Workflow Base：データベース技術に基づくワークフローモデル

國島丈生[†,☆] 横田一正[††]

ワークフロー管理システムは定型的協同作業を支援するグループウェアであり，その基盤技術としてデータベースが重要であると考えられる．しかし，従来ワークフロー管理システムでは，質問言語や一貫性管理などのデータベース技術の応用は十分行われていなかった．本論文では，ワークフローをデータモデルの視点からとらえ，データベース技術と親和性の高いワークフローモデル Workflow Base を提案する．このモデルでは，ワークフローは作業単位を表す活動オブジェクトの集合として与えられ，フローは活動オブジェクト間の入出力関係や集約関係から導出される関連として扱われる．ワークフローの実行に関する意味は，フローに対応するルール集合に基づくプロダクションシステムによって，またワークフローの実体化は，ワークフローの汎化/特化階層上での代入によって，それぞれ規定される．また，Workflow Base におけるワークフローの意味制約とワークフロー演算についても議論する．

Workflow Base: A Flexible Workflow Model with Database Technologies

TAKEO KUNISHIMA[†,☆] and KAZUMASA YOKOTA[††]

Database technologies are indispensable for workflow management system, which is one of the groupware for supporting well-structured asynchronous cooperative work. In this paper, a formal workflow model suitable for database management, *Workflow Base*, is proposed. In this model, a workflow is defined as a set of objects (activity objects). Flows are derived from the relationships between activity objects, such as input-output relationships and part-of relationships. An execution model of the workflow model is given by a production system on the rules corresponding with flows in one-to-one relationships. Instantiation of workflows is defined by an assignment process on generalization/specialization hierarchies of workflows. We also discuss about integrity constraints on workflows, and about a operation set on workflows.

1. はじめに

オフィスにおける定型的協同作業を支援するグループウェアの1つに，業務の流れを計算機によって支援するワークフロー管理システム（Workflow Management System, WFMS）[1]がある．これは，あらかじめ定められた業務の流れ（ワークフロー）を，電子メールなどを用いて計算機により支援しようというシステムであり，定型的オフィス業務の支援，工程管理の基本システムとして注目されている[2]．

ワークフロー管理システムでは，業務の流れの定義や進捗状況，業務に関するさまざまな条件，業務の進行にともなって発生する資源（プロダクト）などをシステムで管理する必要があり，その構築にはデータベース管理システムとの連携が不可欠となる．また印刷・出版など，ある種の工程管理では，過去の業務のノウハウの蓄積・検索が非常に重要であり，データベースとの連携はさらに重要となる．しかし，現在，製品化されているワークフロー管理システムの多くでその基盤として使われているデータベース管理システムは，レポジトリとしての利用にとどまっているものが多い．

データベース技術の積極的な応用によって，ワークフロー管理システムには以下のような利点がもたらされると考えられる．

- 複数のワークフロー定義を管理することによって，互いの重複や矛盾をなくし，効率的な管理が可能になる．

† 奈良先端科学技術大学院大学情報科学研究科
Graduate School of Information Science, Nara Institute of Science and Technology
☆ 現在，岡山県立大学情報工学部情報通信工学科
Presently with Department of Communication Engineering, Faculty of Computer Science and System Engineering, Okayama Prefectural University
†† 岡山県立大学情報工学部情報通信工学科
Department of Communication Engineering, Faculty of Computer Science and System Engineering, Okayama Prefectural University

- 複数のワークフローから特定の個人に関する「ビュー」を生成することによって，個人の仕事の状況を的確に把握することができる．
- ワークフローをオブジェクトの集合としてとらえることによって，ワークフローの柔軟な更新が可能になる．
- ワークフローの実行によって生成された資源も一緒に管理することによって，ワークフローに関連するデータ全体が包括的に管理できる．

このような背景に基づき，本論文では，データベース技術をより積極的に応用したワークフローモデル Workflow Base[3),4)] を提案する．Workflow Base では，ワークフローの定義，割当てが行われて実行されているワークフロー，各ワークフローの進捗状況などを，データベース上で統合して管理する．また，関係代数に類似した汎用的なワークフロー演算子群を持ち，これを用いてワークフロー操作やビュー機能を実現することができる．

本論文の構成は以下のとおりである．2 章では，Workflow Base のワークフローモデルについて形式的な定義を与える．3 章で，ワークフローモデルの実行モデルをプロダクションシステムによって定義した後，4 章で，それまでの定義に基づいて Workflow Base の形式的モデルを与える．5 章では Workflow Base で用意するワークフロー演算子群の定義とその応用例を示す．6 章で関連研究との比較を行う．7 章はまとめと今後の課題である．

2. ワークフローモデル

2.1 活動オブジェクト，ワークフローテンプレート

Workflow Base におけるワークフローは，仕事の単位に対応する**活動オブジェクト**（**activity object**）集合と，活動オブジェクト間に成立する制約集合から構成される．

活動オブジェクトは，以下のように再帰的に定義される．

$$
\begin{aligned}
a &= (I, O, P, S)\\
I &= \{i_1, \cdots, i_n\}\ (n \geq 1)\\
O &= \{o_1, \cdots, o_m\}\ (m \geq 1)\\
P &= \{\text{string}\}\\
S &= \text{WFT}
\end{aligned}
$$

