

## 分散協調環境における部分データベースの複製方法

宮西 洋太郎 高品 智一 渡辺 尚 水野 忠則

静岡大学

複数作業者の協調作業を支援するコンピュータ機能として、近年、グループウェアやCSCW(Computer Supported Cooperative Work)が研究されてきた。質の高い協調作業を行うには共通情報の共有、各作業者の個別作業及び作業結果の交換・調整が必要である。本研究では、情報の共有方法として、各サイトに部分データベースの複製を配置する方法を提案し、サイトを切り離して個別作業を行う場合の複製データ維持負担とアクセス可否の評価、及び個別作業後の結果調整として、データベース一貫性回復についての検討を行う。協調作業の場合、一貫性はデータ内容に関わる判断が必要であり、単純な案を提案しているが、今後更に検討を要する。

## A Method of Partial Database Allocation in a Distributed Cooperation Support System

Yohtaro Miyanishi, Tomokazu Takashina, Takashi Watanabe, and Tadanori Mizuno

Shizuoka University

3-5-1 Johoku, Hamamatsu, 432, Japan

Many studies have been conducted for computer system to support human cooperative work, i.e. CSCW(Computer Supported Cooperative Work) or Groupware. In cooperative work, it is essential to share the common information and to coordinate some conflictions among workers. In our previous work, we had proposed a method of fragmentation and allocation of database addressing degraded operation when a site is isolated from a network. We try to adapt this method to CSCW. In this paper we propose a method of partial database allocation to share common information and a simple method of coordination.

## 1. はじめに

近年、人間の協調作業をコンピュータで支援するという観点で、CSCWやグループウェアについて多くの研究がなされている。

質の高い知的生産の協調作業を行う場合に、

- ①目標等、上位情報の共有、
  - ②各作業による検討作業(個別作業の実施)、
  - ③相互の検討作業結果(意見)の交換、
  - ④相互の検討作業結果(意見)の調整
- 等の要件が必要である。

本稿では、分散コンピュータシステム(以下分散システムと略する)が協調作業を支援する場合を想定し、

- ①情報の共有をデータベース(以下DBと略する)の共有と考え、各サイトへのDBの分割(fragmentation)、複製(replication)及び配置(allocation)方法と複製維持負担の評価方法を提案する、
- ②サイトが故障あるいは意図的に全体から分離した場合のDBアクセス可否を評価する、及び
- ③DBの一貫性(consistency)について検討する、特にあるサイトが全体から分離して個別に動作した後のDBの一貫性の回復についての検討を行う。

協調作業支援における、DBの一貫性回復は、作業者の意見調整の意味があり、通常のトランザクション処理におけるread/writeレベルのそれでは不十分であり、DBの内容に関わる調整方法が必要である。本稿ではデータ融合と称し、いくつかの簡単な実現方法案を提案する。

## 2. 問題の背景

分散システムにおける分散DBの研究が行われている。これらの研究においては、分割配置の方法論<sup>1)</sup>、ネットワーク分断時の複製データの一貫性の検討や関係を利用した近似回答の検討<sup>2)</sup>等がなされている。分散DBの今後の課題として、分割と配置を統合して扱う問題、分散検索の最適化の問題、複製維持方法の問題、故障時の評価、従来の直列可能性での正当性から意味論的な正当性の問題、分散オブジェクト指向DB、分散知識ベース等の課題が挙げられている。<sup>3)</sup>

筆者等は先に、分散システムにおける、データの地域的特性、時間的特性に応じたDBの分割配置方法について提案した。<sup>4)</sup>

この方法は、

・DB分割方式を

- ①分割無し、
- ②重複のある分割、
- ③重複のない分割

・サイトへの配置方式を

- ①全てのサイトに配置、
- ②ランダムなサイトに配置、
- ③特定のサイトに配置、
- ④判断によるサイトに配置

に分類し全体をマトリックス的に概観し、データ特性に応じてそれらの中から分割配置方式を選択する方法であり(表1)、上記方式の選択アルゴリズムの一部を提案した。

ここで「判断によるサイトに配置」とはデータ特性を利用してある判断アルゴリズムによって複製を配置するサイトを決定することを意味する。

表1 DBの分割配置方式

分割 配置	DB分割せず	重複のある 分割	重複の無い 分割
全サイトに 配置	ケース11	ケース12	ケース13
ランダムに 配置	ケース21	ケース22	ケース23
特定サイト に配置	ケース31	ケース32	ケース33
判断による 配置	ケース41	ケース42	ケース43

