

逐次追加型ワークフローにおける記述方式

6X-6

TOREC の提案*

飯塚 京子 梶垣 博章 平川 豊

NTT ソフトウェア研究所

1 はじめに

計算機のダウンサイジングとオープン化により、企業 LAN が急速に浸透している。このような企業ユーザを対象として、ワークフローシステム (WFS) やグループウェアツールの需要が拡大している [1]。WFS とは、ネットワーク上の定型作業の支援システムであり、適用する作業手順に合わせてワークフロー (WF) を定義し、WF に従い作業をサポートする。

WFS の問題点は、システム運用中に WF の修正要求が頻繁に発生することである。これに対し我々は、“逐次追加法” [2] を提案することで問題解決を計った。本稿では、逐次追加法で必要とする WF 記述の性質を検討し、WF の記述方式 TOREC を提案し、その記述評価を行う。

2 逐次追加法

WFS の運用中には、WF に定義していない作業手順を追加しなければならない場合が発生する。なぜならば、WFS の導入前に予想しうるすべての手順を洗い出し、WF を設計することが困難であるからである。予想困難な手順の例として以下がある。「クレジットカードによる支払い処理作業において、カードの不正使用事件が発生した。これまで WF に含まれていなかった、“ブラックリストと照合”する作業を追加する必要が生じた。」この問題を逐次追加法では、ユーザによる WF への作業手順の追加によって解決する。

逐次追加法を用いた WFS の構成を図 1 に示す。作業のサポートを行なう場合、“タスク定義ファイル”、“手順定義ファイル”、“リソース配置定義ファイル”を呼び出し、WF を実行する。WF に手順を追加する場合、“手順定義ファイル”のデータを WF 図に変換して WF エディタに表示し、WF 図の編集を行なう。

逐次追加における要求条件

逐次追加法では、システムに関する知識が乏しいユーザが WF の修正を行なうことを想定している。よって、WF を図的に表示し、容易に編集できる WF エディタが望まれる。そこで本稿では、WF エディタで使用する WF 図について考察する。

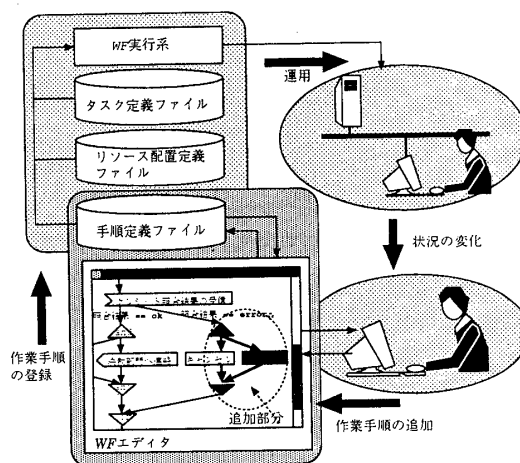


図 1: システム構成図

WF 図に手順を追加する場合、以下の 3 点を考える必要がある。

1. WF 図の上で、手順の追加箇所を探すことを容易にするためには何が必要か。
2. WF 図への手順の描き加えることを容易にするためには何が必要か。
3. 手順の追加により WF 図が大きくなることで発生する弊害は何か、またその解決策は何か。

この点を考慮して、ソフトウェアのインストール作業と通信販売作業の記述による予備実験を行なった。予備実験の結果から、以下の要求条件が WF 図に存在することが明らかになった。

1. 個々の作業者の行なう手順が連結した 1 つの WF 図で記述できる
理由：連結した 1 つの WF 図ならば、追加箇所の特定が容易であるため。
2. 1. 並行作業の記述が可能かつ容易である
2. 分岐の追加が容易である
理由：予備実験から手順の追加事例は、2.1) 新たな手順を既存の WF へ並行に追加する場合、2.2) 新たな手順を既存の WF から分岐させて既存の WF へ合流させる場合、以上の 2 つに分類できた。よって、WF 図がこの要求条件を満たせば、手順の追加は容易になる。
3. 閲覧性が良い
理由：WF 図の規模が大きくなると WF 図を理解することが困難になり、手順の追加が難

*TOREC; Task Order Expression Chart

しくなる。このため、閲覧性が良いWF図を使用することが望ましい。閲覧性に関する考察は、4章で行なう。

3 逐次追加用 WF 図 “TOREC”