ここで I は a への入力，O は a からの出力，P は a の実行主体であるエージェント，S は a を実行するためのワークフローテンプレート（WFT）である．ワークフローテンプレートについては，この後で定義を与える．厳密には，活動オブジェクトは (a, I, O, P, S) の 5 つ組で定義され，その識別子を a と考えるべきだが，ここでは簡略化のために単に a と表記したり，$a = (I, O, P, S)$ のように表記している．また I，O，P が単一要素集合であれば $\{\}$ は誤解がない限り省略する．

ワークフローテンプレート（Workflow Template, WFT） W は活動オブジェクト $a_1, \cdots, a_n$ $(n \geq 0)$ の集合 $\{a_1, \cdots, a_n\}$ として定義される．正確には $(W, \{a_1, \cdots, a_n\})$ であるが，ここでも識別子を外に書き，$W = \{a_1, \cdots, a_n\}$ のように上記の活動オブジェクトと同様に簡略化して表記する．

2.2 作業間のフロー

WFT では，活動オブジェクト間に**水平フロー**（$\Longrightarrow$）と**垂直フロー**（$\longrightarrow$）という 2 種類のフローが定義できる．それぞれの定義は以下のとおりである．ただし $a = (I, O, P, S)$ とする．

$$
\begin{aligned}
a_1 \Longrightarrow a_2 &\stackrel{def}{=} O_1 \supseteq I_2\\
\{a_1, a_2, \cdots, a_n\} \Longrightarrow a &\stackrel{def}{=} \left(\bigcup_{i=1}^{n} O_i \supseteq I\right) \wedge \\
&\qquad \forall j \bigcup_{1 \leq i \leq n, i \neq j} O_i \not\supseteq I\\
a \longrightarrow a_i &\stackrel{def}{=} a_i \in S
\end{aligned}
$$

$a_1 \longrightarrow a_2$ のとき a_2 は a_1 の**子**，a_1 は a_2 の**親**と呼ぶ．水平フローの 2 番目の定義は入力の最小性を保証する．

水平フロー $\{a_1, a_2, \cdots, a_n\} \Longrightarrow a$ が成立するとき，$a_1, \cdots, a_n$ と a の間には**部分水平フロー** $a_1 \rightharpoonup a, \cdots, a_n \rightharpoonup a$ が成立するという．ここで，部分水平フローは次のように定義される．

$$
a_1 \rightharpoonup a_2 \stackrel{def}{=} O_1 \cap I_2 \neq \emptyset
$$

この定義から，$a_1 \Longrightarrow a_2$ が成立するとき，必ず $a_1 \rightharpoonup a_2$ も成立する．

水平フロー $\Longrightarrow$ の推移的閉包は次のように定義される．

$$
\begin{aligned}
a_1 \stackrel{+}{\Longrightarrow} a_2 &\stackrel{def}{=} a_1 \Longrightarrow a_2\\
&\quad \vee ((a_1 \stackrel{+}{\Longrightarrow} a_1') \wedge a_1' \Longrightarrow a_2)
\end{aligned}
$$

同様にして，部分水平フローの推移的閉包 $\stackrel{+}{\rightharpoonup}$，垂直フローの推移的閉包 $\stackrel{+}{\longrightarrow}$ を定義することができる．また，$\rightsquigarrow \stackrel{def}{=} \{\Longrightarrow \vee \rightharpoonup \vee \longrightarrow\}^+$ とする．ここで $+$ は 1 回以上の繰返しを表す．

たとえば，4 つの活動オブジェクト a_1，a_2，a_3，a_4 があったとする．ただし，$O_1 = \{o_1, o_2\}$，$O_2 = \{o_2, o_3\}$，$O_3 = \{o_3, o_1\}$，$I_4 = \{o_1, o_2, o_3\}$ である．このとき以

下の水平フローが考えられる．

$$\{a_1, a_2\} \Longrightarrow a_4,$$
$$\{a_2, a_3\} \Longrightarrow a_4,$$
$$\{a_3, a_1\} \Longrightarrow a_4$$

一方，$\{a_1, a_2, a_3\} \Longrightarrow a_4$ は，入力の最小性を満たさないため，水平フローとは扱わない．

2.3 ワークフロー

ワークフロー（**workflow**）は，すべての活動オブジェクトが互いに連結されているような WFT として定義できる．すなわち，次の制約（**ワークフロー制約**）を満たす WFT W である．

$$\begin{aligned}&\forall a_1, a_2 \in W, a_1 \neq a_2.\\ &\quad a_1 \rightsquigarrow a_2\\ &\vee\ a_2 \rightsquigarrow a_1\\ &\vee \exists a_3.\ ((a_1 \rightsquigarrow a_3 \wedge a_2 \rightsquigarrow a_3)\\ &\qquad \vee (a_3 \rightsquigarrow a_1 \wedge a_3 \rightsquigarrow a_2))\end{aligned}$$