一方、人間の協同作業については、協調の次元階層モデルとして把握することが提案されている。<sup>5)</sup> このモデルでの最上位層である協同(collaboration)層は状態として、

- ①協力(cooperation)、②衝突(conflict)があり、状態遷移を起動するものとして、
- ③調整(coordination)、④競争(competition)があるとしている。

競争と調整により単独作業に比べて高品質の成果が得られる可能性がある、他人からのコメントが新たなアイデアのヒントになるというのは我々の日常経験するところでもある。協調作業支援システムにおいても、これらの側面を考慮すると①作業は各作業者の分担に概略分割されるが、ある作業は複数の作業者に割り当て、各作業者が各自の案を作成する。(競争)

- ②各作業作成の案は整合しているか又は矛盾している、矛盾の場合の対策を考える。(衝突)
  - ③この矛盾を各作業者に説明することにより、各作業者に解決のヒントを提供する。(触発)
  - ④矛盾解決の案を提供する。(調整)
- といった側面を考慮したシステムを構成することが望ましい。

### 3. 問題の定義

本稿で検討するモデルとして、

- ①作業内容:複数人数(J人)による知的生産作業、具体的にはソフトウェア仕様書の協同作業を想定する。
- ②支援システム:1台のサーバマシンとJ台のサイトすなわちワークステーション(以下WSと略する)からなる分散システムとする。各作業者に各WSを1対1で割り付ける(図1)。
- ③運用形態:オフィスにおいてはネットワークに接続されて運用する、この運用状態を協調運転と称する。在宅作業等においてはWSを分離して運用する、この運用状態を独立運転と称する。
- ④DB:本稿でのDBは協同作業対象のソフトウェア仕様書とする。この仕様書は階層的に構成されるが、ここでは簡略のため、変更の少ない層及び作成、変更のある層の2階層で構成されるものとする。前者は一般に作業者が共有する上位の情報であり、レイヤ1(上位仕様)と称することとする、後者は一般に作業者が主に作業対象とする情報でレイヤ2(作業仕様)と称することとする。それらの粒度(granularity)については検討を要するが、ここでは各々がデータアイテム(アイテムと略する)から成っているものとする(図2)。
- ⑤分担:概略の作業分担は決まっているが、いくつかのアイテムについては、複数作業者が並行して作業するものとする。(個別の分担と共通の分担)

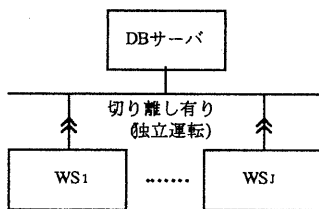


図1 システム構成

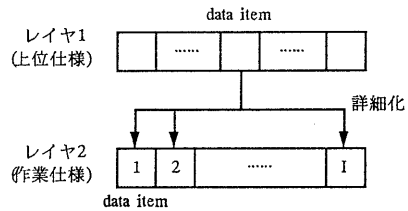


図2 データの構成

### 4. DBの分割配置

DBのread性能と可用性の向上のために、サイトに複製を配置するが、DBの規模が小さい場合には分割せずに、かつ全てのサイトに複製を配置するという方法(表1のケース11)が考えられるが、サイトすなわち複製の数が多くなるに従って、複製の維持に要する負担(格納の負担、read-one, write-all等による一貫性の維持負担)が増大してくる。従って適切な判断基準でDBを分割、配置することが必要となってくる。

前述のDB分割方式において、レイヤ1は各サイトで使用するので分割対象とはしないこととする(表1のケース11)。レイヤ1は上位情報であり、変更も少なくデータ量も少ないので複製維持負担は小さい。レイヤ2を分割対象とし、アイテム1, 2, ..., I (Iはアイテム数)からなるものとする。

