ワークフローにおけるオブジェクトの振る舞いの ペトリネットによるモデリングと ワークスケジューリング*

稲本 惇**

Petri Net Based Object Behavior Modeling and Work Scheduling in Workflow*

Atsushi INAMOTO**

In this paper, we take up two of the most serious technical problems aiming at the practical workflow implementation, namely, how to build up the model of workflow and how to set up and solve the optimum work-schedule problem. First, for precise dynamic modeling of workflow, we establish the timed Petri Net models which are converted from state transition models based on object-oriented modeling technique(OMT). Then, we define the optimum work scheduling problem on those timed Petri Net models and develop the solution of that problem depending on linear programming techniques. Further, we set up supplemental constraints for the linear programming to avoid resource conflicts between objects in workflow. Finally, we present the numerical calculation results of our practical studies in order to prove the feasibility and efficiency of our solution.

1. 緒言

ビジネスシステム分野におけるプロセスリエンジニアリング(BPR)が推進されていく中に、ワークフローオートメーションが注目され、すでに、いくつかの先進的企業において構築されている。ワークフローオートメーションは、ビジネスプロセスのスピードアップやホワイトカラーの生産性向上を目的として、ビジネスプロセスで取り扱われる伝票や文書などのフォームを電子化したうえで、その流れを自動制御、自動監視、自動管理することである。言い換えれば、ワークフローオートメーションは今までメインフレーム上のデータ処理システムではオートメーション化が実現されなかった種類のオフィスワークに関するオートメーション化をねらいとしている。

多くのソフトウェアベンダがワークフローオートメーション構築のためのソフトウェアであるワークフローソフトウェアをグループウェア製品の一貫として開発してきており、製品についてはすでに実用段階にあるといえる⁶⁾。しかしながら、現実のワークフローオートメシ

ンシステム構築にあたっては、(1)システム要求仕様定義のため、ワークフローを分析してモデル化すること、(2)ワークフロー管理の目標となる最適なワークスケジュールを決定すること、がユーザ側に要求されることとなり、これが重要な技術課題となる。

ここで、ワークフローを対象としたプロセス分析やモデリングの方法論に関する研究はまだ充分とはいえず、また、一般的に参照できるプロセスモデルの開発もまだ進んでいないのが現状である。しかしながら、オブジェクト指向分析・設計の分野では、オブジェクトモデリング技法(OMT: Object Modeling Technology)^{3),1)}がビジネスシステムのドメイン分析技法として広く研究されてきており、そのワークフロー分析への応用は可能性が大である。また、ペトリネットによるモデリングは、リアルタイムシステムの振る舞いに関するモデリング技法として古くから研究されてきたが、最近になり分散処理システム開発における要求仕様定義技法として、あるいは、スケジューリング問題のためのモデリング技法として注目を集めてきている。これにつれて、時間ペトリネットや確率ペトリネットなどのいわゆる高水準ペトリネットの開発も進められてきている²⁾。このペトリネットモデルのワークフローへの応用も研究されてきているようである。

* 原稿受付 1996年4月26日

** 三菱電機(株)CCVシステムセンター Mitsubishi Electric Corp.CCV System Center; 2-2-3 Marunouchi, Chiyoda ward, Tokyo 100, JAPAN

Key Words: object oriented approach, scheduling, mathematical programming.

Fig. 1 Basic mechanism of workflow

一方、ワークフローの最適スケジューリングについてもその必要性は認められているもののモデリング技法が確立していないことから具体的な成果が出るまでには至っていない⁴⁾。しかしながら、生産システムや並列コンピュータシステムの分野におけるスケジューリングやシミュレーションの理論と応用の研究は枚挙に暇がない^{10)–12)}。したがって、これらの分野での研究成果をワークフローに活用することは充分可能であると考えられる。

本論文の目的は、ワークフローにおけるモデリングとスケジューリングの二つの重要な技術課題に解決策を与えることであり、ワークフローのモデリングについてはワークフローの時間ペトリネットモデルを確立し、ワークスケジューリングについては線形計画法を主体とした解法を提案する。

以下では、まず、ワークフローを定義し、その基本メカニズムを明らかにする。つぎに、電子伝票を振る舞いを持つオブジェクトとしてとらえ、このオブジェクトの振る舞いを、OMTを用いてモデル化することにより、ワークフローの状態遷移モデルを確立する。また、このオブジェクトモデルに等価な時間ペトリネットを確立し、このモデルがワークフローのスケジューリングのためのモデルとして位置づけることができることを示す。さらに、この時間ペトリネットモデル上でワークスケジューリング問題を定義し、線形計画法による解法を提案する。そして、電子伝票オブジェクト間のリソースの競合を回避するため優先順位制御ルールを見だし、そのルールを線形計画法の制約条件として定式化する。最後に、ここで確立されたワークスケジューリング技法を具体的なワークフローの事例に適用し、このペトリネットモデルによるオブジェクトの振る舞いのモデリング技法とワークスケジューリング技法の実用性、有効性を数値計算により評価する。