2 つの WFT S_1 と S_2 を考える．これらの間の順序関係を部分集合関係，すなわち $S_1 \subseteq S_2$ によって定義する．WFT 中のワークフローのうち，$\subseteq$ による順序関係で極大のものを**極大ワークフロー**という．極大ワークフローにおいて，親を持たない活動オブジェクトを単に**親**という．一般には WFT W の極大ワークフローは複数存在する．

例として，査読プロセスを表すワークフロー（**図 1**）に対応する WFT は**図 2** のようになる．

実際のワークフローのプロセスでは，委員長，委員，査読者，秘書などのエージェント，および「投稿論文」など活動オブジェクトへの入出力などが実行前に実体化されるべきだが，それは 2.5 節で説明する．

2.4 ワークフローの一貫性制約

ワークフロー制約以外にも，WFT での制約には以下のようなものが考えられる．

(1) 閉じた WFT

ある WFT W の任意の活動オブジェクト $a = (I, O, P, S)$ に対して，任意の $a_i \in S$ に対して $a_i \in W$ ならば，W は**閉じている**という．

(2) **非循環** WFT

もし $\forall a \in W. \neg(a \stackrel{+}{\rightarrow} a)$ であれば，W は水平フローに関して**非循環**という．また，$\forall a \in W. \neg(a \stackrel{+}{\longrightarrow} a)$ であれば，W は垂直フローに関して**非循環**という．

(3) 冗長 WFT

任意の $a = (I, O, P, S)$ に対して，もし $O \subseteq I$ であれば，a は**冗長**であるという．冗長な活動オブジェクトを含む WFT を**冗長**であるという．

(4) WFT の**三角性**

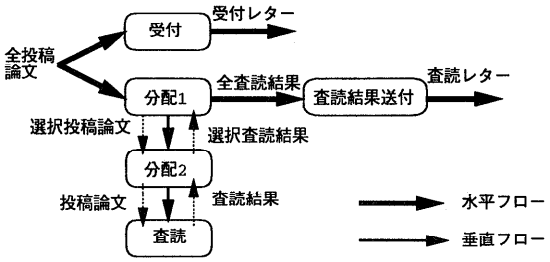


図 1 査読プロセスワークフロー
Fig. 1 A workflow example of review process.

査読プロセス = { 受付, 分配 1, 分配 2, 査読, 査読結果送付 }
受付 = (全投稿論文, 受付レター, 秘書, {})
分配 1 = (全投稿論文, 全査読結果, 委員長, { 分配 2})
分配 2 = (選択投稿論文, 選択査読結果, 委員, { 査読 })
受付 = (投稿論文, 査読結果, 査読者, {})
査読結果送付 = (全査読結果, 査読レター, 秘書, {})

図 2 査読プロセスの WFT
Fig. 2 A WFT of the review process.

もし $\exists a.(a_1 \rightarrow a_2 \wedge a_1 \stackrel{+}{\rightarrow} a \wedge a \stackrel{+}{\rightarrow} a_2)$ であったとすれば，a_1 と a_2 の間には少なくとも 2 本のフローが存在する．このような活動オブジェクト a_1，a_2 を含む WFT は，a_1，a_2 に関して a を通る**三角性**（triangle）を満たすという．

閉じた WFT において，作業全体を表すワークフローは必ず閉じている．すなわち，閉じた WFT は，少なくとも 1 つの閉じたワークフローの定義を持つことが保証される．

2.5 ワークフローインスタンス

前節までの WFT を実際の仕事に適用するためには，WFT 中の入力，出力，エージェント等を実体化しなければならない．

活動オブジェクト $a = (I, O, P, S)$ に対して，I に $I' = \{i_1, i_2, \cdots, i_n\}$，$O$ に $O' = \{o_1, o_2, \cdots, o_m\}$，$P$ に $P' = \{p_1, p_2, \cdots, p_k\}$ を割り当てることを考える．ここで入力，出力のオブジェクトであるメッセージオブジェクトの定義域 $\mathcal{M}$ とエージェントの定義域 $\mathcal{P}$ を考え，$\mathcal{M}$，$\mathcal{P}$ はそれぞれ $\sqsubseteq_{\mathcal{M}}$，$\sqsubseteq_{\mathcal{P}}$ で半順序集合であると仮定する．ただし誤解のない限り $\sqsubseteq$ の下付き文字は省略する．この割当てを次のように書く．

$$\theta = \{I/\{i_1, \cdots, i_n\}, O/\{o_1, \cdots, o_m\}, P/\{p_1, \cdots, p_k\}\}$$

この割当ては次のような型の特化に対応している．

$$I' \sqsubseteq_S I \Leftrightarrow \forall i \in I,\ \exists i' \in I'.\ i' \sqsubseteq i$$
$$O' \sqsubseteq_S O \Leftrightarrow \forall o \in O,\ \exists o' \in O'.\ o' \sqsubseteq o$$
$$P' \sqsubseteq_S P \Leftrightarrow \forall p \in P,\ \exists p' \in P'.\ p' \sqsubseteq p$$

つまり $\sqsubseteq_S$ は Smyth 順序で，この成立が割当てであ

全投稿論文/{ 論文 $_1$, 論文 $_2$, ⋯, 論文 $_n$}
委員長/上林
委員/{ 牧之内, 植村, 出中, 増永 }
秘書/{ 高田 }
⋮

図 3 査読プロセス WFT に対する割当ての例
Fig. 3 An example of assignments to the review process WFT.

