

メタレベル記述可能なワークフロー支援システム

井坂源樹*¹ 布川博士*² 宮崎正俊*³ 白鳥則郎*⁴

*¹ 東北大学電気通信研究所

*² 宮城教育大学理科教育研究施設

*³ 東北大学大学院情報科学研究科

*⁴ 東北大学大学院情報科学研究科/電気通信研究所

e-mail: motoki@shiratori.riec.tohoku.ac.jp

URL <http://www.shiratori.riec.tohoku.ac.jp/~motoki/>

ワークフローとして定型的作業を支援するためには、各々の作業を電子化することに加え、全体の作業手順も記述しなければならない。従って、定型的作業自身を記述できる枠組み（モデル）とそのための記述言語が必要となる。

本稿では、我々が提案してきた定型的作業の記述モデルであるワークフローのダブルモデルを述べ、更にこのモデルによりワークフローがワークフロー自身を扱えるメタレベル記述が可能なワークフローを提案する。メタレベル記述により、ワークフローが作業の進行に応じて変更可能であることや組織間にまたがるような複合的かつ大規模な作業が支援可能なシステムを構築できることを議論する。

Metalevel discription in Workflow Management System

*¹Motoki ISAKA *¹ *²Hiroshi NUNOKAWA *³Masatoshi MIYAZAKI *⁴Norio SHIRATORI

*¹Research Institute of Electrical Communication, Tohoku University

*²Institute for Science Education, Miyagi University of Education

*³Graduate School of Information Sciences, Tohoku University

*⁴Graduate School of Information Sciences, Tohoku University/Research Institute of Electrical Communication

We have developed a model for procedural activities named <R<DFAP>> model. The advantages of this model are natural descriptiveness of procedural activities and easiness of derivation of workflow management system which supports those activities.

In this paper we propose metalevel description in workflow management system by extending the model. In addition, we argue metalevel description enables a workflow management system to change itself dynamically and support compounded and large scale activities among many organizations.

1. はじめに

複数の作業員によってある定められた順番により進行する作業を支援するシステムとしてワークフロー支援システムに関する研究がここ数年盛んに行われている[1][2]。しかし、それらの多くは単に書類を電子化し、組織における一定のルール（作業手順）により書類を転送したり、管理を行なうシステムを作成したものが多く、

我々も定型的な書類を扱う作業（定型的作業）を支援するワークフロー支援システムを開発してきた。このシステムを構築するために、定型的作業のモデル化を行い、その支援のシステム記述モデルを明らかにした。また、実際の作業の記述を行いその実用性の評価を行った[3]。

本稿では、我々の提案したワークフローのモデルを拡張することにより、ワークフローがワークフロー自身を扱えるメタレベル記述が可能なワークフローを提案し、メタレベル記述により、ワークフローが作業の進行に応じて変更可能であることや組織間にまたがるような複合的かつ大規模な作業が支援可能なシステムを構築できることを議論する。

2. 定型的作業とそのワークフロー記述[3]

2.1 定型的作業の定義

本稿では、次の(1)~(5)を満たす作業を定型的作業と呼ぶ。

(1)定まった役割

定型的な作業を行う作業員やソフトウェアはある役割を持ち、行う作業は定まっているとする。

(2)データの枠組み

用いられる書類の形式が定められているので、書類に求められるデータの種類は定まっている。例えば、発注書には発注者名、連絡先、希望する製品名、などがある。実際の作業では、定型的数据が定型のフォーム上にあてはめられ書類という形で用いられている。

(3)定型のフォーム

定型的作業を行う上で、その作業員間で受け渡される書類の形式と種類は作業業務により定まっている。

(4)定型の処理

作業で用いられる書類への記入手続き、記入

内容に対する処理は定まっているとする。

(5)定型的作業プロトコル

作業手順が定まっているものとする。ここでいう作業手順とはどの役割からの書類には、どのような処理をするかをいう。この作業プロトコルとは、単に書類の受渡し手順を指すものではない。

一般に定型的な作業は、(1)~(5)の組み合わせで遂行される。作業員はある書類を完成させ、それを次の作業員に送る。書類を受け取った作業員は、送られてきた書類をもとに新たな書類を作成したり、その書類の内容を更新することで作業が進行する。