2. オブジェクトモデリング

2.1 ワークフローとオブジェクト

「ワークフローとは、役割に割り当てられた作業（ワーク）とその作業の対象となる電子伝票（電子化された伝票）との相互作用の連鎖プロセスである」と定義する⁸⁾。

たとえば、生産システムでは、生産プロセス全体がワークフローであり、その中に、生産計画策定プロセス、受注—出荷プロセス、見積もり回答プロセス、資材購入プロセスなどいくつかのワークフローを含むことができる。そこでは、経理担当、経理課長、出荷担当、受注担当などが「役割」であり、伝票発行、検印などの業務単位が「作業」である。また、入金伝票、材料伝票、作業指示書、工事命令書などが作業の対象となる「電子伝票」である。

ワークフローの基本メカニズムはFig. 1であるとして、この基本メカニズムをもとに、以下のオブジェクトモデリングを進めることとする。なお、この基本メカニズムで表現しているワークフローのシナリオは下記のとおりである。

- (1) ある役割（たとえば、経理担当）に電子伝票（たとえば、出金伝票）を送られることにより、その作業（たとえば、出金作業）が起動される。
- (2) 作業が完了すると電子伝票に作業完了の操作（たとえば、検印）をする。
- (3) この操作により電子伝票は状態遷移（たとえば、発行→出金完了）する。
- (4) 電子伝票は状態遷移すると次の役割（たとえば、経理課長）に送られ、その作業（たとえば、検認）を起動する。

2.2 オブジェクトの定義

まず、ワークフローにおけるオブジェクトのクラス構成は、Fig. 2のクラス構成図に示すように、最上位のクラスとしては「電子伝票 F 」、「役割 R 」と「作業 W 」が位置づけられ、「電子伝票」のサブクラスとして $F_1, \dots, F_i, \dots, F_m$ が、「役割」のサブクラスとしては、 $R_1, \dots, R_j, \dots, R_n$ が、「作業」のサブクラスとして、 $W_1, \dots, W_k, \dots, W_p$ が位置づけられている構成である。（以下では、オブジェクトを表すのに鉤括弧「」を用いる）。

つぎに、ワークフローにおけるそれぞれのオブジェクトの構造を以下に定義する。「電子伝票 F_i 」は、データ属性 A_i と操作（オペレーション） $V_{ih} \in V_i$ を持つとともに、状態 $S_{ih} \in S_i$ を持ち、外部からの操作イベントにより状態遷移する動的オブジェクトであるとする。一方、「役割 R_j 」と「作業 W_k 」は状態遷移しない静的なオブ

Fig. 2 Object-class hierarchy model of workflow

Fig. 3 Object communication model of workflow

ジェクトであるとする。

2.3 オブジェクト関係モデル

ワークフローにおいて、各々の「作業」はそれぞれ一つの「役割」に割り当てられているという静的な関係が存在している。一方、「電子伝票」と「作業」間および「電子伝票」と「役割」間には次のような動的な関係が存在している。

- (1) 「電子伝票」は「作業」に対して起動アクションを起こし、結果として、「作業」から（請願された）完了イベントを受け取る。
- (2) 「電子伝票」は「作業」を起動する前に、その「作業」が割り当てられている「役割」をリザーブし、「作業」完了イベントでその「役割」をリリースする。

Fig. 3 のオブジェクト間通信モデルは、ワークフローにおけるオブジェクト間の関係のうち、上に述べた動的な関係をモデル化したものである。Fig. 3 のモデルでは、以下のことがモデル化されている。

- (1) 「電子伝票 F_i 」の「操作 $V_{i,k-1}$ 」において、「作業 W_{k-1} 」への起動アクション $G_{i,k-1}$ を起こす。
- (2) 上の起動アクションに対応する「作業 W_k 」から「電子伝票 F_i 」の「操作 $V_{i,k}$ 」に対して請願された

イベント $E_{i,k}$ が送り返される。

- (3) 「電子伝票 F_i 」の操作 $V_{i,k}$ において、前の「作業 W_{k-1} 」の「役割 R_{j-1} 」をリリースしたのち、次の「作業 W_k 」に必要な「役割 R_j 」をリザーブし、「作業 W_k 」の起動アクション $G_{i,k}$ を起こす。