ることの必要十分条件となっている．

2 つの活動オブジェクト $a_1 = (I_1, O_1, P_1, S_1)$ と $a_2 = (I_2, O_2, P_2, S_2)$ の間の順序関係は以下のように定義される．

$$a_1 \sqsubseteq a_2 \stackrel{def}{=} I_1 \sqsubseteq_S I_2 \wedge O_1 \sqsubseteq_S O_2 \wedge P_1 \sqsubseteq_S P_2 \wedge S_1 \sqsubseteq_S S_2$$

ここで，$S_1 = \{a_1\theta, \cdots, a_n\theta\}, S_2 = \{a_1, \cdots, a_n\}$ である．このとき，S_1 と S_2 との間に，割当て θ に関する等式 $S_1 = S_2\theta$ が成り立つと定義する．

WFT W に対する割当て θ は次のようになる．

$W\theta = \{a_1\theta, \cdots, a_m\theta\}$

割当ての行われた WFT $W\theta$ を，WFT のワークフローインスタンス（Workflow Instance）という．ただし，WFI はそれ自身 WFT と見なすことができる．そこで，Workflow Base では，特に断らない限り，WFI もまた WFT と見なし，両者の区別はしない．この結果，WFT の集合には，割当てによって順序が付けられる．

図 2 の WFT「会議査読」を，具体的な会議の査読プロセスを表す WFI に実体化する例を示す．国際会議「CODAS」の委員長が「上林」，委員が「牧之内」「植村」「田中」「増永」，秘書が「高田」であったとする．このとき，**図 3** に示す割当てによって WFT「会議査読」は実体化されて「CODAS 査読」という WFI が生成され，その実行が始まる．

3. Workflow Base の実行モデル

3.1 活動オブジェクトの実行モデル

直観的には，活動オブジェクト $a = (I, O, P, S)$ は，a が入力 I を受け取ると，P は必要な仕事を実行し，O を出力することを表す．そのとき a は必要なら I を分割して S にその依頼を行い，実行を監視し，結果を合成して出力 O を生成する．すなわち，活動オブジェクト $a = (I, O, P, S)$ は I から O への関数と考えることができ，以下のように定義される．

(1) $I \in 2^{\mathcal{M}}$, $O \in 2^{\mathcal{M}}$
(2) $P \in 2^{\mathcal{P}}$

水平フロー	プロダクションルール
$a_1 \Longrightarrow a_2$	$a_2 \Leftarrow a_1$
$a_1 \Longrightarrow a_3$ かつ $a_2 \Longrightarrow a_3$	$a_3 \Leftarrow a_1, a_2$
$a_1 \Longrightarrow a_2$ かつ $a_1 \Longrightarrow a_3$	$a_2 \Leftarrow a_1$ かつ $a_3 \Leftarrow a_1$
a_1	$a_1 \Leftarrow .$

図 4 水平フローを表すプロダクションルール
Fig. 4 Production rules for horizontal flows.

(3) $a \in \mathcal{A}$, $S \subseteq \mathcal{A}$, ただしここで $\mathcal{A}$ は活動オブジェクトの集合であり，再帰的に定義される．

$$\mathcal{A} := 2^{\mathcal{M}} \times 2^{\mathcal{M}} \times 2^{\mathcal{P}} \times 2^{\mathcal{A}}$$

P も $\mathcal{M}$ 上の関数として定義できるが，ここでは異なった定義域 $\mathcal{P}$ を考えることにする．つまり入出力が同一でも，異なるエージェントに割り当てられている活動オブジェクトは異なるものとして扱われる．

3.2 WFT の実行モデル

WFT の実行モデルは，水平フローと垂直フローにそれぞれ対応した 2 種類のモデル P-box と B-box から構成される．

水平フローは**図 4** のようにプロダクションルールとして定義できる．

$\Leftarrow$ の右側は条件部であり，左側が行動部である．このルールは前向きに評価される．つまり，それぞれの条件部は対応する活動オブジェクトの終了条件を受け取ることによって成功となる．すべての条件部が成功することによって行動部が活性化される．このようなプロダクションルールの集合がプロダクションシステムであり，WFT 中の **P-box** に格納される．活動オブジェクト自身も，事実（fact）というプロダクションルールとして扱われ，P-box に格納される．

一方，垂直フローは Prolog のような論理プログラミング言語の確定節の集合として表現される．つまり活動オブジェクト a が子オブジェクト $a_1, a_2, \cdots, a_n$ を生成したとき，その実行モデルは

$a \leftarrow a_1, a_2, \cdots, a_n$

と表現される．直観的には，a から制約/束縛情報が $a_1, a_2, \cdots, a_n$ に伝播され，それらすべての子オブジェクトの実行が成功で終了すると，a の実行が終了する．つまりこれらは後向きに評価される．これら確定節の集合は WFT 中の **B-box** に格納される．動的に生成される子オブジェクトも 1 つの活動オブジェクトであるので，その子オブジェクト自体は P-box に格納される．

例として，図 2 の実行モデルを示す．ここで注意すべきことは，{ 査読 } のように指定されている子オブジェクトが，委員長が全査読結果を受け取った後，動的に複数生成されなければならないことである．した