このように一連の定型的作業は書類の受け渡しにより進められ、書類の受け渡し順序及び、使用される書類はその業務手順によって定められている。

例として図1に示す商品の受発注を分析する。この商品受発注という一連の作業において、(1)卸し、小売り、買い手、事務といった役割、(2)作業に関わるデータの枠組み、(3)見積書や請求書、納品書などの定まったフォーム、(4)フォームへの必要事項の記入、金額の計算などの書類作成に必要な処理、(5)図1の1~10のように書類に対する処理と受渡しの順番が定まっている。従って、商品受発注は本稿でいう定型的作業であるといえる。

2.2 定型的作業のワークフロー記述モデル[3]

前節で定型的作業を定義したが、これをワークフローとして定式化する定型的作業のワークフロー記述モデルを述べる。ワークフローにおける各作業は以下で定義されるRおよびDFAPを用いてタプル<R <DFAP>>によりモデル化ができる。これをワークフローのタプルモデルと呼ぶ。

(1)役割 (Roles, $R = \{r_1, r_2, \dots, r_j\}$)

定型的な作業を行う上で、作業員は何らかの役割を担っており、作業員は役割に応じて作業

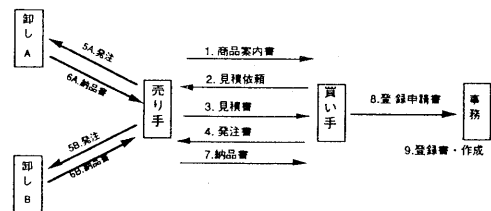


図1. 商品受発注における作業手順

を行う。実際の作業者を作業業務の内容により、役割という形で抽象化したものである。

例： r_1 =買い手, r_2 =売り手, r_3 =卸しA, r_4 =卸し

(2)フォーム (Forms, $F=\{f_1, f_2, \dots, f_k\}$)

定型的作業に用いられる定型的な書類の枠組みである。フォームに作業によるデータが記入されることによって書類となる。

例： f_1 =見積り依頼書, f_2 =見積り書..... f_k

(3)データの枠組 (Data, $D=\{d_1, d_2, \dots, d_i\}$)

作業をすすめる上で必要なデータの種類である。

例： d_1 =氏名, d_2 =連絡先, d_3 =商品名,..... d_i

(4)アクション (Actions, $A=\{a_1, a_2, \dots, a_m\}$)

各書類に対する処理。Roleが記入を行うという処理,記入された内容に対する処理がある。また、書類の完成、すなわち作業の終了により次の作業のもとへ送るなどの処理がある。

また、ある書類が期限までに完成されなかったら自動的にその旨を伝えるなどの全体の作業における書類の管理機能も記述される。

(5)プロトコル (Protocol, $P=\{p_1, p_2, \dots, p_n\}$)

$p_1=\langle r_1, D_1, F_1, A_1, r_2 \rangle, \dots, p_i=\langle r_i, D_i, F_i, A_i, r_{i+1} \rangle, \dots, p_n=\langle r_n, D_n, F_n, A_n, nil \rangle$

但し、 $r_i, r_j \in R, D_i \subset D, A_i \subset A, F_i \subset F$ である。

$p_i=\langle$ その作業を行う役割(role), その作業に必要なデータの集合, その作業に必要なフォームの集合, その作業に必要な処理の集合, 次の作業を行う役割 \rangle から成り立つ。

このように作業プロトコルはある段階の作業の記述において、使用するフォーム、行うべきアクションを記述する。すなわち、書類の受け渡し順序のみを記述したものではない。

定型的作業は以上の(1)~(5)により一般的に記述できる。例えば、図2に示すワークフローは以下

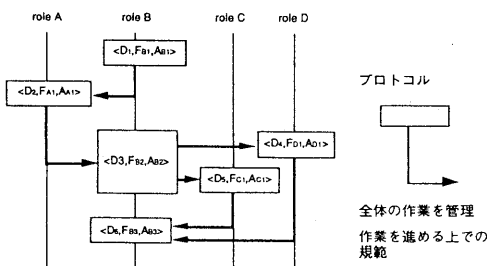


図2. $\langle R \langle DF \rangle \rangle$ によるワークフロー記述

のように記述される。

$W=\langle \langle \text{roleA, roleB, roleC, roleD} \rangle \langle D F A P \rangle \rangle$