#### 4.1 データ特性の表現

筆者等は先に、データの統計的特性をデータ特性に使用する一つの方法として、

- ・アイテムiのアクセス発生率 $\lambda(i)$ を時間特性として利用する、
- ・そのアクセスがサイトjでなされる条件付確率 $P(j|i)$ を地域特性として利用する、

といったDBの分割配置方法を提案した。この考え方を協調作業支援システムに適用する。

単位時間内[Sec]において発生する作業すなわちデータアクセス(readまたはwrite)において、それがデータアイテムiであり、ノードjにおいてなされるという発生率 $\lambda(i, j)$  [1/Sec]をここでは一致発生率と称することとする。これは、

$$\lambda(i, j) = P(j|i) \cdot \lambda(i) \quad (1)$$

であり、一致発生率は、データアイテムiの発生率 $\lambda(i)$  [1/Sec]とそのアクセスがサイトjすなわち

作業者jで行われるという条件付き確率 $P(j|i)$ [1]との積で表される。ただし $[j]$ はディメンションを表す。

$P(j|i)$ は作業者jとアイテムiとのアクセス上の関連の度合を表すものであるが、簡単のため作業者とアイテムの分担(責任または権限)の度合をも表すものとする。

分担が明確に一人の作業者jに定まっているアイテムiについては $P(j|i)=1$ である。

これらの $\lambda(i)$ 、 $P(j|i)$ は目的の作業が進捗するに従って変化するものであるから、適切な期間(例えば週一回)で再設定することが必要であろう。レイヤの再構成も同様である。

## 4.2 DB分割配置方式の選択

表1においてどのケースを選択するかという問題がある。先にケース42の選択方法として下記の2項目を数値的に評価する方法を提案した。<sup>4)</sup>

- ・対象データに地域的特性が大きい、かつ
- ・アクセスが複数ノードに分散している。

本稿の協調作業の場合に、同様の考えを適用すると、上記の条件は各々、

- ・アイテムの作成、変更担当者がほぼ決っている。(誰もかどれでも、ということではない)
- ・一つのアイテムに複数の作業者が対応する。(競争を行う)

という条件となり、数値的評価を行わなくても、ケース42が適していると判断した。また分割の単位としては最小単位すなわちアイテム単位とした。

## 4.3 複製データの配置サイトの決定

### (a) 複製データを持たせるか否かの判断。

アクセス発生率 $\lambda(i)$ が閾値を越えるアイテムiについては複製をもつこととする。

### (b) 複製を配置するサイトの決定。

各iについて、 $P(j|i)$ の大きい順にjを並べ直す。並べ直したjを $j_{1i}, j_{2i}, \dots$ とする。従って大きい順に並べ直した $P(j|i)$ は、

$$P(j_{1i}), P(j_{2i}), P(j_{3i}), \dots \quad (2)$$

となる。上位のものから、一定の閾値 $P_0$ を越えるまで次式でrを増加して加算する。

$$\lambda(i) \cdot \sum_{r=1}^{R_i} P(j_{ri}) \geq \lambda_m \cdot P_0 \quad (3)$$

$$\lambda_m = (1/I) \cdot \sum_{i=1}^I \lambda(i) \quad (4)$$

ここで、 $\lambda_m$ は全アイテムの平均アクセス(作業)発生率である。上記の式が成立する最初のrを $R_i$ とし、このときのサイト $j_{1i}, j_{2i}, \dots, j_{R_i}$ にアイテムiの複製データを配置する。すなわちアイテムiがよくアクセスされるサイトの順に $R_i$ 個の複製を配置する。これをアイテム毎に複製配置サイトを整理した表を表2に示す。

また、そのとき「複製の重み」 $w(j_{ri})$ を、 $P(j_{ri})$ に比例させて定義する。

$$w(j_{ri}) = P(j_{ri}) / \sum_{r=1}^{R_i} P(j_{ri}) \quad (5)$$

(r = 1, 2, 3, \dots, R\_i)

次にサイト毎に複製保持アイテムを整理した表を表3に示す。表3の $i_{sj}$ はサイトj(j=1,2,3,\dots,J)について表2の複製配置サイト $j_{ri}(r=1,2,3,\dots,R_i)$ を調べサイトjに一致するアイテムiを並べたものである。

