ルールベースの電子メールによるワークフローの実現

垂水浩幸[†] 田渕篤[†] 吉府研治[†]

本論文では，著者らの開発したグループウェア開発支援環境「育組」（いくみ）と，UNIX上のルールベース電子メール基盤「め組」（めぐみ）によるワークフロー定義，制御，監視の方式について述べる．このシステムでは，め組のルールによる制御機能を利用して，ルールによる帳票回覧を実現し，それによって業務の連携を図ってワークフローの運用を実現する．ワークフローの進捗は各作業者の着信，開封，受理，送信の各タイミングで記録され，監視・報告される．ワークフローは育組によってチャート形式で定義できる．育組ではワークフローの分岐条件等もほとんどマウス操作だけで定義できるので，エンドユーザの力で定義できる範囲は広くなっている．このように定義されたワークフローはルールに変換され，め組に提供される．本システムの特徴は，ワークフローを集中制御するサーバを持たず，また組織間のリンクは電子メールのみで良い点にあり，これによって組織をまたいだワークフローの運用も容易になる．

Workflow Implementation with Rule-Based E-Mail

HIROYUKI TARUMI,[†] ATSUSHI TABUCHI [†] and KENJI YOSHIFU [†]

Method of workflow definition, execution, and supervision with the groupware development environment *IKUMI* and rule-based e-mail system *MEGUMI*, developed by the authors, is described. In this system, workflow is realized by the formed e-mail circulation with the rule-based control in MEGUMI. The workflow progress is supervised by the log records, which is made at the event of receiving, opening, accepting, and sending e-mail messages at each worker. The workflow is defined with a graphical chart drawn with IKUMI. Since the chart can be drawn almost only with mouse operations, even if the definition includes conditional branch, end-users are much empowered in defining workflow. The workflow definition is transformed into rules for the MEGUMI system. In this system, there are no central workflow control server, and only e-mail links are assumed between workgroups. This makes it easier to establish inter-organizational workflow management.

1. はじめに

本論文では，著者らの開発したグループウェア開発支援環境「育組」[1),2)]（いくみ）と，UNIX上のルールベース電子メール基盤「め組」（めぐみ）によるワークフロー定義，運用，管理について述べる．

これまで行われてきたグループウェアに関する研究開発には，（1）電子メール，電子掲示板などの非同期通信に基づくツール[3)]，（2）在席電子会議などのリアルタイム通信に基づくツール[4)]，（3）共同執筆，意思決定支援などの特定用途向けツール[5)]，などがあるが，これらはいずれも単一の作業フェーズの支援に留まっており，複数の作業フェーズからなる大規模な実際の協調作業を総合的に支援するには至っていない．

一方，最近ワークフローシステムが注目されている．ワークフローシステムでは，複数の作業フェーズからなる協調作業の定義を容易化し，その運用を制御し，運用状況を監視・報告する．これらのシステムでは，帳票記入等の簡単なビジネス業務に分野が限られてはいるが，ある程度規模の大きい協調作業の支援が可能である．

しかしながら，既存のワークフローシステムは制御を集中的に行うサーバの存在を前提としてるため，すべてのユーザ端末はサーバとの結合を要求される．従って，組織をまたぐ業務のような，より開放的な運用には向いていない．

本論文で述べるシステムは，ワークフローの定義を支援する育組と，ワークフローの運用制御・監視を行う電子メールシステムめ組によって構成される．ワークフローは，電子メールを制御するルールによって記述される．このシステムは，単一業務制御サーバの存在を仮定しないことが特徴であり，複数組織をまたぐ協調作業の場合であっても，組織間に電子メール以外

† NEC 関西C&C研究所
Kansai C&C Research Laboratories, NEC Corporation

の通信リンクを張る必要がないという利点を持っている．

本論文では育組のワークフロー定義方式，ルール生成方式，およびめ組によるワークフロー制御・監視方式について述べる．なお，育組は，帳票処理に代表される簡単なワークフロー定義に限らず，前記(1)～(3)のグループウェアツールを統合する環境を目標としているので，著者らはこれをグループウェア開発支援環境と呼んでいる．

以下，2章で電子メール基盤め組について説明した後，3章にて育組によるワークフロー定義方式とその運用方式について述べる．4章では他のワークフローシステムと比較する．

2. 電子メール基盤「め組」

2.1 概　　要

め組には，以下の特徴がある．

1. OVAL[3] 等と同じく，半構造型の電子メールである．ただし，帳票様式のユーザインタフェースを持つ(図1)．
2. 図形，イメージ，音声などのマルチメディアデータを扱える．
3. メールの形式は，MIME[6] に基づいて従来のUNIX メールを拡張したものである．
4. 回覧機能，優先度付け機能（特急メール），返信等に関する期限管理機能を有する．
5. HyperScheme 言語[7] によるルールが記述でき，これによってフィルタリング(メールの選択的受信，選択的ファイリングなどの機能)や，メール着信時のアプリケーション起動などの柔軟な処理が可能である．
6. メールの流通状況（回覧がどこまで達しているか，受信者による開封状況等)を監視できる．(トレース機能と呼ぶ)
7. これらの機能は電子メールツールとしてユーザに直接提供されるだけでなく，API(Application Program Interface）としても提供され，各種アプリケーションへの電子メール機能組み込みが可能である．

