

## マルチエージェントによるプロジェクト管理システムの開発

吉府 研治      喜田 弘司      朝倉 敬喜      宮下敏昭  
{yoshifu, kida, asakura, miyasita}@hml.cl.nec.co.jp

NEC ヒューマンメディア研究所

プロジェクト管理業務である作業割当、進捗管理、日程調整をマルチエージェントを用いることにより自動化するプロジェクト管理システム Done-Do-Rhythm(ダンドリズム)について報告する。本システムは、プロジェクト定義、進捗率表示を行う業務計画指示エージェント(VD-Ag)、あらかじめ定義したプロジェクトガントチャートに従って、タスクの割当依頼、進捗促進を促す催促、進捗遅れによる関連タスクの日程調整交渉を行うプロジェクト管理エージェント(PRO-Ag)と、作業員自身のスケジュールを管理し、タスクに関する情報を PRO-Ag とやりとりするスケジュール管理エージェント(Pochet)から構成される。本システムは、ORB を用いたエージェント通信基盤 INA/LI を使い、Microsoft WindowsNT4.0/95 上で実装した。このシステムを評価した結果、作業割当時間短縮、進捗管理・日程調整コスト削減に関して有効性を確認した。

## Development of Project Management System based on a Multi-Agent System

Kenji Yoshifu      Koji Kida      Takayoshi Asakura      Toshiaki Miyashita

NEC Corp., Human Media Research Labs.

This paper proposes a project management system based on a multi-agent system called Done-Do-Rhythm, to support task assignment, progress management and project scheduling. This system is composed of three kinds of agents, PRO-Ag, VD-Ag, and Pochet. VD-Ag provides a GUI for business planning, which helps a manager to define project schedules and to make visible on worker's job progresses. PRO-Ag assigns tasks to workers according to project definition, urges workers, and coordinates project schedules when a task is delayed. Pochet manages worker's schedules, communicates with PRO-Ag, and exchanges information about task. This system is built on our multi-agent platform called INA/LI and is implemented on Microsoft Windows NT4.0/95. We evaluated this system, and confirmed effectiveness of resource allocation, progress management and schedule coordination.

## 1. はじめに

我々は、マルチエージェント技術を利用したオフィスシステムの研究開発を行ってきた[1][2]。今回は、特にプロジェクト管理業務である作業割当、進捗管理および日程調整をマルチエージェントにより自動化するシステムについて報告する。

従来のプロジェクト管理システムは、プロジェクトの設計、進捗管理、スケジュール調整シミュレーション機能を有している[3][4]。しかし、これらの機能は、作業者のスケジュールをあまり考慮しない作業割当、進捗状況の非能動的な確認・催促、日程調整時に管理者の多大な労力を要する手動シミュレーションなど問題点が多く存在する。管理者と作業者との間で作業依頼・進捗問合せ・日程変更依頼などの情報を効率的にやりとりする仕組みが望まれていた。

これらの仕組みは、我々が研究開発を行ってきたマルチエージェント通信基盤(INA/LI)[1][2]を用いることで、実現可能であると考え、システム開発を行った。

本稿では、プロジェクト管理業務をマルチエージェントにより自動化するシステム Done-Do-Rhythm<sup>1</sup>(ダンドリズム)[5]について紹介する。また、このシステムを Microsoft Windows NT4.0/95 上で実装、評価したので、その内容についても報告する。

## 2. PJ管理支援

### 2.1. 概要

我々が想定しているプロジェクト(PJ)管理業務の例を説明する(図1)。

(1)線表作成: PJ管理者は、自分のマシンを用いて管理したいプロジェクトのガントチャートを作成する。作成項目としては、個々のタスク・サブタスクの内容・期間、作業割当者名などがある。