表2 アイテム毎の複製配置サイト

アイテム	複製配置サイト	サイト数
1	$j_{11}, j_{21}, \dots, j_{R11}$	$R_1$
2	$j_{12}, j_{22}, \dots, j_{R22}$	$R_2$
⋮	⋮	⋮
i	$j_{1i}, j_{2i}, \dots, j_{ri}, \dots, j_{R_i i}$	$R_i$
⋮	⋮	⋮
I	$j_{1I}, j_{2I}, \dots, j_{R_I I}$	$R_I$

表3 サイト毎の複製保持アイテム

サイト	複製保持アイテム	アイテム数
1	$i_{11}, i_{21}, \dots, i_{S11}$	$S_1$
2	$i_{12}, i_{22}, \dots, i_{S22}$	$S_2$
⋮	⋮	⋮
j	$i_{1j}, i_{2j}, \dots, i_{sj}, \dots, i_{S_j j}$	$S_j$
⋮	⋮	⋮
J	$i_{1J}, i_{2J}, \dots, i_{S_J J}$	$S_J$

## 5. 複製の維持負担(コスト)の評価

複製はデータの可用性及び応答性を高めるために行われるが、そのためには常時、複製の一貫性を維持するために処理資源を消費している。このように協調運転状態における、複製を維持するための負担(コスト)を記憶容量及び処理負荷の面で評価する。

### 5.1 記憶容量の負担

各々のサイト  $j$  に複製される複製データ容量を求める、式(1)において  $j_{ii}$  が  $j$  に一致するアイテムのサイズを集計する。

サイト  $j$  における複製データ量  $c_j$  は、アイテム  $i$  のデータ長を  $l_i$  として、

$$c_j = \sum_{i=1}^I \sum_{r=1}^{R_i} \delta_{jj_{ri}} \cdot l_i \quad (6)$$

ただし、 $\delta_{jj_{ri}}$  は  $j = j_{ri}$  ならば 1、そうでなければ 0 の値をとるものとする。

$c_j$  に単位長あたりの負担(コスト)を乗じることによりサイト  $j$  における記憶容量負担を求めることができる。

### 5.2 処理能力の負担

複製の維持は read-one write-all によるものとする。この場合には単純なアクセス発生率ではなく、read 発生率  $\lambda_r$ 、write 発生率  $\lambda_w$  とに分けて考える必要がある。協調運転時における、サーバ以外のサイト全体及び一つのサイトでの評価を行う。簡単のため、各サイト、サーバでの各アイテムの read、write については同一の負荷と想定する。

1回のアクセス(readまたはwrite)につき、CPU時間  $t_c$ 、ディスク時間  $t_d$ 、ネットワーク時間  $t_n$  (厳密では無いがネットワークのためのCPU時間も含む)とする。

アイテム  $i$  のアクセスが到来した時、いずれかのサイトに複製が存在する確率  $Pe_i$ 、及び存在しない(ネットワークを通してアクセスが必要)確率  $Pai$  は、

$$Pe_i = \sum_{r=1}^{R_i} P(j_{ri}) \quad (7)$$

$$Pai = 1 - \sum_{r=1}^{R_i} P(j_{ri}) = 1 - Pe_i \quad (8)$$

である。これらと表2、表3を用いて負荷(duty)の

評価を行う、負荷から適切な演算により負担(コスト)を求めることができるもの考える。

#### (a) read操作の全体負荷

readの場合、複製の維持という観点からは負荷は生じない、通常のアクセスによる負荷のみを評価する。協調運転状態であるので、アクセス対象のアイテムはサーバを含めれば、いずれかのサイトに複製(実体)が存在している。read操作による負荷を、発生率とアイテムにわたってシステム全体で評価する。readのCPU負荷  $\rho_{cr}$ 、ディスク負荷  $\rho_{dr}$ 、ネットワーク負荷  $\rho_{nr}$  は、

$$\rho_{cr} = \sum_{i=1}^I \lambda_r(i) \cdot t_c \quad (9)$$

$$\rho_{dr} = \sum_{i=1}^I \lambda_r(i) \cdot t_d \quad (10)$$

$$\rho_{nr} = \sum_{i=1}^I \lambda_r(i) \cdot Pai \cdot t_n \quad (11)$$

となる。上記の式では  $\rho_{cr}$  に  $Pe_i$  が影響していないが、 $t_n$  の中にはアクセスされたサイトでのCPU時間が含まれているので、厳密にはCPU負荷に  $Pe_i$  の影響がある。また応答時間の評価では  $Pe_i$  が直接の影響を与える。