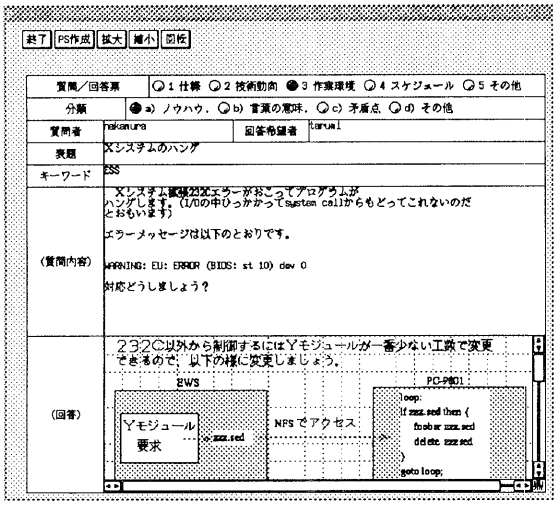

図1　め組の帳票型ユーザインタフェース
Fig. 1　Form style E-mail interface in MEGUMI.

2.2 構　　成

め組の諸機能は，各利用者ごとに常駐するサーバプロセス（megumi-server）によって実現される．megumi-server は sendmail より利用者側に位置する．メールハンドラおよび各アプリケーションは，megumi-server とは別のプロセスであり，megumi-server とプロセス間通信を行う．クライアント側にはライブラリ(libmegumi)が提供されており，このライブラリの C++関数インタフェースが API となる*．このライブラリを用いてメールハンドラや各種アプリケーションの構築が可能である(図2)．

megumi-server： 論理的には各利用者ごとに一つずつ存在するメール管理プログラムであり，(1)受け取ったメールに対するフィルタリング等の自動処理，(2)メールフォルダの管理，(3)サーバ間相互通信(この通信も原則電子メール）によるメールの着信等状況の報告，(4)メールハンドラあるいはアプリケーションに対するメール到着・締切接近等の様々な通知，(5)ルールに基づく送信先の決定，等の機能を有する．システムコスト等運用上の都合に配慮して megumi-server は複数の利用者に対してサービスを行うよう設計されており，実装上は各利用者ごとに一つずつ存在しているわけではない．しかし，メール管理はあく

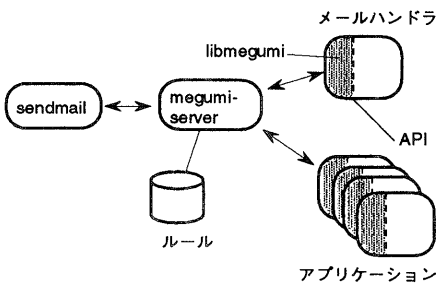

図2　各利用者ごとのめ組の構成(概略)
Fig. 2　Configuration of MEGUMI system for a User.

* API としては他に HyperScheme も利用できる．

まで利用者単位である．言い換えれば，メールを回覧する場合，回覧の各ステップとなるユーザの megumi-server において次の回覧先までの転送を制御している．よって，メールの回覧によって実現されるワークフローは分散制御されることになる．

メールハンドラ： 利用者がメールを送受信，編集するためのユーザインタフェースである．め組では簡単な標準メールハンドラを提供するが，API を用いたプログラミングにより，催促の形態，編集の方式等を変更した別のメールハンドラを作ることも容易である．め組の帳票メールだけでなく，一般の UNIX メールもこのハンドラで取り扱える．

アプリケーション： め組では，アプリケーションにメール機能を埋め込むことができる．例えば，ソフトウェアテスト管理ツールにテスト報告書発信機能を埋め込み，テスト結果集計ツールにテスト報告書受信機能を埋め込むといった使い方ができる．アプリケーションから送受信されるメールは，一般のメールと区別されるため，メールハンドラでは取り扱われない．sendmail から受け取ったメールの交通整理は megumi-server が行い，適切なアプリケーションに配信される．

libmegumi： メールハンドラおよびアプリケーションを構成するためのライブラリで，megumi-server との通信機能のほか，メール送受信，帳票解析，催促，トレース要求等の機能を C++ の関数として提供する．

2.3 ルールとその起動タイミング

ルールは HyperScheme で記述されているが，これは原則として育組によって生成され，配布されるもので，一般ユーザは HyperScheme を記述する必要はない．もちろん，能力のあるユーザならば HyperScheme ルールを自分で追加したり，修正したりすることは可能である．

例えば，【“評価” フィールドの値が 1 ならば，“FL 005” という ID のフロー（メール転送の 1 ステップの流れ）を選択し，さもなくば “FL 006” のフローを選択する】というルールは**図 3** のように書ける．

メールの持つ諸条件（帳票の種類等）によって呼び出されるルールは異なる．このような，適用すべきルールの選択を行うためのルールをメタルールと呼んでいる．メタルールの例を次に示す．

```
(gg-eval-rule "receive"
  ("exl"  1.0 1.0 "FL001")
  ("FormA" 1.0 1.0) "exl-1.0-C-1.scm")
```

この例は，【exl という ID のワークフロー定義（バージョン 1.0 以上 1.0 以下）の FL 001 という ID のフローにおいて FormA（バージョン 1.0 以上 1.0 以下）という帳票のメールを受け取った（receive）とき，exl-1.0-C-1.scm という名前のファイルに格納されているルールを実行する】である．

```
(if (equal?
      (substring (gg-get-field gg-current-mail "評価") 1 ))
    (gg-select-flow "FL005" #t)
    (gg-select-flow "FL006" #t)
)
```

図 3 ルールの例
Fig. 3 An example of Rule.

