

分散環境におけるプロセスの名前管理機構

3H-2

に関する考察 — その2 —

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

計算機資源は、名前を用いて呼ぶ (call by name) 要請がある。その際、対象と名前との間に曖昧性があると、正しくアクセスできない。このような名前管理の必要性は大きい、これまでに、それほど研究されてこなかった。

そこで、筆者らは、昨年以来、名前の役割を調べている。目標は、計算機諸資源の命名法およびその管理法にある。が、当面は、特にOS制御で重要となるプロセスの名前管理に的を絞っている。これまでに、名前管理の一手段として、naming domain (後述) という概念を導入し、これを名前の構造を明らかにし、名前管理を行なうための指針としてきた。[1]

今回は、これを用いた名前の制御と名前管理機構の検討を行なったので、ここに述べる。

2 これまでの経過

2-1 考察の前提と要請事項

名前管理を考える範囲として、分散環境を前提とする。そのために、網の伝達遅延や回線障害の可能性を考慮する。プロセス間通信は、メッセージ授受形態を考える。

対象を名前で呼ぶ場合、用途に応じて、使い易い(形式の)名前を用いることができれば便利である。また、プロセスの名前からその属性が引き出せるとよい。制御の面からは、名前で呼ぶ時の対象が正しく一意に決まることが望ましい。

これから、名前管理プロセスの処理は、以下のものを設定した。

- (1) 識別名(後述)同士の変換
  - (2) 或る属性を満たすプロセスの探索
  - (3) 呼側(caller, 対象を名前を使って呼ぶ人)の用いる識別名から対応する対象(object)へのアクセス
  - (4) naming domainの要素の一意性の保証と管理や、naming authorityとの関連
  - (5) プロセスの状況の変化(生成、消滅)に基づく、名前の追加、更新、削除などの管理。
- このうち、主に、(1)-(3)の機構を述べる。

2-2 名前の機能の解析

この節は、前回も述べたことを簡単に再録する。

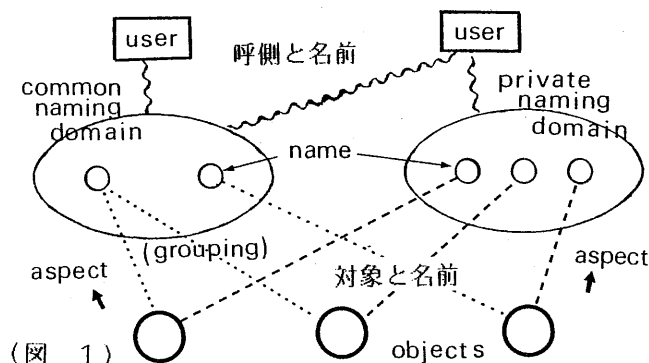
名前は、対象を参照し、区別するための目的で使う。これには、対象の捉え方と名前、呼側と名前の両側面を考える必要がある。(図 1)

対象を区別することは、各々が一意に識別可能ということである。一意に識別するためには、どの範囲で一意になるかを定める必要がある。そこで、名前が一意になる範囲をここでは、naming domain と呼ぶ。この要素は、名前を表わす。

対象の捉え方には、対象を個として捉える場合と、対象の或る一側面、つまり意味付けによる区別との二面がある。ここでは、前者を識別名、後者を意味付け名、と呼ぶ。

いろいろな意味付けによる名前を組み合わせることにより、対象の記述ができる。(descriptive name)

naming domain そのものに名前を付けることにより、名前の構造化が可能になる。



(図 1)

3 名前管理機構

3-1 naming domainの使用方法と特徴

対象を個として区別する場合を考える。呼側の目的に応じた対象の呼称を使えるようにするために、それに合うような naming domain を使うことにする。同じ呼称を用いる時は、共通の、また、異なる呼称の時は、個別の naming domain を用いる。