#### (b) write操作の全体負荷

writeにおいては、あるアイテムを1回アクセス(write)する通常の負荷に加えて、複製を維持するための負荷が生じてくる。それらを含めて全体の負荷を評価する。write操作による負荷を、発生率とアイテムにわたってシステム全体で評価する。

writeのCPU負荷を  $\rho_{cw}$ 、ディスク負荷を  $\rho_{dw}$ 、ネットワーク負荷を  $\rho_{nw}$  と表す。1回のアクセスでサーバへのwrite操作及び複製の数のwrite操作(write-all)を行うので、

$$\rho_{cw} = \sum_{i=1}^I (R_i + 1) \cdot \lambda_w(i) \cdot t_c \quad (12)$$

$$\rho_{dw} = \sum_{i=1}^I (R_i + 1) \cdot \lambda_w(i) \cdot t_d \quad (13)$$

$$\rho_{nw} = \sum_{i=1}^I (R_i + 1 - Pe_i) \cdot \lambda_w(i) \cdot t_n \quad (14)$$

となる。

#### (c) サイトでの負荷

上記の結果は全体での負荷評価であり、次にサイト毎の評価を行う。サイトから見れば複製を多

くのアイテムについて保持すれば、自己の複製からreadできるアイテムが多くなり平均としてreadアクセス応答時間が短縮できるが、write-allによるwriteが多くなり負荷が高まる。あるサイトjについて、アイテム $i_{sj}$ の複製を保持することによるwrite-allによる負荷は、表3とアクセス発生率を用いて、

$$\rho_{cwj} = \sum_{s=1}^{S_i} \lambda w(i_{sj}) \cdot tc \quad (15)$$

$$\rho_{dwj} = \sum_{s=1}^{S_i} \lambda w(i_{sj}) \cdot td \quad (16)$$

によって概略評価できる。

## 6. 独立運転サイトのサービスの低下

先に提案した方法により、特定サイトが独立運転時に、発生したアクセスが当該サイトに複製データを持つアイテムならばサービスを受けることができ、持たないならばサービスを受けることができない、という状況を評価することができる。

サイトjにアイテムiの複製を持っている確率 $q_{ij}$ は、複製を持って入れば1、持っていないならば0である、次式で表される。

$$q_{ij} = \sum_{r=1}^{R_i} \delta_{jjr} \quad (17)$$

独立運転時に特定サイトjにおいてアイテムiがサービスを受けられる率 $\lambda'(i,j)$ は、複製が存在する場合には $\lambda$ がそのまま維持され、複製が存在しない場合には $\lambda$ は0となりアクセスが発生してもサービスは行われない。従って、

$$\lambda'(i,j) = \sum_{r=1}^{R_i} \delta_{jjr} \cdot \lambda(i,j) \quad (18)$$

ここで、 $j=j_r$ の場合のみ $\delta_{jjr}=1$ 、それ以外は0である。これを全てのiについて評価するため、サービス継続比率SLとして次ぎの評価指数を提案する。

$$SL = \frac{\sum_{i=1}^I \lambda(i,j)}{\sum_{i=1}^I \lambda(i,j)} \quad (19)$$

前述の一定閾値 $P_0$ を大きくすれば $R_i$ が小さくなり、それだけ複製を持たせるサイトの数が少なくなり、その結果 $\lambda' = 0$ のiが多くなり、SLは小さくなる。

## 7. データの一貫性

独立運転時の一貫性を検討する。この場合にはネットワークが切断されているので、一貫性は一時保留の状態である、すなわち楽観的同時実行制御方式6)の前半に相当する。ネットワークが接続された時点で全体一貫性の回復処理が必要となる。複数作業員が同一アイテムをwriteする際に、衝突が発生した場合でも、通常のトランザクション処理とは異なり、アイテムの作業を無効(abort)とはしないで、できる限り作業結果を活かすことが必要と考えられる。その理由は前述のように、複数の作業員が互いに別の作業結果を持ちより、より高品質の作業結果を得ようとすることは、本来的に別々のデータ内容を期待しているという面が有るから、また例えばどちらかが正当な作業結果を持ってきた場合に、どちらも無効とすると手戻り作業となり効率が悪くなるからである。