本稿では、逐次追加用の WF 図 “TOREC(Task Order Expression Chart)” を提案する。

TOREC は、7 種類のノードとアークを用いて WF を表現する (図 2 参照)。各ノードの役割を以下に説明する。

- “task”
タスク定義ファイルに記述された作業を示す
- “and-branch” と “and-marge”
並行作業の開始位置と待ち合わせ位置を示す
- “or-branch” と “or-marge”
分岐位置と分岐の合流を示す
- “send-message” と “receive-message”
他の作業者へのメッセージの送受信を示す

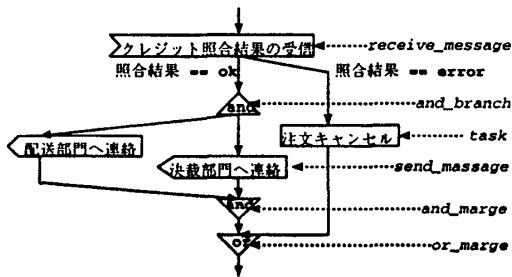


図 2: TOREC 記述例

TOREC は、“and-branch” と “and-marge”、“or-branch” と “or-marge” を用いることで、個々の作業者の WF を分断せずに、並行作業と分岐の追加を容易に行なえる。

4 評価

先に示した要求条件に関して、従来提案されている代表的な図的表記法と TOREC を比較する。

SDL[3] 通信ソフトウェアの下流仕様の記述言語。実行順序を問わない作業を並列に記述する場合は、複数の WF 図に分断される。

MSC[4] 通信ソフトウェアの上流仕様の記述言語。WF に分岐がある場合、分岐条件ごとに異なる WF 図に分断される。

PTN(PetriNet) 上記の条件全てを満たす。

R-Net[5] リアルタイムシステムでのメッセージの受信から送信までの動作記述する図。メッセージの送受信で WF 図は分断される。

要求条件 1,2 に関する比較結果を図 3 に示す。比較の結果から、PTN と TOREC 以外の図的表記法は逐次追加に適さないことが明らかになった。

	SDL	MSC	PTN	R_Net	TOREC
1 WF 図の枚数*)	×	×	○	×	○
2.1 並行記述**)	△	○	○	○	○
2.2 分岐記述**)	○	△	○	○	○

*) ×は複数枚、○は1枚の図で記述できること
**) △は並行/分岐するとWF図の枚数が増える

図 3: 記述性の評価表

WF 図の閲覧性評価

最後に、要求条件 3 で挙げた閲覧性に関する評価を行なう。本稿では、WF 図の閲覧性を “閲覧度 = タスクの数 ÷ WF 図を構成する全要素の個数” とする。閲覧度が大きければ WF 図の閲覧性は良いと言える。但し、ここで言うタスクとは WF の作業手順の単位となる作業のことを意味する。TOREC では、“task”、“send-message”、“receive-message” といったノード 1 つが、1 つのタスクに相当する。

我々は、PTN と TOREC に対して通信販売の WF 図に 13 の手順を逐次追加する記述実験を行なった。記述実験の結果を図 4 に示す。この結果から、PTN に比べて TOREC の閲覧性は高いと言える。

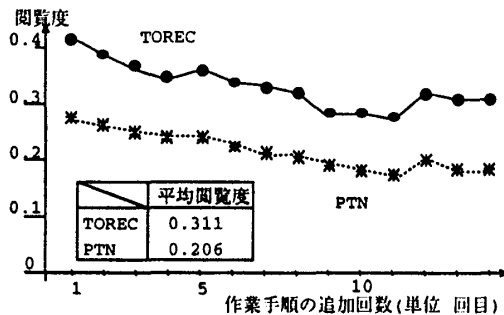


図 4: 閲覧度の比較グラフ

5 おわりに

本稿は、逐次追加法に使用する WF 図 TOREC を提案し、他の図的表記法との比較を行なった。その結果 TOREC が、逐次追加法における手順の追加において有効であることが明らかになった。

今後は、TOREC で記述した WF の正当性の検証について検討していく予定である。

参考文献

- [1] 河合, 土屋, “情報共有の窓が開く - 本格化するグループウェア,” 日経コミュニケーション, No.206, pp.60-82, (1995).
- [2] 飯塚, 桧垣, 平川, “逐次追加による AP 構築手法” 情報処理学会第 50 回全国大会 (6), pp.215-216 (1995).
- [3] ITU:Z.100(1993), “CCITT Specification and Description Language(SDL),” ITU-T Jun. (1994).
- [4] Z120(1993), “Message sequence chart(MSC),” ITU-T, (1994).
- [5] M.Alford, “SREM at Age of Eight!; The Distributed Computing Design System,” IEEE computer, vol.18, no.4, pp.36-46, (1985).