ワークフロー管理システム「Flowmate」（５）
－マルチサーバー

5M－5

斉藤 隆　馬嶋 宏*　堀内 孝*　新田 淳　秋藤 俊介　塔下 哲司**

(株)日立製作所システム開発研究所 *(株)日立製作所ソフトウェア開発本部 **日立西部ソフトウェア(株)

1．はじめに

ワークフロー管理システムにおいてマルチサーバ環境が今後重要になると考える。試験的に導入されたワークフロー管理システムが大きな成果を上げた[1]ために、ワークフロー管理システムが管理する業務の範囲が拡大することが予想されるからである。

マルチサーバ環境を実現する方式は、業務の流れを記述したビジネスプロセス定義（以降、BP定義と呼ぶ）の記述内容およびその管理方式により以下の２つがある。

（１）BP定義転送方式

① 一連の業務の流れを１つのBP定義に記述
② BP定義に業務を管理するサーバを記述
③ BP定義を書類に付加

（２）BP定義連携方式

① 一連の業務の流れを、管理するサーバ単位に分割して、BP定義に記述
② BP定義に他のBP定義と連携するための情報（連携情報と呼ぶ）を記述
③ BP定義を１つのサーバで管理（他サーバに転送しない）

BP定義連携方式には、BP定義転送方式にない以下の長所がある。

- 実世界の分業体制をBP定義の連携モデルとして表現できる
- 部署がその部署内の業務をBP定義として記述して管理することができる

我々は、前記長所により分業指向・現場指向のワークフロー管理システムを実現できると考え、BP定義連携方式によるマルチサーバ環境を実現した。

Workflow Management System "Flowmate"(5)- Multi-Server -
Takashi SAITO[1], Hiroshi MAJIMA[2], Takashi HORIUCHI[2],
Zyun NITTA[1], Shunsuke AKIFUJI[1], Tetsuji TOUGE[3]
1 Hitachi, Ltd. Systems Development Laboratory
2 Hitachi, Ltd. Software Development Center
3 Hitachi Seibu Software, Co., Ltd.

2．BP定義の連携モデル

我々は、「サーバは部署が所有し、部署内の業務の流れを管理するために使う」という考察から、以下の結論を出した。

- １つのBP定義は、部署内の業務の流れを記述する
- BP定義を連携して、複数の部署にまたがる業務の流れを記述する

我々は、複数の業務の流れの連携をモデル化し、以下２つのBP定義の連携モデルを作成した。

（１）直列型連携モデル

ある部署で処理を終えた書類を他の部署が引き継いで処理をするモデル。（図１参照）

手続型プログラミング言語のGOTO文[2]を摸して、連携元BP定義中のシンクノードに連携情報を記述する。連携情報には、以下の情報を記述する。

- 連携先BP定義を管理するサーバ名
- 連携先BP定義名
- 連携するソースノード名

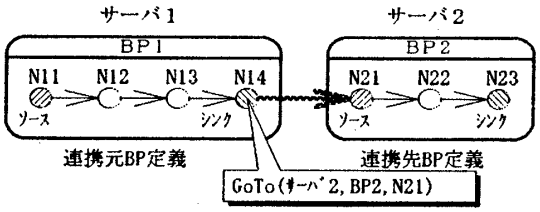

図１ 直列型連携モデル

（２）階層型連携モデル

ある部署で処理途中の書類を、他の部署が一時的に処理した後に、再び元の部署において処理を継続するモデル。（図２参照）

手続型プログラミング言語の関数呼び出し[2]を摸して、連携元BP定義中の連携位置にCALLノードを置き、そこに連携情報を記述する。CALLノードは、BP定義の階層型連携を実現するために新たに設けた制御ノードである。連携情報として記述する内容は直列型連携モデルと同じである。

