

## 分散処理システムにおける名前管理の一方式

古宇田フミ子 田中 英彦 元岡 達  
( 東京 大学 工学部 )

### 1 はじめに

計算機の諸資源は計算機間の結合により、単独利用だけでなく共同利用が可能になる。このためにプロトコルの標準化が進められているが、これと並んで諸資源の命名法や管理の重要性も忘れてはならない。

計算機諸資源の呼称は、通常、各々の計算機に特有のものであり、網構成で利用する場合必ずしも調和的でなく不便な面が生ずる。だからといって網内すべての資源に統一的な名前付けを行なうのは、制約が大きい。各計算機に特有な命名法を保持しつつ、かつ網においても有効に対処し得る名前管理方式が必要となる。

そこで、本文では、分散処理系での名前管理システムを、名前の捉え方そのものから考察を行ない、それに基づいた構成法の一つについて論ずる。

この分野における他の研究としては、次のようなものがある。

ISO/TC97/SC21 (1984年までは、SC16) ではOSIモデルに基づくネーミングとアドレッシング方式を検討している。[SC16 N1390など]

Grapevine (XEROX) [Birrell82] は網構成での電子郵便などのメッセージのやりとりに関する名前付けやその管理を扱った。

分散データベース管理のための名前付けや管理は、Clearinghouse (XEROX) [Oppen81, 83] や、IBMのR\* [Lindsay80] が有名である。

IEEEプロジェクト802ではネットワークのアドレスを提案している。

本文の構成は、2章で名前管理の問題点を幾つか挙げ、ここでの方針を述べる。3章では、名前の捉え方とこれに基づく構成法および、言語Pascalを用いた構成例を示し、4章ではこの考察を行ない、仮想的な使用の一例を紹介する。

### 2 分散名前管理の問題点と方針

#### 2-1 問題点

1. 名前とアドレスの観点 : 網では、目的とする対象と、これの存在場所とを厳密に区別しなければならぬ。そこで、本論では、名前は対象を示すもの、アドレスは場所を示すものと定義した [Shoch78] の立場を基本とする。(名前は、what we seek、アドレスは、where it is を示す、と述べている。)

プロセス間通信では、ホスト差を意識しないこと、つまり、通信の透明性が重要とされ、研究が行なわれてきたが、宛先の指定法として、「ホスト名、ホスト内番号、(その他の情報)」の形式を採るものが多い。(例ARPA)これは、ホスト差を意識しない、と言いつつ「ホスト名」を書くことにより利用者にその差を意識させることになる。

真の透明性を実現するために、利用者には、アドレスの概念を一切排除し、目的とする対象は名前で示す方式が必要となる。この方式を採れば、対象指向プログラミングとの整合性が良く、ローカルな対象指向言語を自然な形で分散処理系に拡張することに役立つ、と考えられる。

2. 利用者の用いる名前とシステムで用いる名前の使い分けの必要性 : 処理系で用いる名前を利用者にも見せる場合がある。この方式は、処理系に向けた名前体系なので、利用者には使いにくいことや、利用者の恣意が働く余地がある。利用者用とシステム処理用の名前を異なる形にすれば、処理系にも、利用者にも使い易い名前が可能になる。単に名前を分けるだけでは、悪意対策には不十分なので、二つの名前の間にアクセス資格等の何らかの保護機能を組み込む必要がある。

3. 網での名前付け方式 : 網での名前付け方式は、一般には、絶対的名前付け方式と、相対的名前付け方式の二通りがある。これは、名前の一意性の範囲を基準にした分類であり、前者は考えている範囲全体で名前を一意に識別するように構成し、後者は、ある部分で一意とする構成である。

絶対的名前付けは、皆が同じ名前を使うので、名前管理は容易となるが、網全体で一意性を保証するため

の方策が必要となる。例えば、XEROXのclearinghouse [Oppen83]では、同姓のユーザ・ネームに対する識別法としては、birthmark を付けて区別している。