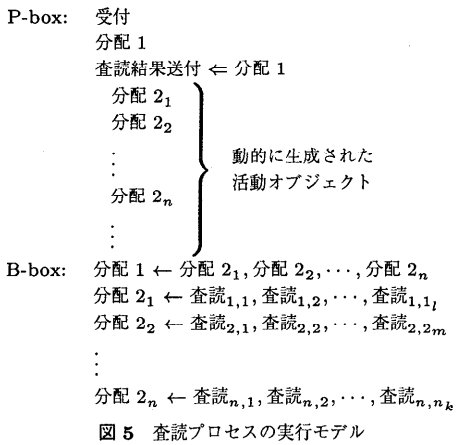


図 5 査読プロセスの実行モデル
Fig. 5 An execution model of the review process.

がって，この例の実行モデルは**図 5** のようになる．まず 3 つのプロダクションルール「受付」「分配 1」「送付 ⇐ 分配 1」が P-box に生成される．「分配 1」が起動された後で，「分配 2_1」「分配 2_2」⋯「分配 2_n」が生成され，その活動オブジェクトが P-box に，実行を制御するためのルール（確定節）が B-box に生成される．各活動オブジェクト「分配 2_i」が「分配 2」を起動するとそれに対応した「査読」のための活動オブジェクトとルールがそれぞれ P-box と B-box に置かれる．

実行時に P-box に挿入される活動オブジェクトは新たに生成されたものであって，既存のオブジェクトとは異なっている．したがって，プロダクションシステムとしての P-box の動的な更新は既存のものの保守的拡張であって，その意味論を変更したり新たに競合を生じさせるものではない．

4. Workflow Base

前章までに定義した各種情報が Workflow Base に格納される．議論を単純化するために，WFT/WFI，活動オブジェクト，メッセージオブジェクト，エージェントなどの識別子は大域的であるとする．

Workflow Base は WFT の集合と $(\mathcal{M}, \sqsubseteq_{\mathcal{M}})$, $(\mathcal{P}, \sqsubseteq_{\mathcal{P}})$ によって定義される．ここで，2 章の定義を一般化し，各 WFT W は特化した上位 WFT W' の識別子と割当て情報を含むようにする．その結果，Workflow Base は次のように構成される．

定義情報： $W = (W', \theta, \{a_1, \cdots, a_m\})$
$a_1 = (I_1, O_1, P_1, S_1)$
⋮
$a_n = (I_n, O_n, P_n, S_n)$

実行モデル： P-box: ⋯
B-box: ⋯

Workflow Base の一貫性制約には，ワークフロー制約，2.4 節で定義したもの以外に，

- WFT の汎化階層：
  $(\mathcal{M}, \sqsubseteq_{\mathcal{M}}), (\mathcal{P}, \sqsubseteq_{\mathcal{P}})$ に基づいて WFT 間の順序が矛盾なく定義されている．

が考えられる．

5. Workflow Base のワークフロー演算子群

本章では，Workflow Base 上でのワークフロー操作のための演算子群について述べる．この演算子群により，一貫性制約を保ちつつワークフローを操作することが可能であり，ワークフローの再構成やビュー機能を実現することができる．また，ワークフローシステムの実現に依存しないため，この演算子群を通してワークフローにアクセスすることにより，相互運用性も実現できる．

5.1 WFT に対する演算

5.1.1 選　　択

Workflow Base 上の**選択**（selection）演算としては，活動オブジェクト集合（WFT）に対するもの，WFT の集合に対するものの 2 種類が考えられる．

WFT に対する選択 $\sigma_{expr} W$ は，WFT W から，*expr* を満たす WFT を返す．*expr* は次のような要素を含む式である：

(1) 被演算子を $a.A$（A は活動オブジェクトの属性名）あるいは定数，演算子を算術比較演算子，集合比較演算子などとする項．
(2) 論理演算子．$\wedge$, $\vee$, $\neg$ など．

$\forall a \in W$ に対して *expr* を評価し，真になった活動オブジェクトからなる集合を結果として返す．W が省略された場合，W としてすべての活動オブジェクトの集合 W_{all} が仮定される．この仮定は，以降の他の演算子に関しても適用される．

5.1.2 グルーピング，アングルーピング

グループ化（grouping）演算 ωW は，WFT $W = \{a_1, \cdots, a_n\}$ に対して次の活動オブジェクト a' を返す演算である．

$$a' = \left(\bigcup_{i=1}^{n} I_i, \bigcup_{i=1}^{n} O_i, \bigcup_{i=1}^{n} P_i, W \right)$$

直観的には，WFT W をひとまとまりの仕事と見なし，W を副作業とするような活動オブジェクト a' を作成する．つまり，W を「グルーピング」して 1

つの活動オブジェクトとする．

平坦化（flattening）演算 Ωa はグループ化の逆演算である．活動オブジェクト $a=(I,O,P,S), S\neq\emptyset$ に対し，WFT S を返す．