ルールの実行のタイミングには，メールの着信直後，送信直前の他，開封時，受理時がある．それ以外にワークフロー起動時に実行されるルールがある．

ここで，開封時とは，（ユーザまたはプログラムが）libmegumi を利用してメールファイルを最初にオープンしたときである．また，受理時とは，開封後に libmegumi の受理関数が呼ばれたときと定義している．

受理は「確かに受け取った，了解しました．」という意味の，作業者間あるいはアプリケーションプログラム間の慎重な確認作業のために用いるインタフェースとして準備した．これはメールを開封したが実際には読んでいない場合と，読んで理解した場合を区別するためである．

受理関数を呼び出すために必要なユーザの手続きを複雑にするようにメールハンドラを実装すれば，受理確認は信頼度を増すがユーザには操作負担になる．逆に簡単な手続きで受理関数が呼び出されるようなメールハンドラでは，ユーザの負担は減るが，習慣的・無意識的に受理手続きを発行するユーザが出て来るため，受理の信頼度は低くなる．もちろん開封時に受理関数も自動的に呼び出すようにメールハンドラを実装すれば，開封と受理は同じ意味になる．これらの実装法は応用分野によって使い分けることができる．

2.4 ワークフロー監視

前述したように，め組では，帳票電子メールを回覧することによりワークフローを実現する．ここで，回覧経路の他，各作業者の作業分担（すなわち帳票記入分担）もルールとして定義されている．ワークフローの監視機能は次のように実現される．

各作業者の megumi-server では，着信，開封，受理，送信のルール起動と同じタイミングでワークフロー進捗を報告する．この報告先は予め決められた特定の（例えばメール発信者や業務管理者の）megumi-server であり，そこでロギングされる．このログはあらかじめ設定された読み出し権を持つユーザやプログラムが，

遠隔で読み取ることができる．

ワークフローの途中で別の帳票が新たに起票される場合には，起票操作を含めてルール化される．このように派生した帳票メールの進捗ログも一括して管理される．

3. 育　　組

3.1 育組の構成

育組はワークフローチャートを編集し，ワークフローの実現に必要なめ組用のルールを生成するツールである．以下，本節では特に断らない限りメールとはめ組のメールを意味する．

育組は複数のツールにより構成される．これらのツールを大きく分けると，ワークフローを定義するための編集系ツールと，定義済のワークフローをルールに変換するための生成系ツールとに別れる．プログラミング言語に例えると，前者がエディタ，後者がコンパイラと言える．各ツール間の関係を図4に示す．GGtmpl, GGrole, GGbulletin, GGagent, GGwf, GGenvが編集系ツール，GGtrans, GGinstallが生成系ツールである．

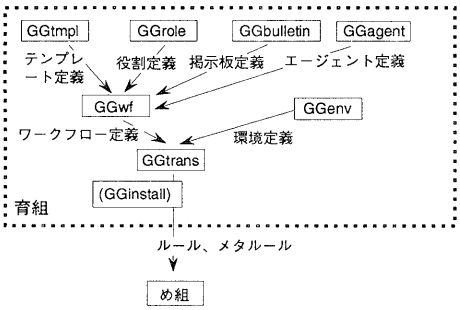

図4 育組のサブツール間の関係
Fig. 4 Configuration of IKUMI.

3.2 ワークフロー定義

ワークフロー定義を行う編集系ツールについて説明する．

3.2.1 GGtmpl

電子メール，電子掲示板に掲示するデータ等の形式(すなわち，帳票形式)を視覚的に定義するツールである．

GGtmpl (図5) は，画面上で帳票部品（ラベル，テキスト，日付，多肢選択，数値，図形，イメージ等を記入するフィールド部品）を帳票シート上にタイリング配置し，各部品の属性を個別に定義できるようになっている．フィールドの属性としては例えば罫線の有無，数値記入の場合の最大値，多肢選択の場合の選択可能数等がある．これらの属性は，各部品をマウスでダブルクリックしたときに現れるサブウィンドウで定義する．

GGtmplは，これらの定義を反映した「帳票定義」を出力する．この帳票定義は，独自の簡易言語によるものである．め組のメールはこの帳票定義をそのまま使用することができるので，ワークフロー機能を使わないポイント・ツー・ポイントの帳票型電子メールを利用する場合にもGGtmplを利用できる．

3.2.2 GGrole

育組で定義するワークフローは，処理主体であるノードと，ノード間のメールの流れを表すフローによるグラフ構造で表現される．育組で用いるのは，作業者あるいは役割をノードとする作業者中心型[8]のワークフロー記述モデルである．ノードには役割ノード，掲示板ノード，エージェントノード等があるが，GGroleはそのうち役割ノードを定義するツールである．

