムリストを指定できるようにした。'default に対するアルゴリズム指定を省略すると、report が指定されたのと同様である。

3.3.2 define-gim-plan

define-gim-plan は、組織目標未達の場合のアルゴリズム適用順を特に指定する。この場合も、ワークフロー定義者がアルゴリズム適用順や各種パラメータを指定する。たとえば、

```
(define-gim "inalip://pc01/gim")
(define-gim-plan
  '((force-end :pri 20
    :deadline "ME 17:00:00")))
```

とすると、inalip://pc01/gim で指定される GIM に蓄積されている組織目標が未達である場合、あるいは組織目標が変更された結果目標未達となった場合に、force-end アルゴリズムを適用する。

たとえば、受注処理ワークフローの場合、組織目標としては月間の売上件数や金額が考えられる。GIM はこの目標と売上実績 (これは、ワークフローの完了時に更新される) を格納しており、BPT エージェントはこれらを参照して、月末近くで目標未達の場合には再計画を行う。

なお、define-gim-plan では、define-plan の場合と異なりノードを指定しない。したがって、ノード指定の必要なアルゴリズム (alt など) は利用できない。

3.3.3 管理者による、アドホックな再計画

指定した再計画アルゴリズムを BPT に強制的に起動させるプロトコル FORCE-REPLAN を提供する。たとえば、FORCE-REPLAN 中で

```
:replan ('N02 '((force-end :pri 20
  :deadline "1997/7/22 17:00:00")))
```

と書くと、force-end アルゴリズムを強制的に起動できる。各種パラメータは管理者が指定する。このプロトコルの送付権限は、ワークフローの管理者に限定する。

本機能は次のような場合有効である。ワークフローの起動時に終了予想時刻が管理者に対して表示されるが、管理者がその時刻に不満であるような場合、それより早い時刻を指定した force-end アルゴリズムによる再計画を強制起動することができる。

3.3.4 作業員による、アドホックな再計画

たとえば、ある作業員が予定どおり業務ができなくなったとき、当該作業員が自主的に代理作業員を見つけ、代理業務の依頼の根回しが終わっているような場合、ワークフロー管理者を通さず、当該作業員の指示によって再計画が行われると便利である。このような機能を提供することは、トップダウンなワークフロー管理にこだわらず、オフィスの仕事の進め方の実態に合わせることにつながり、重要であると考えている。

そこで、キャンセルや業務の延期依頼のプロトコル中で特定の再計画アルゴリズムの起動を Pochet が指定できるようにした。BPT エージェントは、あらかじめ定義された再計画アルゴリズムの適用順ではなく、新たに指定されたアルゴリズムを採用する。これにより、Pochet 側から予約の取消、業務の中断等の際に明示的に代理人を指定できる。

この場合のアルゴリズム・パラメータ指定は作業員が行うことになる。現在自由に指定できるようになっているが、将来的には制限も必要と考えられる。何を

どう制限すべきかは実用経験を踏んだうえで考えて行く必要がある。

3.4 例

たとえば、「担当者 → 係長 → 課長 → 部長」というワークフローの場合、課長や部長のノードには hurry を、係長や担当者のノードには pri を指定しておくことにより、課長や部長の都合を優先しつつ締切を守るような再計画を行うことができる。同じフローでも急ぎでない業務の場合には各ノードに delay を指定すると、締切遵守にさほどこだわらない再計画ができる。このように、業務や各作業者の性格に応じた柔軟な再計画が可能である。

4. 評価

WorkWeb System の根幹をなす BPT エージェントと Pochet を Windows NT, Windows 95 の環境で動作するよう開発した。エージェント通信基盤には INA/LI⁶⁾を用いた。開発言語は Visual C++ と Delphi である。