個々のワークフロー・ドメイン分析において、これら「電子伝票」、「役割」と「作業」間の動的な関係は、Table 1 の状態遷移表を用いて以下のように仕様定義（モデル化）することができる。この表において、縦軸は「作業」とその「作業」が割り付けられている「役割」であり、横軸は「電子伝票」の状態である。

- (1) イベント定義：各々の「作業」完了イベントが、いずれの状態にあるいずれの「電子伝票」に対して送られるべきかを決定する。Table 1 では、「作業 W_{k-2} 」は「電子伝票 F_i 」の状態 $S_{i,k-1}$ に対して、「作業 W_{k-1} 」は状態 $S_{i,k}$ に対して実行されるとして、「作業 W_{k-2} 」の行の状態 $S_{i,k-1}$ の列に $E_{i,k-1}$ と記し、「作業 W_{k-1} 」の行の状態 $S_{i,k}$ の列に $E_{i,k}$ と記している。

Table 1 State transition table

Role	Work	Form: F_i									
				S_{i1}	...	S_{ih-1}	S_{ih}	S_{ih+1}	...		
R_i	W_i										
.	.										
.	.										
R_{j-2}	W_{k-2}										
R_{j-1}	W_{k-1}										
R_j	W_k										
.	.										
.	.										
R_n	W_p

Fig. 4 Lifecycle model of a form object

- (2) 起動アクション定義：各々の「作業」がいずれの状態にあるいずれの「電子伝票」から起動されるべきかを決定する。Table 1 では、「作業 W_k 」が「電子伝票 F_i 」の状態 S_{ih} から、「作業 W_{k-1} 」が状態 S_{ih-1} から起動されるとして、「作業 W_k 」の行の状態 S_{ih} の列には G_{ih} 、「作業 W_{k-1} 」の行の状態 S_{ih-1} の列に G_{ih-1} と記している。

2.4 オブジェクトの振る舞いのモデル

「電子伝票 F_i 」のライフサイクルは、生成の状態 S_{i0} を経て、最終の消滅状態に至るサイクルであり、Fig. 4 の状態遷移モデルで表現できる。すなわち、「電子伝票 F_i 」は、前「作業 W_{k-1} 」完了時点での請願されたイベント E_{ih} により、前状態 S_{ih} から状態 S_{ih+1} に遷移し、この状態遷移の結果が次の「作業 W_k 」の起動アクション G_{ih} を起こす。この状態遷移モデルから、「電子伝票 F_i 」のライフサイクルモデル LC_i は 4 項の組み (S_i , E_i , G_i , R_i) で記述できることになる。ここで、

$S_i = (S_{i1}, \dots, S_{ih}, \dots)$: 「電子伝票」の状態の集合
 $E_i = (E_{i1}, \dots, E_{ih}, \dots)$: 「作業」から「電子伝票」への請願イベントの集合
 $G_i = (G_{i1}, \dots, G_{ih}, \dots)$: 「電子伝票」から「作業」への起動アクションの集合
 $V_i = (V_{i1}, \dots, V_{ih}, \dots)$: 「電子伝票」の状態遷移を引き起こす操作の場合

である。ここで、操作 V_{ih} は次のプロセス代数式で表現できる。

$$V_{ih} : S_{ih} = E_{ih}[Q_{ih}(A_i)].G_{ih}.S_{ih+1} + E_{ih}[Q'_{ih}(A_i)].G'_{ih}.S'_{ih+1} + \dots \quad (1)$$

ただし、 $Q_{ih}(A_i)$ はデータ属性の集合 A_i をパラメータとする条件式であるが、上のプロセス代数の表現しているシナリオは以下のとおりである。

- (1) $Q_{ih}(A_i) = \text{true}$: イベント E_{ih} に対して状態 S_{ih+1} に遷移する (起動アクションは G_{ih})
- (2) $Q'_{ih}(A_i) = \text{true}$: イベント E_{ih} に対して状態 S'_{ih+1} に遷移する (起動アクションは G'_{ih})

Fig. 5 Petri Net model of dynamic behavior of a form object

ョンは G'_{ih})

すなわち, (1) 式では状態遷移の分岐が存在するものとして, モデル化されているが, 以下では状態遷移の分岐はないものとして議論を進める.

3. ペトリネットモデリング

3.1 ライフサイクルモデル

2.4において, OMTを用いて,「電子伝票」のライフサイクルが状態遷移モデルで表現できることを示した. 本節では, そのモデルがいかにペトリネットモデルに変換できるかを議論する. ここで, ライフサイクルモデルにペトリネットモデルを用いる理由は以下のとおりである.

- (1) オブジェクトの振る舞いに関する状態遷移, 操作, 起動アクションとイベント送信を同時に図式表現することができる.
- (2) オブジェクト間の状態遷移の同期 (待ち合わせ) やオブジェクト間のイベント送受信といったオブジェクト間の協調をより簡便に使いやすくモデル化できる.
- (3) ペトリネット上で開発されている解析手法や最適化の手法を利用することができる.