但し、 $D=D_1 \cup D_2 \cup D_3 \cup D_4 \cup D_5 \cup D_6$

$F=F_{A1} \cup F_{B1} \cup F_{B2} \cup F_{B3} \cup F_{C1} \cup F_{D1}$

$A=A_{A1} \cup A_{B1} \cup A_{B2} \cup A_{B3} \cup A_{C1} \cup A_{D1}$

$P=\langle \langle \langle \text{roleB, } D_1, F_{A1}, A_{A1}, \text{roleA} \rangle, \langle \text{roleA, } D_2, F_{A1}, A_{A1}, \text{roleB} \rangle, \langle \text{roleB, } D_2, F_{A1}, A_{A1}, \text{roleA} \rangle, \langle \text{roleD, } D_4, F_{D1}, A_{D1}, \text{roleB} \rangle, \langle \text{roleC, } D_5, F_{C1}, A_{C1}, \text{nil} \rangle \rangle \rangle$

また、図1で示した商品受発注をワークフローとして図として表現したものが図3である。

3. 定型的作業を記述するワークフローのメタレベル記述

3.1 メタレベル記述

メタレベル記述とは、タプルモデルにおける各要素(R,D,F,A,P)がその一部としてタプル $W=\langle R' \langle D' F' A' P' \rangle \rangle$ をその要素として持つことである。以下ではそれぞれの要素についてメタレベル記述の意味を考察する。

(1)Rが $W=\langle R' \langle D' F' A' P' \rangle \rangle$ を要素とする場合

$R=\{r_1, r_2, \dots, r_i, \dots, r_j\}$ において、

$r_i=\langle R' \langle D' F' A' P' \rangle \rangle$ の時、すなわち、

$W=\langle \{r_1, \dots, \langle R' \langle D' F' A' P' \rangle \rangle, \dots, r_j \} \langle D F A P \rangle \rangle$

役割として抽象化した作業内容は、実際にはいくつかの役割により行われるワークフローとして行われることを意味する。

物品購入を行うワークフローを考える。購入する商品が予算内かの確認や金銭管理は予算の管理を行うワークフロー $W_{\text{金銭管理}}$ が一元的に管理すべきである。物品購入ワークフロー $W_{\text{物品購入}}$ において“金銭管理者”という役割(role)を予算管理ワークフローが行うのが妥当であり、R

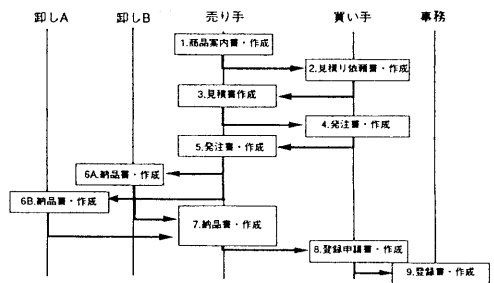


図3. 商品受発注のワークフロー

金銭管理者がW_{金銭管理}を要素とすることで実現される。

(2)Dが<R' <D' F' A' P'>>を要素とする場合

$D = \{d_1, d_2, \dots, d_i, \dots, d_k\}$ において、
 $d_i = \langle R' \langle D' F' A' P' \rangle \rangle$ のとき、すなわち、
 $W = \langle R \langle (d_1, d_2, \dots, \langle R' \langle D' F' A' P' \rangle \rangle, \dots, d_k) F A P \rangle \rangle$
Dは任意のアクション($a \in A$)からアクセス可能である。すなわち、Dとしてのワークフロー<R' <D' F' A' P'>>は任意のアクションにより任意の作業の段階で起動できることを意味する。

例：ワークフロー<R <D F A P>>の進行にともない、あるroleが別のワークフローの担当になった場合、そのroleが d_i としてのワークフロー<R' <D' F' A' P'>>を起動することで実現できる。

(3)Fが<R' <D' F' A' P'>>を要素とする場合

$F = \{f_1, f_2, \dots, f_i, \dots, f_l\}$ において $f_i = \langle R' \langle D' F' A' P' \rangle \rangle$ の時、すなわち、
 $W = \langle R \langle D \{f_1, f_2, \dots, \langle R' \langle D' F' A' P' \rangle \rangle\} A P \rangle \rangle$