90 から 133 MHz 程度の Pentium を搭載したパーソナルコンピュータを 10 Mbps のイーサネット で接続した環境で実験したところ、ワークフロー 1 ノードあたり 2~3 秒で予約 (INQUIRE および FUTURE-REQUEST の処理) が完了した。したがって、たとえば 3 ノードのワークフローの場合、予約には約 10 秒弱の時間がかかることになる。ワークフロー再計画が起動された場合、やはり 1 ノードあたり 2~3 秒程度の時間が再計画のために必要である。たとえば、3 ノードのワークフローの最初のノードで delay アルゴリズムが起動され、後続のすべてのノードでそれが受理された場合に約 10 秒弱の時間がかかる。したがって、再計画に失敗するなどして複数のアルゴリズムが適用されたり、ループが検出される場合には 1~2 分の時間がかかることもありうる。以上の時間には、Windows OS におけるイベント処理やタスクスイッチングの時間が含まれている。この性能は、現状でも人間による交渉より十分早いという意味では十分に考えているが、ノードあたりに要する時間については改善の余地が残されている。

Pochet はすでに研究所内で実用化されており、各個人のスケジュールデータは十分入力されている。BPT エージェントは、現在のところ実際のワークフローシステムと結合されるに至っていないが、紙ベースで処理されているワークフローに関するデータを手入力し、Pochet と通信させてシミュレーションを行うことはできる。我々の研究所では管理職の出張が多いため、

論文や特許の査閲・承認手続きを締切までに完了するためには原稿をいつ上司に提出すればよいかの問題になることが多かったが、BPT エージェントを用いることによって、提出時期の見積りが容易になった。また、再計画機能により、原稿提出後に発生した管理職の出張にも対処できる。

5. 議論

最近、ワークフローの動的変更に関する研究が発表されているが、多くはワークフロー定義の変更にとまらなう処理についての研究である^{1),5),7)}。WorkWeb System はワークフローの定義変更をとまらなう、実運用時のアドホックな問題に対処できる。

よって本稿で述べた意味のワークフローの再計画に関する研究は他に類がないが、市場のワークフロー製品には類似の機能が部分的に導入されている。たとえば、Staffware¹⁴⁾では、ある作業者の業務が締切までに終わらなかった場合に起動すべき例外処理が定義できる。しかしながら本稿で述べたように再計画処理を系統的かつ柔軟に実現しているものはない。

また、マルチエージェントを用いたスケジュール管理についての研究については、会議のスケジュール調整についての研究が多く発表されている^{2),15),20)}。会議のスケジュール調整とワークフローのスケジュール調整は方式も異なっているが、ロバストネスという観点ではワークフローへのエージェント応用の方が容易であるといえる。なぜならば、会議時間調整の場合、ダブルブッキング等のトラブルを防ぐためには、エージェントへのスケジュールデータ入力にはつねに完全であることを要求される。一方、ワークフローの場合は多少の入力漏れがあっても締切までに各担当者が業務をこなせるかどうかには大きな影響が出ないことも多く、出張や時間のかかる作業などが入力されていればかなりのトラブルに対し事前の対処が可能になる。

エージェントを用いたスケジュール計画にはジョブショップスケジューリング¹¹⁾のような生産計画問題への応用もある。このような問題との最大の相違点は、意志の異なる複数の管理者によって管理されている複数のワークフローを処理しなければならず、このため全体として何が最適なのかを決定する尺度が一通りに定まらないことである。すなわち、ジョブショップスケジューリングは探索問題であるのに対し、WorkWeb System では交渉問題が本質である。交渉支援という立場から、今後、WorkWeb System の導入により他の管理者との業務の調整が円滑に行われるようになったかどうかという観点で評価していく必要がある。こ

れは今後の課題である。

WorkWeb System のようなシステムでは各作業者が自分の予定を入力してくれることが前提となっている。この点を問題とする意見をいただくことも多いが、我々は以下の理由により楽観視している。