役割とは，電子メールを介した一連の協同作業の中

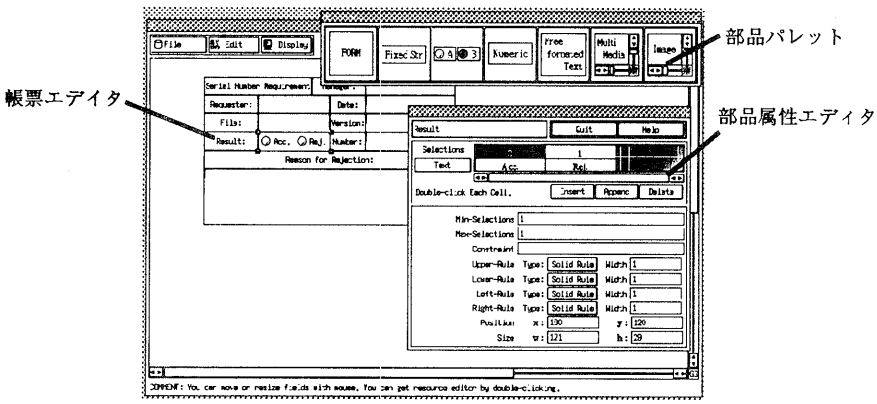

図5 GGtmplの画面例
Fig. 5 An example of GGtmpl Window.

で，個々の参加者が果たす作業義務であり，例えば「課長」「出庫担当者」「添削者」等がある．GGrole では役割の名前とアイコンのみ定義し，各役割の具体的な仕事の内容は後で GGwf 側で定義する．

3.2.3 GGbulletin

掲示板ノード[9]の仕様を定義するツールである．我々の提供する掲示板は，誰でもアクセス可能な掲示板ではなく，プロジェクトごとにアドホックに生成可能で，高度にカスタマイズ可能なものである．掲示板の詳細については，本論文では省略する．

3.2.4 GGagent

エージェントノード[2]の仕様を定義するツールである．育組におけるエージェントとは，役割を自動化したもので，電子メールを送受信する能力を持つプログラムである．エージェントはユーザとのインタラクションを原則として行わない．例えば，帳票のあるフィールドを自動的に埋めるプログラムや，電子メールによるアンケートを自動集計するプログラムはエージェントである．エージェントの詳細については，本論文では省略する．

3.2.5 GGwf

GGrole, GGbulletin, GGagent で定義したノードの間を，GGtmpl で定義した帳票形式の電子メールが転送される手順をチャートによって視覚的に定義する(図6)．

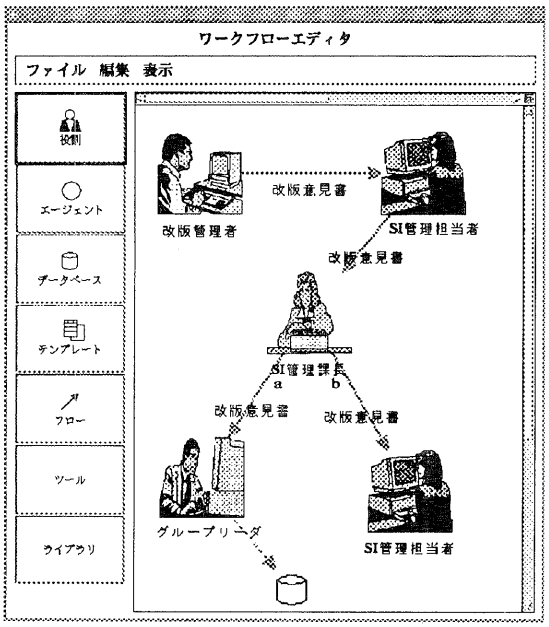

図6 GGwf の画面例
Fig. 6 An example of GGwf Window.

個々のフローについて，そのフローを流れる情報の形式である帳票を指定することができる．各フローの帳票形式は異なっていても良く，複数の帳票を順次利用するような業務も定義可能である．

さらに，チャートには表現できない，各ノードにおける処理の詳細を定義する．ここでは，図6中央にある役割（SI 管理課長という役割）の場合を例として説明する．

GGwf の画面上で，この役割ノードをダブルクリックすると，役割ノード詳細定義ウィンドウ(**図7**)が現れる．このウィンドウでは，受信した帳票の処理（表示，編集，プログラムへの入力等），記入範囲の指定，締切の指定，分岐条件の指定等を行う．このうち記入範囲の指定と，分岐条件の指定は画面上で帳票イメージを見ながらほとんどマウス操作でできるようになっており，極力スクリプト言語の使用を避けている．このことにより，エンドユーザの力でもより複雑なワークフローを定義できる可能性が高くなる．

例えば，分岐に関する定義は次のようにして行う．複数の流出フローから一つのフローを選択する場合の判定条件は，(1)帳票が流入した時点で記載されていた内容(記載内容)，(2)このノードでユーザが帳票に記入した内容(記入内容)，(3)プログラム実行の場合，プログラムの終了コードの値，(4)ユーザへのダイアログボックスによる問い合わせ，の4種から選択でき