(2)作業割当: PJ管理サーバは、上記ガントチャート情報に基づき、作業割当者に対して作業割当を行う。

(3)進捗報告: 作業者は各自のマシンに作業の進捗状況を入力する。

(4)進捗管理: PJ管理サーバは各作業者が入力した進捗状況を管理し、管理者に提示する。

(5)遅延報告: 作業者は自分の都合で割当作業が締切日より遅れそうなときにPJ管理サーバに遅延を報告する。

(6)再計画: PJ管理者はプロジェクトの都合で納期を早める必要がある場合、ガントチャートの最終タスクの締切日を早める。

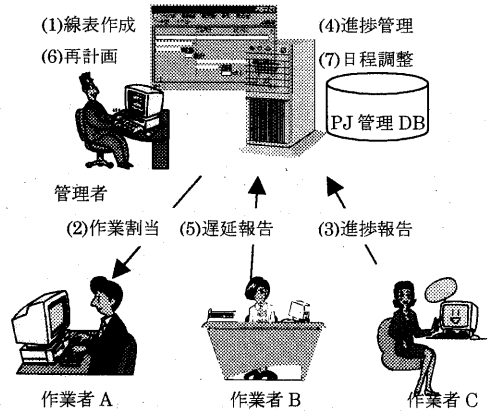


図1 PJ管理の一般手順

(7)日程調整: PJ管理サーバは、(5)遅延報告や(6)再計画がなされた場合に、その影響を受けるタスクに割り当てられた作業者に対して日程調整を行う。

### 2.2. システムの狙い

特に以下の点をシステムのねらいとする。

#### 1. 作業者のスケジュール、作業実績を考慮した計画作成支援

PJ計画を立てる際に、管理者が計画で決めたタスクをトップダウン的に割り当てるのではなく、作業者の個人スケジュールを参照しながら対話的に作業者を割り当てる手段を提供する。このとき、作業者の作業実績を管理者に提示する手段も提供する。これらにより、無理なPJ計画による混乱を減少させ、PJ計画時の日程が遵守されやすくなる。

#### 2. 自動進捗管理

各作業者が個人で管理している作業の進捗状況、完了予定日などをPJ管理サーバが定期的に収集して進捗データを管理し、進捗が遅れている作業者に対して、催促・警告を自動的に行う。これらにより、従来管理者が自発的に行ってきた進捗管理が自動化され、管理コストが減少する。

#### 3. 横断的な日程調整

作業遅延、納期変更に伴うPJの日程調整を、あらかじめ設定しておいた戦略に基づいて自動的に行う。また他のPJ兼務者、非PJの一般の工数も考慮した調整を行う。これらにより、オフィスで働く作業者の作業実態に即した日程調整が可能となる。

## 3. マルチエージェントによるPJ管理システム

<sup>1</sup> 「リズムカルに Done/Do の管理をする」の意味。商標出願中。

### 3.1. システム概要

Done-Do-Rhythm は、以下に詳述する業務計画指示エージェント (VD-Ag)、PJ管理エージェント (PRO-Ag)、スケジュール管理エージェント (Sch-Ag:Pochet) から構成される。図 2 に Done-Do-Rhythm の全体構成を示す。

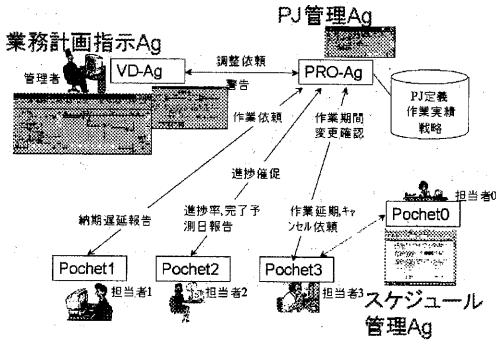


図 2 Done-Do-Rhythm の構成

#### (1)業務計画指示エージェント(VD-Ag)

PJ管理用の GUI で PJ管理者が用いるエージェント(図 3)。タスク期間、割当作業者などの定義や進捗率・完了予測日間合せのタイミングの定義を行う。スケジュール管理エージェントから収集した進捗率が予定より低ければ、催促メッセージを表示する。また、日程調整結果をダイナミックに表示する。

