

## オフィス情報処理におけるフォームフローシステム

北海道大学 工学部 田中 譲  
(財)日本情報処理開発協会 滝沢 誠

### 1. 序

近年、生産業務と比較した時のオフィス業務の生産性改善の遅れが問題となっている。オフィス業務の生産性向上と高度化とを計算機と通信システムとを用いた情報システムによる試みはオフィス情報システム(以降 OIS と記す)と呼ばれている [COOK80, BAUMS80, ELLIC80, LADDIS80]。[COOK80]は、情報制御ネット (ICN) モデル [ELLIC80] を用いて、各オフィス業務間の仕事の流れ(制御フロー)と、各仕事の必要とする情報の格納媒体の関連(情報フロー)とを表し、オフィス業務の解析と合理化 (streamlining) とを試みている。[LADDIS80]は、オフィスにおけるフォーム(オフィスでの情報単位、e.g. 書類)の流れに着目して、その流れをメッセージフロー問題として解析し、オフィス生産性を定量化しようとしている。[BAUMS80]は、自動化された仕事の流れと、管理者による制御との関係をマトリネットによる表もろうとしている。

上述した現在までのOISへの試みにおける問題点は、情報システムの核となるデータベースシステム(以降DBSと記す)とオプス情報処理(業務)との関連が明らかでないことである。本論文では、この問題の解決を目指している。OISにおけるDBSの第1の特徴は、従来の統合DBSと比較して、データベース内の全てのデータが、オプス全体で共有される必要がない、即ち全データを1つに統合する必要がない点である。各オプス業務内で用いられるデータは、主にこの業務内でローカルに利用される。このデータ共有の単位を部局と呼ぶ。部局とは、例えば部、課、プロジェクト、グループ等である。データは部局内では、一貫性を保つために統合管理される。しかし、概して部局間では、データの一貫性に対する制約はゆるい。例えば、ある部から他の部に書類が届けられた時、送られた部でのこの書類への更新は前の部に知らされる必要はない時である。この様に、OISでは、部局ごとに必要とするデータを統合し、部局間で共有するデータに対して比較的ゆるい統合をオプス全体で行なえばよいことになる。又、部局は互いにオーバーラップすることも階層関係を持つことも許される。以上のことは、[HAMM80]におけるfederated DBS概念と類似している。

オプスにおいて処理される情報単位をフォーム(form)と呼ぶ。フォームは、オプス従業員から視ると、書式づけられた印刷された

書類であると共に、他従業員又は部局に対する情報転送単位である。フォームとは、情報の表現であるとともに、情報伝達単位である。

OISにおけるDBSを考えるうえで他の重要な点は、データの共有関係である。ある部局から他の部局にフォームが転送された後、このフォームに対する一方の処理が他方に影響を与えるかどうかの問題である。フォームのコピーを取り、これを更新する場合には、フォームの共有は生じない。一方、フォームを共有している場合には、ある部局による更新は、他の部局によっても視られることになる。この共有と非共有の関係を明らかにすることは、オススでのDBSを考えるための基本となる。

以上より、OISにおけるDBSを考える上で、次の3点が重要な概念となる。

- 1) データベースのオスス全体との統合から、必要な部局ごとの統合化 (統合DBSから federated DBSへ)。
- 2) オスス情報処理の単位としてのフォームについて、その情報表現と、情報の転送(フロー)とを考えねばならない。
- 3) フォームには、部局間で共有されるもの(共有フォーム)と共有されないもの(非共有フォーム)とがある。

上記3点に基づいて、本論文ではOISにおけるDBSとしてのフォームフローモデルについて論じる。又、これはこのモデルの思

想と概要を述べる。3.では、このモデルの数学的基礎づけを行なう。4.では、フォームフロ-モデルの例を示す。

## 2. フォームフロ-モデルの思想と概要

オフィス情報システム(OIS)における情報処理単位はフォームである。OISのDBSの位置づけを明らかにするためには、フォームとフロ-と、DBSとの関係づけが必要になる。