5.1.3 ワークフローの構成

先の定義によれば，一般にWFTはワークフローではない．そのため，WFTからワークフローを得るための演算**ワークフロー構成**（get workflow）$\eta_{(F,W)}W'$（ただし $W\subseteq W'$）を用意する．ここで F はフローおよびその推移的閉包の記号からなる集合である．

たとえば，$\eta_{(\{\stackrel{+}{\Longrightarrow}\},W)}W'$ は，次のWFTを返す．

$$\{a'|\forall a\in W.\ a\stackrel{+}{\Longrightarrow}a', a'\in W'\}$$

これは，ワークフローの定義により，明らかにワークフローである．特に，$F=F_{all}\stackrel{def}{=}\{\stackrel{+}{\Longrightarrow},\stackrel{+}{\longrightarrow},\stackrel{+}{\rightarrow}\}$ のとき，この演算は，W を包含するような W' の極大ワークフローを求める演算を表す．

図2において，投稿論文のうち論文番号A-100に関する査読状況を調べたいとする．この論文の査読フローは，以下のような演算によって得られる：

$$\eta_{(F_{all},W_{all})}\ \sigma_{A-100\in I}\ W_{all};$$

その結果，査読委員 taro のところで作業が滞っていたとする．そこで，taro が他にどのような仕事を現在抱えているかを検索する．このための演算は次のようになる：

$$\sigma_{taro\in P}\ W_{all};$$

5.2 分割，連結

2つのワークフロー W_1，W_2 を**連結**（concatenation）するためには，適当な活動オブジェクト $a_1\in W_1$，$a_2\in W_2$ を選び，a_1 から a_2 にメッセージオブジェクトを受け渡すようにしてやればよい．すなわち，連結演算 $\Gamma(a_1,a_2)$ は $I_2:=I_2\cup O_1$ として定義できる．

ワークフロー W の**分割**（split）を行うには，W 中の活動オブジェクトを disjoint な2つの集合に分割すればよい．すなわち，ワークフローの分割演算 $\gamma(W,W_1)$ は次のように定義できる．

$$\gamma(W,W_1)\stackrel{def}{=}W-W_1$$

このとき，一般には得られた2つのWFT W_1，$\gamma(W,W_1)$ はワークフローではない．また，WFT $W'=\{a_1,a_2\}$，$a_1.S=W_1$，$a_2.S=W_2$ もワークフローであるとは限らない．

$W_1,W_2=\gamma(W,W_1)$ がそれぞれワークフローであり，かつ

(1) W_1，W_2 をまたぐような垂直フローが存在しない．

(2) $\exists a\in W_1, W_2\subseteq a.S$

(3) $\exists a\in W_2, W_1\subseteq a.S$

のいずれかが成り立つときかつそのときに限り，W' はワークフローになる．直感的には，W_1，W_2 をまたぐフローがすべて水平フローであるか，W_2（または W_1）中の活動オブジェクトがすべて W_1（または W_2）中のある活動オブジェクトの子になっているとき，かつそのときに限り，W' はワークフローになる．

上記3条件のうち，(2) および (3) はグルーピング演算と同等の操作をしていることになる．

5.3 活動オブジェクトに対する演算

次のような演算がある．

- new(a,I,O,P,S)

  新たな活動オブジェクトを生成する．

- destroy(a)

  活動オブジェクト a を消去する．

- $a\theta$

  活動オブジェクト a に割当て θ を適用し，得られた活動オブジェクトを返す．

- foreach a in s do *Operator*

  WFT s 中の各オブジェクト a に対し，*Operator* を実行する．*Operator* は活動オブジェクトに対する任意の操作であってよい．たとえば，taro の担当する作業をすべて jiro に委譲するような場合，次のような操作を実行する：

  $$\text{foreach } a \text{ in } \sigma_{taro\in P}\ W_{all} \text{ do } a.P:=\{jiro\};$$

活動オブジェクトは，メソッドによる内部の隠蔽は行っていない．したがって，通常の集合演算や前述のWFTに対する演算によって，属性の値を読み出したり書き換えたりすることが可能である．

5.4 WFT集合に対する演算

Workflow Baseでは，複数のWFTが存在し，かつその間に割当て θ に基づく汎化/特化階層が成立している．そのため，WFT集合に対する演算が有用である場合がある．たとえば，あるワークフロー定義に基づいて実際に行われている作業は，Workflow Baseでは定義の実体として扱われるため，それらすべての実体に対する演算を行うことによって作業の進捗状況の調査などが行える．

WFT集合に対する選択 $\sigma'_{expr}S$ は，WFT集合 S から式 *expr* を満たすWFT集合を返す演算である．ここで *expr* は次のような要素を含む式である．

- 活動オブジェクト，またはWFTを被演算子とする集合比較演算．$\subseteq$, $\in$ など．
- WFTの階層に関する述語．$\mathrm{wft}(W')$, $\mathrm{wfi}(W')$, $\mathrm{general}(W')$, $\mathrm{special}(W')$ など．
- 論理演算子．$\vee$, $\wedge$, $\neg$ など．