- (1) 今後のモバイルコンピューティングの発達によって、外出先で発生した要件を手軽に入力し、その結果が即座に計算機ネットワーク上の他の計算機に通知される仕組みができると考えられる。
- (2) かつてスケジュール管理システムが社会的に受け入れられないことを指摘した³⁾Grudin が最近の論文で発表した調査結果⁴⁾によれば、少なくとも一部の米国の大手企業では従業員全員による電子スケジュール管理が積極的に使われている。
- (3) 本システムの導入により、負荷分散の適正化、作業実績の客観的評価、無理なスケジュールの事前察知などの効果があり、末端の作業者にとってもメリットがあるため、スケジュール入力が促進される。
- (4) 会議開催については通知した時点でスケジュールに自動登録することも可能であり、これについては実現済である^{8),10)}。その他にも電子メールなどから自動的にスケジュールを獲得することが可能である。

6. おわりに

本論文では、従来のワークフローシステムでは複数の業務の相互干渉が考慮されていないという問題点をあげ、それを解決するものとして WorkWeb System を提案した。従来は個人のスケジュールや他のワークフローの予定を勘案した計画が不可能であったが、WorkWeb System ではエージェントを用いてこの問題を解決している。よって、優先的なワークフローや急な出張が発生した場合に、他のワークフローの再計画を行うなど、相互干渉を考慮することができる。

マルチエージェントシステムを用いたことにより、意志決定が分散化する。たとえば、組織の都合（達成目標）と個人の都合の調和をとって、仕事の裁量性を最大限に許しながら全体としての目標を達成する計画策定を交渉ベースで行うのに適した実現方式である。場合によってはエージェントが嘘をつくこともありうる。これについての考察は別論文¹⁹⁾で行っているが、有用性の実証評価は今後の課題である。

WorkWeb System はその全体が完成してはいないが、今回報告した機能だけでもワークフローの完了日

が事前に見積りできる、担当者のお出張などによるワークフロー進行上のトラブルを早期に予期できる等の効果がある。

本稿で提案した機能は製品¹³⁾への導入を検討している。また、プロジェクト管理への応用²²⁾も検討している。

謝辞 熱心に議論していただき、適切な助言をいただいた真名垣昌夫所長、宮井均部長、宮下敏昭課長をはじめとする NEC ヒューマンメディア研究所の諸氏に感謝します。

参考文献

- 1) Ellis, C., et al.: Dynamic Change Within Workflow Systems, *Proc. Conf. on Organizational Computing Systems (COOCS '95)*, pp.10-21, ACM (1995).
- 2) Garrido, L. and Sycara, K.P.: Multi-Agent Meeting Scheduling Preliminary Experimental Results, *Proc. Int. Conf. on Multi Agent Systems (ICMAS '96)*, pp.95-102 (1996).
- 3) Grudin, J.: Why CSCW Applications Fail: Problems in the Design and Evaluation of Organizational Interfaces, *Proc. ACM 1988 Conference on Computer Supported Cooperative Work (CSCW '88)*, pp.85-93 (1988).
- 4) Grudin, J. and Palen, L.: Why Groupware Succeeds: Discretion or Mandate?, *Proc. 4th European Conference on Computer-Supported Cooperative Work (ECSCW '95)*, Marmolin, H., et al. (Eds), pp.263-278, Kluwer Academic Publishers (1995).
- 5) Herrmann, T.: Workflow Management Systems: Ensuring Organizational Flexibility by Possibilities of Adaption and Negotiation, *Proc. Conf. on Organizational Computing Systems (COOCS '95)*, pp.83-94, ACM (1995).
- 6) Ishiguro, Y., Tarumi, H., et al.: An Agent Architecture for Personal and Group Work Support, *Proc. Int. Conf. on Multi Agent Systems (ICMAS '96)*, pp.134-141 (1996).
- 7) Jaeger, T. and Prakash, A.: Management and Utilization of Knowledge for the Automatic Improvement of Workflow Performance, *Proc. Conference on Organizational Computing Systems '95*, pp.32-43, ACM (1995).
- 8) 片岡 章ほか：エージェント交渉による会議時間調整支援，第 54 回情報処理学会全国大会論文集，3V-4 (1997)。
- 9) 喜田弘司ほか：エージェント間交渉によるスケジュールの調整方式，情報処理学会グループウェア研究会，GW-21-16 (1997)。
- 10) 喜田弘司ほか：オフィスワークを支えるスケジュー

ールエージェント—スケジュールエージェント Pochet の評価, 第 55 回情報処理学会全国大会 論文集, 5Y-5 (1997).