Fig. 4の「電子伝票 F_i 」のライフサイクル状態遷移モデルは, 次の手順により, Fig. 5に示す「電子伝票 F_i 」の振る舞いペトリネットモデルに変換できる.

- (1) 状態 S_{ih} をプレース S_{ih} に, 操作 V_{ih} をトランジション V_{ih} に置き換え, プレース S_{ih} とトランジション V_{ih} を順次アークでつなぐ.
- (2) 「作業 W_k 」に対応する作業トランジション W_k を外部トランジションとして追加する.
- (3) 起動アクションのためのプレース G_{ih} とイベント受信のためのプレース E_{ih} を追加して, それらプレースと操作トランジション V_{ih} , 作業トランジ

ション W_k をアークで結ぶ.

- (4) 「役割 R_j 」をリソースに対応するプレース (初期マーキング付き) を追加して, トランジション V_{ih} とアークで結ぶ.

このようにして「電子伝票 F_i 」ライフサイクルモデル $LC_i = (S_i, E_i, G_i, R_i)$ から変換されるペトリネットモデルは $PN_i = (P_i, T_i, C_i, \mu_i^0)$ の4項モデルである. ここで,

P_i : プレース S_{ih}, G_{ih}, E_{ih} の集合

V_i : トランジション V_{ih} の集合

C_i : プレースとトランジションを結ぶアークの集合

μ_i^0 : プレースの初期マーキング

である.

3.2 オブジェクト間協調モデル

ワークフローにおいて, 異なる「電子伝票」間の「作業」起動の同期を「電子伝票」側ですべて制御するためには「電子伝票」間の協調が必要になる. 「電子伝票」間の同期のモデル化は, Fig. 6のように二つのイベント送信用のプレースと必要なアークを追加することにより表現できる. ここでは, 「電子伝票 F_i 」の状態 S_{ih} のタイミングと「電子伝票 F_{i+1} 」の状態 S_{i+1h} のタイミングの同期を取ることを目的として, それぞれにプレース H_{ih} とプレース V_{i+1h} を追加している.

4. ワークスケジューリング

4.1 時間ペトリネットモデル

ワークスケジュール問題を取り扱うのにあたり, 「電子伝票」で構成される「ジョブ」を以下のように定義する.

- (1) 一つのジョブは, m の「電子伝票 F_i 」で構成され, 定周期に投入されるワークフローの単位である. したがって, 複数のジョブが同時にワークフロー上に存在することになる.

Fig. 6 Petri Net model of synchronization between form objects

Fig. 7 Petri Net model of dynamic behavior of a job

- (2) ジョブを構成する各「電子伝票 F_i 」は、共通に p の作業ステージを持ち、各ステージ k において対応する「作業 W_{ik} 」が起動されるとする。
- (3) ジョブを構成する各「電子伝票 F_i 」の同じ作業ステージ k において起動される「作業 W_{ik} 」には同一の「役割 R_j 」が割り付けられているとする。たとえば、「電子伝票 F_i 」の作業ステージ k において起動される「作業 W_{1k} 」に「役割 R_j 」が割り付けられるとすれば、他の「電子伝票 F_{i+1} 」の同じ作業ステージ k において起動される「作業 W_{ik} 」についても「役割 R_j 」が割り付けられることになる。

この定義に従えば、ジョブのペトリネットモデルは、Fig. 7 のペトリネットモデル図に示すように、3. で確立

した「電子伝票 F_i 」対応のペトリネットモデル $PN_i = (P_i, T_i, C_i, \mu_i^0)$ を「電子伝票」間の協調のためのプレース $H_i = (H_{i1}, \dots, H_{ip})$ を用いて連結し、対応する作業トランジション W_{ik} を加えたものになる。ここで、プレース H_{ik} は、「電子伝票 F_i 」と次の「電子伝票 F_{i+1} 」間での作業ステージ k における「役割 R_j 」の競合を避けるための協調用プレースである。また、冗長性を回避するため、ジョブモデルにプレース S_{ih} を組み入れていない。また、作業ステージ間の「役割」の競合については、4.3 において回避のための制約条件を設定することとして、Fig. 7 のモデルでは「役割」の競合はあえて組み込んでいない（「作業」の数と「役割」の数等しいとしている）。

最終的に、ジョブのペトリネットモデル $TRN = (P, T, C, \mu_0)$ は、 $3mp$ 個のプレース $P = (P_1, \dots, P_m)$ と