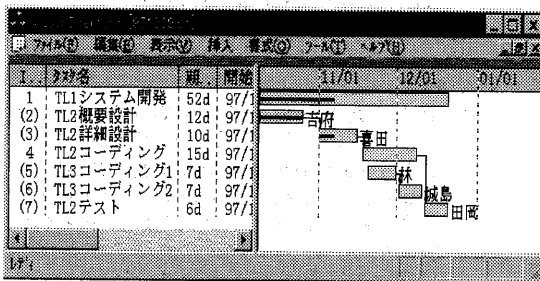


図3 業務計画指示エージェント画面例

#### (2)PJ管理エージェント(PRO-Ag)

個々のPJを管理し、動的にPJを再計画するエージェント。各エージェント間の通信のやりとりをグラフィカルに表示する GUI を有する(後述)。PJ管理エージェントは、スケジュールに対して仕事を依頼し、進捗率・完了予測日を問合せ。PJの締切を守るよう、納期を予測し、スケジュールを調整する。忙しくて仕事を受けてくれない人がいれば、別の作業者に仕事を依頼する。別の作業者の選び方としては、あらかじめ指定しておく場合と、進捗状況の良い人を管理者が選ぶ場合がある。また、「再計画アルゴリズム」として、「職位の高い人の

スケジュールを優先する」、「納期最優先」などを使うことにより、各組織業務の運用形態にあわせた再計画が可能である。

#### (3)スケジュール管理エージェント(Pochet)

個人のスケジュールというリソースを管理するエージェント(図 4)。PJ管理エージェントから仕事の依頼や進捗率・完了予測日の問合せを受けると自動的に応答する[6]。

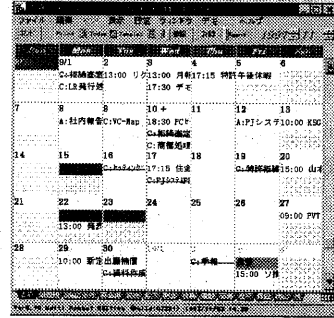


図 4スケジュール管理エージェント画面例

### 3.2. システム詳細

#### 3.2.1. 通信方法

INALI エージェント通信基盤では、各エージェントに URL のような固有のアドレスがつく。

業務計画指示 Ag:inalip://machine1/people/ysf  
 スケジュール管理 Ag:inalip://machine2/people/tkn  
 PJ管理 Ag:inalip://machine3/pro/1

各エージェントは、上記のアドレスを用いて他のエージェントと通信する。各エージェント間の通信の様子は、プロジェクト管理エージェントの GUI でもあるエージェント通信ブラウザ(図 5)を通して参照できる。

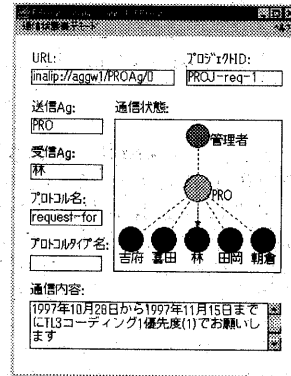


図 5 Ag 通信ブラウザ(PJ管理 Ag-GUI)

このブラウザでは、通信中のエージェントを色分けしたノードで表現し、使用している通信プロトコルと通信内容をテキストで表現する。

### 3.2.2. 通信プロトコル

各エージェント間で用いられる通信プロトコルの記述形式はKQML[7]準拠する。各プロトコルの:content 部に通信に必要な情報を埋め込む。以下に各エージェント間で用いる通信プロトコルの分類を示す。

- (1)PJ 管理 Ag→スケジュール管理 Ag
  - 1. 作業依頼関連(予約、実行、変更など)
  - 2. 問合せ関連(作業実行可能性、進捗率など)
  - 3. 催促関連
- (2)スケジュール管理 Ag→PJ 管理 Ag
  - 1. タスクキャンセル関連(予約、実行)
  - 2. タスク遂行状況報告関連(遅延、完了)
- (3)業務計画指示 Ag→PJ 管理 Ag
  - 1. 起動関連(新規、再起動)
  - 2. PJ 管理データ関連(取得、変更)
- (4)PJ 管理 Ag→業務計画指示 Ag
  - 1. 警告
  - 2. 管理者調整要求

例えば、作業依頼(予約)の:content 部は図 6 のように表現される。

```

:content (future-request
:request-id id
:order-time time
:deadline time
:priority value)
  
```

図 6 作業依頼:content 部

### 3.2.3. 動作方法

Done-Do-Rhythm の動作方法について、各フェーズごとに述べる。

- (1)プロジェクト設計、起動
- (1) get-data
  - (2) new-project
  - (3) inquire(acceptable)
  - (4) future-request
  - (5) request-for-task

