

M-53 ピア・ツー・ピアネットワークのためのボトムアップ結合可能な名前サービス A Bottom-up Combinable Name Service for Peer-to-Peer Network

上野 裕一†
Yuichi Ueno

1 はじめに

最近、Peer-to-Peer(P2P) ネットワーク [1] による分散処理が大きく発展しつつある。P2P ネットワークは、中央の制御機構を持たず、ネットワーク・ノード間をアドホックに連係し計算機リソースを互いに利用しあう。このような処理手法は、自発的なコミュニティ形成や、ネットワーク帯域・計算機リソースのより有効な利用を可能にする [2]。

しかしながら、現状では、P2P ネットワークの構成は一時的である。P2P ネットワークをグループウェアなどのユーザ間に長期的な関係をもたらすアプリケーションに適用するためには、アドホックに構成されるネットワークの上に、グループワークのライフサイクルの期間を通して、関連する計算機リソースを恒常的に利用可能な状態に維持するための様々な高位サービスを必要とする [1]。そのような高位サービスの一つは名前サービス/ディレクトリサービスである。

従来の分散型名前サービス/ディレクトリサービスである DNS(Domain Name System)、LDAP[3]、Spring Name Service[4] では、複数のサービスが、大域的な単一名前空間を階層ごとに分割して管理している。このような管理モデルは、全体の一貫性を持つように予めトップダウンに設計された上で運用されねばならないため、P2P ネットワークのアドホックな構成方法とは本質的に相容れない。

本論文では、完全非集中制御のもとで局所名前空間をボトムアップに合成できる特徴を持った新しい分散型名前サービスを提案する。我々はこの名前サービスを、共同作業指向ネームサービス (Collaborative Work Oriented Name Service; 以下 CWONS) と名付ける。CWONS は全体的な一貫性や中央の制御機構を前提とせず機能するので、P2P ネットワークと親和性が高く、P2P ネットワークの利用やグループウェア・アプリケーション構築に多くの恩恵をもたらす。本論文では CWONS の基本設計および Java プラットフォーム [5] 上のプロトタイプ実装について述べる。さらに、CWONS の利点について考察する。

2 設計

2.1 分散アーキテクチャ

CWONS は P2P ネットワークにおける分散型名前サービスであるので、一定範囲のピア・ノードが管理するオブジェクトに対する名前付けに責任を持ちつつ、他の CWONS と連携して動作する。左記の責任範囲をドメインと言う。

我々は CWONS 連携の設計原則として、複数の CWONS 同士は主従関係を持たずに対等に動作するべきであるという指針を設定する。これは前節で考察したように、P2P ネットワークでは事前に大域的な構造を前提とできないことに対応するためである。

さらに、我々はこの指針を次の 2 点の機構に展開した: (1) ドメイン間連携の局面で横断ドメインを設定し、横断ドメインでは関連するドメインそれぞれの局所名前空間の一部を切り出して合成した名前空間を生成する (2) 横断ドメインの設定操作は、関連する CWONS 同士が対等に結合する完全な非集中制御で行う。前者については、第 2.2 節に、後者については、第 2.3 節において説明する。図 1 は CWONS を含む分散シス

テムの構成を表す概念図である。

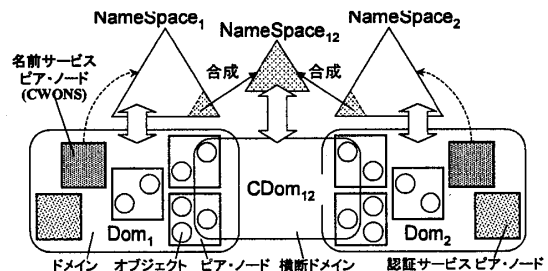


図 1: 局所名前空間の対等合成による名前空間ボトムアップ生成

なお、アプリケーションのセキュリティ上の要請により、名前サービスはしばしばアクセス制御機能を備える必要がある。CWONS においても、ドメイン内に認証サービスを提供するピア・ノードがある場合には、それを利用できる。

2.2 名前空間と基本操作

CWONS における名前空間はネーミングコンテキストの有向グラフによる階層的な名前の集合である (図 2)。

CWONS は通常の名前サービスと同様に、ネーミングコンテキストに対する基本操作として *bind*, *unbind*, *lookup*, *list*, *createSubcontext* の各操作を実行できる。これらの操作はそれぞれ、オブジェクトを結合する、オブジェクトの結合を解除する、結合しているオブジェクトを照会する、結合しているオブジェクトの名前を列挙する、新しいサブコンテキストを生成して結合する、という処理を行う。