OISでのDBSの第1の特徴は、データが各部署内で高い局所性をもつて用いられることである。部署内で必要なデータの一致性は、部署内において保たれる必要があるが、部署間で共有されるデータの一致性保持要求は比較的ゆるいものである。例えば、他の部署に廻った書類の内容変更を瞬時に知る必要はない時が多いし、知る必要性さえない時もある。これ等のことは、DBSの立場から見ると、データベースから導出されたスナップショットが新たなデータベースとして共有されていく場合である。即ち、データは各部署内で一致性を保つために統合される必要があるが、部署間で共有されるデータに対してはそれほど強い統合を求められない。オフィス全体のDBSは従来の中央集権的な統合DBSではなく、分権的な federated DBS [HAMM180]が必要となる。

オフィスで処理される情報単位はフォームである。フォームとい

言葉には、次の2つの意味がある。

- i) 出力フォーム (output form)
- ii) フォームリレーション (form relation)

前者は、日常我々が用いる書式づけるれた書類に対応している。このフォームは、書類としての情報表現であると共に、他の情報転送単位でもある。

後者は、フォームの持つ情報のシステム表現としてのリレーションである。システム内で、フォームの処理と管理とは、このフォームリレーションに対して行われる。一般に、1つのフォームリレーションは、複数の出力フォームを持ち得る。フォームリレーションと出力フォームとのマッピングでは、特に repeating group の処理が問題になる。マッピングとしては、以下の点が問題になる。

- 1) フォームリレーションから出力フォームの定義
- 2) 出力フォームを通してのフォームリレーションのアクセス
- 3) 出力フォームと出力紙の制御
- 4) フォームアクセスの制御 (authorization & authentication)

フォームは、オラス情報処理の単位であるとともに、部署間又は部署内の情報転送単位である。例えば、ある部員から部長への書類の提出がこれにあたる。このフォーム転送をフォームフローと呼ぶ。フォームフローの単位はフォームリレーションである。

フォームのフローは、目的部署での新たな情報の生成をもたらす。

フォームとしては、次の2種を明確に分ける必要がある。

i) 共有フォーム

ii) 非共有フォーム

共有フォームとは、ある書式を通じて、複数部署又は従業員が同一の情報を視れるものである。例えば、会議の出欠案内の表に他人の出欠状況を視れるものは共有フォームである。共有フォームを表すフォームリレーションは、リレーションのセーに対応している。即ち、ある従業員によるフォームの更新は、同じフォームを保持している他の従業員にも伝搬することになる。非共有フォームとは、これを保持している従業員により、ローカルに更新出来るものである。これを表すフォームリレーションは、リレーションから導出されたスナップショットに対応している。このスナップショットを再び共有することも可能である。

## 3. フォームフローモデル

前章に述べた思想に基づき、OISのためのデータモデルを提案する。本論文では、フォームの情報単位としての性質を論じ、出力形態としてのフォームに関しては別の機会に論じる。したがって、フォームは、リレーションとしてモデル化し、これをフォームリレーションと呼ぶ。

以下に述べるフォームフローモデルは、OISにおける次の3つの概念を取り込んでいる。

- i) federated Data Base System,
- ii) フォームリレーションの共有と非共有,
- iii) フォームリレーションの流れ。

ここで言うフォームの流れとは、次の2種類のことを意味する。

- i) 他のフォームリレーションから、新たなリレーションをビューとして定義(共有フォームリレーションの定義)
- ii) 他のフォームリレーションから、新たなリレーションをsnapshotとして導出(非共有フォームリレーションの導出)。

i) では、フォームの転送後、このフォームを作成するのに用いたこのフォームリレーションを変更すると、転送先のフォームの内容も変更され、逆に、転送先でこのフォームを変更すると、転送元のフォームリレーションも変更される。これに対し ii) では、転送元のフォームリレーションと、転送されるフォームとは、全く独立である。i) で転送されるフォームを仮想フォームと呼び、ii) のそれを実フォームと呼ぶ。仮想フォームリレーションに対する更新問題は、オブジェクトモデルに基づいたデータ抽象化技法 [TANAY 80] により解決することができる。