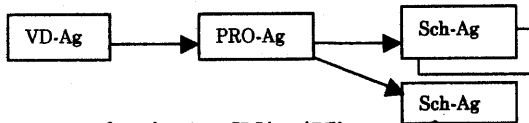


図 7 プロジェクト設計、起動フェーズ

PJ 管理者は、VD-Ag を用いて PJ を設計する。各タスクに作業者名を割り当てる際に、VD-Ag は *get-data* を発行して過去のプロジェクトの実績を参照する。このとき VD-Ag は過去の実績を表示する(図 8)。この実績は、PRO-Ag が各作業者の Sch-Ag から収集した作業実績(通算工数、遅延日数など)を集計したものである。ここで管理者が作業者を割り当てる際に優先項目(遅延が少ないなど)を指定すれば、その項目が優れている順番に作業者をリストアップするので、その中から管理者は割り当てる作業者を

名前	タスク名	開始日	終了日	通算工数	遅延日数	遅延率	遅延日数	遅延率
吉府	TL1TEST	97/9/1	97/10/5	30	26	5	25	
田岡	TL3詳細設	97/07/25	97/08/14	20	10	10	50	
吉府	TL3概要設	97/06/28	97/07/11	12	10	-2	-17	
林	TL2開発計	97/08/27	97/08/28	2	2	0	0	
田岡	TL2テスト	97/10/10	97/10/24	15	0	0	0	
林	TL2コーディ	97/09/03	97/09/20	10	20	-10	-90	
吉府	TL2設計	97/07/25	97/08/05	30	22	8	12	

図 8 過去のプロジェクト実績

決定すればよい。リスト上位の者を自動割当することも可能である。

次に、管理者は PJ 管理データの収集・作業者催促タイミング、日程調整アルゴリズムを設定する(設定内容については後述)。

PJ 計画作成が終了すると、VD-Ag は *new-project* を発行する。この命令の:content となるプロジェクト動作定義内容を図 9 に示す。

```

(new-project
(define-schedule-file-contents (
1,TL1システム開発,78,97/10/01,98/01/30,,0,
2,TL2設計,17,97/10/01,97/10/24,,喜田,5,
3,TL2コーディング,37,97/11/01,97/12/26,2,"吉府",5,
4,TL2テスト,1,19,98/01/05,98/01/26,3,朝倉,5,))
(define-project-file-contents (
(define-action (
(INQ (PROGR NON 0 2 0))
(URG (PROGR ON 0 36 0))))
(define-priority "5")
(define-plan '(shift-term pri "4"))))
  
```

図 9 プロジェクト動作定義内容(一部略)

PRO-Ag は次のステップの順で Sch-Ag に作業依頼を行う。

1. PJ 定義内容を基に各作業者の Sch-Ag に割り当てたタスクが実行可能かを問い合わせる(*inquire :acceptable*)。すべての Sch-Ag から実行可能であるとの回答がある場合は 2 のステップに進む。作業者の Sch-Ag から、割り当てたタスクが実行不可能との回答がある場合は、事前に定義した別の作業者に問い合わせを行う。別の作業者も実行不可能のときは、PJ 定義に基づく初期タスク依頼が失敗したことを VD-Ag(PJ 管理者)に報告する。
2. 問合せた作業者の Sch-Ag に対してタスクの仮予約(*future-request*)を行う。
3. PJ の最初のタスクの開始日が到来したときに、最初のタスクを処理する Sch-Ag に作業開始依頼(*request-for-task*)を発行する。

#### (2) 能動的進捗管理

PRO-Ag は、VD-Ag であらかじめ定義したタイミングで関連するエージェントに対して問い合わせる(*inquire*)。問合せる項目として進捗率(

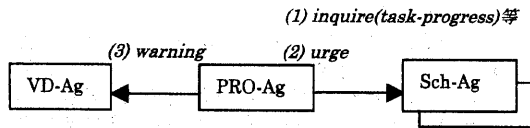


図 10 能動的進捗管理フェーズ  
task-progress)、完了予測日 (completion-expected)、通算工数 (spend-time)、残業時間 (over-time)がある。次に、PRO-Ag が獲得した問合せ項目の値があらかじめ定義した催促条件 (例 進捗率 < 30%、通算工数 > 10 時間など) に該当したときに、PRO-Ag は Sch-Ag に催促メッセージ (urge) を発行し、VD-Ag に警告メッセージ (warning) を発行する。