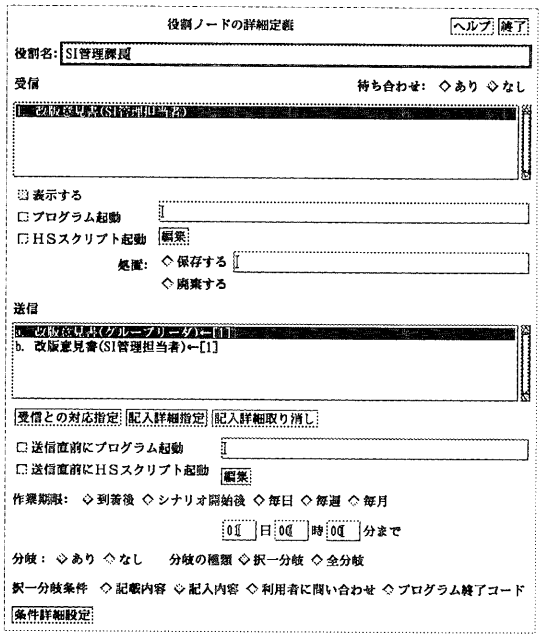

図7 役割ノード詳細定義ウィンドウ
Fig. 7 Window for detail role definition.

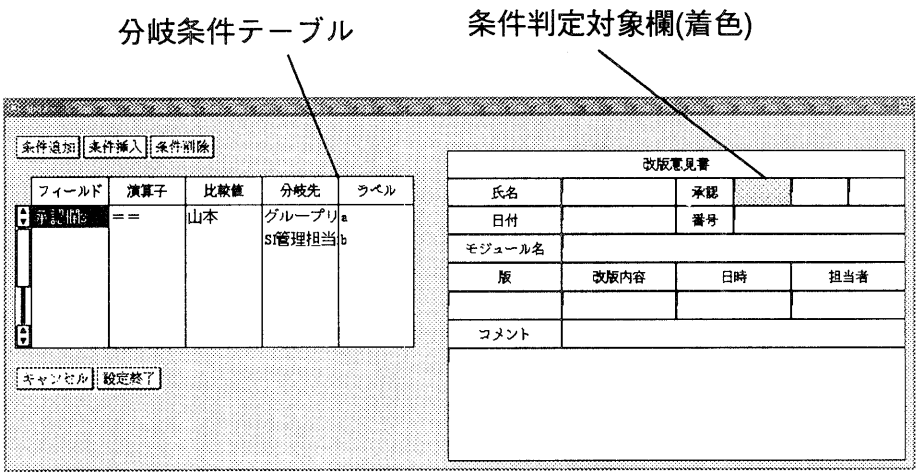

図8 分岐条件定義ウィンドウ
Fig. 8 Window for conditional branch definition.

る．このうち，記載内容，および記入内容による判定の場合，さらに分岐条件の指定も別のウィンドウ（図8)で行う．このツールでは，右側に出ている参照帳票中のフィールドをクリックしたあと，左側の分岐条件テーブル上でマウス操作を行い，そのフィールドに書かれている内容と規定値との比較による分岐条件テーブルを作成する．

この例では，承認欄3(図の帳票中着色されている部分）に「山本」と記載されていれば，グループリーダ(分岐ラベルa)へ，それ以外の場合にはSI管理担当者(分岐ラベルb)へ進む．キーボードから入力する必要があるのは「山本」という文字列のみである．

3.2.6 GGenv

GGenvは，アプリケーション実行環境に関する定義全般を行うツールである．例えば，各役割を務めるユーザのメールアドレス，組織構成，マシン・ネットワーク環境等が環境として考えられる．環境をシナリオの定義から分離することにより，シナリオの可搬性が向上し，例えば人事異動等の際の対応が取り易くなる．

現状では，各役割を実際に担当するユーザのメールアドレスを定義する機能のみ提供している．

3.3 ルール生成と実行

3.3.1 GGtrans

GGtransは，以上の各編集系サブツールから出力された定義をマージし，検証した後，め組用のルールを生成する．

め組を用いたワークフローの運用には，(1)あらかじめ，ルールを各ノードに配布(**事前配布方式**)，(2)ルールを実行時の電子メールに内蔵（**メール内蔵方式**)，の二つの方式がある．GGtrans利用時にはいずれかの方式を指定する必要がある．

ここで，事前配布方式には，

- ワークフロー定義が更新されたときに更新の手間がかかる
- 一度きりしか実行しないようなワークフローの場合に効率が悪い
- 役割担当者，グループのメンバを，ワークフロー定義時にすべて決めておかなくてはならない

といった短所があるが，半面

- 何度も繰り返される定型業務を効率良く処理できる
- 事前に配布されたルールが正当で適切なものかどうかチェックする余裕が十分にある

という利点がある．メール内蔵方式の場合は長所短所が逆になる．

メール内蔵方式の場合，メールのサイズが大きくなるという問題がありそうであるが，実際にはMIMEのexternal-body方式（ルール部分を外部ファイルとして共有)が利用できるので，その問題は回避できる．

育組では，両者を場合によって使い分けることを想定し，両方の方式をサポートする．

事前配布方式の場合，各ノードごとにルールを生成する．生成されたルールはGGinstallというツールで各ノードのmegumi-serverに配布される．役割ノードに対して生成されるルールには，以下のものがある．

