分散環境における名前管理システム†

古宇田 フミ子†† 田中 英彦††

分散環境において，利用者に分散透明性を提供可能な，新しい分散処理向きの名前管理構成法を提案する．計算機網は，多様な資源の共同利用を可能にしたが，異機種計算機資源の扱い等，必ずしも，利用者向きの使用環境が提供されていない．ここで提案する名前管理構成法では，名前付領域の概念を導入し，利用者に計算機資源の構成方式の差異を意識させないことを可能にし，この概念を用いた名前利用管理により，分散環境で問題となる通信時の情報不一致対策を持つ．プロセス間通信は基本的な情報伝達手段であり，プロセスの更新はその管理者のみから生ずる等の理由から，名前管理構成法の考察対象を，網上のプロセス間通信を利用者が用いる場合の名前管理方式に絞った．プロセス間通信時，必要となる名前の種類や名前付領域の数，種類，構成法を検討し，利用者に分散透明性を提供し，分散環境で特に問題となる伝達遅延による情報不一致対策を含む名前管理の構成をモデル化した．特に，更新中に通信が生ずる場合の状況を調べた．その結果，利用者には識別名および記述名の二種類の名前を提供する必要があること，多様な計算機資源において分散透明性を提供するためには，名前付領域は少なくとも，四種類必要であること，これらの組合せとその名前利用管理法により，伝達遅延による情報不一致管理が可能であることが分かった．

1. はじめに

この論文は，分散資源利用時の名前管理構成法を，利用者指向で実現するためのものである．

計算機網は，単一計算機処理と異なり，計算機資源やOS等が多様であり，その名前付けやアドレス方式は異なることが多い．各資源は遠隔に配置されることから，一部の情報変化は瞬時に伝わらず，各資源に対する情報は必ずしも最新とはならない．

資源の利用は名前やアドレスを介して行われる．計算機網における名前やアドレスの従来の研究は，主にその分類や役割が主であった．

Shoch は，「名前」，「アドレス」，「ルート」の意味を各々，何を指すか，どこにあるか，どのように到達するか，と定義した[7]．

Hauzeur は，この定義では不十分であるとして，これらを分類し直した．名前には構造，時間，数の三つの属性があり，通信には名前とルートが基本要素で，アドレスはこれらの間の橋渡しである，と述べた[4]．しかし，この論文は OSI 基本参照モデルの旧版を基にした議論である．

国際標準化案 OSI 基本参照モデル ISO 7498-3[11] は，物理層から応用層までの名前とアドレス構成の標準化の定義と記述である．名前はプリミティブ名，記述名，一般名があり，各々にタイトルと識別子がある[5]．アドレスは名前の一部であり，(N)-層と$(N-1)$-層の間で同じ機能を表すサービスアクセスポイントの集合を指す．ルートの概念は用いていない．名前管理情報はディレクトリ管理に任せるとし，明確でない．

DASH プロジェクト[1] では，実体 (entity) を恒常的なものと一時的なものに分けた．名前はグローバル名，ローカル名の二種類であり，グローバル名にはシステムレベルとサービスレベルの名前があると分類した．システムレベル名の形態の必要条件も述べている．

網で統一した名前付けを与え，管理する方式がある．この方式は，Clearinghouse[6] における〈組織，ドメイン，ローカル名〉の3層形式等に見られるように，名前形式が制限され，使用文字列が長くなる．また，一意性の管理に birthmark の付加が必要なこと等，網を利用する場合の利用者には複雑で負担が大きい．

このように，名前体系は従来から議論されているが，定義やシステム内の名前構成に留まり，利用者向きの考察は不十分であり，これらの相互関係も明確でない．

ここで提案する分散処理向き名前管理は，利用者の便宜を考慮した名前管理の構成法を目的とし，(1) 異なる名前体系においても，利用者に自由な名前付け (分散透明性 (transparency) の実現) を可能とする，(2) 利用者に「問い合わせ」用の名前を提供し，従来のような「正確な」名前を知らなくとも資源利用を可能にする，(3) 名前管理構成法では，利用時の伝達遅延の影響を考慮する，等の特徴を持つ．本文で

† The Name-managing System in the Distributed Environments by FUMIKO KOUDA and HIDEHIKO TANAKA (Department of Electrical Engineering, Faculty of Engineering, University of Tokyo).

†† 東京大学工学部電気工学科

は，そのモデルを構成し，システム上の関係を明らかにする．

本文の構成は，2章で問題提起とその基本的取り組み方，3章で名前管理モデルの提案，4章でその例の検証，5章，考察である．

2. 分散処理向き名前管理法の提案

2.1 名前管理の構成法

まず，名前管理と名前付けとの違いを明らかにしたい．名前管理は付けられた名前がどの対象を指すかを管理することである．名前付けは名前をどう付けるかであり，付けた後の管理とは峻別されるべきである．

名前管理の構成には，名前の分類や役割の検討だけでなく，利用面からの検討も欠かせない．すなわち，名前付けを，計算機構成や網での統一的名前付け（例えば，Clearinghouse の名前付）等に拘束されず，利用者やシステムに自由に任せることができるように名前管理を構成する必要がある．これは，利用者に名前付けで分散透明性を可能にすることである．アドレスは分散を意識させるので，利用者には示さず，名前のみ用いる．