### 7.1 操作の履歴

独立運転時にデータベースを操作(readまたはwrite)する際に、全てをデータ操作履歴として記録しておく。このデータ操作履歴を使用して、衝突発生時にもアボートせずに、出来る限り操作を活かせる方策を検討する。

n回の操作履歴を、

$$o_1(x_1), \dots, o_n(x_n) \quad (20)$$

とする、oは操作すなわち readまたはwriteである。xは操作されたアイテムを表す。

簡単のため作業員が2人の場合を考える  
サイト1についてn回の操作があった、

$$o_{11}(x_{11}), \dots, o_{1n}(x_{1n}) \quad (21)$$

サイト2についてm回の操作があったとする。

$$o_{21}(x_{21}), \dots, o_{2m}(x_{2m}) \quad (22)$$

### 7.2 データ融合の必要性

複数作業員からの作業結果すなわち操作履歴が衝突した(同一のアイテムにwriteが重複した)場合に、データの内容を調べ、矛盾があればそれらの調整案を支援システムから提供することを検討する。この矛盾データ内容の調整を本稿ではデータ融合(data fusion)と称することとする。

(a) oの中に同一のアイテムにwriteが重複していない場合、データ更新をコミットする。

(b) oの中に同一のアイテムにwriteが重複している場合、

(b-1)重複したアイテムについてのwriteのデータ内容が一致しているときデータ更新をコミットする。

(b-2)重複したアイテムについてのwriteのデータ内容が一致していないとき、単純な(read write)レベルでのトランザクションの直列化(serialization)では、あまり意味が無いと考えられる。このような場合に作業者の意見調整としてデータ融合を行い、調整案として使用者に提示することが考えられる。

このような状況は独立運転時のみならず協調運転時においても高位レベルでの意見調整の問題がある。

データ融合の処理を全自動的に行うよりは使用者との対話型で行う方が適切と考えられる。従って支援システムは次のような調整支援機能を備えることが望ましい。

- ・内容矛盾の相互への通知支援機能
- ・内容矛盾の発生理由説明支援機能
- ・選択(保留)の通知支援機能
- ・選択の理由説明支援機能

### 7.3 データ融合の方法案

データ融合自体で一つの研究問題であると考えるが、本稿では単純な方法を提案する、今後の研究を必要とする。

#### (a) データが数値的内容の場合

- ・加重平均値による融合

$$x = w1 * x1 + w2 * x2 \quad (23)$$

ここでw1、w2は更新しようとしているアイテムに関するサイト1、2各々における重み(権限、責任、信頼を代表させる)で、前述の「複製の重み」w(jri)を適用することも考えられる。x1、x2はこのアイテムについてサイト1、2各々が更新を提案してきたデータの内容である。xは融合されたデータの内容である。

- ・選択による融合

w1とw2の大きい方のx1、x2を選ぶ

- ・データの完全性(integrity)による融合

x1とx2のどちらが完全性に合致(近い)しているかを調べ、合致(近い)している方を選択する。完全性の計算式をfとして、

$$f(x1, X), f(x2, X) \quad (24)$$

を計算し比較し、選択する。ここでXは、x1,x2以外のデータアイテム値の集合である。計算式fは全データアイテム(1,2,3,...,I)の値の間での関係式であり、対象となるDBの内容に依存し様々である。今後の研究課題である。

#### (b) データが文書的内容である場合

内容の表現に関係する研究は形式的仕様記述、制限自然言語による要求記述等の研究が行われている。<sup>7), 8), 9), 10)</sup>

いずれかの方法により、データ内容は論理的表現すなわち以下のコンピュータ処理が可能な形になっているものとする。

データ融合の方法案を以下に示す。

- ・権限(責任)の大きい方を選択する方法

この方法の場合、選択はデータ内容が数値的内容の場合と同様に簡単に行うことができるが、責任の無い作業の方が、競争に参画する意欲が出ないので協調作業本来の相乗効果が発揮しづらい。