この2種のフォームリレーションは、任意のフォームリレーション上に階層的に定義され得る。

フォームフローモデルは、フォームリレーションを表わすノードとフォームリレーションの定義と導出の関係を表わすアークとからなる高階グラフにより表わされる。ノードを情報単位と呼び、アークをフォームフローと呼ぶ。この高階有向グラフをフォームスキーマ図と呼ぶ。

我々は、オフィスにおける事務処理を

1. 処理や管理の単位となる情報単位の集合  $I$ .
2. 情報の流れの集合  $F$ .



3. 部局の集合  $D$
4. 情報処理を行う行動の集合  $A$ .
5. 情報処理行動を情報の流れに対応づける写像  $\alpha$ .
6. 各部局で見るべきべき情報と、許される情報処理を規定する写像  $\beta$

の6つの基本概念から構成されているものとして、6項組  $(I, F, D, A, \alpha, \beta)$  をこれをモデル化する。

$I$  は情報単位と呼ばれる要素の有限集合である。 $I$  の各要素  $n$  には、その表現  $\mathcal{J}(n)$  を考えることができる。 $\mathcal{J}$  としては、 $I$  を関係の集合  $R$  の中へ写す写像

$$\mathcal{J}: I \rightarrow R$$

を考える。 $\mathcal{J}(n)$  は  $n$  に対するフューリレーションである。 $I$  は互いに排反な  $I_e, I_a, I_v$  に分割される。

$$I = I_e \oplus I_a \oplus I_v$$

$I_e$  は、オフィス外の情報単位の集合を表わす。外部情報単位は内部の情報単位を導出したリ、逆に内部の情報単位より、外部情報単位が出力として導出される。このように、 $I_e$  は、オフィスの入出力情報を表わす。スキーマ図では、外部

情報単位を三方形で囲んで表現する。

$I_a$  は実情報単位と呼ばれる要素の集合である。実情報単位  $n$  に対する  $\mathcal{J}(n)$  は実フォーミュレーションであり、実データとして、DBMS によって物理的に一つの関係として蓄積されている。実情報単位は、正方形で囲んで表現する。

これに対して  $I_v$  は仮想情報単位と呼ばれる要素の集合である。仮想情報単位  $n$  に対するフォーミュレーション  $\mathcal{J}(n)$  は、システム中に物理的に一つの関係として蓄積されているのではなく、他の情報単位  $n_1, n_2, \dots, n_k$  に対応する関係  $\mathcal{J}(n_1), \mathcal{J}(n_2), \dots, \mathcal{J}(n_k)$  より定義されるビューの一つである。スキーマ図において、仮想情報単位は菱形で囲んで表現する。

$F$  はフローの集合と呼ばれ、 $(I, F)$  は  $I$  をノードの集合、 $F$  を

$$F \subset 2^I \times I$$

なる高階の有向辺の集合とする高階有向グラフと定義する。

$F$  は情報のフローの集合  $F_I$  とフレーム（書類の書式の持組）のフローの集合  $F_F$  とに分割される。

$$F = F_I \oplus F_F$$

情報フローは、実情報単位を、いくつかのソースの情報単位から導出するものである。生成された実フォーマリレーションとソースのフォーマリレーションは互いに独立で、情報の共有一括管理は行なわれない。情報フローは、書類の物理的な転送、コピー、ディスプレイへの情報の表示、電子郵便等に対するものである。上述のことから、 $F_I$  を

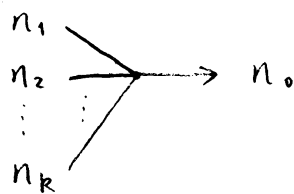
$$F_I \subset 2^I \times (I_a \cup I_e)$$