利用者に自由な名前付けを可能にするためには，利用者に必要な名前の種類やこれらを満たす名前管理の構成法を明らかにし，名前管理で分散の多様性や情報の伝達遅延対策を処理する必要が生ずる．例えば，伝達遅延による情報不一致対策は，利用者に必然的に見える情報不一致以外は，従来からの同期手法等を用いて管理可能なので名前管理システム内で取り扱い，やむを得ない場合のみ利用者に知らせる方式が必要である．

ここで，名前管理構成法を考察するための資源を絞る．分散データベースシステムや分散ファイル管理における名前管理は，同時更新の問題やレプリカの管理を扱う必要があり，複雑になるが，プロセスの場合は，単純化できる．すなわち，あるプロセスの更新は，そのプロセス管理者からのみなので，これらの問題が生じない．プロセス間通信は，情報伝達手段として，分散処理に基本的で不可欠な要素と考えられ，通信では，対象間の名前やアドレス管理の差や伝達遅延が陽に表れる．さらに，名前管理処理では，対象の更新通知等，プロセス間通信機構を利用して処理する．

このような理由から，プロセス間通信を含めたプロセスの名前管理システム構成方式を考察する．

以下において，対象，資源はいずれもプロセスを指すが，適宜使い分けることにする．

2.2 利用者に必要な名前の種類と対象の関係

利用者が分散した計算機資源を使用する場合は，対象を識別できること，対象の存在や機能を問い合わせることや，使用可能性を確認すること等を行う．

このような識別や問い合わせには名前を用いるが，識別と問い合わせでは利用目的が違うので，名前付けの考え方が異なる．対象を識別する場合は，個々に区別可能であればよい．これを識別名と呼ぶ．一連番号や無意味な文字列等で表される．問い合わせや確認は，識別名で行うこともできるが，利用者は必ずしも識別名を知っているわけではない．そこで，別の手段が必要になる．その手段として，対象の持つ機能や性質を述べる方法ならば問い合わせが可能になるであろう．一つの対象が持つ種々の機能や性質を述べる名前を記述名と呼び，個々の機能を指す単語を意味付名と定義する．すなわち，記述名は意味付名の集まりから構成される．一つの対象は区別されるための名前（識別名）と，その機能，性質を記述する名前（記述名）で参照される．記述名の構成要素はいろいろな側面から表現し得る．例えば，機能，利用者のアクセス権，等である．この各側面の要素を意味付名と呼ぶ．

2.3 名前管理領域の設定

利用者に必要となる識別名と記述名の二種類の名前により，分散透明性と情報伝達の不一致対策を可能とする名前管理の構成法を考察したい．

プロセス間通信では，利用者の通信相手の指定に（物理的）場所の差を意識せず使用できる名前体系が存在すれば，利用者に対する分散透明性を実現しやすい．一方，OS や名前管理等のシステムでは分散を意識して処理しなければならないから，利用者とシステムとで用いる識別方式は異なったものが必要となる．

利用者が必要とする資源を表す名前を，システム固有のアドレスや名前とは別に，利用者向けに付けることができれば，計算機の違いによる名前付けの差を意識することなく資源の利用が可能になる．

そこで，名前管理の基本要素として，名前付領域（naming domain, n.d. と略記する.）と呼ぶ名前管理表を導入する．この目的は，（1）利用者の用いる名前をシステムが提供する名前形式に制限されず，自由にする，（2）同じ名前で二つの対象を指すことや名前で指す対象と異なる対象のアクセスを防ぐ，（3）対象の消滅や移動等の更新時に問題となる，情報伝達遅延による情報の一貫性欠如に対する対策を可能にす

ること，である．

n. d. の要素，すなわち，名前が，識別名か，記述名か，意味付名かにより n. d. を各々識別名 n. d.，記述名 n. d.，意味付名 n. d. と呼ぶ．同じ n. d. の利用者数，すなわち，n. d. の利用者が単数か複数かにより，個別 n. d. と共通 n. d. とに区別する．n. d. の名前を集めた n. d. の n. d. (スーパ n. d.) も可能である．

名前付領域を複数用いると同じ対象を表す識別名や記述名が複数生ずる．この時，名前，時刻，名前付領域，対象の間には，次の関係が観察される．(fp fd gs gm ds dm を写像とする．冗長な関係も書いた．) ここで対象はシステム内の内部アドレスやポインタ等で示されるがこの部分の具体化は行わない．

- 識別名や記述名から対象を指す写像 fp, fd：
  - fp：識別名 (n. d. 名，時刻) ⟶ 対象
  - fd：記述名 (n. d. 名，時刻) ⟶ 対象

ある時刻におけるある n. d. 中の名前はある対象を示す．

- 識別名間の関係，識別名 1 から識別名 2 への写像 gs, gm：
  - gs：識別名 1 (n. d. 1，時刻 k) ⟶ 識別名 2 (n. d. 2，時刻 k)
  - gm：識別名 1 (n. d. 1，時刻 k) ⟶ {識別名 2 (n. d. 2，時刻 k)} (集合)

時刻 k の n. d. 1 の識別名 1 は同じ時刻の n. d. 2 の識別名 2 と同じ対象を表す．gm の識別名 2 は集合である．

- 記述名から識別名への写像 ds, dm：
  - ds：記述名 (n. d. 1，時刻 k) ⟶ 識別名 (n. d. 2，時刻 k)
  - dm：記述名 (n. d. 1，時刻 k) ⟶ {識別名 (n. d. 2，時刻 k)} (集合)