$2mp$ 個のトランジション $T = (T_1, \dots, T_m)$ に C と μ^0 を加えた4項組で表されることになる。ここで、 P_i と T_i は、それぞれ「電子伝票 F_i 」対応のプレースとトランジションの集合：

$$\begin{aligned} P_i &= (H_{i1}, G_{i1}, E_{i1}, H_{i2}, \dots, G_{ip}, E_{ip}), \\ |P_i| &= 3p, i = 1, \dots, m \\ T_i &= (V_{i1}, W_{i1}, V_{i2}, \dots, V_{ip}, E_{ip}), \\ |T_i| &= 2p, i = 1, \dots, m \\ G_{ih}: &\text{イベント受信プレース} \\ E_{ih}: &\text{起動アクションのプレース} \\ V_{ih}: &\text{操作トランジション} \\ W_{ih}: &\text{作業トランジション} \end{aligned}$$

である。また、 C はプレース P とトランジション T の接続行列であり ($|C| = 2mp \times 3mp$)、 μ^0 は初期マーキングベクトル ($|\mu^0| = 1 \times 3mp$) である。 C^+ を出力行列 ($|C^+| = 2mp \times 3mp$) とし、 C^- を入力行列 ($|C^-| = 2mp \times 3mp$) とすると $C = C^+ - C^-$ が成立する。

さらに、スケジューリングでは時間が重要な要素となることから、各プレースにトークンの遅延時間を割り当てた時間ペトリネットモデルを採用する。このモデルにおけるプレースの遅延時間を次のようにモデル化した。

- (1) プレース G_{ik} における遅延時間 ρ_{ik} は「作業 W_{ik} 」の起動条件が満足されてから、実際に「作業 W_{ik} 」起動のアクションを起こすまでの作業起動調整時間とする。
- (2) プレース H_{ik} の遅延時間は τ_{ik} は、前段の「電子伝票 F_{i-1} 」における「作業 W_{i-1k} 」終了後の当該「電子伝票 F_i 」の「作業 W_{ik} 」開始までの「役割 R_j 」の待ち時間である。
- (3) プレース E_{ik} の遅延時間 d_{ik} は、「作業 W_{ik} 」の作業時間に対応する。

4.2 スケジューリング問題

ワークスケジューリング問題は、「ジョブの処理時間 π が最短となるように、ジョブを構成する各「電子伝票 F_i 」に対応する各「作業 W_{ik} 」の起動調整時間 ρ_{ik} を決定することである。」と定義する。このワークスケジューリング問題を4.1で確立したジョブ対応時間ペトリネットモデル TPN 上で定式化する。

ジョブの時間ペトリネットモデル TPN に「役割 R_j 」に関する競合を組み込んでいないことから、 TPN は強連結のマークグラフである。したがって、 TPN の接続行列 C に対して、 $(|P| - |T| + 1 = mp + 1)$ 個のプレース・インバリエント・ベクトル β_l が存在する ($|\beta_l| = 3mp \times 1$)。すなわち、

$$\beta_l^T \times C^+ = 0, \quad l = 0, \dots, mp \quad (2)$$

である。また、単位時間における各トランジションの平均発火ベクトルを σ ($|\sigma| = 1 \times 2mp$)、 π をジョブの処理時間とすると、一つのジョブ内ではトランジションは1回ずつ発火することから、 $\sigma = (1/\pi) \times [1, 1, \dots, 1]^T$ が成立する。さらに、ジョブ内の平均マーキングベクトル $\bar{\mu}$ は、単位時間にプレースに出力されるトークン数にそのプレースにおける遅延時間をかけたものであるから、

$$\bar{\mu} = C^+ \times D \times \sigma \quad (3)$$

となる。ここでは、 D はプレース遅延時間対角行列 ($|D| = 3mp \times 3mp$) であり、次式で与えられる。

$$D = \begin{bmatrix} D_1 & & \\ & D_2 & \\ & & \ddots \\ & & & D_m \end{bmatrix}, \quad i = 1, \dots, m$$

さらに、 D_i は「電子伝票 F_i 」ごとのプレース遅延時間対角行列であり ($|D_i| = 3p \times 3p$)、

$$D_i = \begin{bmatrix} \tau_{i1} & & & & \\ & \rho_{i1} & & & \\ & & d_{i1} & & \\ & & & \tau_{i2} & \\ & & & & \ddots \\ & & & & & \tau_{ip} \\ & & & & & & \rho_{ip} \\ & & & & & & & d_{ip} \end{bmatrix}, \quad i = 1, \dots, m$$

と表現される。(3)式の両辺に(2)式で求められるプレース・インバリエント・ベクトル β_l を掛け合わせると、

$$\beta_l^T \times \mu^0 = \beta_l^T \times D \times C^+ \times \sigma, \quad l = 0, \dots, mp \quad (4)$$