また、使用可能な表現体系が固定されるので、異なる文字体系からは使用できない。例えば、ローマ字による名前付けが行なわれている場合は、ギリシャ文字や漢字の名前では参加できない。

相対的な名前付けの場合、名前の付け方は、名前を付け得る全空間を共通部分が無く、しかも全体を覆うように分割して各々で一意とする方法や、この空間の一部分を他との重なりを許して切抜き、そこで一意とする方法がある。後者は、同一の対象に別名という形で名前が複数個付き得るので、冗長性が生じ、分散処理系向きの名前付けが可能になる。

網では、情報の伝達遅延や伝達順序の非決定性、不着の可能性がある。分散系での名前の更新には、このことの考慮が必要となる。更新の方法として、(1)更新時はすべての更新操作が完了するまで次の状態に進めない。(2)更新操作が完了した所から、次の処理に進む。更新中には、一時的に情報の不一致が生ずる、がある。一例として、分散データベースシステムにおける2フェーズコミットは前者であり、前述のclearinghouseは一時的な不一致は認める方式を採っている。更新の時間管理はタイムスタンプ(timestamp)方式を用いることが多い。

## 2-2 分散名前管理の方針

1. アドレスでなく名前を扱う。名前付けが、場所ではなく対象に依存していることや、生成、消滅など状態の変化が理解しやすい、と言う理由でプロセスの名前付けを、動的名前付けの面から、調べる。

2. 相対的な名前付けを採用する。同一対象に別名を許す、複数の名前体系を設定し、各利用者には使い易い名前を提供する。網であることを考慮した名前体系の決定法や異なる体系間の最適な関係を探る。

3. 分散処理系で動的名前付けに対処し得る機構を入れる。名前の更新に関しては、効率を考えて、更新済みの所から、次の処理に進む方式とする。網の伝達遅延や不着に対しては、C-listとA-listの両方を用いた保護機能を組み込んだ。通常はどちらか一方だけで保護機能を果たすことができるが、ここでは、分散系であることを考慮して、このような方式にした。タイムスタンプは用いず、C-listとA-listのケーパビリティの変化で更新管理を行なう方式にした。

4. プロセスは、核プロセス、システムプロセス、ユーザプロセスの三種類を考える(図2-2-1)。プロセスの場所は固定とする(ホスト間を移動しない)。プロセス間通信は、メッセージ授受方式で、OSのセッションレイヤに相当するものを考える。ユーザプロセスのプロセス間通信は計算機の差を意識しなくて済む方式、すなわち、相手計算機を問わず同一手法を用いる。

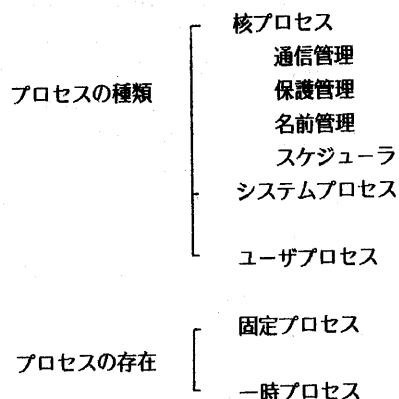


図2-2-1 プロセスの分類

## 3 分散処理系での名前の捉え方と構成

### 3-1 名前の概念的捉え方(この節では[分散84]での考察の要点のみ述べる。)

名前は対象を捉える場合の識別子である。対象の捉え方には、個として捉える場合と、或る観点(例えば、プロセスの機能は何か。)から対象を捉え、この観点の中で対象をクラス分けして識別を行なう場合(例えば、TSS機能やファイル処理。)との二通りを仮定する。個として識別する場合の識別子を識別名、或る観点から捉える場合の一つのクラスの識別子を意味付け名と呼ぶ。意味付け名は対象の或る一面を表すだけである。異なる意味付け名を組み合わせることにより、対象の性質を記述することが可能になる。意味付け名を組み合わせた識別子の列を記述名と呼ぶ。