時刻 k の n. d. 1 の記述名と n. d. 2 の識別名は同じ対象を表す．dm は一つの記述名と複数の識別名を対応させる．

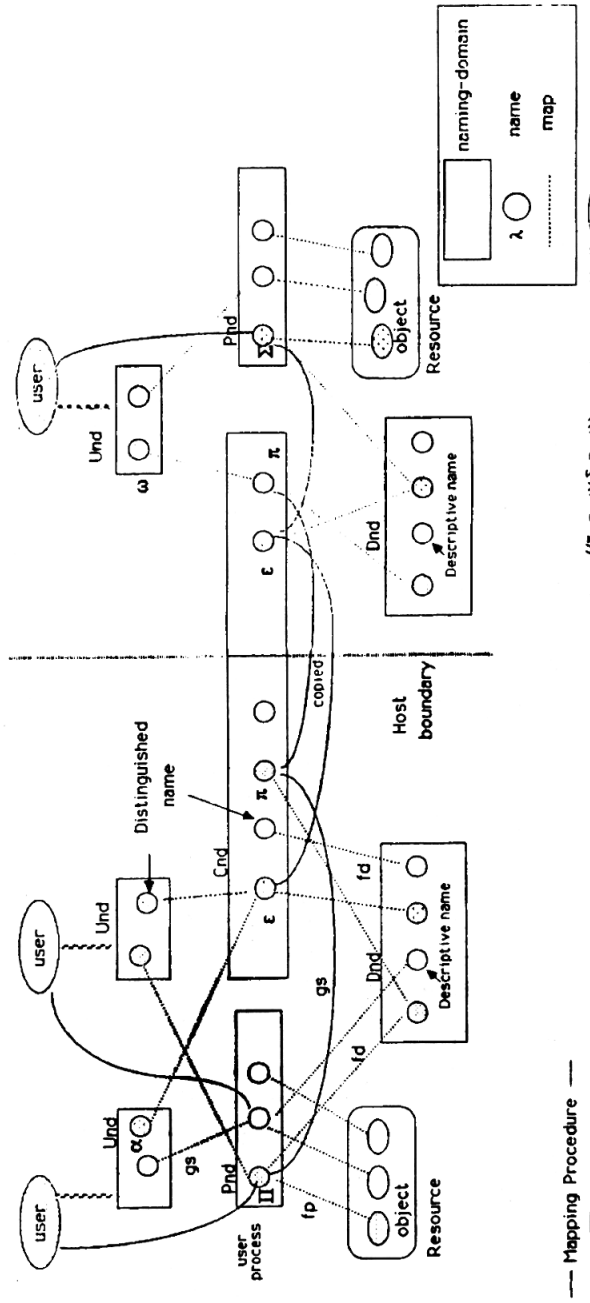

図 1 プロセス間通信を考慮した名前付領域と通信時の名前変換手順
Fig. 1 Several kinds of naming-domains, which are Und, Pnd, Cnd and Dnd. Names connected with dashed lines indicate the same object. The mapping procedure is also shown.

以上の関係を利用して，分散環境に向く構成を，プロセス間通信を含めて検討する．利用者に自由な名前付けを可能にし，分散透明性を図る，という目的から，名前付領域を利用者用，通信系に用い，分散で重要なホストの自律性も考慮に入れホスト用にも一つ作るという方針を採ると，以下のようになる（図 1）．

●利用者用n.d.：利用者が用いる識別名n.d. 個別または複数利用により，個別 n.d. または共通 n.d. を構成する．個別の場合は使用する名前付けは利用者の自由であるが，共通の場合は何らかの規則が必要である．

●通信用n.d.：通信の宛先を示す識別名n.d. 通信管理システム間で共通に用いて対象の識別を行うので，共通 n.d. である．この n.d. 中の名前には ISO 基本参照モデル「名前とアドレス」[11]中の用語，システムタイトルに対応する何らかの場所情報が必要であろう．名前形式には立ち入らないが，これは例えば計算機網 Junet のアドレス体系(書式例, suzuki@tansei.junet)に対応させ得よう．

●ホストごとの n.d.：分散環境では，ホストの自律性が重要である．ホスト内の全プロセスの名前を登録してある識別名 n.d. を設ける．

●記述名の n.d.：記述名は利用者が機能，役割を知るために使用する．共通 n.d. にすると便利である．

以上の n.d. を各々 Und, Cnd, Pnd, Dnd と呼ぶ．n.d. 間の関係付けは，同じ対象を表す識別名や記述名間の写像となるが，資源対象の利用時に，n.d. 間で名前の変換回数を少なくするように作る必要がある．

対象存在ホストには管理用の Pnd があるので，これを利用して，対象を参照する．すなわち，

fp：識別名 (Pnd, 時刻)⟶対象

である．

利用者 n.d. は，対象が存在するホストでは直接 Pnd に写像可能である．ホスト内になければ通信を介して対象資源を利用するので，それぞれ，

gs：識別名 (Und, 時刻)⟶識別名 (Pnd, 時刻)

または，

gs：識別名 (Und, 時刻)⟶識別名 (Cnd, 時刻)