1. メタルール．
2. ワークフローの開始時に起動されるルール．
3. 帳票の起票時に起動されるルール．
4. メールの着信時，開封時，受理時，送信時に起動されるルール．

```
(gg-set-mail-attribute gg-current-mail "GGNdeadline"
    '("reply" "yamamoto@hare.obp.cl.nec.co.jp"
        (gg-compute-deadline 'LOCAL 10000 0)))
(gg-call-maileditor gg-current-mail '() '())

(define ggtmp-selected-flow
    (cond ((string=? (gg-get-field ggtmp-M005 "承認欄3") "山本")"FL003")
          (else "FL004")))

(define ggtmp-selected-mail
    (assoc ggtmp-selected-flow '(("FL003" ggtmp-M005)("FL004" ggtmp-M007))))

(define ggtmp-selected-to
    (assoc ggtmp-selected-flow '(
      ("FL003" "nakata@hare.obp.cl.nec.co.jp")
      ("FL004" "tanaka@hare.obp.cl.nec.co.jp")
)))

(if (gg-cflowmail? ggtmp-selected-mail)
  (begin
    (gg-set-mail-attribute ggtmp-selected-mail "GGNto" ggtmp-selected-to)
    (gg-send-without-rule ggtmp-selected-mail)
  )
  (begin
    (gg-select-route ggtmp-selected-mail ggtmp-selected-to)
    (gg-set-mail-attribute ggtmp-selected-mail "GGNflowID"
          (symbol->string ggtmp-selected-flow))
    (gg-simple-send ggtmp-selected-mail)))

(if (not (gg-call-maileditor gg-current-mail '("承認欄3") '("TE15")))
  (gg-quit-rule #f))

(if (not (gg-call-maileditor gg-current-mail '("承認欄3" "TE15") '()))
  (gg-quit-rule #f))
```

図9 生成されたルールの一例
Fig. 9 An example of generated rule set.

生成例として,図6中央のSI管理課長ノードのルールの一部を図9に示す.

一方,メール内蔵方式の場合には,ルールを帳票別に整理して出力する.

3.3.2 運　　用

ワークフローは,始点ノードのユーザが,シナリオ起動コマンドを実行すると開始される.

実行時にやりとりされるメールには以下のIDが埋め込まれており,各ノードが電子メールを受け取ったときに,それがどのシナリオのどのフローに属するものかを判定できる.各ノードのmegumi-serverは,その判定に基づき,配布されたルールの中から適切なものをメタルールによって選択し,実行する.

シナリオID ワークフロー定義の単位をシナリオと呼んでいるが,このシナリオごとに一つつけられるIDである.

スレッドID 一つのシナリオは何回も実行される.例えば,購買要求処理というシナリオは,ものを一つ買う要求が発生するたびに実行される.そのような一回一回の実行をスレッド(thread)と呼んでいる*.各スレッドにはIDが付与される.

フローID 一つのシナリオには一般に複数のフローが存在する.これら一つ一つのフローにIDが付与される.

3.3.3 ルールのセキュリティ

ルールの安全性については十分な配慮が必要である.極端な話だが,メール内蔵方式で,悪意のあるユーザが,メールにファイル消去のルールを潜ませて来た場合,正直にルールを実行すると大変な被害を被る.防止策にはいくつかの方法が考えられるが:

- 特定の送信者のみルールを内蔵できるという限定を加えて,デジタル署名技術を用いて発信者を同定し,ルールの内容に改竄が加えられていないことをチェックすることは可能だが,これでは電子メール通信の持つ開放性が制限される.例えば,アンケートメールを制御するためのルールは,広

* 他システムではワークフローインスタンスあるいは案件(case)と呼ばれる場合がある.

い範囲の人（例えば社内全体）から送られて来たものであっても実行したい．

- 悪意がない場合でも，過失で誤ったワークフロー制御を行うルールを内蔵したメールを送ってしまうことがあるので，改竄されていなければ安心というわけではない．

等の問題がある．

そこで，育組では事前配布とメール内蔵の各方式について以下の制限を設ける予定である（未実装）．

- メール内蔵方式の場合，そのメール自身以外のデータに副作用を与えるルールは内蔵できない（ATOMICMAIL[10]と同様）．また，ルールが改竄されていないことをデジタル署名技術によって保証する．
- 事前配布方式の場合，ルール配布者の正当性とルールが改竄されていないことを配布時に保証すると同時に，処理対象のメール以外のデータに副作用のあるルールが配布された場合は，ユーザに確認を求める*．

3.4 合図フローの実装

フローには帳票フロー（formed flow）のほかに，合図フロー（beacon flow）がある．この点について補足する．

合図フローは利用者には見えない情報の流れで，あるノードでの処理の終了を別のノードに通知する働きを持つ．合図フローはワークフロー定義図中では破線の矢印で表現している．

合図は，複数作業者の連携による作業で，作業者間の連絡に必ずしも文書情報の交換を必要としない場合に有用である．例えば単に作業の完了を知らせれば良い場合，作業者がそれだけのために電子メールを書かされたり，読まされたりするのは煩雑である．例として，**図10**で，Role AからRole Cに進捗報告が送られるときには，必ずRole Cは最新の予算資料が必要であると仮定する．この作業が定型化されているならば，Role AからRole Cに進捗報告が送られるとき，Role AからRole Bに対して「予算資料を作成してRole Cに送付してください」というメールを毎回送るのは煩雑である．そこで，合図によってそれを知らせ，Role Bのmegumi-serverでは合図を受け取ったときに自動的に予算資料作成用のツールを起動可能状態にして待機させるようにルールを設定しておけばよい．

このような伝達は既存のワークフローシステムではワークフロー制御サーバの役目であったが，育組・めぐみでは分散制御となるため，合図が必要になる．

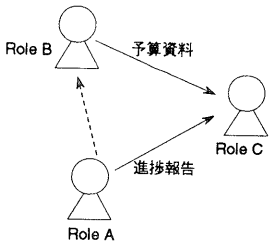

図10 合図の使用例
Fig. 10 An example of beacon message.