- 11) Liu, J. and Sycara, K.P.: Multiagent Coordination in Tightly Coupled Task Scheduling, *Proc. Int. Conf. on Multi Agent Systems (ICMAS '96)*, pp.181-188 (1996).
- 12) 松井一郎: 業務の連携を自動化, 時間短縮と管理を実現, ワークフロー管理ソフトが日本でも利用可能に, 日経コンピュータ, 94年5月2日号, pp.57-67 (1994).
- 13) 中川路充ほか: ワークフローシステム Flexflow のエージェントによる実装, 情報処理学会グループウェア研究会, GW-20-5 (1996).
- 14) 日本ユニシス: Staffware のワークフローパターン (2), http://www.unisys.co.jp/product_info/staffware/pattern2.htm (1998年6月現在存在する WWW ページ).
- 15) Sen, S. and Durfee, E.H.: On the design of an adaptive meeting scheduler, *Proc. 10th IEEE Conf. Artificial Intelligence Applications*, pp.40-46 (1994).
- 16) 高木晴夫ほか: マルチメディア時代の人間と社会—ポリエージェントソサエティ, 日科技連出版社 (1995).
- 17) 垂水浩幸: グループウェア・ワークフローの研究動向, 電子情報通信学会知能ソフトウェア工学研究会, KBSE-97-30 (1998).
- 18) 垂水浩幸, 金政ふじ, 小笠原章夫: ワークフロー技術とその応用, 計測と制御, Vol.34, No.12, pp.932-936 (1995).
- 19) Tarumi, H., Kida, K., Ishiguro, Y., Yoshifu, K., and Asakura, T.: WorkWeb System - Multi-Workflow Management with a Multi-Agent System, *Proc. ACM International Conference on Supporting Group Work (Group '97)*, pp.299-308 (1997).
- 20) Wada, Y., et al.: An agent oriented schedule management system - IntelliDiary, *Proc. 1st Int. Conf. and Exhibition on the Practical Application of Intelligent Agents and Multi-Agent Technology*, pp.655-667 (1996).
- 21) 横井康二, 荒尾辰之: ワークフローシステムによる新契約査定業務の改革, 日立評論, Vol.77, No.6, pp.43-46 (1995).
- 22) 吉府研治, 喜田弘司, 垂水浩幸: マルチエージェ

ントによるプロジェクト管理システム, 情報処理学会グループウェア研究会, GW-22-3 (1997).

(平成 9 年 8 月 28 日受付)

(平成 10 年 4 月 3 日採録)



垂水 浩幸 (正会員)

昭和 35 年生. 昭和 63 年京都大学大学院工学研究科情報工学専攻博士後期課程修了. 同年日本電気 (株) 入社. ソフトウェア工学, ヒューマンインタフェース, グループウェア

の研究開発に従事. 平成 9 年 4 月より京都大学助教授. 平成 2 年情報処理学会学術奨励賞, 情報処理学会創立 30 周年記念論文賞, 平成 9 年度坂井記念特別賞各受賞. 日本ソフトウェア科学会理事. 工学博士.



喜田 弘司 (正会員)

昭和 42 年生. 平成 5 年岡山大学大学院工学研究科情報工学専攻修士課程修了. 同年日本電気 (株) 入社. グループウェア, エージェントの研究開発に従事.



柳生 弘之

昭和 37 年生. 昭和 60 年大阪電気通信大学工学部電子工学科卒業. 同年 (株) NEC 情報システムズ入社. 形態素解析, MERMAID, グループウェア等の開発に従事.



石黒 義英 (正会員)

昭和 39 年生. 平成 2 年京都大学大学院工学研究科電気工学第二専攻修士課程修了. 同年, 日本電気 (株) 関西 C&C 研究所入所. 平成 9 年より同社ヒューマンメディア研究所勤務. オフィスにおけるエキスパートシステム, グループウェア, エージェントシステム等の研究に従事.