となる．ここで，存在ホストにおいても，Und⟶Cnd⟶Pnd と写像する方法も可能であるが，変換回数が増えるので場合を分ける上述の方法を採用した．

Cnd は論理的には一つであるが，通信管理用という目的を考慮し，各ホストに，必要な部分の写しを置く．

Dnd を用いる場合も同様に対象がホスト内にある時は Pnd へ，ない時は Cnd への写像とする．すなわち，

ds：記述名 (Dnd, 時刻)⟶識別名 (Pnd, 時刻)

または，

ds：記述名 (Dnd, 時刻)⟶識別名 (Cnd, 時刻)

である．

2.4 プロセス間通信時の名前管理処理の問題と解決法

分散処理では網の伝達遅延による情報不一致の問題が大きい．この問題は避けられないので，不正アクセス対策も必要になる．しかし，通信のセキュリティ対策は，OSI の標準化案 (ISO 7498-2)[10] 等での検討や，データの加工に関する暗号化[2] 等の提案等，利用者から隠すことが可能なので，扱わない．

伝達遅延や処理系の遅れ等のため利用者にまで影響を与える場合がある．すなわち，利用者に利用可能な資源の状態が変化した時，変化情報が伝達遅延のために相手の利用者に伝わらず，例えば，同じ名前で二つの対象を指す，名前はあるが，対象は存在しない，名前で指す対象が実際には異なる対象を指す等の現象が生じている時に，相手側がその時持っている変更前の情報を基に通信を始める場合である．資源の更新がなければ，各場所の情報は不変なので，網の伝達遅延があっても通信は正常に行われる．資源に関する情報を持たなければ，利用することがないので問題はない．

ここで，伝達遅延問題を利用者から見えなくし，分散透明性を支援するため，正確性という概念を導入する．正確性を，(1) 利用者が手元の情報に基づいて通信を行うと，通信可能ならば，通信情報は相手に正しく届き，通信不能ならば，その旨を利用者に知らせ得る，(2) 間違いアクセスの場合はこれを防ぐ機能があることと定義する．

利用者に関係する不正アクセスには，利用者が名前を間違えた場合と利用者が相手のアクセス権を誤った場合がある．前者は名前管理で名前付領域を調べることで解決する．後者のアクセス権はケーパビリティを利用し，名前登録時，利用者間のケーパビリティを記述名に含めることで解決を図る．従来手法として，アクセス資格チェック用のケーパビリティの概念[3] がある．主側 (主体) か，従側 (対象) にケーパビリティ・リスト (各々，C-list, A-list と呼ぶ.) を置けば十分であるとされている．ところが，一方のみに置くと伝達遅延の影響等を受けるので冗長化して C-list と A-list の両方を置く．ケーパビリティ自体を暗号化する方式もある[9] が，ケーパビリティは正しいと仮定する．

不正アクセス対策を含めると，利用者の通信に必要な名前管理表は，C-list, A-list, Und, Pnd, Cnd, Dnd, これらの間の写像，となる．この時，識別名を用いる

場合の参照順は，C-list，Und/Pnd（プロセス対のため），Und/Pnd から Cnd への写像，自ホスト Cnd，相手ホスト Cnd，Cnd から Pnd/Und への写像，Pnd/Und，A-list となるはずである．しかし，更新中は上記の問題点にもあるように，これらの管理表の更新は伝達遅延のため少しずつ遅れて行われるので，全体として眺めると，新，旧の情報が混じり合っている．このような環境では表同士の値が食い違い，必ずしも参照順に通信を進めることができない．そのため，プロセスの名前管理では，参照に関する制御が必要になる．一方，名前付領域に登録してある相手の名前の更新はプロセス対の相手の管理者（名前管理プロセス）からのみになるので，更新の同期制御は不要となる．そこで，ケーパビリティ・リストだけでなく，名前付領域中の名前にも状態をつけ，名前の参照，変換時に名前の状態値により，通信続行の可否を判断する方式を採る．

名前管理表を抽象化して，名前確認点 n と呼び番号 n は端から順に 1～8 と付ける．名前確認点における名前付領域の名前の状態は，normal, empty, moved, delete, moving, binding とし（名前の状態の状態遷移は**図 2** に示す．），ケーパビリティと名前変換写像の状態は（以後，これも名前の状態と呼ぶ．）True, False を仮定する．

名前確認点での判断基準を，次のように決める．通信方向を示すため，通信情報の属性に，通信が，進みか戻りの区別と名前確認点番号を加える．ただし，進みは相手側に向かい，戻りは要求提出側に戻る向きで名前確認点を一つ移動することとする．進みの条件は，名前の状態が normal, moved, binding または，True である．戻りの条件は，moving, delete, empty または False である．moved の意味は，対象が移動してきたホストでは，対象が存在し，名前管理表に書き込み処理中であることを示す．他ホストでは，利用者には場所の移動を知らせる必要がないので，moved は，Cnd のみに必要な状態であり，名前管理に転送先が変わったことを示す．moving は対象の移動前のホストで，対象が移動処理中である時に用いる状態で，特に Cnd で moving の場合は通信情報を移動先のホストに移す処理を行う．

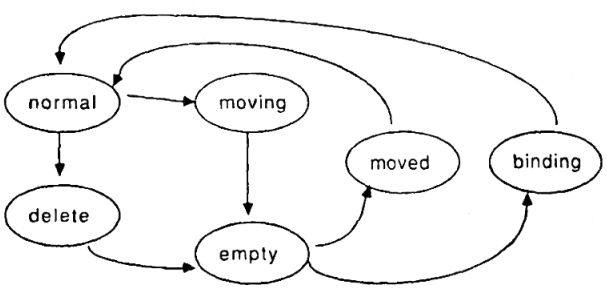

図 2 名前確認点における名前の状態の状態遷移
Fig. 2 State transition of a name in the naming domains.