CWONS では特定のネーミングコンテキストを横断ドメインにおけるルートとし、横断ドメイン上の各ルート・ネーミングコンテキストを経由して横断的に名前解決を行うことで、複数の局所名前空間を単一化する (図 2 における C_{12} , C_{22})。こうした横断名前解決を設定したネーミングコンテキスト以下の部分名前空間上では、上記各基本操作はドメインを横断して働き、他のドメインにあるオブジェクトを自分のドメインにあるオブジェクトと同様に操作することが出来るようになる。なお、同一の名前に複数のオブジェクトが対応する場合の競合解消については本稿では割愛する。

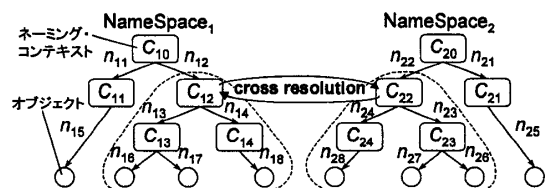


図 2: 局所名前空間のドメイン間横断による名前解決

2.3 横断ドメイン設定操作

横断ドメインの設定操作は、横断ドメインを構成しようとする各 CWONS 間で相手 CWONS の ID を事前に共有しておくことで、図 3 に示す 3 フェイズからなるプロトコルを用いる非集中制御により達成することができる。

フェイズ 1: 横断ドメイン生成 各クライアント (一般的なピア・ノードが実装する) は、ユーザ操作などにもとづいて、横断ドメインを構成しようとする全ての CWONS の ID 集合 $\{X, Y, Z\}$ と、横断ドメインのルートとなるネーミングコンテキストの ID ($Root_X, \dots$) を引数に、CWONS に対し

†富士ゼロックス (株) 中央研究所, Corporate Research Center, Fuji Xerox Co., Ltd.

て join 操作を要求する。各 CWONS はこの操作に対して、横断ドメイン $ID(CD_x, \dots)$ および横断ドメイン用認証キー ($Auth_{yx}, Auth_{zx}, \dots$) を生成する。さらに、各 CWONS はこれらデータをまとめて横断ドメイン結合要求メッセージ ($M_{xy}, M_{xz}, \dots$) を生成する。この操作は各 CWONS が非同期的に実行する。

フェイズ 2: 横断ドメイン結合 各 CWONS は第 1 フェイズで生成した横断ドメイン結合要求メッセージを他の CWONS に送信しあう。この操作も非同期的である。第 1 フェイズが終了していない CWONS に対するメッセージは処理可能になるまで送信ないし受信バッファに留め置かれる。各 CWONS は、それぞれ独立して生成した横断ドメイン ID はじめメッセージに含まれる各種情報をこのフェイズで互いに関連付ける。なお、横断ドメインの同値性は CWONS の ID 集合で判断する。

フェイズ 3: 横断ドメイン確立 各 CWONS は横断ドメインを構成する他の全ての CWONS から横断ドメイン結合要求メッセージを受信した時点で、横断ドメイン確立メッセージ ($Commit$) を送信する。さらに、CWONS は横断ドメイン確立メッセージを他の全ての CWONS から受信し終えた時点で横断ドメインが確立したものとして有効化させる。このフェイズによって、各 CWONS 間で同期を取ることになる。

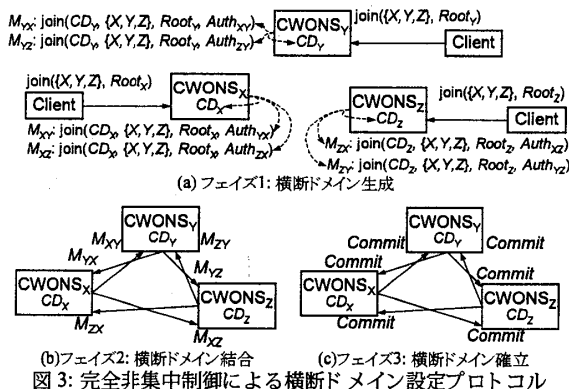


図 3: 完全非集中制御による横断ドメイン設定プロトコル

既存ドメインへの追加操作および、既存ドメインからの脱退操作 (*disjoin*) も同様の非集中制御で達成できる。

なお、CWONS では、図 4 に示すように、セッション・トークンを認証しそれをもとにアクセス制御を行うという、キーパリティ方式を採用している。CWONS は横断ドメイン用の認証キーを用い、ユーザに代行してドメイン間の認証を行う。