WFT W に対し，WFTに関する述語は以下のように評価される．

- $\mathrm{eval}(\mathrm{wft}(W')) = \mathbf{true}$ iff $W \sqsubseteq W'$, $\neg\exists W_1.\, W \sqsubseteq W_1 \sqsubseteq W'$
- $\mathrm{eval}(\mathrm{wfi}(W')) = \mathbf{true}$ iff $W' \sqsubseteq W$, $\neg\exists W_1.\, W' \sqsubseteq W_1 \sqsubseteq W$
- $\mathrm{eval}(\mathrm{general}(W')) = \mathbf{true}$ iff $W \sqsubseteq W'$
- $\mathrm{eval}(\mathrm{special}(W')) = \mathbf{true}$ iff $W' \sqsubseteq W$

汎化（general）演算 $\vartheta(W,\theta)$ は以下のように定義される．

$$\vartheta(W,\theta) \stackrel{def}{=} \{W' | W'\theta = W\}.$$

半順序関係 $\sqsubseteq$ における W の親WFTからなる集合を返す演算である．また，特化（special）演算 $\Theta(W,\theta)$ は次のように定義される．

$$\Theta(W,\theta) \stackrel{def}{=} \{W' | W' = W\theta\}.$$

半順序関係 $\sqsubseteq$ における W の子WFTからなる集合を返す演算である．

これらの演算の推移的閉包 ϑ^+ と Θ^+ は，それぞれすべての先祖WFT，すべての子孫WFTを返す演算である．

図5のWFTにおいて，「査読$_{1,1}$」，…，「査読$_{n,n_k}$」はすべて，半順序関係 $\sqsubseteq$ 上において「査読」の直接の下位になる．したがって，同じ会議の他の査読委員からtaroに査読が来ていないか調べるには，次のようにすればよい．

$$\underset{taro \in P}{\sigma}\ \underset{\mathrm{wfi}(\text{査読})}{\sigma'}\ W_{all};$$

5.5 その他の演算

上記の演算，および通常の集合演算を用いて，有用な演算を定義することができる．そのいくつかを示す．

一般にWFTは垂直フローによって階層的に構成される．したがって，仕事の概要を見るために，WFTから子の活動オブジェクトを隠して見せるような機能（**隠蔽**），またこの逆の機能（**提示**）が用意されていると便利である．WFT W 中の活動オブジェクト a の子孫をすべて隠蔽する演算 $\psi(W,a)$ は次のように定義される：

$$\psi(W,a) = (W - \underset{(\stackrel{+}{\longrightarrow},\{a\})}{\eta} W) \cup \{a\}$$

逆に，WFT W 中の活動オブジェクト a の子孫をすべて提示する演算 $\Psi(W,a)$ は次のように定義される：

$$\Psi(W,a) = W \cup \underset{(\stackrel{+}{\longrightarrow},\{a\})}{\eta} W$$

6. 関連研究

6.1 Workflow Management Coalition

Workflow Management Coalitionは，1993年にワークフロー技術の標準化のために設立された団体である．ここの発行している文書の中に，ワークフローモデルに関する用語集とその定義集[5]があり，ワークフローの基本的モデルが与えられている．このモデルには，階層構造，有限オートマトンを基礎とする実行の意味，ワークフローの実体化など，現在ワークフローに関する共通認識の得られている概念が含まれている．しかし，このモデルでは，いずれの概念にも厳密な定義は与えられていない．

6.2 Regatta

Regatta[6]は，タスクの要求，合意（commitment），質問を構成要素とする階層的なワークフローモデルを用意しており，フローの動的変更，インクリメンタル自動実行，ワークフローの実体化，個人のto-doリストなど，強力なワークフロー記述能力を持っている．これらの特徴は，本稿のモデルと共通点が多い．しかし，これらのモデルで用意されているワークフローモデルには厳密な意味が与えられておらず，またワークフロー操作はad hocな機能しか用意されていない．

6.3 Statechart

Statechart[7]は，プロセスの並行実行やメッセージのブロードキャスト機能を持った階層的な有限オートマトンである．その実装システム[8]は，データベースによる管理を行い関係代数に基づく質問言語を用意している，などの点で本稿のワークフローモデルと類似している．しかし，協同作業支援のためのモデルとして見た場合，Statechartには，構造が静的である，エージェントに関する議論がない，プロセスモデルの実体化に関する議論がない，などの問題点があげられる．

7. おわりに

現状のワークフローでは，一般に，次のような問題点を持っている．

(1) 例外処理への対応．一時的なフローの変更といった機能面の問題，また例外処理のための機能を盛り込んだ結果，モデルが複雑化するという問題など．

(2) ワークフロー定義を変更したときに発生する副作用の扱い．実行中のワークフローインスタンスに変更を反映させる方法，また反映させるかどうかの是非など．

(3) 複数のワークフロー管理システム間の相互運用性．

本論文で提案したWorkflow Baseは，上に示した現状のワークフローの持つ問題点に対して，ある一定範囲での解決方法を与えるモデルであるといえる．すなわち，Workflow Baseにおけるワークフローは，動的に決定される2種類の制約（水平フロー，垂直フロー）によって定義されるため，これらの制約で記述できる範囲の一時的変更には容易に対処できる．また，汎用的なワークフロー演算子群によって，複数のワークフローから新たなワークフローを合成したり，逆にワークフローを分割するといった操作は，容易に行うことができる．