通信時，名前管理では，名前確認点で名前の状態を参照し，進みか戻りの判断をし名前確認点を移動する．これらを 4 章のモデルで確認する．

3. 分散処理における名前管理モデル

3.1 プロセス間通信を含めた名前管理構成のモデル

プロセス間通信はプロセスの間でメッセージ（以下，通信情報と呼ぶ）のやりとりをするという単純なモデルを仮定する．ケーパビリティによるアクセス権はプロセス対で決まるので，通信は一対一のみを考える．

プロセス間通信を支援するため，各ホストに核プロセスとして，作り付けの名前管理プロセスと通信管理プロセスを仮定する．すなわち，名前管理は各ホストの自律性を重視し，分散管理とする．このプロセスの識別名はあらかじめ与える．

名前管理システムに関連するプロセス間通信処理は，プロセスが動的との仮定から，（1）利用者が名前管理プロセスに，ある機能を持つ対象の存在の問い合わせをする，（2）利用者同士でメッセージの交換をする，（3）利用者が他のプロセスの存在状態の変化（生成，消滅，移動）を名前管理プロセスに知らせる，（4）対象の名前の登録，結合時に名前管理プロセス同士の通信で処理すること，等の種類が挙げられる．これらは，対象が変化したため名前の状況を変える名前更新管理（(3), (4)）と名前を利用する時の名前利用管理（(1), (2)）に分けられる．

利用者に対応する利用者プロセスを置き，(2)は利用者プロセス同士の通信形態とする．Und には通信可能な相手プロセス名を利用者が用いる識別名の形で書く．

3.2 名前管理のモデル

名前管理は名前確認点の参照や更新をし，次の名前確認点へ離散的に移るという形にモデル化できる．

通信時は，利用者が通信のため提示した相手

の名前に基づき，2.4 節の判断基準に従って，通信情報を進める．通常の名前変換の手順は図1のとおりである．名前更新管理のうち，削除と移動は，図4のようである．

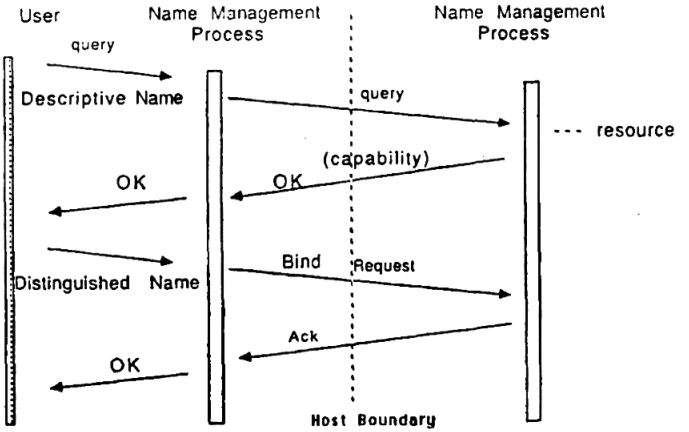

図 3 名前の登録と結合手順
Fig. 3 Procedure of registering and binding names.

対象の存在確認は記述名を用いて，図3のように名前管理プロセス間で問い合わせを行い，この二つのプロセス間で，通信アクセス可能ならば，各々の名前を結合する．その結果，図1の関係が成立する．名前管理では，名前の登録，結合時に名前付領域内の名前の一意性チェックも行う．この部分は，次章で述べる問題点の外に当たるので，名前確認点モデルでは扱わない．

4. 検　　証

4.1 検証の対象

更新で問題となるのは変化情報が相手側に伝わる前に相手から通信が生ずる場合である．そこで，名前管理プロセスの名前更新管理中（削除，移動）に，相手から通信要求が生じた場合の状況を名前確認点のモデルで調べる．名前確認点は更新側のホストにある名