ここでは、プロセスをユーザ、システム、核の三種類に分けた。ユーザプロセスは各々用途が異なるので、個別に識別用のnaming domain

を持つ。システムプロセスは、一つのホストで共通な識別用のnaming domainと、用途に応じた個別の識別用naming domainを持つ。一つのプロセスは、複数のnaming domainを所有してよい。核プロセスの処理がホスト境界を越えて行なわれるものでは、識別用のnaming domainは考えている網で共通に用いる必要がある。一方、核プロセスの処理がホスト内で閉じる場合は、その範囲で共通にすればよい。

naming domainをこのように用いることにより次のような特徴が生ずる。

- (1) 名前の指す対象の範囲が明確になる。名前と対象の間の曖昧性が無くなり、capabilityの強化にも繋がる。
- (2) 同じ対象を異なる名前を用いて、利用できる。
- (3) 違うnaming domainを用いることにより、異なる対象を同じ名前と呼べる。

対象を或る観点から意味付けして捉える場合、観点(aspect)毎に、naming domainが定まる。この場合は相互理解の必要があるので、すべての呼側で共用する。

観点としては、場所、機能、capability等を考える。

### 3-2 naming domain間の関係

識別名は、呼側にとって都合のよい対象の呼び方なので、この間の変換をする機構が必要になる。

写像の方法としては、

- (1) 各々の識別用のnaming domain間で対応付ける
- (2) 特定の識別用naming domainとの間で対応付ける。

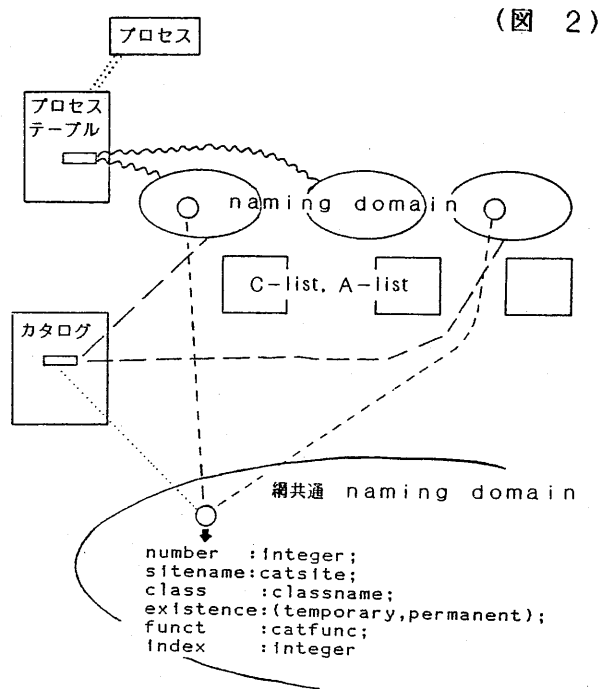
などの方法がある。naming domainの要素数が多い場合は、(1)の方法では写像の数が膨大になり、処理が大変なので、ここでは、(2)を採用した。特定のnaming domainとしては、網共通のnaming domainを用いた。

写像の機構には、網共通のnaming domainの要素(名前)が、どのnaming domainに入っているかを示すカタログがある。

各naming domainとともにcapabilityの表(C-list, A-list)もあり、不正なアクセスを防ぐ機構を入れた。

プロセステーブルには、所有するnaming domainの名前の一覧がある。

これらを用いて、呼側から対象へのアクセスや、識別名の変換を行なう。(図2)



意味付け naming domainの要素を組み合わせた名前(descriptive name)の表と(2)による識別名との対応表を作り、2-1 (2)、すなわち、属性を問い合わせることにより、プロセスのアクセスができるようにした。

### 3-3 naming domainの要素

識別名の付け方は、原則的には、(1) 全く意味の無い名前(例えば通し番号)を用いれば十分であるが、(2) 機能や場所を考慮した名前付けを行なえば、処理の経済化を図ることができる。(2)では、意味付けnaming domainの要素を利用することができる。

ここでは、網共通のnaming domainでは、要素は、主として、(2)を用い、これだけでは一意に区別が付かない時は、(1)による通し番号を用いた。システムプロセスの用いる名前は、(2)を利用し、ユーザプロセスの使う名前は、夫々に任せた。

## 4 今後の方針

現在、ここで述べた機構を言語PASCALを使って記述を行っている。これを基に、シミュレーションをし、評価を行なう予定である。

### 参考資料

- [1] 古宇田、田中、元岡：分散環境における諸資源の命名法および名前管理機構の構成法；分散処理システム研究会21-4 (1984)