名前の使用目的の一つは対象を一意に区別することである。一意に識別する範囲を系全体とすれば、絶対的な名前付けとなり、一部に限れば、相対的な名前付けとなる。一意に識別する範囲を、ここではnaming domain(以下n. d.と略す。)と定義する。相対的な名前付けを採るのでn. d.は複数存在する。識別名のn. d.を識別名n. d.と呼び、一つの観点に入る意味付け名の集合を意味付け名n. d.と呼ぶ。

ぶ。n. d. を管理するn. d. のn. d. を考える。これには、識別名n. d. の名前の集合としてのn. d. と、意味付け名n. d. の観点の名前の集合としてのn. d. の二種類ある。相対的名前付けの場合は誰がその識別子で区別するかが問題となる。そのため、n. d. の利用者を明確にする。

意味付けによる識別名の表わす内容は、状況によっては指す範囲が異なることがある。例えば、整数という語は整数論と計算機とでは意味が異なる。このようなことから意味付け名には説明を付ける。

この説明には、

- (1) 体系的な分類。例えば、semantic networkによる表現を用いて位置付けを決め、
- (2) 個々の意味付け名の指すものの微妙な違い(すなわち、inheritanceの差)は、それぞれに記述する。

という方針を採る。

以上の概念は、図3-1-1のように表わされる。

set of aspects

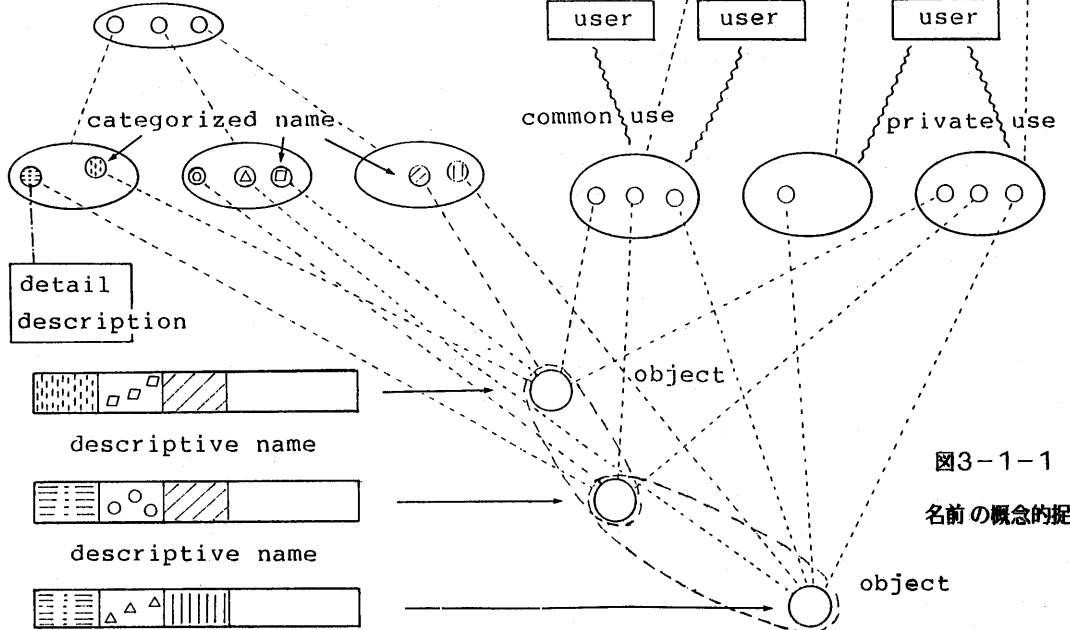


図3-1-1  
名前の概念的捉え方

同じ対象に対し、利用者毎に異なる名前が付き得るので、この名前の中に「同じ対象である。」という関係が付く。この点に注目して写像関係を設定した。

一般に2つの集合の間の写像は、(1) 1:1 (2) 1:多 (3) 多:1 (4) 多:多 の4通りの関係がある。同じ対象を表す識別名n. d. の要素間の関係は、2つのn. d. 間ではすべて(1)にした。関係すべてを利用する完全網を作ることもできるが、現実の管理の面からはこのうちの一部の対応関係を使うだけでよい。そこで、識別名やその別名、記述名、対象や利用者間の基本的関係を次のように見た。f, g, h, u, v とwを写像とすると、以下のようになる。