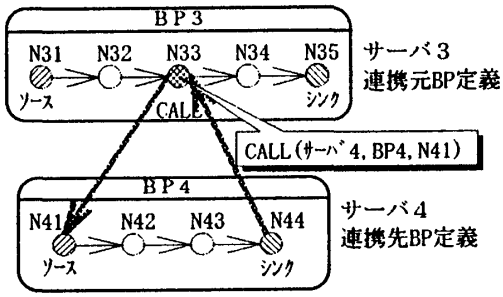

図2 階層型連携モデル

3．マルチサーバ環境の実現

BP定義連携方式に基づいた図3のようなマルチサーバ環境を実現するために、以下の拡張をサーバおよび書類に加えた。（図4参照）

- BP定義の連携処理を行う連携デーモンをサーバ上に新たに用意した
- 連携デーモンが処理する書類を入れるためのデーモントレイを用意した
- 書類に転送情報と戻りスタックというBP定義連携に必要な情報を格納する領域を用意した

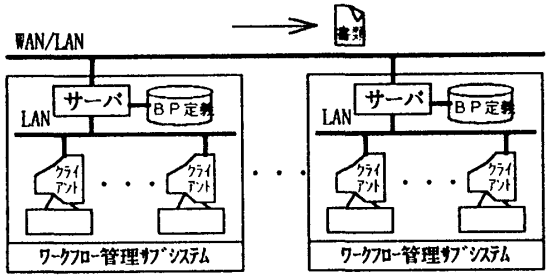

図3 マルチサーバ環境

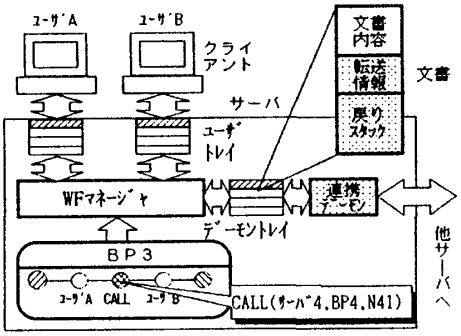

図4 マルチサーバ環境におけるサーバの構成

4．BP定義の連携処理

BP定義の連携処理は、WFマネージャと連携デーモンによって行われる。以下にそれらの処理を述べる。

4．1 WFマネージャの処理

WFマネージャは、シンクノードとCALLノードにおいて、以下に示すBP定義の連携処理を行う。

（1）シンクノードにおける処理

① シンクノードに連携情報がある場合：
連携情報を書類中の転送情報に設定する
シンクノードに連携情報がない場合：
書類中の戻りスタックから値をポップし、その値を書類中の転送情報に設定する

② 書類をデーモントレイに入れる

（2）CALLノードにおける処理

① CALLノードに記述された連携情報を書類中の転送情報に設定する

② 書類中の戻りスタックにカレントのサーバ名、BP定義名、ノード名を設定する

③ 書類をデーモントレイに入れる

4．2 連携デーモンの処理

連携デーモンは、書類中の転送情報を参照して、以下の2つの処理をする。

- デーモントレイ中の書類を他のサーバ上の連携デーモンに転送する
- 他の連携デーモンから転送された書類を、BP定義のソースノードもしくはCALLノードに投入する

5．おわりに

BP定義連携方式に基づくマルチサーバ環境の実現方式について述べた。ワークフロー管理システムの適用範囲は今後ますます拡張されて、複数企業にまたがる業務にまで拡張される可能性がある[3]。これに対応するためには、分業指向と現場指向の特徴をもつBP定義連携方式に基づくマルチサーバ環境が必須であると考える。

参考文献

[1]日経情報ストラテジー：組織の生産性を画期的に高めるグループウェアの衝撃，日経ＢＰ社，pp.70-81，1994年10月

[2]Wirth.N.:Algorithms+Data Structures=Programs, Prentice-Hall, 1976

[3]日経コンピュータ：情報を軸に企業連携へ，日経ＢＰ社，pp.46-67，1994年10月3日