が成立する。さらに、 $C^+ \times \sigma = (1/\pi) \times [1, 1, \dots, 1]^T$ であるから、

$$\beta_l^T \times \mu^0 = \left(\frac{1}{\pi} \right) \times \beta_l^T \times D \times [1, 1, \dots, 1]^T, \quad l = 0, \dots, mp \quad (5)$$

が得られる。プレース遅延時間対角行列 D に替えて、プレース遅延時間ベクトル δ を用いると、(5)式は、

$$\beta_l^T \times \mu^0 = \left(\frac{1}{\pi} \right) \times \beta_l^T \times \delta, \quad l = 1, \dots, mp \quad (6)$$

$$\beta_0^T \times \mu^0 = \left(\frac{1}{\pi} \right) \times \beta_0^T \times \delta \quad (7)$$

となる。ここで、 δ と δ_i は次式のとおりである。

Table 2 Work assignment table

Stage		k		k'		k''		k'''	
Role		R_j		R_j		R_j		R_j	
$F1$									
$Fm-k''' + k$				$Wm-k''' + k k'$		$Wm-k''' + k k''$		$Wm-k''' + k k'''$	
$Fm-k'' + k$		$Wm-k'' + k k$		$Wm-k'' + k k'$		$Wm-k'' + k k''$			
$Fm-k' + k$		$Wm-k' + k k$		$Wm-k' + k k'$					
Fm		$Wm k$							

$$\delta = \begin{bmatrix} \delta_1 \\ \delta_2 \\ \vdots \\ \delta_p \end{bmatrix}, \quad \delta_i = \begin{bmatrix} \tau_{i1} \\ \rho_{i1} \\ d_{i1} \\ \vdots \\ \tau_{ip} \\ \rho_{ip} \\ d_{ip} \end{bmatrix}, \quad i=1, \dots, m$$

ここで、すべての i, K について、 d_{ik} が既知あり、 τ_{ik} と ρ_{ik} は未知数であり、

$$\rho_k \geq 0, \quad \tau_{ik} \geq 0, \quad k=1, \dots, p, i=1, \dots, m \quad (8)$$

である。さらに、

$$\begin{aligned} \beta_0^T &= [0, 1, 1, 0, 1, 1, 0, 1, 1, \dots, 0, 1, 1] \\ \mu^{0T} &= [0, 0, 1, 0, 0, 0, \dots, 0, 0, 1, 0, 0, 0, 0, \dots, \\ &\quad 1, 0, 0, 1, 0, 0, \dots, 1, 0, 0] \end{aligned}$$

であることを利用すると、(7) 式は次式になる。

$$\pi = \frac{1}{\pi} \sum_{i=1}^m \sum_{k=1}^p (\rho_{ik} + d_{ik}) \quad (9)$$

最終的に、ワークスケジューリング問題は、(6) 式と (8) 式の制約条件もとで、(9) 式で与えられる π が最小になる ρ_{ik} と τ_{ik} を求める線形計画問題に帰着することになる。

4.3 競合回避条件の設定

4.2 では、「役割」に関する競合は発生しないこととしたことにより、ワークスケジューリング問題は単純な線形計画問題に帰着できた。しかしながら、通常の場合に

は、作業ステージの数 p よりも「役割」の数 n の方が小さく、一人の「役割」に複数の「作業」が割り付けられる。すなわち、「役割」リソースの競合が「作業」間で発生することになる。したがって、ワークスケジューリング問題に、このことを制約条件として加えなくてはならない。

以下では、Table 2 に示すジョブ対応の作業割付表を作成し、Table 2 のもとに「作業」の優先起動順位を決定し、それに従う制約条件式を設定する。この表において、行はジョブを構成する各「電子伝票」であり、列は作業ステージである。ジョブの定義に従い、各作業ステージには、「電子伝票」に共通に四つの「役割」が割り付けられている。表における第 2 行は作業ステージに割り付けられている「役割」を示している。この第 2 行から、一つの作業ステージ k について、その「役割 R_j 」がジョブ内で結合している他の作業ステージ k', k'', k''' を見いだすことができる。これらの作業ステージに起動される「作業 W_{ik} 」、「作業 $W_{i+k-k'k'}$ 」と「作業 $W_{i+k-k''k''}$ 」と「作業 $W_{i+k-k'''k'''}$ 」間において、「役割 R_j 」に関する競合が発生することになる。このリソース競合を以下の優先起動順位決定ルールによって回避する。