a) 名前から対象へのアクセス。すなわち、名前をアドレスに変換する関係。ここでは、アドレスや実際の道筋については、考察の範囲外なので、これらを、fによって、代表させる。

$$f(n. d. 名, name\ in\ n. d., time) = object(address)$$

b) 別名間の対応。異なるn. d. で同じ対象を指す名前の関係。

$$g(n. d. 名, name\ in\ n. d., n. d. 名, time) = alias$$

c) 利用者とn. d. の関係

$$h(user, situation, time) = n. d. 名$$

d) 識別名から記述名への写像。

$$u(n. d. 名, name\ in\ n. d., time) = descriptive\ name$$

e) 記述名から或る識別名への写像 ( { · } は多値を意味する。 )。  
 $v(\text{descriptive name, time}) = \{ (n. d. \text{名, name in } n. d., \text{time}) \}$

f) 識別名から対象への写像。  
 $w(\text{descriptive name, time}) = \{ \text{object}(\text{address}) \}$

g) に関しては、別名は複数あるので管理上都合のよい見方をする。一つの n. d. の要素 (すなわち識別名) を基準とすると、他の複数の n. d. の要素への多値写像とも見ることができる。この逆写像は1価と見る。このようなことから、写像 g の構造としては、

- (1) 1対1用  $g_1$  [ n. d. の要素の番号, n. d. の要素の番号 ]
- (2) 1対多用  $g_2$  [ 一つの n. d. の要素の番号, { n. d. の要素の番号の集合 } ]

の2組を考える。(図3-1-2)

写像 u と v の場合も g と同様になる。

### 3-2 分散処理系での名前管理

網では、情報の伝達遅延や回線障害の可能性があるので、各々のホストでデータがまとまっていることや、データに冗長性を持たせることが必要となる。これを踏まえて、識別名の名前管理を以下のように構成する。

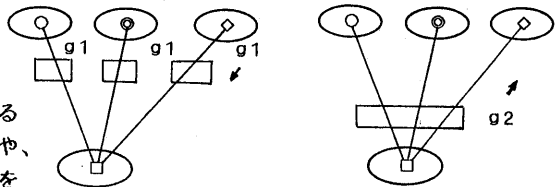


図3-1-2 写像 g

- (1) ホストの自立性を重視し、ホスト内に存在する全プロセスを一意に識別するために用いる n. d. を設け、これを Pnd と呼ぶ。
- (2) 網で情報交換するための識別子はどのホストからも理解できる必要があるので、各ホストの通信管理プロセスすべてに共用の識別名 n. d. を設け、これを Cnd と呼ぶ。

この二つの n. d. を基本とし、プロセスの用途に応じた識別名 n. d. を付け加える。核プロセスの場合を考える。スケジューラは、ホスト内に存在するプロセスのうち active なもののみ扱う (状態(2)後述) ことから、識別名は Pnd の一部を用いればよい。これを Snd とする。通信管理プロセス (以下 CO MP) は Cnd を持つ。名前管理プロセス (NAP) と保護管理プロセス (CAP) はホスト内プロセスの名前管理と、他ホストにあるプロセスの名前管理を、そのホストにある NAP や CAP に依頼する必要があるので、Cnd と Pnd を共有する。

システムプロセスについては、コピーや写像の数を減らし得ることや、アクセス権の制御は C-list や A-list で行なえることから (例えば或るプロセスからはアクセスできないプロセスの識別名は C-list をアクセス不可とする)、システムプロセス間の識別名は、なるべく共有する方針を採る。システムプロセスの性格上常駐プロセスと一時プロセスを区別した方が便利である。そこで、前者に対する識別名 n. d. とし、Fnd を、後者に対しては、Vnd を用いる。

ユーザプロセスはプロセス自身が、生成、消滅したり用途が多様に渡ったりするので、個別に n. d. を持つことにする。(Und と呼ぶ。) ユーザプロセスの間で共用 n. d. を持ってもよい。このことによりユーザプロセスにとって使い易い、自由な形式の名前付けや、動的名前付けが可能になる。以上の関係をまとめると、表3-2-1 のようになる。

表3-2-1 プロセスと n. d.