合図フローは，megumi-server間で自動的にやりとりされる電子メールによって実現できる**．このメールは利用者には見せない．

なお，帳票フローがデータフローを，合図フローがコントロールフローを意味するわけではない．実業務をデータフローとコントロールフローに分解することは不自然である．A氏からB氏に書類が渡されれば，何も言わなくてもそこには「処理してくれ」あるいは「読んでおいてくれ」という暗黙のコントロールフローがあるし，A氏からB氏に「仕事の続きをお願いします」との伝達，すなわちコントロールフローがあれば，BさんはAさんの持っていたデータを何らかの形で入手しなければ仕事の進めようがない．

育組の帳票フローは合図フローよりもデータフロー的な要素が強いが，コントロールフローを必ず含んでいる．合図フローはコントロールフロー的な要素が大部分であるが，実は，データ（参照ファイル名等）も運搬してよい．

4. 比　　較

4.1 Staffware等との比較

1993年後半から1994年前半にかけて，ワークフローを売り物にしたシステムが欧米の大手コンピュータメーカから相次いで発表された．例えば，Recognition社のFloWare，IBMのFlowMark/2，UnisysのStaffware，DECのTeamLinks，OlivettiのIBIsys，BullのFlowPATH等である[11]．

これらの製品は，従来オフコンや汎用機で実現されていたオフィス業務をパソコンやワークステーションで行えるようにし，その際の業務連携の仕掛けとしてワークフロー機能を導入したものである．業務の進捗管理，進行は単一の業務制御サーバによって行われて

* この確認の方式はユーザインタフェース上課題が多く，今後検討を要する．

** 電子メール以外の実現も考えられるが，本論文では省略する．

おり，電子メールによる連携ではない*．分散サーバの実現を可能としているものもあるが，実際にはサーバが途中で他のマシンに引っ越しをしているに過ぎないなど，事実上の単一サーバ構成である．

育組・め組はこれらのシステムと比較すると以下の特徴を有している．

1. Internetのメールを基盤とした疎な結合を前提としているため，組織をまたいだ協調作業を容易に支援できる．
2. トレース機能は，め組ではメールの送受信(仕事の開始，終了)だけではなくメールの「開封」「受理」などの細かいレベルで監視できる．監視の状況を表示するツールはAPIを用いて柔軟に構築できる．
3. 育組では各作業者の帳票記入範囲や分岐条件まで視覚的に指定でき，エンドユーザでも定義できるワークフローの範囲が広い．
4. MIME標準に従っているため，UNIXの既存のメールシステム上に容易に構築でき，かつ他のメールシステムとの共存も可能である．

特に，第一の特徴は，方式の本質的な違いに起因しており，電子メールを基盤とすることで実現できた特徴である．

4.2 Action Workflowとの比較

本節以降では，論文として公表されている学術成果と比較する．

Action Workflow[12]は，すべてのビジネスプロセスをProposal, Agreement, Performance, Satisfactionの四ステップからなるループの階層で表現するという，独特のアプローチでワークフローを表現する．

この方式は表現力があるが，フロー図の視認性があまり良くない，データの流れが見えない，西洋流の契約指向のアプローチであり，日本のオフィスには馴染まない等の問題がある．

このシステムも単一業務制御サーバ型であり，育組・め組とは性格を異にしている．

4.3 OM-1との比較

OM-1[13]においては業務の手順，データの流れ，担当者の割り当てなどをすべて表現できる．この表現方式は完成度が高く，表現能力も高い．しかしそのためチャートが複雑になるという欠点がある．育組のように作業者中心型の場合は，すべてをチャートだけで表現できるわけではないが視認性が良い．二つのチャート方式のどちらが最終的にエンドユーザに受け入れられるかは今後の実績を確認しないと評価できない．

なお，文献13)によれば，OM-1においても，ワークフローの制御は単一サーバによる．ただし，制御のためのやりとりはすべて電子メールによって実現するものとされており，組織間結合に必要なネットワーク構成の容易さの点では，育組・め組と同程度であると言える．育組・め組では，制御を分散させているため，各組織の自律的な運用ができる可能性がある．

4.4 Regattaとの比較

Regatta[14]は強力なワークフロー記述能力を持っており，階層的なワークフローの定義や種々のワークフロー部品の利用が可能である．育組・め組でもこれらの機能は提供する[2]が，ワークフロー記述方式の詳細に関する比較は本論文の範囲を越えるので別の機会に述べる．また，Regattaにおけるワークフローの制御方式については不明であり，この点についての比較は現状ではできない．