と定義する。 $F_I$  に対して、

$$(\langle n_1, n_2, \dots, n_k \rangle, n_0) \in F_I$$

は、ソースフォーマリレーション  $\mathcal{J}(n_1), \mathcal{J}(n_2), \dots, \mathcal{J}(n_k)$  より関係演算を用いて得られるある関係が、実情報単位  $n_0$  に送り、 $\mathcal{J}(n_0)$  の更新が行なわれることを意味する。

情報フローは実線の矢 ( $\rightarrow$ ) によって表わし、高階の有向辺  $(\langle n_1, n_2, \dots, n_k \rangle, n_0)$  は、



と表わす。

フレームフローは、仮想情報単位を、いくつかのソースの

情報単位に対するフォーミュレーション上に定義されるビュー  
 リレーションとして定義する。定義された仮想フォーミュ  
 レーションと、ソースのフォーミュレーションとの間には情  
 報の共有がなされ、一方における更新は、他方に影響を及ぼ  
 ず。F<sub>F</sub> は、

$$F_F \subset 2^{I-I_e} \times I_V$$

と定義され、

$$(\{n_0, n_1, \dots, n_k\}, n_0) \in F_F$$

は、 $n_0 \in I_V$  に対する  $\mathcal{J}(n_0)$  が、 $\mathcal{J}(n_1), \dots, \mathcal{J}(n_k)$  のビュー  
 として定義されることを意味する。フレームフローは、スキ  
 ーム図では、破線( $\cdots \rightarrow$ )で表わされる。

$\mathcal{D}$  は、部局の集合である。どの部局で、どの情報単位が見  
 え、どの情報処理行動が実行を許されるかは、後述の  $\beta$  が規  
 定する。

A は情報処理行動の集合である。

$\alpha$  は、各情報処理行動  $a \in A$  に、情報フローの有限系列  
 $f_1, f_2, \dots, f_n$  を対応させる。系列中の情報フロー

$$f_i = (\{n_1^i, n_2^i, \dots, n_{j_i}^i\}, n_0^i)$$

に対し,  $a$  は  $f_i$  を  $j_i+1$  個の関係変数を含む関係演算式  $e$  に写す.

$$\alpha: A \rightarrow F_I^*$$

$F_I^*$ :  $F_I$  の要素の有限系列の集合

$$\forall a \in A$$

$$a: F_I \rightarrow E$$

$E$ : 有限個の関係変数を含む関係演算式の集合

$a \in A$  に対し,

$$\alpha(a) = (f_1, f_2, \dots, f_n)$$

とすると,  $a$  の情報処理行動  $a$  は,

$$\mathcal{L}(n^i_0) \leftarrow a(f_i) (\mathcal{L}(n^i_1), \mathcal{L}(n^i_2), \dots, \mathcal{L}(n^i_{j_i}))$$

なる更新を  $i = 1, 2, \dots, n$  についてこの順に実行するという意味を持つ.

$F_F$  に対しては,

$$\delta: F_F \rightarrow E$$

が定義されているものとする.  $f = (n_1, n_2, \dots, n_k, n_0)$

$\in F_F$  に対して,  $\delta(f)$  は  $k$  個の関係変数を含む関係演算式で,  
 $\nu_2 - \mathcal{Q}(n_0)$  は

$$\mathcal{Q}(n_0) = \delta(f)(\mathcal{Q}(n_1), \dots, \mathcal{Q}(n_k))$$

と定義される

$\beta$  は権限を表わす写像で,

$$\beta: \mathcal{D} \rightarrow 2^{I-I_e} \times 2^A$$

と定義される.  $d \in \mathcal{D}$ ,  $I_d \subset I - I_e$ ,  $A_d \subset A$  とし,  $\beta(d) = (I_d, A_d)$  とすると,  $I_d$  は部局  $d$  で見ることにできる情報単位の集合を表わし,  $A_d$  は部局  $d$  で許される情報処理行動の集合を表わす.