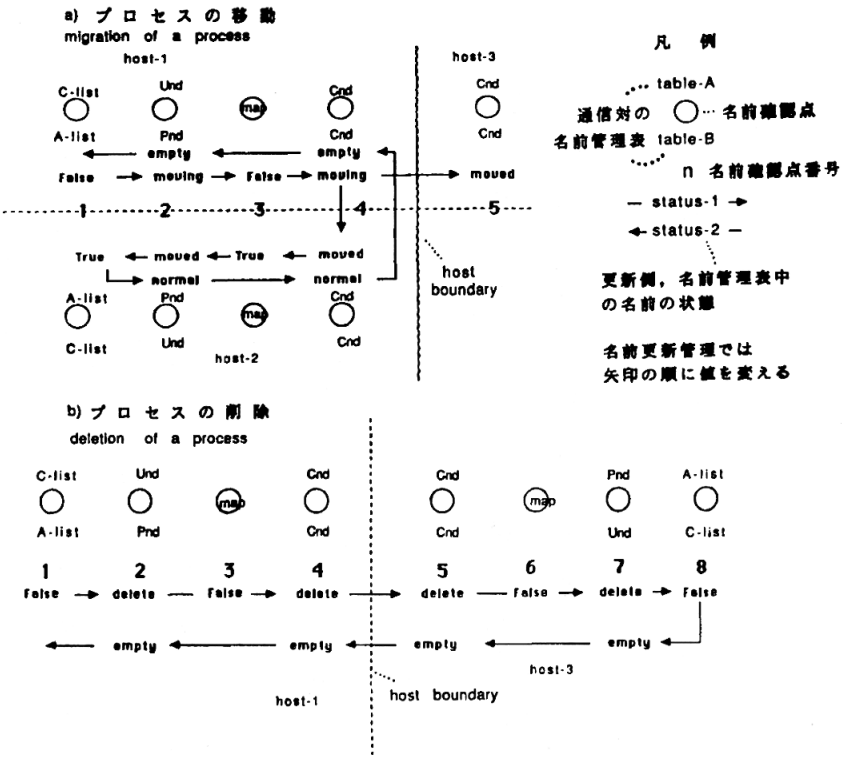

図 4 更新処理と名前確認点
Fig. 4 Update procedures and checkpoints of name status (CPNS). CPNSs are shown in bold numbers.

前管理表を名前確認点1～4，通信要求側のそれを5～8とする(図4)．他の場合，すなわち，名前と対象（両方向）の結合時に，処理済み側から通信要求が生じた場合は，更新と同様に扱える．また，対象の問い合わせやその登録については，この処理中に対応相手からの通信は生じないので，問題ない．

4.2 検証モデルと検証プログラム

移動モデル：ホスト1のプロセスAがホスト2に移動中に，ホスト3のプロセスBからAに通信要求が生ずる．

削除モデル：ホスト1のプロセスAを削除中に，ホスト3のプロセスBからAに通信要求が生ずる．

Bからの通信は，名前確認点8から，2.4節の判断基準に従って進む．

名前更新管理は，名前確認点1から4まではこの番号順にプロセスAの名前の状態を更新する（更新値は図4の値である．）．続いて，移動の場合はホスト3のCndの名前を移動先の名前に書き換え，この名前の状態をmovedとし，プロセスAがホスト2に移動した時点でホスト2で必要なAとBの対の名前管理表を作る．削除はホスト3のプロセスAの名前の状態を更新する．最後に，ホスト1に戻り，更新を完成させる．

検証プログラムでは，名前確認点を処理の単位として，通信の名前利用管理と名前更新管理が，同時開始の場合と通信を遅らせた場合を調べる．各々の名前確認点において，必ず更新処理をし，その時点で法 n ($n=2, 3, 4$) による乱数を発生させ，1が出た時には，(逆向きの) 通信処理も行うことにすると，更新速度と通信速度の比は，$n:1$ と見ることができる．この関係を反対にすると，更新速度と通信速度の比は，$1:n$ になる．

名前確認点4と5の間はホスト境界であるが，ここでの遅延は特に加えない．

4.3 結　　果

図5に移動と削除の典型的な結果を示した．図5の縦軸は時間の経過，横軸は名前確認点である．名前利用管理での通信を進める判断を，プロセスBからの通信として，折線で示した．名前更新管理で更新を行った時は，更新後の名前の状態をその時刻の線分上に記した．これを明記していない場所の名前の状態は，ホスト1，3では normal または True を，ホスト2は empty または False を仮定した．

一例として，移動で速度比1：3の場合を見る．名前確認点8から3までは normal なので，プロセスBからの通信情報は進む．ところが名前確認点2では moving なので戻りになる．名前確認点4も，moving なので通信情報をホスト1からホスト2に転送する．しかし，ホスト2では，名前更新管理の方が遅く，名前確認点4の名前管理表が empty なのでホスト3に戻る処理を行っている．

更新速度と通信速度の関係は，(1)同時に始めた場合は，i)更新速度が通信速度より速いと更新が進み，通信は行先変更の判断が名前確認点5～8，すなわち，自ホスト内で可能である．ii)通信速度が更新速度より速い場合，未更新が続くので，相手ホストに達してから更新状況が分かる．(2)通信速度が更新速度より速い場合でも，通信開始が遅い場合は上記 i) と同様に判断できる．

対象削除の場合，名前確認点の名前の状態は delete か，False なので通信は必ず戻りになる．対象移動で，移動先での名前更新処理が通信より遅い場合と，ホスト1内で通信が戻りになったが，更新処理の遅れのため，名前確認点4が moving でない場合は，通信が行先不明として戻される．それ以外は，移動先に転送される．

5. 考　　察

5.1 名前付領域について

一般に n.d. の概念が現状のように OSI 基本参照モデルのネットワーク層のレベルでしか存在しない場合は，利用者は各種の情報を明示して，階層化された長い文字列の識別名を使う必要がある．例えば，OSI 基本参照モデル ISO/DIS 7498-3 の (N)-PAI では，(P-selector, S-selector, T-selector, list of network addresses) 形式を用いている．

利用者に分散透明性を提供するためには n.d. を複数用いて網向きに階層化された構造を隠す必要がある．ここで提案した多層 n.d. は OSI 基本参照モデルの応用層やローカル環境に置くものであり，用途別に名前形式を可変とし，このことを利用して分散透明性を図るものである．2章で述べた構成法で，もし，Cnd のみの構成であると，(1) 名前付けを利用者間で必ず共通にしなければならない，(2) 論理的な場所情報を含める必要がある，という制約が生じる．(1)では利用者が用いる名前付け方式が制限され，(2)は対象が移動した時には全利用者にその旨を知らせる必要がある．Cnd とともに Und を設けると，上記(1)，(2)の問題が解決する．すなわち，(2)は対象