フォームとして<R' <D' F' A' P'>>が記述されるということは、roleはフォームに対して処理を行えるため、<R' <D' F' A' P'>>を編集することができることを意味する。すなわち、作業者が<R' <D' F' A' P'>>を新規に作成したり、<R' <D' F' A' P'>>= $\langle R \langle D F A P \rangle \rangle$ の場合は現在進行中のワークフロー<R <D F A P>>を変更可能であることを意味する。

これにより、あらかじめ定められた手順によるワークフローの進行以外に作業の進行状況に対応した作業支援やその動的な変更が可能となる。これは[4]で提案されている開放型ワーク

フロー支援システム(開放型WFMS)に対応する。

(4)Aが<R' <D' F' A' P'>>を要素とする場合

$A = \{a_1, a_2, \dots, a_i, \dots, a_m\}$ において、
 $a_i = \langle R' \langle D' F' A' P' \rangle \rangle$ の時、すなわち、
 $W = \langle R \langle D F \{a_1, a_2, \dots, \langle R \langle D F A P \rangle \rangle, \dots, a_m\} P \rangle \rangle$

Wにおいて、 $p_s \in P$ により指定される $r_i \in R$ において、その r_i のみでWを実行できることを意味する。すなわち、これは(2)の特別な場合であり、 p_s によって指定される特定の作業においてのみ、他のワークフローを呼び出すことができるのを意味する。

図2に示した商品受発注のワークフロー(組織間のワークフロー)におけるrole“買い手”がある研究室であったとする。この場合、商品の注文を行うのに“発注書”を作成するのに、研究室として書類に記入を行うだけであるが、そのためには研究室内では、どの機種が適当なのか、学生の意見は、スタッフの意見は、予算に収まるのかといったことを行うワークフロー(組織内のワークフロー)があるはずである。このワークフローは“買い手”である研究室固有のフローとして記述されるのが妥当である。

組織間と組織内のワークフローは作業の位置付けを異にするものであり、別のワークフローとして互いが関連性を持つよう記述されねばならない。これらの作業の形態はAの要素としてのワークフローとしてメタレベル記述をすることにより記述可能となる。

(5)Pに<R <D F A P>>を要素とする場合

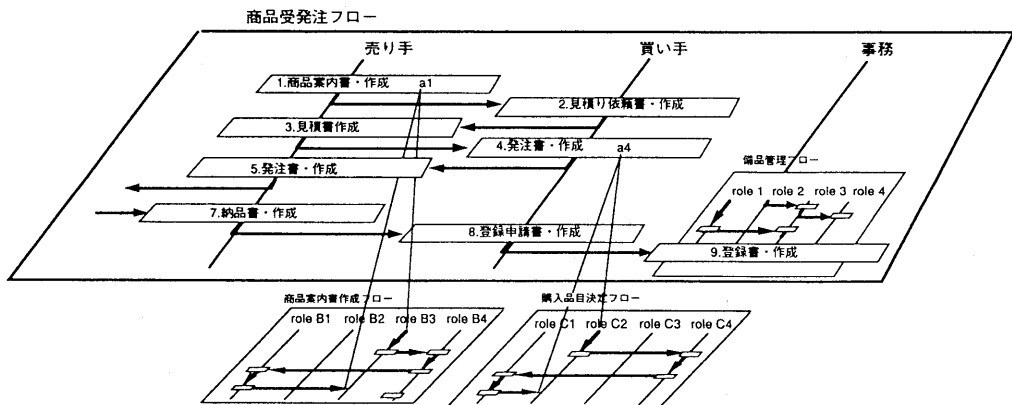


図4. メタレベル記述の適用例

$P=\{p_1, p_2, \dots, p_i, \dots, p_n\}$ において、 $p_i \in P$ の時、すなわち、 $p_i = \langle r_x, D_s, F_i, A_u, r_y \rangle$ と記述されるので、 p_i にメタレベル記述を行うのは、 p_i の任意の要素が $\langle R' \langle D' F A' P \rangle \rangle$ となる。