5. おわりに

育組・め組を用いた電子メールベースのワークフロー定義，運用，管理の方式について述べた．育組・め組の特徴は以下のようにまとめられる．

- Internetメール接続のみを前提とし，ワークフローを分散制御することにより，異組織間結合や，分散拠点対応が容易である．よって，外注や分散開発の盛んな製造業でも広範なアプリケーションが構築できる．
- 単に書類のありかを示すだけでなく，担当者が開封したか未読かを含めて，分散環境でのきめ細かい業務監視が可能である．
- エンドユーザによる基本的範囲でのワークフロー定義が可能である．特に分岐条件の定義等においても，帳票を直接操作する感覚で視覚的な定義ができる．
- 上記に加え，さらに高度のアプリケーションを構築したい場合には，C++, HyperSchemeの両者によるAPIを利用することができ，カスタマイザビリティが高い．

今後は，1章で述べた多種のグループウェアツールとの結合などを含めたトータルなシステムとして実現し，評価して行く予定である．め組は，事業部門でのドキュメント管理に応用されている[15),16)]．

謝辞 本研究を進めるにあたり，貴重なコメントを頂いたNEC関西C&C研究所の真名垣所長，宮井部長を始めとする諸氏に感謝します．

* 著者の知る限り，育組・め組と同じ方式をとるものはないが，実現の詳細を公表していないベンダもあるので，断言はできない．

参考文献

1) 垂水浩幸, ほか："GG" におけるワークフロー設計支援方式, 情報処理学会グループウェア研究会, GW-4-7 (1993).
2) Tarumi, H. et al.: IKUMI: A Groupware Development Support System with Visual Environment, in *Advanced Information Systems Engineering (Proceedings of CAiSE * 94)*, Wijers, G. et al. (eds.), Lecture Notes in Computer Science, Vol. 811, Springer-Verlag, pp. 66-79 (1994).
3) Malone, T. W. and Fry, Ch.: Experiments with Oval: A Radically Tailorable Tool for Cooperative Work, *Proc. CSCW'92,* ACM, pp. 289-297 (1992).
4) 渡部和雄, ほか：マルチメディア分散在席会議システム MERMAID, 情報処理学会論文誌, Vol. 32, No. 9, pp. 1200-1209 (1991).
5) Conklin, J. and Begeman, M. L.: gIBIS: A Hypertext Tool for Exploratory Policy Discussion, *Proc. CSCW'88,* ACM, pp. 140-152 (1988).
6) Borenstein, N. and Freed, N.: *MIME (Multipurpose Internet Mail Extensions) Part One*: Mechanisms for Specifying and Describing the Format of Internet Message Bodies, RFC1521 (1993).
7) 佐治信之, 景山辰郎：HyperStation：分散オブジェクト指向言語の構想, 第 45 回情報処理学会全国大会論文集, 2 Q-1 (1992).
8) 垂水浩幸, 岩崎新一：ワークフローシステム, 日本ソフトウェア科学会チュートリアル"CSCW" テキスト (Sep. 1994).
9) 吉府研治, ほか：グループウェアサービス基盤の構築～グループ情報管理部品～, 第 48 回情報処理学会全国大会論文集, 1 J-8 (1994).
10) Borenstein, N. S.: Computational Mail as Network Infrastructure for Computer-Supported Cooperative Work, *Proc. CSCW'92,* ACM, pp. 67-74 (1992).
11) 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).
12) Medina-Mora, R., Winograd, T., et al.: The Action Workflow Approach to Workflow Management Technology, *The Information Society,* Vol. 9, pp. 391-404 (1993).
13) Ishii, H. and Ohkubo, M.: Message-Driven Groupware Design Based on an Office Procedure Model, OM-1, *J. Inf. Process.*, Vol. 14, No. 2, pp. 184-191 (1991).
14) Swenson, K. D.: Visual Support for Reengineering Work Processes, *Proc. Conference on Organizational Computing Systems,* ACM, pp. 130-141 (1993).
15) 金政ふじ, ほか：電子メール基盤「め組」を利用したドキュメント管理システム「究仙」, 情報処理学会ソフトウェア工学研究会 (情報処理学会研究会報告, Vol. 94, No. 18), 97-6 (1994).
16) 金政ふじ, ほか：電子メール基盤「め組」を利用したソフトウェアプロセス管理—究仙—, 情報処理学会ソフトウェアプロセスシンポジウム論文集, pp. 87-96 (1994).

(平成 6 年 8 月 30 日受付)
(平成 7 年 4 月 14 日採録)

垂水 浩幸 (正会員)

1960 年生. 1988 年京都大学大学院工学研究科博士後期課程研究指導認定退学. 同年 NEC 入社. ソフトウェア工学, ユーザインタフェース, グループウェアの研究に従事. 現在, 関西 C & C 研究所主任. 日本ソフトウェア科学会理事. IEEE Computer Society, ACM 各会員. 工学博士. 共編「オブジェクト指向コンピューティング」(岩波書店).

田渕 篤 (正会員)

1964 年生. 1988 年京都大学大学院工学研究科修士課程修了. 同年 NEC 入社. グラフィカルユーザインタフェースおよびグループウェアの研究に従事. 現在, 関西 C & C 研究所勤務. 日本ソフトウェア科学会, 人工知能学会各会員.

吉府 研治 (正会員)

1969 年生. 1991 年大阪大学工学部精密工学科卒業. 同年 NEC 入社. 現在 NEC 関西 C & C 研究所勤務. グループウェアおよびユーザインタフェースの研究に従事.