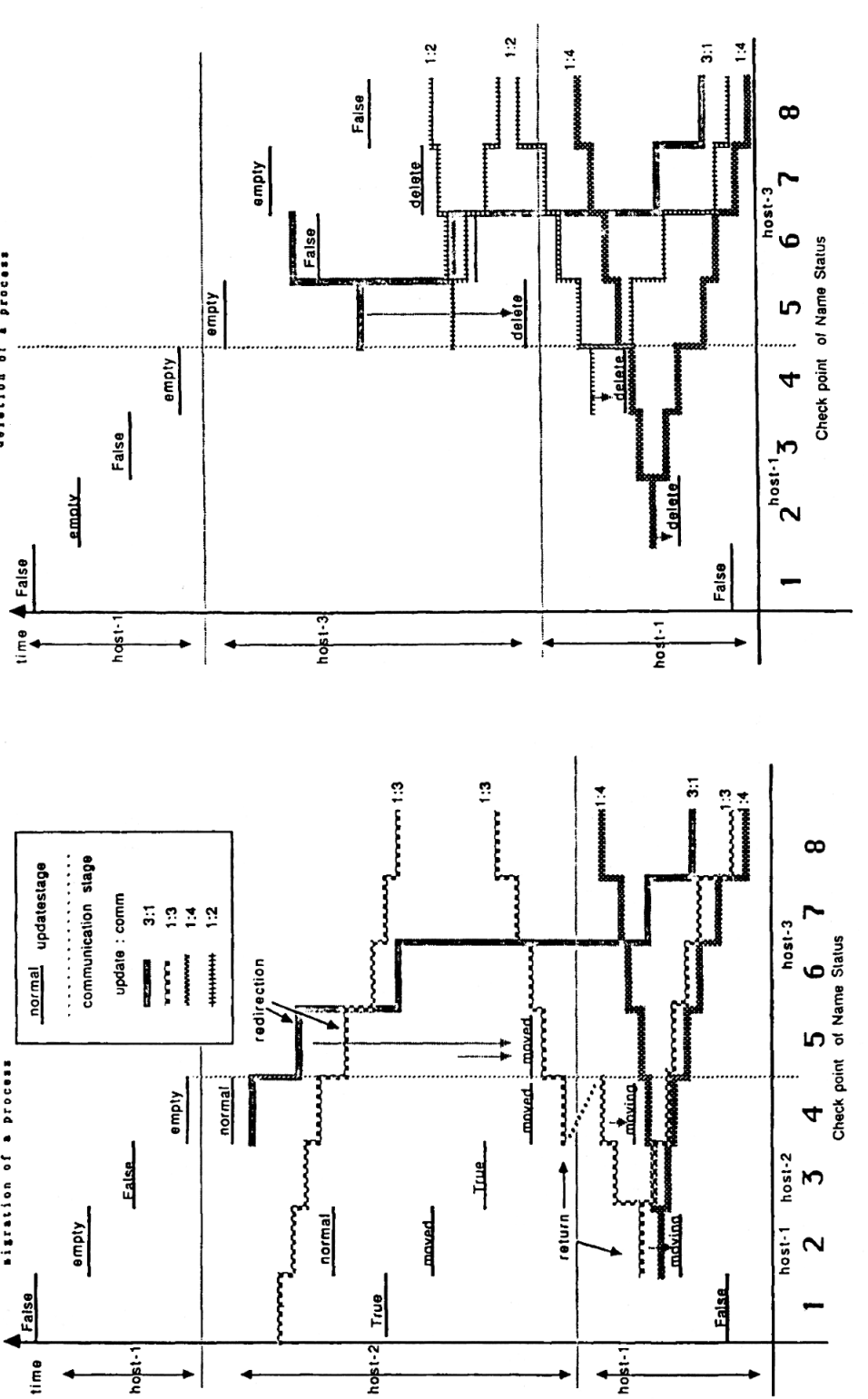

図 5 更新と通信がともに生じた場合の処理
Fig. 5 The name-management processing when a communication from the peer occurs in the execution of updating of an object.

の移動情報は Cnd のみの変更で済み，Und の中の名前は変える必要がない．利用者にとっては分散透明性が図れる．(1)は利用者に独自の名前付けを可能にする．現在，計算機上の使用文字は暗黙の了解として英単語ということになっており，また，これに慣らされて，特に疑問を持たない状況にあるが，Und の概念を用いると，利用者の文字の種類を変えることもできる．例えば，Und に対しては漢字，カナの名前付けも可能になり，分散透明性以外の利用価値も生ずる．Und と Cnd があれば十分とも見なし得るが，分散環境においては，ホストの自律性，局所性は重要である．このため，ホスト内の識別名管理用の Pnd も必要である．