ただし，Workflow Baseで現状のワークフローの持つ問題点をすべて解決できているわけではない．上記の問題点それぞれについて述べると以下のとおりになる．

- (1)については，作業の入出力の変更にともなうフローの変更，作業の詳細化などにともなうフローの階層に関する変更などは容易に対処できるが，何らかのイベント発生にともなうフローの一時的変更については論じていない．これについては，ECAルール[9]のようなイベント付きルール機構をWorkflow Baseに導入することを検討している．
- (2)の問題は，現在のWorkflow Baseのモデルでは解決できていない．
- (3)の問題も十分には解決できていない．ワークフロー演算子群によって，ワークフローに関するデータの検索を汎用的に行うことは可能になっている．しかし，ワークフローの定義の異種統合や，定義の異なる複数のワークフローを連動して実行させる機能などに関しては，十分な機能は用意されていない．

(2)や(3)の問題点に関しては，現在のWorkflow Baseの枠組みだけでなく，ワークフロー上での協同作業をマルチエージェントを用いて形式的にモデル化する関連研究[10),11)]などの成果も含めて解決していくことを考えている．

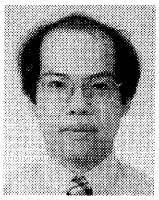
謝辞 研究全般にわたり有益なご助言をいただく京都大学大学院工学研究科の上林彌彦教授，岡山県立大学情報工学部横田研究室の皆様，奈良先端科学技術大学院大学情報科学研究科の渡邉勝正教授ならびに渡邉研究室の皆様に感謝します．なお，本研究の一部は科学研究費特定領域研究「高度データベース」による．

参考文献

1) Georgakopoulos, D., Hornick, M. and Sheth, A.: An Overview of Workflow Management: From Process Modeling to Workflow Automation Infrastructure, *Journal of Distributed and Parallel Databases*, Vol.3, No.2, pp.119–153 (1995).
2) 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，1994/5/2, pp.57–67 (1994).
3) 国島丈生，上林弥彦：ワークフロー管理システムWorkFlowBaseにおけるワークフローデータモデル，情報処理学会研究会報告，DBS104-41 (1995).
4) Kunishima, T. and Yokota, K.: Flexible Workflow Framework for Supporting Collaborative Work, *Proc. International Symposium on Cooperative Database Systems for Advance Applications*, Kyoto, pp.412–419, World Scientific (1996).
5) Workflow Management Coalition: Workflow Management Coalition Terminology & Glossary, Issue 2.0, available from http://www.wfmc.org/ (1996). Document Number WFMC-TC-1011.
6) Swenson, K.D.: Visual Support for Reengineering Work Processes, *Proc. ACM Conference on Organizational Computing Systems*, pp.130–141 (1993).
7) Harel, D.: Statecharts: A Visual Formalism For Complex Systems, *Science of Computer Programming*, Vol.8, pp.231–274 (1987).
8) Harel, D., Lachover, H., Naamad, A., Pnuell, A., Politi, M., Sherman, R., Shtull-Trauring, A. and Trakhtenbrot, M.: STATEMATE: A Working Environment for the Development of Complex Reactive Systems, *IEEE Trans. Softw. Eng.*, Vol.16, No.4, pp.403–414 (1990).
9) McCarthy, D.R. and Dayal, U.: The Architecture of An Active Data Base Management System, *Proc. ACM SIGMOD Intl. Conf. Management of Data*, pp.215–224 (1989).
10) Yokota, K., Kunishima, T. and Nakanishi, H.: Hypothetical Query Processing for Workflow Management, *Proc. Seminar on Logical Databases and the Meaning of Change*, Dagstuhl, pp.44–49 (1996). UPMAIL Technical Report No. 129, Uppsala University.
11) Kunishima, T. and Yokota, K.: An Agent-

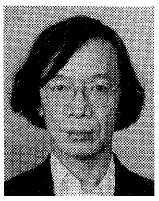
Based Coordination Model on Workflow Databases, *Proc. 4th International Conference on Object-Oriented Information Systems*, pp.361–371, Springer-Verlag (1997).

(平成 9 年 6 月 13 日受付)
(平成 10 年 9 月 7 日採録)



國島 丈生（正会員）

1967 年生．1991 年京都大学大学院工学研究科情報工学専攻修士課程修了．1994 年同研究科情報工学専攻博士後期課程単位取得認定退学．同年より奈良先端科学技術大学院大学情報科学研究科助手．1997 年 12 月より岡山県立大学情報工学部情報通信工学科助教授，現在に至る．データベース，ワークフロー，グループウェア等に興味を持つ．博士（工学）．電子情報通信学会，ソフトウェア科学会，ACM，IEEE 各会員．



横田 一正（正会員）

1972 年京都大学理学部卒業．1985 年沖電気工業（株）入社．1985 ～ 1995 年新世代コンピュータ技術開発機構（ICOT）出向．1995 年京都大学大学院工学研究科助教授．1997 年から岡山県立大学情報工学部情報通信工学科教授，現在に至る．工学博士．著書に「新データベース論」，「ゲーデルの世界」等．高度データベース，知識表現，論理型言語等に興味を持つ．日本ソフトウエア科学会，人工知能学会，ACM，IEEE 等会員．