これは形式的には、

$\langle R \langle D F A \{p_1, p_2, \dots, \langle r_x, D_s, F_i, A_u, r_y \rangle, \dots, p_n\} \rangle \rangle$

但し、 $\{r_x, D_s, F_i, A_u, r_y\} = \langle R' \langle D' F A' P \rangle \rangle$ 、と書くことができる。

このように(1)~(4)で述べた記述法はすべてプロトコル(p)で行うことが可能であるが、性質や機能により分類した $\langle R \langle D F A P \rangle \rangle$ モデルの簡潔性を乱すことになるため意味がない。

3.2 適用例

メタレベル記述例を図4に示す。

“商品受発注フロー”は商品受発注における“売り手”“買い手”などの組織にまたがるワークフローを記述している。“商品案内書作成フロー”“購入品目決定フロー”はそれぞれの組織内での商品受発注における書類を作成するためのワークフローであり、それぞれの役割の固有ものであり、商品受発注の一つのアクション(a)として記述される。また、商品購入後の備品管理は事務が一括して行うべきであり、“備品管理フロー”として商品受発注フローの役割として記述される。

また、前節で述べたようにワークフローをフォームとして扱うことが可能なので、各作業の段階で作業の進行にともなうフローの変更を行える。

4. ワークフロー支援システムの記述

4.1 $\langle R \langle D F A P \rangle \rangle$ の記述実験[3]

東北インターネット協議会 (TiA) 事務局における運用作業に適用した[6]。まず、我々のモデルに基づき分析し定型的作業として表した。全体の作業は43個の $\langle R \langle D F A P \rangle \rangle$ で示せた。また、これらのワークフローは互いに関連があり、完全に独立した作業ではない。その関連性をメタレベル記述により記述することができた。

4.2 支援システムの実装[3]

4.1で述べた43個のワークフローの内6つを支援システムとして実装した。これにより、 $\langle R \langle D F A P \rangle \rangle$ モデルによりシステム化が可能であることがわかった。定型的作業を支援するワークフロー支援

システムを記述することは、2章で述べた $\langle D F A P \rangle$ をプログラムとして記述することである。DFAP記述言語は分散環境記述言語であるDeLis上の実装されている[5]。また、Rは以下のようにして実現された。

(1) 作業者が人間である場合。

作業者である人間は書類を介して作業を行う、すなわち人間とのインタラクションを行うGUIとして表示することによりRoleが実現される(図5)。このフォームはユーザインタフェースでありデータが埋め込まれ書類そのものである。

(2) 作業者がソフトウェアである場合

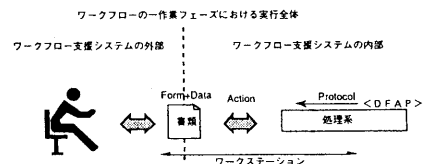
ソフトウェアが $\langle D F A P \rangle$ と作業を行う、すなわちフォームはroleであるソフトウェアに対するアプリケーション・インタフェースとして記述することを実現される(図5)。詳しくは[3]を参照されたい。

4.3 メタレベル記述例

定型的な作業に使用するデータを複数の複数のワークフローで利用することが頻繁に発生する。たとえば、同じ台帳群を複数の人が同時に利用することなどがある。この場合は台帳を一元的に管理するワークローが行うべきである。これをDBMSを用いたワークローがroleとして作業を行うシステムを実装した[3]。

また、東北インターネット協議会 (TiA) 事務局における運用作業のワークフローのうちでも、例えば“文書作成配布”“作業依頼”“文書作成”があるが、現在のところこれらは独立のワークフローとして実装されている。しかし、後者の2つは“文書作成配布”のフローにおける一作業をして行われるワークフローであるので、メタレベル

(a) Roleが人間の場合



(b) Roleがアプリケーション

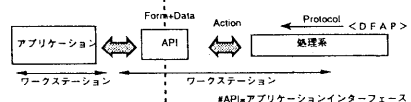


図5. Roleの実現

記述により統一的なワークフローとして実装することができる。

5. 考察

ワークフロー型の協調作業支援に関する研究は多く行われている[1][2]。作業をワークフローとしてとらえ、作業手順を明確化、作業の流れをスク립ト化し、計算機による定型作業として支援する言う点では我々の作成したシステムと同じである[7][8][9]。