手順 1 一つのジョブ内の最後発である「電子伝票 F_i 」対応の「作業 W_{ik} 」完了後に、「電子伝票 $F_{i+k-k'}$ 」の「作業 $W_{i+k-k'k'}$ 」が起動されるように条件設定する。すなわち、次式の制約条件を設定する。

$$\sum_{h=1}^k (\tau_{ph} + \rho_{ph} + d_{ph}) \leq \sum_{h=1}^{k'-1} (\tau_{i+k+k'h} + \rho_{i+k-k'h} + d_{i+k-k'h}) + \rho_{i+k-k'k} \quad (10)$$

ここで、左辺はジョブが投入されてから「作業 W_{ik} 」が完了するまでの時間であり、右辺は「作業 $W_{i-k'+kk'}$ 」が起動されるまでの時間である。手順2 つぎに、「電子伝票 $F_{i+k-k''}$ 」対応の「作業 $W_{i-k''+kk''}$ 」完了後に、「電子伝票 $F_{i+k-k''}$ 」の「作業伝票 $W_{i-k''+kk''}$ 」が起動されるように、次式の制約条件を設定する。

$$\sum_{h=1}^{k'} (\tau_{i+k-k'h} + \rho_{i+k-k'h} + d_{i+k-k'h}) \leq \sum_{h=1}^{k''-1} (\tau_{i+k-k''h} + \rho_{i+k-k''h} + d_{i+k-k''h}) + \rho_{i+k-k''k} \quad (11)$$

手順3 さらに、「電子伝票 $F_{i+k-k''}$ 」の「作業 $W_{i-k''+kk''}$ 」完了後に、最先発の「電子伝票 $F_{i+k-k'''}$ 」の「作業 $W_{i-k''' +kk'''}$ 」が起動されるように、次式の制約条件を設定する。

$$\sum_{h=1}^{k''} (\tau_{i+k-k''h} + \rho_{i+k-k''h} + d_{i+k-k''h}) \leq \sum_{h=1}^{k'''-1} (\tau_{i+k-k'''h} + \rho_{i+k-k'''h} + d_{i+k-k'''h}) + \rho_{i+k-k'''k} \quad (12)$$

最終的に、ワークスケジューリング問題は、(6)式、(8)式に(10)~(12)式を加えた制約条件のもとで(9)式を最小にする線形計画問題となる。

5. 数値計算例

ワークスケジューリングの数値計算事例として選択した事例の要点を以下に示す。

- (1) ジョブを構成する「電子伝票」の数： $m=8$ とする。
- (2) 「役割」の数： $n=1 \rightarrow 8$ とケースにより変化させることとする。
- (3) 各「電子伝票」における作業ステージ数： $p=8$ として、各作業ステージにおける作業時間は、「電子伝票」共通に、Table 3のとおりとする。

Table 3 Working-time table in case-study

Work-1	Work-2	Work-3	Work-4	Work-5	Work-6	Work-7	Work-8
3	5	8	2	4	2	2	4

- (4) 「役割」の数、「役割」の割付方式により、六つのCASEを設定した。

CASE-1: 八つの「作業」の「作業」ごとに一人の「役割」を割り当てたCASE (このCASEでは「役割」リソースの「作業」間での競合は発生しない)。

CASE-2: 「役割」数を4とし、各作業ステージに順次「役割」を割り当てたCASE。

CASE-3, 4: いずれも「役割」数は3としているが、CASE-3は「役割」を順次割り付けたCASEであり、CASE-4はそれ以外の割付方式を採用したCASE。

CASE-5: 「役割」数を2としたCASE。

CASE-6: 「役割」数を1とした特殊CASE (このCASEではワークスケジューリングは無意味となる)。

これらCASE-2からCASE-5の四つのケースについて、競合回避の制約条件を付加した線形計画問題を解き、結果をガントチャートに表して、得られたワークスケジュールの最適性と正当性を検証した。このうち、CASE-2については結果のガントチャート Fig. 8に示している。このCASEにおいては、「役割」の数は4であるが、チャートにおいて「役割」を色調で表現している。また、Table 4には、六つCASEの数値計算結果の総括として、以下の項目をデータとして示している。

- (1) ジョブの最短投入周期 (Job Dispatching Period)
- (2) ジョブの処理時間 (Job Processing Time)

- (3) ジョブにおける「役割」の平均負荷率 = (総作業時間 / 「役割」数) / ジョブ投入周期

6. 結言

以上、本論の成果をまとめると次のとおりである。

- (1) ワークフローの基本メカニズムの設定。
- (2) ワークフローの主要オブジェクトとしての「電子伝票」、「役割」と「作業」の設定とオブジェクト構成 (Fig. 2)と構造の定義。
- (3) 「電子伝票」ライフサイクルのOMTによる状態遷移モデルの構築 (Fig. 3)。