対象は一時的なものや常駐のものがあるので対応する識別名も変化する。このような動的名前付けに関するプロセスの状態 (状態遷移とは一致しない) は、次の3つを考える。

- (1) プロセスの生成から消滅まで (生きている間を表す。)
- (2) プロセスが active (run, ready, wait 状態) である。
- (3) プロセスが通信登録を行ない情報交換可能な期間。(OSI の dialog unit [ISO/DIS8326] に当る。)

使用プロセス	識別名 n. d.	記述名
核	scheduler	Snd
	名前管理	Pnd と
	保護管理	Cnd
	通信管理	Cnd
system	Fnd と Vnd	ldes rdes
user	個別 n. d.	

この何れかの状態遷移が起こるとき、プロセスの名前の登録や更新、抹消が行なわれる。

分散環境での動的名前付けを行なうためには、プロセスの存在状態の他に、情報の伝達遅延や順序の変化、

不着などを考慮しなければならない。情報の受け側から、伝達の状況を分類すると、

- (1) 順序や内容が正しく伝達される。
- (2) 内容は正しいが、順番が変わって到着する。
- (3) 到着しない。
- (4) 内容が変化して到着する。

(4)については、回線の問題であるので下のレイヤで解決可能である。そのためここでは考えない。

遅延や不着の対策としては保護機能が必要となる。保護機能の実現法については、[分散84]で述べた。それは、C-listとA-listとを併用する方式であり、C-listとA-listとを関連ある対象間で置く、というものである。

### 3-3 言語Pascalによる構成

n, d. の論理構造 (record型) は次のようになった。 : (1) n, d. の利用者名。この名前は名前管理プロセスの用いる識別名で書く。(2) 識別名。このn, d. の利用者が共通に理解する識別子であり、この中で一意に区別がつく。(3) A-listとC-list。管理を容易にするため識別名と利用者に関するA-listとC-listも組入れた。(4) 写像g1。

表 3-3-2 写像 g

cndまたはpndへの写像も組み込み、名前変換の手間を省いた。以上が、一般的n, d. の構造であるが核プロセスの場合は、権限が大きいことなどから、保護機能を省き、使用者が限定されているので、利用者名も除いた。写像は独立して設けた。(図3-3-1) 写像gの種類 : (1) 写像の表を設けたもの。(表3-3-2) (2) 直接ポインタなどで指すもの、の二種類ある。

写像の名前	行き先	写像のタイプ
dm1	descriptive name -> pnd	<g2>
	pnd -> descriptive name	<g1>
dm2	descriptive name -> cnd	<g2>
	cnd -> descriptive name	<g1>
nm	snd -> cnd	<g1>
	cnd -> snd	<g1>
psul	pnd -> und	<g2>
und内	und -> pnd	<g1>
culr	cnd -> und	<g2>
und内	und -> cnd	<g1>

処理 : 名前管理プロセスの処理としては、(1)核プロセスの識別名とその関係の初期化。動的名前管理では、(2)プロセス生成時の新たな識別名に対する関係の作成。(3)プロセスをactiveにした時の登録。(4)通信可能状態にするためのbind処理。(5)プロセスの状態が変化した時の更新(状況により状態(3)のまま、または、状態(2)に変化)。(6)プロセスの消滅に伴う識別名の変化の処理、がある。静的名前管理には、(7)識別名間の交換。(8)記述名から対象を引く。(9)識別名から対象を引く、などがある。n, d. と写像テーブルと、対象間の関係の一例を図3-3-3に示す。