しかし、これらのアプローチでは単に書類を電子化し作業手順によって作業間でやりとりできるシステムを構築したものである。

一方、我々は単なる書類の電子化とそのやりとりの手順を電子化しただけではなく、ワークフローの概念を定めるとともに定型的作業をワークフロー記述を行うモデルを提案し、それを支援システムとして記述する手法を提案した。実際、作業を行うのは現場の作業者であり、作業を知るのは彼らである。従って、作業の支援システムを構築する場合にまず作業者が彼らの作業を分析し表現できる必要がある。現段階では図3に示したワークフローの図的表現を用いる程度にしか分析方法を提案していないが、ワークフローという概念が作業者が用いて自分達の作業を表現する手段となる。さらに作業者が表現したワークフローをもとにシステムを構築する側で<DFAP>のうちシステムとして記述し、作業支援が行える部分が支援システムとなる。

このように、我々のアプローチは実際の作業の分析とシステムの構築が自然と行えるモデルを提案したという点で重要である。また、従来のアプローチでは、具体的な作業モデルがないために、作業をある作業レベルで切り出したワークフローしかシステム化できなかった。一方、我々のワークフローのモデルはメタレベルでの記述もできるため、大規模かつ複合的な作業もワークフローとして統一的に扱え、その支援を行うことができる。

6. まとめと展望

本稿では定型的作業をワークフローという概念を用いてモデル化しその記述モデルを明らかにした。更に、そのモデルに基づきメタレベルでのワークフローの必要性とその記述手法を提案し

た。実際、複数の組織にまたがって行われる大規模な作業は、各組織ごとに作業を行った結果が他の組織に作業の結果として伝達され次の作業が行われる。組織間では、対外的にはある種の作業行う役割として認識され、内部でどのような作業が行われているかはわからない。よって、組織内の作業を一つのワークフローとして全体のワークフローから扱えることが重要である。

また、2章で定義した要素がすべて固定的である定型的作業を対象としが、実際にはある抽象度レベル（例えば書類の流れに着目）では定型的な枠組みのなかで進行する作業も、他の作業との関連など作業を制約する条件や作業の進行状況により生ずる作業手順の変更などがあり、実際には完全には定型にはならない。本稿では、作業者であるユーザが作業の変更をユーザが支援システムに伝える方法をメタレベル記述によりフォームとしてワークフローを扱えることを示した。今後、実際にユーザ自身がワークフローの変更を行えるようなシステムを本稿で提案した手法に基づき行うことによりモデルおよびシステムの評価を行う予定である。

参考文献

- [1]F. H. Lochovsky, "Supporting Organizational Activities : Status AndProspects", Proceedings of The IISF/ACM JAPAN International Symposium in "Compuers as our Better Partners" Ed : H.Yamada et. World Scientific
- [2]C.A Ellis,S.J.Gibbs, G.L.ReIn, Groupware:Some Issues and Experiences, Communication of the ACM, January 1991/Vol.34, No1
- [3]井坂,布川,樋地,宮崎, 定型的作業のモデル化と支援システムの実装, 電子情報通信学会技術研究報告Vol.94 No.373(OFS94-30~36),1994
- [4]国島,上林, 協調作業の「場」に基づく開放型ワークフロー管理システム,第51回情報処理学会全国大会講演論文集, pp.6-179~6-180,1995
- [5]三石,布川,宮崎,野口, 分散環境のための言語系 DeLis, 情報処理学会研究会報告93-PRG-10,pp.57-64,1993
- [6]佐藤, 布川, 樋地, 脇山, 太田, 野口, インターネット運用作業のワークフロー化の考察とその電子化の試み, 第48回情報処理学会全国大会講演論文集, pp.6-219~6-220,1994
- [7]市村,松下 : PilotMail, 発言と行動の管理に基づいた協同作業支援電子メール,情報処理学会論文誌,Vol.33 No.7, 1992
- [8]松尾,服部,橋本,貫井, エージェントメールシステムのワークフロー制御への適用,情報処理学会研究会 94-GW-7 pp.59-66,1994
- [9]垂水,田淵,吉府 : ルールベースの電子メールによるワークフローの実現,情報処理学会論文誌, Vol.36 No.6, pp1322-1331,1995