### (3) P J 再計画

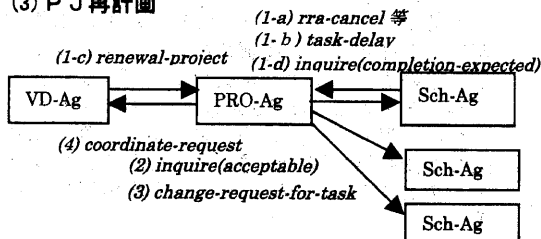


図 11 P J 再計画フェーズ

P J 再計画が自動実行されるのは、1-a) 作業者が Sch-Ag で依頼作業をキャンセルする時 (rra-cancel)、1-b) 作業者が Sch-Ag で依頼作業の遅延を報告する時 (task-delay)、1-c) 管理者が VD-Ag で納期短縮など P J 計画を変更する時 (renewal-project)、1-d) PRO-Ag が Sch-Ag に問合せた完了予測日が関連タスクの作業開始予定日を超える時 (inquire(completion-expected)) である。PRO-Ag は、上記イベントが発生すると、VD-Ag で管理者があらかじめ又は P J 稼働中に設定した日程調整アルゴリズムを動作させる (図 12)。

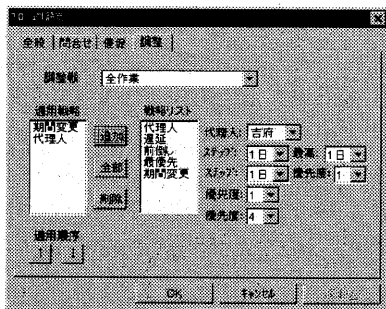


図 12 日程調整アルゴリズム設定  
選択できるアルゴリズムには、代理人、遅延、前倒し、最優先及び締切遵守がある<sup>1)</sup>。

それぞれの調整方法を以下に示す。  
代理人：当初予定の作業者に代えて、別の作業者に作業を依頼する。

遅延：後のタスクに日程を搬寄せる。

前倒し：前のタスクに日程を搬寄せる。

最優先：該当タスクの優先度を局所的に上げる。

締切厳守：プロジェクトの最終サブタスクの締切厳守を最優先に考え、プロジェクト内の各作業についてすべての作業優先度を上げる。

これらのアルゴリズムは、全てのタスクまたは個別のサブタスクに対して適用できる。例えば、全てのタスクに遅延戦略を適用するが、あるサブタスクにだけ代理人戦略を適用する、ということが可能である。

実際に各タスクを遅延・前倒しさせる期間は、調整アルゴリズム設定時に決定するパラメータと、P J 全体の残り作業期間から算出する。

上記で選択したアルゴリズムに従い、作業者の Sch-Ag に対して作業依頼 (inquire:acceptable) または作業変更依頼 (change-request-for-task) を行う。

以上の P J 再計画が成功すれば、PRO-Ag は管理者の VD-Ag に対して、再計画後の P J の新規日程を出力する。失敗すれば、エージェントによる日程調整交渉が失敗した理由を VD-Ag に報告する (coordinate-request)。

## 4. DoneDoRhythm 実装と評価

### 4.1 構成

Done-Do-Rhythm は、マルチエージェント通信基盤 INALI 上で実装した。この通信基盤は、NEC-ORB[9]ベースで実装されている。表 1 にソフトウェア諸元を示す。

表 1 ソフトウェア諸元

共通	開発言語	VisualC++4.2, Delphi2.0
		データベース
サーバ	OS	Windows NT4.0
	ミドルウェア	NEC-ORB Ver3.0
	AP	ORB サーバプロセ ディレクトリ Ag P J 管理 Ag
クライアント	OS	Windows NT4.0 / 95
	ミドルウェア	NEC-ORB Ver3.0
	AP	ORB クライアントプロセ 黒板プロセ 業務計画指示 Ag スケジュール管理 Ag