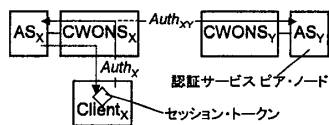


図 4: ドメインを横断するアクセス制御

3 実装

我々はプロトタイプとして、CWONS のサーバ部分/クライアント部分の両方および、独自の認証サービスを Java 2 プラットフォーム Standard Edition Version 1.3.1[5] 上に実装した。P2P ネットワークは、Java RMI[6] 上に構築した独自基盤を用いている。オブジェクトやサービス等を識別する ID は、セキュアランダムを用いて生成する乱数ベースの UUID[7] をもとに構成している。CWONS 管理下のオブジェクトの永続化は、ozone Version 1.0.1[8] を同一の Java 仮想機械上で動作させることで実現している。このような実装のため、各ピア・ノードは Java 仮想機械 1 個で動作できる。

プロトタイプでは、CWONS は個人用の名前サービスとして機能し、個人の管理下にある計算機リソースをドメインにとりまとめる。これにより、複数の計算機を所有し利用するユー

ザは、各計算機上のリソースを個人ドメインの名前空間に配置し透過的に取り扱うことができる。

さらに、CWONS の機能により、複数の個人が集まってグループワークを行う局面で横断ドメインを即興的に構成できることから、個々人がグループワークの文脈でリソースを仮想的に“切り出して持ち寄り”共有することができる。

なお、適切なラッパー・クラスを用意することで、通常ファイルのみならず OS が実行中のプロセスの内部状態など任意の計算機リソースを利用できる。これにより、アプリケーション・プログラムをグループで共同操作したり、利用状況にもとづくウェアネス提供を行う機能を実現できる。

4 考察

CWONS は P2P ネットワークをドメインに分割し、ドメイン内のオブジェクトを名前操作可能にする責任を集中的に持つ。つまり、CWONS は元来平坦な構造を持つ P2P ネットワークにおけるハブとして機能する。CWONS を導入することで、CWONS の管理下にあるオブジェクトの発見については P2P ネットワーク全域に照会を行う必要がなくなることから、ネットワークの利用効率を高めることができる。その一方で、CWONS 同士の連携については、全体的な一貫性を持つ大域的な名前空間も中央の制御機構も必要としないため、P2P ネットワーク全体の非集中処理と対立しない。

また、CWONS により、計算機リソースを大域的な単一名前空間から独立に共有可能となることから、特定の共有サーバスペースを前提とせずにグループワークを組織できるようになる。この特性により、ユーザに対してより即興的かつ自発的なスタイルでグループワークの組織化を許す、新しいグループウェアの構築が可能となるであろう。

5 おわりに

我々は本論文において、完全非集中制御のもとで局所名前空間をボトムアップに合成できる特徴を持った新しい分散型名前サービス “CWONS” を提案し、名前空間合成操作およびサービス間連携設定プロトコルを示した。CWONS は P2P ネットワークの利用効率を高めるとともに、より即興的・自発的にグループワークを組織化できる新しいグループウェア実現の端緒となるであろう。

今後の課題は、属性情報追加によるディレクトリサービスへの拡張、一時的にオフラインになるピア・ノードへの対応、および、より柔軟な P2P 基盤である JXTA[9] 上への移植である。

謝辞

古川泰之所員、藤本正和所員、増田佳弘マネージャ、堀切和典マネージャは創造的な議論を通して本技術の考案に大いに貢献した。各位に深く感謝する。

参考文献

- [1] L. Gong. Peer-to-peer networks in action. *IEEE Internet Computing*, Vol. 6, No. 1, pp. 37–39, Jan./Feb. 2002.
- [2] L. C. Wilson. Peer-to-peer computing: Finding new value. Technical Report D02-2366, SRI Consulting Business Intelligence, Feb. 2002.
- [3] M. Wahl, T. Howes, and S. Kille. *Lightweight directory access protocol (v3)*. Internet RFC 2251, Dec. 1997.
- [4] S. Radia, M. N. Nelson, and M. L. Powell. The spring name service. Technical Report SMLI TR-93-16, Sun Microsystems Laboratories, Inc., Nov. 1993.
- [5] Sun Microsystems, Inc. *Java(TM) 2 SDK, Standard Edition Documentation Version 1.3.1*, 2001. URL: <http://java.sun.com/j2se/1.3/docs/index.html>.
- [6] Sun Microsystems, Inc. *Java(TM) Remote Method Invocation Specification*, 1999. URL: [ftp://ftp.java.sun.com/docs/j2se1.3/rmi-spec-1.3.pdf](http://ftp.java.sun.com/docs/j2se1.3/rmi-spec-1.3.pdf).
- [7] Open Group. *DCE 1.1: Remote Procedure Call*, 1997.
- [8] The ozone Database Project. ozone - The Open Source Java ODBMS. URL: <http://www.ozone-db.org/>.
- [9] L. Gong. JXTA: A network programming environment. *IEEE Internet Computing*, Vol. 5, No. 3, pp. 88–95, May/Jun. 2001.