以上の n. d. はすべて，識別名 n. d. である．識別名は一文字間違えただけでも利用不可となる．この場合，別の手段で対象を利用できる方法が必要である．例えば，一対一の通信で，通信相手にワイルドカードが許されない時，通信相手の名前が一部でも曖昧であると，完全に通信不能になる．この時に，相手の機能を述語として指定し，通信できるようにすると救済になる．資源の存在の有無を知るため，自然言語や機能キーワードを用いた問い合わせ機構も必要である．このような時は，資源の機能問い合わせ用に記述名の n. d. (Dnd) が共通 n. d. として必要になる．以上の考察から，四種の名前付領域は分散透明性実現には最小限必要な構成要素と考える．

現在稼動している網 Junet のアドレス形式は，ここでの提案における Cnd に対応すると見なせる．Junet では，これをそのまま利用者に見せているので，上記二つの制約を強く意識しなければならない．Und の構造を加えると便利であろう．

5.2 正確性の実現

正確性の概念は，分散環境で，利用者の持つ通信用の情報は（見かけ上，常に）正しいことを保証するための利用者への必要性であり，データ通信における信頼性，確実性などのシステムに関する定義とは異なる．

名前確認点に名前の状態を入れて通信の向きを判断し，利用者の不正使用に対してはケーパビリティを用いて正確性の実現を図った．このため，対象の更新時に，通信が生じた場合，この通信は，通信可能ならば，利用者には更新状態を知らせずに実行でき（例，移動），通信相手が消滅する場合等，通信続行不能な更新の場合のみ利用者への通知が可能になった．

分散環境では，更新や通信の生ずる時刻はまちまちなので，通信続行の判断を種々の状況で行う必要がある．利用者に分散透明性を提供するために，名前確認点の数を多くしたため，これが可能になったと考える．

6. おわりに

網上のプロセス間通信を抽象的に捉え，これを考察対象とした．利用者に必要な名前の種類の検討と，利用者に名前で分散透明性を実現可能にするための構成法の検討をし，利用者には識別名と記述名が必要なこと，本文で導入した名前付領域が最低四種類必要なことが分かった．名前付領域を組み合わせた構成法に基づいて，網上では避けられない伝達遅延対策として必要な名前管理法を構成し，プロセスの更新時に通信が生じた場合においても対処可能であることを，名前確認点モデルを用いて，確認した．

今後の課題は，ここで用いたモデルの厳密化を行いより詳細な構成法の検討を行うこと，利用者に必要な名前として提案したが，ここでは考察しなかった記述名や名前登録管理法の具体化と，プロセス間通信以外の分散資源の名前管理構成法への拡張等，である．

参考文献

1) Anderson, D. P., Ferrari, D., Rangan, P. V. and Tzou, S.-Y.: The DASH Project :Issues in the Design of Very Large Distributed Systems, Report No. UCB/CSD 87/338, p. 30 (1987).
2) Birrell, A. D.: Secure Communication Using Remote Procedure Calls, *ACM Transactions on Computer Systems*, Vol. 3, No. 1, pp. 1-14 (1985).
3) Dennis, J. B. and Van Horn, E. C.: Programming Semantics for Multiprogrammed Computations, *CACM*, Vol. 9, No. 3, pp. 143-155 (1966).
4) Hauzeur, J.: A Model for Naming, Addressing, and Routing, *ACM Transactions on Office Information Systems*, Vol. 4, No. 4, pp. 293-311 (1986).
5) Jacobs, K.: OSI Addressing Strategies, *Computer Communications Review*, Vol. 17, No. 3, pp. 7-12 (1987).
6) Oppen, D. C. and Dalal, Y. K.: The Clearinghouse : A Decentralized Agent for Locating Named Objets in a Distributed Environment, *ACM Transactions on Office Information Systems*, Vol. 1, No. 3, pp. 230-253 (1983).
7) Shoch, J.: Inter-network Naming, Addressing

and Routing, *Proc. IEEE Computer Conference COMPCON 78 fall*, pp. 72-79 (1978).
8) Tanenbaum, A.S. and Renesse, R.V.: Distributed Operating Systems, *Comput. Surv.*, Vol. 17, No. 4, pp. 419-470 (1985).
9) Tanenbaum, A.S.: Using Sparse Capabilities in a Distributed Operating System, *The 6th International Conference on Distributed Computing Systems*, pp. 558-563 (1986).
10) Information Processing Systems—Open Systems Interconnection Reference Model—Part 2: Security Architecture, ISO/DIS 7498-2 (1987).
11) Information Processing Systems—Open Systems Interconnection—Basic Reference Model—Part 3: Naming and Addressing, 1987. Final Text of DIS 7498-3, ISO/IEC JTC 1/SC 21 N 2872 (1988).

(昭和63年3月14日受付)
(昭和63年7月15日採録)

古宇田フミ子（正会員）

昭和27年生．昭和49年お茶の水女子大学理学部数学科卒業．東京大学工学部電気工学科に勤務し，分散処理に従事している．電子情報通信学会会員．

田中　英彦（正会員）

昭和18年生．昭和40年東京大学工学部電子工学科卒業，昭和45年同大学院博士課程修了，工学博士．同年東京大学工学部講師，昭和46年助教授，昭和62年教授．昭和53年～54年ニューヨーク市立大学客員教授，現在に至る．計算機アーキテクチャ，並列推論マシン，知識ベースマシン，オブジェクト指向計算システム，分散処理，CADなどの研究を行っている．「計算機アーキテクチャ」，「VLSI コンピュータ I，II」（いずれも共著），「情報通信システム」著．電子情報通信学会，人工知能学会，日本ソフトウェア科学会，IEEE，ACM 各会員．