実情報単位には, これを蓄えているファイルの種類を指定するファイル属性が付随する. ファイル属性には,

- (1) データベース関係
- (2) ワークファイル
- (3) プリントファイル
- (4) ディスクレイファイル
- (5) 電子郵便

等があり, (3), (4), (5) の属性を持つファイルへの更新は,

出力動作を意味するものとする。(3), (4), (5)の属性ファイルには、出力書式属性が与えられ、これにより、フォームの出力を行なう。実情報単位で出力ファイル(3), (4), (5)の属性を持つものは、フォームフロースキーマ図において、2重の正方形で囲んて表現する。

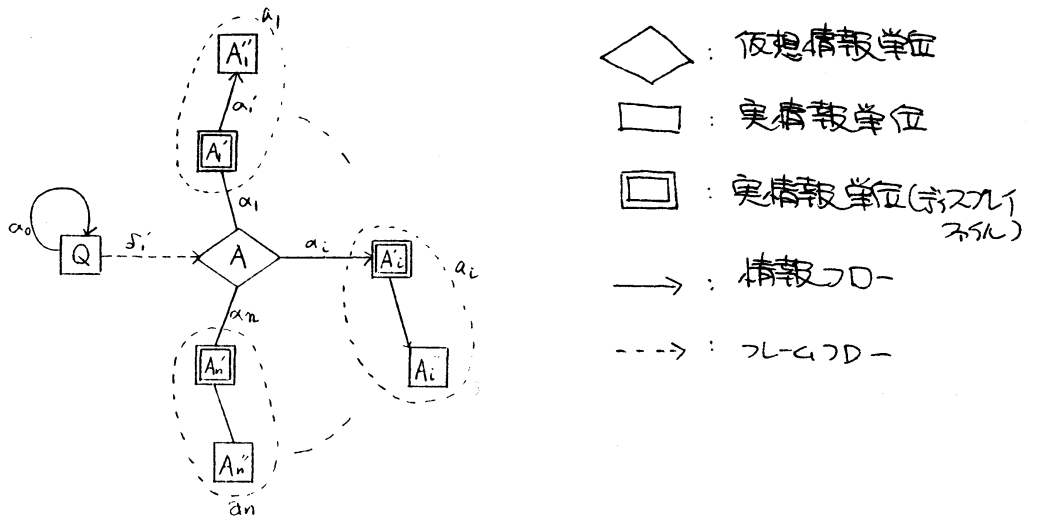


図4.1 (3)のファイルフロースキーマ図(a)

#### 4. フォームワーカーモデルの例

本章では、フォームワーカーモデルの例について述べる。

##### 例1 会議開催案内とその出欠の回覧

部内会議案内が、部の総務( $a$ )から、全部員( $a_1, \dots, a_n$ )に出され、各部員はその出欠を記入して総務に返す例を考えてみる。又、各部員は会議開催情報(eg. 場内、日時、目的)のデータベースを取るものとする。これを図4.1に図示する。 $a$ は会議情報を実情報単位 $Q$ に生成し、これから会議案内フォーム( $A$ )をつくり、全員に送る。各部員( $a_i$ )は、 $A$ を通して会議案内内容を知るとともに、その時点での出欠状況を確認する。各員は、 $A$ を通して自分の出欠を $Q$ に書き込む。この書き込みは、データベース $A$ を通しての更新となる。各員は、自分のデータベース( $A_i$ )に、 $A$ を通して現在の $Q$ のスクリーンショットを導出し、この中から必要な部分を自分のローカルファイル( $A'_i$ )に出力する。