- ・データの完全性による選択方法

上位情報の内容との整合性(継承関係)が正しいか調べ、正しい方を選択する。(上位と矛盾しているものを棄却する)

上位情報の内容との整合性が両者とも正しい場合、より詳細を述べている方を選択する、または両者の詳細部分を併合して新しい内容とする。この場合にアイテムはさらに細分化され、粒度を再設定し、上位情報を改訂することが必要となるであろう。

例えば、あるデータアイテムについてそれをブレイクダウンする作業であれば、作業員1及び2のブレイクダウン結果を比較して、ブレイクダウンが詳細の方を選択、あるいは両者の詳細の部分を選択して一つにまとめるといった方法が考えられる。

## 8. まとめ

本稿では、筆者等が検討している分散システムにおける資源配置方法を協調作業支援システムに適用することを試みた。前者についても、今後検討すべき課題が多くあるが、協調作業支援システムに適用しようとするとき別の観点からの課題も少なくないことが把握できた。

特に協調作業の高度な支援(作業者お互いの知的触発を引き起こし、一層レベルの高い作業成果を得るための支援)のためには作業者の作業結果(意見)の相違をread/writeレベルではなく、データ内容まで立ち入った高位レベルでの調整判断が必要となってくる。

### (a) 資源配置に関する課題

今後の課題は、資源配置に関しては以下の項目の検討を進めていきたい。

- ・データの配置に続き、プロセスについての配置問題の検討
- ・ケース43以外の分割配置方式の検討
- ・データモデルの拡張  
複数属性の関係モデルの分割配置、近似回答の検討等
- ・複製配置方式の精査、検証シミュレーションの実施
- ・複製維持コストについての詳細検討
- ・サービス品質低下についての詳細検討
- ・一貫性の検討

### (b) 協調作業支援に関する課題

協調作業支援システムに関しては、以下の項目の検討を進めていきたい。

- ・データ内容に関わる一貫性、完全性の検討  
属性の関係をルールとして利用
- ・これに関して知識処理、オブジェクト指向等の研究成果の活用
- ・シミュレーションによる検証
- ・プロトタイプによる検証

最後に、データ融合についてのヒントを教示頂いた三菱電機(株) 立花康夫 博士に謝意を表します。

## 参考文献

- 1) S.Ceri, B.Pernici and G.Wiederhold: Distributed Database Design Methodologies, *Proc.IEEE*, Vol.75, No.5, pp.533-546 (1987)
- 2) S.B.Davidson: Replicated Data and Partition Failures, *Distributed Systems*, S.Mullender, ed., ACM Press, pp.265-292(1989)
- 3) M.T.Ozsu and P.Valduriez: Distributed Data Management-Unsolved Problems and New Issues, *Readings in DISTRIBUTED COMPUTING SYSTEMS*, T.L.Casavant, ed., IEEE Computer Soc. Press, pp.512-544 (1994)
- 4) 宮西, 高品, 渡辺, 水野: 分散システムにおける独立性の観点からの資源分散方法の提案, 情報処理学会第48回全国大会 7D-1 (1994)
- 5) 岡田, 松下: 協調の次元階層モデルとグループウェアへの適用, 情処情報メディア&グループウェア研究会, Vol.93, No.95 (1993)
- 6) 水野, 斎藤, 福岡, 落合 訳: 分散システム コンセプトとデザイン, 電気書院(1991)
- 7) 広井, 佐伯: リフレクティブ言語を用いた形式的仕様記述の支援, 情処ソフトウェア工学研究会, Vol.94, No.6 (1994)
- 8) 大崎, 石原, 関, 嵩: 自然語仕様から代数的仕様への変換における表現式の構文規則の生成, 情処ソフトウェア工学研究会, Vol.94, No.6 (1994)
- 9) 蓬菜: 形式的仕様記述プロセスにおけるモジュール協調, 情処ソフトウェア工学研究会, Vol.94, No.6 (1994)
- 10) 原林, 河合, 椎野, 武内: 制限自然言語によるソフトウェア要求記述とその解析, 情処ソフトウェア工学研究会, Vol.93, No.107 (1993)