Fig. 8 Optimal work-schedule of case-study: CASE-2

Table 4 Data table case-study

CASE-#:		CASE-1	CASE-2	CASE-3	CASE-4	CASE-5	CASE-6
Case Description	No. of Roles	8	4	3	3	2	1
	Role:	No Conflict $R_1 - r_8$	R_1, R_2, R_3, R_4	R_1, R_2, R_3, R_4	R_1, R_2, R_3, R_4	R_1, R_2, R_3, R_4	No Scheduling R_1, R_2, R_3, R_4
	Work Assign:	A role R_j is assigned to each W_k	W_1	W_1	W_1	W_1	W_1
			W_2	W_2	W_2	W_2	W_2
			W_3	W_3	W_3	W_3	W_3
			W_4	W_4	W_4	W_4	W_4
			W_5	W_5	W_5	W_5	W_5
			W_6	W_6	W_6	W_6	W_6
			W_7	W_7	W_7	W_7	W_7
Results	Job Dispatching Period:	64	80	104	104	136	240
	Job Processing Time:	94	150	195	195	255	240
	Average Load of Roles	32%	75%	77%	77%	88%	100%

- (4) 「電子伝票」を中心としたワークフローペトリネットモデルの構築 (Fig. 4).
- (5) ワークスケジューリングのための時間ペトリネットモデルの構築.
- (6) ワークスケジューリング問題の定義と線形計画問題への帰着.
- (7) ワークスケジューリングにおける「役割」の競合回避のための制約条件の設定.
- (8) ワークスケジューリングの数値計算事例における有効性評価.

今後、ビジネスシステム現場におけるワークフローオートメーション化の進展に対して、本論文の成果は、以下

の点において、寄与できるものとする。

- (1) ワークフローのペトリネットモデルをシステム要求仕様定義に用いることにより、システム分析・設計作業の精度が向上する.
- (2) ペトリネットモデルによるワークフローの事前シミュレーションが可能となり、システム分析段階でのビジネスプロセス設計に役立つ情報が得られる.
- (3) さらに、ワークスケジューリングを適用することにより、日々のビジネスプロセスの改善が可能となり、ワークフローオートメーション化の効果を助長する.

最後に、今後のワークスケジューリングの研究課題として、以下の項目を提案したい。

- (1) ワークフローにおける条件分岐などを考慮に入れ、ワークフローの確率的ペトリネットモデルの構築と確率的ペトリネットモデル上でのワークスケジューリング問題への取り組み
- (2) ワークフローの途中で作業的割り付けが変更になった場合などのリスケジューリング問題の定式化とその解法の開発
- (3) 本格的なワークスケジューリングの最適化を目指す最適作業割り付け問題の定式化とその解法の開発

参考文献

- 1) J. Rumbaugh et al.: *Object-Oriented Modelling and Design*, Prince-Hall Inc. (1991)
- 2) 青山ほか：ペトリネットの理論の実践；システム制御情報ライブラリー13（システム制御情報学会編），朝倉書店（1995）
- 3) S. Shlaer et al.: *Object Lifecycle Modelling the World in States*, Prince-Hall Inc. (1992)
- 4) 牧野ほか：ライフサイクル図を用いたオブジェクトの協調的な振る舞いの設計；情報処理学会・ソフトウェア工学研究報告（95-SE-103），pp. 153-160（1995）
- 5) 佐々木ほか：オブジェクト指向方法論 OMT における動的モデルと機能モデルの整合性の検証；情報処理学会・ソフトウェア工学研究会，（95-SE-103），pp. 145-152（1995）
- 6) 行徳ほか：ワークフロー管理システム構築支援ツール“CoWorker”；三菱電機技報，Vol. 69, No. 5, pp. 13-18（1995）
- 7) 稲本：ビジネスシステムにおけるワークフローモデリングとスケジューリング；情報処理学会・グループウェア'94 シンポジウム論文集，pp. 75-80（1994）
- 8) 稲本：ワークフロープロセスのオブジェクト志向分析；情報処理学会・オブジェクト指向'95 シンポジウム論文集，pp. 173-180（1995）
- 9) 青山：分散処理システムの開発とネット理論；情報処理，Vol. 34, No. 6, pp. 731-746（1993）
- 10) 長尾，熊谷：ペトリネットの FA 制御への応用；情報処理，Vol. 34, No. 6, pp. 761-760（1993）
- 11) 太田ほか：時間付きプレースをもつペトリネットの繰り返し工程スケジューリング問題への応用；計測自動制御学会論文誌，Vol. 25, No. 6, pp. 92-94（1989）
- 12) K. Onaga, et al.: Qualitative analysis of periodic schedules for deterministically timed Petri Net systems；*IEICE Trans*, Vol. E76-A, No. 4, pp. 580-591（1993）