実情報単位  $Q$  : 会議データベース  
 $A'_i$  :  $a_i$  のデータベース  $A_i$  :  $a_i$  のパーティクル

仮想情報単位  $A$  : 会議開催案内のフォーム

情報フロー  $\alpha_0$  :  $a$  における会議開催の起案  
 $\alpha_i$  :  $a_i$  における、自分のデータベース( $A_i$ )への会議情報の出力  
 $\alpha'_i$  :  $a_i$  が必要とする会議情報をパーティクル( $A_i$ )に格納

フォームワーカー  $\delta_i$  : 会議開催案内のフォームをつくる。

図4.1 (b) 会議開催案内



例) 書類の作製

次の例は書類の作製である[図4.2]。ある部員(a<sub>1</sub>)によ、て起草され、これが課長(a<sub>2</sub>)、部長(a<sub>3</sub>)によ、て修正認可される。各時点で上司に対して書類を提出する時は、そのコピーを各自で保管する。部長は、各課から出されてきた書類(F<sub>1</sub>'<sub>2</sub>)から只のやり取り書類をつくり全員に見せる。各員は、上司による修正を、各時点で知ることが出来る。

a<sub>1</sub>から提出された書類F<sub>1</sub>'<sub>1</sub>に対するa<sub>2</sub>の更新(修正)は、仮想メモリレーションF<sub>1</sub>'<sub>2</sub>を通して、a<sub>1</sub>も視ることが出来る。a<sub>2</sub>からa<sub>3</sub>に提出された書類F<sub>1</sub>'<sub>3</sub>と、a<sub>3</sub>がまとめたF<sub>1</sub>に対する更新は、実メモリレーションF<sub>1</sub>の更新に及び、a<sub>1</sub>とa<sub>2</sub>とはこの更新を視ることが出来る。

- 実情報単位 F<sub>1</sub>: 書類 リレーション  
 A<sub>1</sub>' A<sub>2</sub>', B<sub>1</sub>', B<sub>1</sub>'', C<sub>1</sub>'': デスタブルなシル  
 A<sub>1</sub>'', A<sub>2</sub>'', F<sub>1</sub>' : パーソナルなシル
- 仮想情報単位 F<sub>1</sub>'<sub>1</sub>: a<sub>1</sub>の起草した書類のルール  
 F<sub>1</sub>'<sub>2</sub>: a<sub>2</sub>からa<sub>3</sub>に提出した書類のルール  
 F<sub>1</sub>'<sub>3</sub>: a<sub>3</sub>からa<sub>3</sub> " "  
 F<sub>1</sub>: a<sub>3</sub>が F<sub>1</sub>'<sub>2</sub>, F<sub>1</sub>'<sub>3</sub> からまとめた書類のルール
- 情報フロー α<sub>0</sub>: a<sub>1</sub>による書類の起草  
 α<sub>1</sub>: F<sub>1</sub>'<sub>1</sub>のデスタブル (a<sub>1</sub>に表示)      α<sub>2</sub>: F<sub>1</sub>'<sub>2</sub>のデスタブル (a<sub>2</sub>に表示)  
 α<sub>1</sub>' : F<sub>1</sub>'<sub>1</sub>のコピー      α<sub>2</sub>' : F<sub>1</sub>'<sub>2</sub>のコピー  
 α<sub>3</sub>: F<sub>1</sub>'<sub>2</sub>のデスタブル (a<sub>3</sub>に表示)      α<sub>3</sub>: F<sub>1</sub>のコピー (a<sub>3</sub>にパーソナルなシル)  
 α<sub>3</sub>' : F<sub>1</sub>'<sub>2</sub>のコピー      α<sub>3</sub>' : F<sub>1</sub>'<sub>3</sub>のデスタブル (a<sub>3</sub>に表示)
- ルールフロー δ<sub>1</sub>: 書類の起草ルールの定義      δ : F<sub>1</sub>'<sub>2</sub>とF<sub>1</sub>'<sub>3</sub> による F<sub>1</sub>の定義  
 δ<sub>1</sub>' : F<sub>1</sub>'<sub>1</sub>とF<sub>1</sub>'<sub>2</sub>の定義      図4.2 (a) 書類作製 (例2)