## 4 考察

### 4-1 相対的な名前付け

(1) 名前付けを相対的に行なうことにより、対象を一意に識別する範囲は網全体である必要がなくなり、絶対的な名前付けを行なっているclearinghouse [Oppen83]のような名前全体を管理する機構が不要になる。名前から対象へは、別名という形で連結性があるので、手練ることができる。

(2) 対象の識別名は管理の面からは、名前の階層化が必要となる。ここで提案した方式も名前の階層を2層とした。そのうち上位層は、名前管理表(具体的には、n, d. に組込。)埋め込んだので、普通は、利用者にとって、必要な識別名のみで利用することができる構造となった。そのため、名前を毎回、階層の形で提示しなくとも済む、という利点が生ずる。

(3) 名前付けの権限を利用者としたので、利用者には、名前の形式や使用文字種の制限は無くなった。

(4) 相対的な名前付けではあるが、n, d. の使用により、この体系内では、名前の一意性が保証される。さらに、利用者に必要な識別名のみ登録されるので、誤りや、悪意の処理が防げる。

COMP  
NAP  
CAP

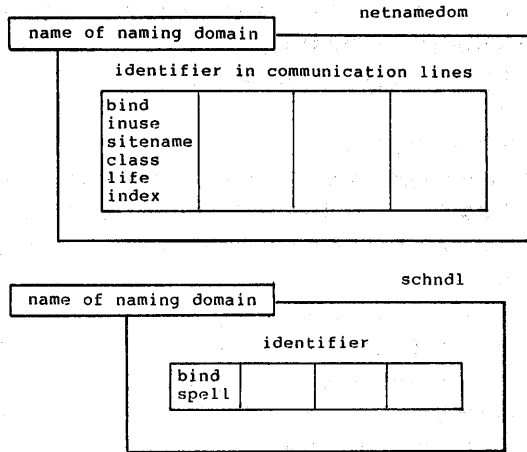


図3-3-1 n. d. の構造 核プロセスの場合