<sup>1)</sup> [8]の再計画アルゴリズム部品をベースにしている。

## 4.2. 評価実験の対象プロジェクト

このシステムを評価するために、表2のソフトウェア開発のプロジェクト管理業務を対象とした。

表2 対象プロジェクト業務について

開発期間	3か月
開発要員	5人
タスク数	4
サブタスク数	2

また、プロジェクト管理サーバ、管理者及び開発要員のクライアントマシンはすべて同じLAN内で稼働させた。

## 4.3. 評価

上記プロジェクトを実際に設計・運営し、以下の評価結果を得た。

### <速度>

実験の結果、133MHz クラスの Pentium マシンでは速度的な問題はほとんどない。プロジェクト管理 Ag の立ち上げから5人に対する作業依頼の処理（空き時間問合せ、仮予約、作業開始依頼）は60秒程度で終了する。これは、従来の電子メールによる作業依頼の処理と比較するとかなり短縮されている（電子メールの場合はリプライのタイミングによるが数分から数日かかる）ので、作業割当に要する時間が短縮された。また、作業者からの遅延報告を受けた場合や管理者による再計画を行う場合の処理に関しても同様の時間で処理が終了する。たまた通信エラー等で処理が中断されるが、その場合でもうまくエラー処理ができれば十数秒で処理が完了するので問題はなかった。

### <プロジェクト設計のしやすさ>

業務計画指示 Ag の GUI は Microsoft Project for Windows95 などの一般的なプロジェクト管理ソフトウェアに類似なため、これらのソフトのユーザであればプロジェクト設計に問題はなかった。各種データの催促条件の設定に関しては、現在の段階で想定できるものだけしか用意していないので、複雑な条件の設定（作業者に応じて催促タイミングを変えるなど）は現在できない。また従来[5]固定的だった日程調整のアルゴリズムを選択的に設定できるので、より実状に促した調整戦略が組める。

### <進捗管理コスト>

管理者が設定したタイミング・条件で作業者の進捗率・完了予測日を自動収集し、必要に応じて自動的に作業者に催促メッセージが出されるので、管理者が逐次進捗を確認する作業がなくなり、進捗管理コストが削減された。ただ、進捗率をまったく入力しない作業者がいる場合、誤って催促してしまう場合があった。

### <再計画の有効性>

人為的に P J 管理 Ag に再計画をさせてみたが、こちらの期待どおりにプロジェクトの日程は再計画された。例えば、既に作業割当した作業者 A に対してプロジェクト以外の優先度の高い作業負荷を与えると、P J 管理 Ag が作業者 A の依頼作業の完了予測日が遅れを察知し、作業者 A 以降の作業者に日程変更依頼・別の作業者への依頼など、適切な再計画が行われた。

ただ、複数の作業者がすべて既に優先度の高い別の作業を入れているなどの場合は、日程調整の際にプロジェクト管理 Ag が再計画不能になる場合があった。

## 5. おわりに

本稿では、プロジェクト管理業務である作業割当、進捗管理、日程調整をマルチエージェントを用いることにより自動化するプロジェクト管理システム Done-Do-Rhythm(ダンドリズム)について報告した。このシステムにより、効率的な作業割当、進捗管理コストの減少、オフィスでの作業実態に即した日程調整が可能となる。

## 参考文献

- [1] Ishiguro.Y., et al.: An Agent Architecture for Personal and Group Work Support, Proc. of International Conference on Multi Agent Systems (ICMAS'96, pp.134-141(1996))
- [2] 石黒義英ほか、マルチエージェントオフィスシステムとその実現、日本ソフトウェア科学会「マルチ・エージェントと協調計算ワークショップ」(MACC'95)
- [3] プロジェクト管理ソフト、日経情報ストラテジー、No.55, pp88-111(1996)
- [4] Microsoft Project for Windows95 ユーザーズガイド(1995)
- [5] 吉府研治ほか、マルチエージェントによるプロジェクト管理システム、情報処理学会研究報告、97-GW-22-3(1997)
- [6] 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、情報処理学会研究報告、97-GW-21-17 (1997)
- [7] Finin T., et al: DRAFT Specification of the KQML Agent-Communication Language, The DARPA Knowledge Sharing Initiative External Interfaces Working Group(1993)
- [8] 垂水浩幸ほか、WorkWeb システム: ワークフロー動的再計画の方式、情報処理学会研究報告、97-GW-21-18 (1997)
- [9] 分散オブジェクト基盤 ORB, <http://www.sw.nec.co.jp/OT/product/nec-orb/orb.html>