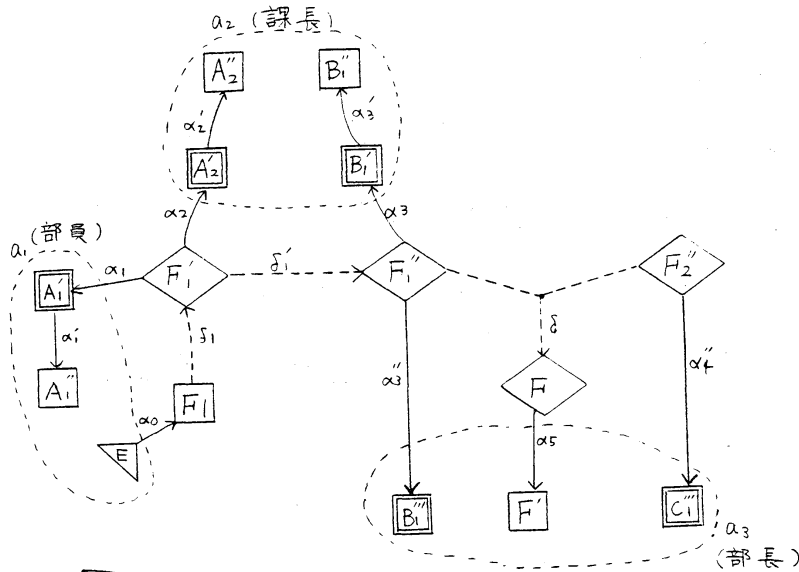


図4.2 例2のスキーマ図式 (b)

### 5. 今後の課題

本論文では、オフィス情報システム(OIS), CAE/CADを中心とした今後の計算機アプリケーションにおけるデータベースシステム利用のモデルとして、フレームワークモデルの提案を行った。このモデルは、federated DBS, フレームワークの共有、フレームワークの柔軟なつなぎを基本概念としている。これにより、OIS等のDBSとの関係が不明確であったアプリケーションにおけるDBSの位置づけを明らかにできたと考えている。更に、OISの設計用ツールとしても利用できる。

今後の課題としては、以下の点がある。

- 1) OISにおけるコンカレントオペレーションの記述。全てのオペレーションを許すのではなく、各部署に対して許されるオペ

レゾリューション集合の定義が必要になる。

- 2) SBA [ZLOOM80]におけるトリガ機能のモデル化。
- 3) 仮想フォーマリレーションに対する更新に対して、オブジェクトモデルとデータベース化技法[TANAY80]の適用。
- 4) 分散型データベースシステムとの関連の明確化。
- 5) CADへの適用

### 参照文献

- [BAUM80] Bauman, L.S. and Coop, R.D., "Automated Workflow Control: A key to Office Productivity," AFIPS Conf. Proc., 1980, pp. 549-554.
- [COOK80] Cook, C.L., "Streamlining Office Procedures - An Analysis Using the Information Control Net," ibid, pp. 555-565.
- [LADD80] Ladd, I. and Tschritzer, D., "An Office Form Flow Model," ibid, pp. 533-537.
- [ELLIC80] Ellic, C.A. and Nutt, G.J., "Office Information Systems and Computer Science," ACM Computing Surveys, Vol. 12, No. 1, Nov. 1980, pp. 27-60.
- [HAMM80] Hammer, M. and McLeod, D., "On Database Management System Architecture," Infotech State of the Art, 1980, pp. 177-201.
- [TANAY80] Tanaka, Y. and Takizawa, M., "Object Model and Data Abstraction," in preparation.
- [ZLOOM80] Zloof, M., "A Language for Office and Business Automation," Proc. of the CA Conf., Atlanta, Georgia, Me. 1980, pp. 249-260.