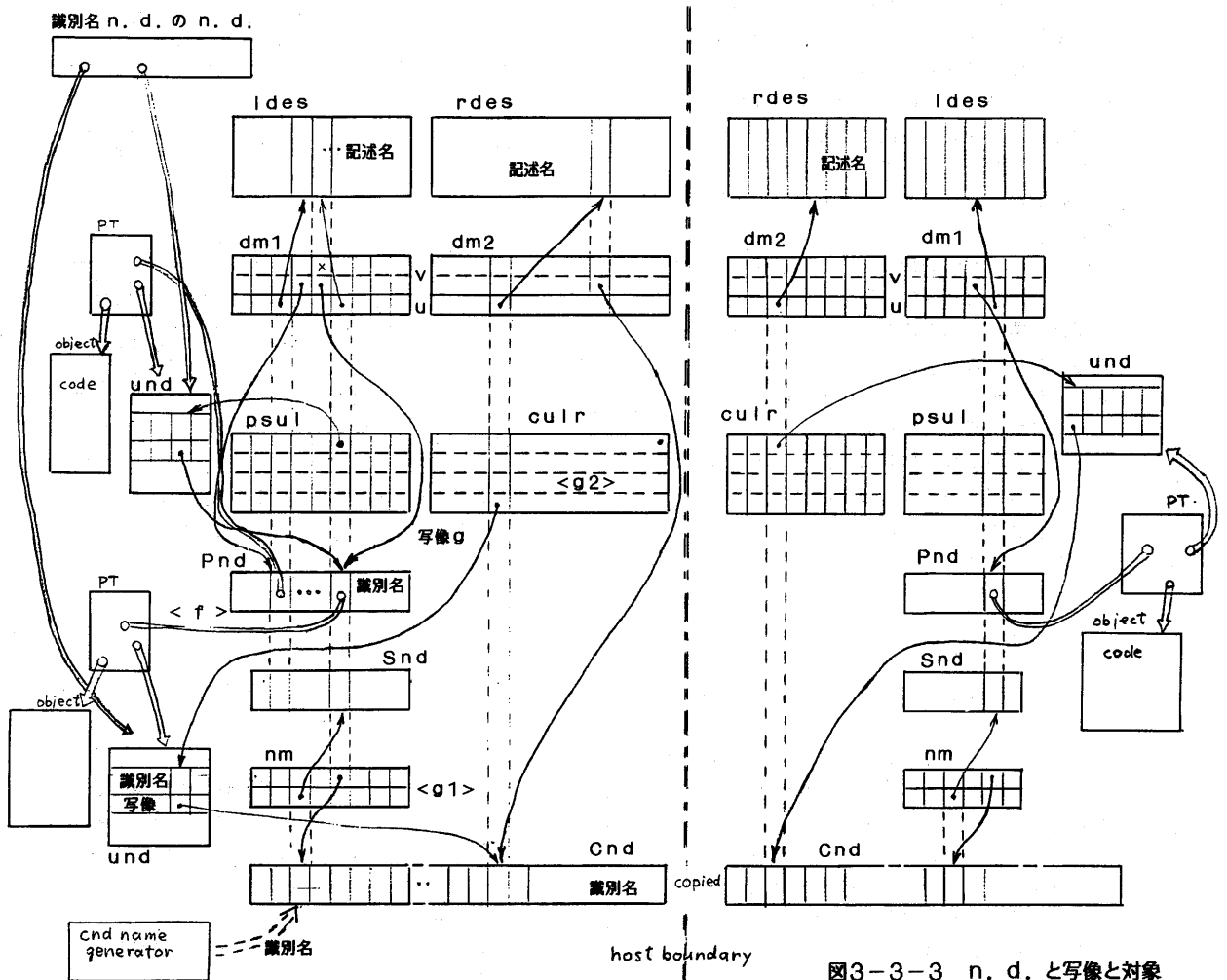


図3-3-3 n. d. と写像と対象

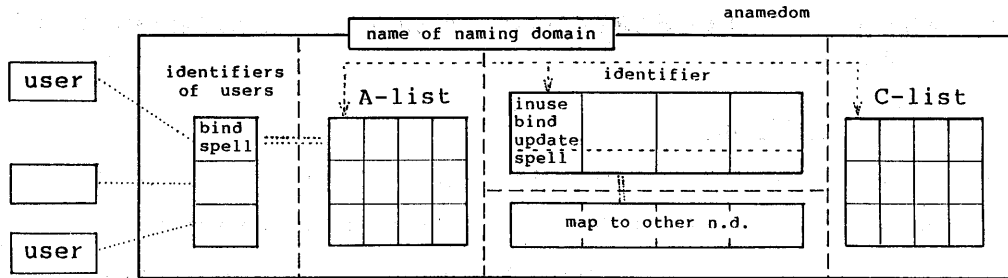


図3-3-1 一般的 n. d. の構造

#### 4-2 分散処理系での名前管理方式

(1) n. d. は、或る意味でデータベースと見ることができるので、分散名前管理も分散データベースシステム方式の一部を流用できる。データベースの分散配置法には、種々の議論がある。ここでは、名前の一意性や、必要な名前のみを用いることに注目したため、n. d. の分散配置法は完全な重複や分割にはならない。n. d. の写しとしての繰り返しを持つのは、cndのみである。個々の名前については、同一対象に対する複数の識別名は名前の形が異なる繰り返しと言える。3-1で述べた写像関係を関係表の形に書き下すと、関係代数演算により、処理を進めることができる。

(2) 分散系になると、更新は、写しを持つ場所だけでなく、(1)で述べた個々の名前の繰り返しを持つ場所すべてについて行なう必要がある。効率と安全性は相反するものであるが、ここでは、遅延や不着対策を重視し、二重の保護機能を入れ、しかも効率低下を防ぐため、更新操作が終了した場所から、次の処理に進む方式となった。

#### 4-3 名前付け構造の比較

名前から対象への写像について調べる。clearinghouse [Oppen83] は、3層の識別名から、性質<性質名、性質の型、性質の値>への写像を持つ。型がitemかgroup かに応じて記述子または、名前の集合に行く。数種のserverによりこれを実現している。R\* [Lindsay80] の名前はprint nameからcontext やhandleを利用して、システムでユニークなSWNという4層形式の名前に解きほぐし、これにより対象の記述への写像を行なっている。何れも、名前は層形式で示し、一段で対象の記述に到る。

ここで提案した方式は、識別名は前述のように必要な部分(の一層)でよいが、名前から対象へは、数個のn. d. を経て、到達する。対象と対象の記述名を分けたことにより、記述名も一種の識別名として機能し、記述名によるアクセスが可能になる。

#### 4-4 分散名前管理実現への一案(仮想的な使用例)

国立大学の大型計算機センタ間を結ぶ計算機網N1ネットワークがある[N1]。これは、異機種計算機間の接続になることが多いので、一つの計算機から、他の計算機に入って利用する場合、両方の計算機の異なるコマンド体系を理解する必要がある。読むべきマニュアルの量が増えて、負担が重い。これは、現在のN1が単に基本的な通信レベルのみを提供し、計算機の差異は、すべて利用者に任せているため、と言える。そこで、N1システムにここで提案した分散処理向き名前管理システムを(仮想的に)載せ、ジョブ変換機能を持つシステムプロセスも載せることにし、名前管理を異種のコマンド名の体系に適応を試みる。例えば、A大学のみ存在する特殊な有限要素法をB大から使用させよう場合、B大側では、記述名による問い合わせ、および結合手順、ケーパビリティチェックを経れば、B大側で決めた識別名と、計算データを提出するだけで、A大学の計算機のコマンドを知らなくとも、使用できる可能性が生ずる。

### 5 おわりに

分散処理向き名前管理の一案について述べた。提案した方式の特徴は、相対的な名前付けを採ることにより、(1)名前付けの権限を利用者に持たせ、使い易く、利用者には計算機の差を意識しなくて済むような名前を提供できるようになった。(2)記述名からも対象を引けるのでフレキシビリティを持つ。また、(3)名前管理

の構造を明確化し、分散処理系での管理に向く方式にした、などである。

今後は、言語Pascalによる定量的評価を行ない、この考え方に基づいた分散名前管理方式を網上に実装して行く予定である。

#### 参考文献

- [1] Oppen, D.C., Dalal, Y.K. : The Clearinghouse report. OPD-T8103, 1981
- [2] Oppen, D.C., Dalal, Y.K. : The Clearinghouse : A Decentralized Agent for Locating Named Objects in a Distributed Environment , ACM. Transactions on Office Information Systems Vol.1, No.3 July 1983 pp230-253
- [3] Bruce Lindsay : OBJECT NAMING AND CATALOG MANAGEMENT FOR A DISTRIBUTED DATABASE MANAGER , RJ2914 (36689) 8/29/80  
Computer Science
- [4] Birrel, A.D., Lelin, R., Needham, R.M. and Schroeder, M.D. : Grapevine : An Exercise in Distributed Computing Communications of the ACM April 1982, Vol.25, No.4
- [5] ISO/TC 97/SC 16 N1390 : Draft answer to question on principles of naming and addressing  
( WG1 Q 13 ) March 1983 ISO/TC97/SC16/WG 1
- [6] IEEE project 802 DRAFT Appendix D PART2 : LOCAL NETWORK ADDRESSING  
May 1 1982
- [7] SC16 N236
- [8] John F. Shoch : Inter-Network Naming, Addressing, and Routing  
COMPCOM 78 fall
- [9] ISO/DIS 8326 : Information processing systems - Open systems interconnection - Basic connection oriented session service definition  
August 1984 ISO/TC 97
- [10] ISO/DIS 8327 : Information processing systems - Open systems interconnection - Basic connection oriented session protocol specification  
August 1984 ISO/TC 97
- [分散84] : 分散処理システム研究会 21-4 1984.2
- [N1] : VOS3 N1ネットワークシステム使用の手引 8090-3-129-20