

P2P 技術を適用した画面転送・遠隔操作システムの開発

三浦 元喜[†] 志築 文太郎[†] 田中 二郎[†]

ピアツーピア (peer-to-peer, P2P) 技術を適用した画面転送・遠隔操作システム “comDesk” の設計と実装, 比較実験について述べる. P2P 技術をコミュニケーションを目的とする画面転送・遠隔操作システムに適用することにより, あらかじめサーバを準備しておく必要性がなくなり流動的な環境への適応性が向上する. また, Peer 間の対等な関係により画面転送の方向や転送開始操作を行うホストの自由度を高めることが可能となる. またこれらの効果をより高めるため, 直感的なインタフェースと利用者単位の管理機構, 転送状況をリアルタイムに把握する機構を導入した. 利用者はドラッグ&ドロップによる転送開始操作と転送画面の再転送操作を統一的な枠組みで行うことができる. comDesk が備える自由度の高い転送開始操作により, 様々な利用場面に柔軟に対応することが可能となる.

Implementing a Remote Desktop System based on P2P Techniques

MOTOKI MIURA,[†] BUNTAROU SHIZUKI[†] and JIRO TANAKA[†]

We have designed and implemented a remote desktop system based on P2P (peer-to-peer) techniques, named “comDesk”, which enables users to exchange the desktop images as well as to control remote hosts. Due to the P2P mechanisms, the users can start communication without pre-located servers. Also the equality of peers is well suited to bi-directional transfers and operations. comDesk provides a visual interface, which enables users to specify both the source and the destination hosts freely, with simple dragging operations. The users can not only understand the current transferring sessions, but also control the sessions. P2P features and our design make the system simple, and increase the usability and interoperability. Consequently comDesk is effective for a variety of situations based on desktop sharing.

1. はじめに

ノートパソコンなどの小型で可搬性に優れた計算機の普及によって, 固定した場所にとらわれずに計算機を移動して作業を行う環境が容易に実現されつつある. 例えば, 作業用の机があるオフィスからミーティングルームに移動して会議を行ったり, 出張先において一時的な会合を持つことは日常的に行われている行為である.

このような状況において, その場に集まっている計算機間での情報交換を円滑に行うことは, 会合そのものを効率よく進める上で重要であるため, 会議支援システムとして様々な研究が行われている. 例えば, スタンフォード大の Interactive Workspace Project¹⁾ では, 会合を支援するため大画面表示装置を備えた iRoom²⁾ と呼ばれる部屋を構築し, EventHeap³⁾ と呼ばれる通信基盤を利用した様々な研究を行っている. しかし,

iRoom では, EventHeap を処理するサーバが, あらかじめ環境に配備されている必要があるため, どこでもすぐに利用できるというわけではない.

我々は人間が移動可能な計算機を携えて一時的に集まるといった「流動的な環境」においても, 利用者間の情報交換を円滑に行えるようにすることが重要であると考えている. ここで述べる「流動的な環境」とは, 計算機同士が相互に接続するためのネットワークは存在するが, 場所が不定で固定的な資源を想定できない環境を想定している. このような「流動的な環境」において計算機と利用者が参加する「場」を構成し, 情報交換を円滑かつ効果的に行うため, 我々はピアツーピア (peer-to-peer, 以下 P2P)^{4,5)} 方式に着目した画面転送・遠隔操作システム comDesk (communicable Desktop system) の研究を行ってきた. 本研究の中途結果についてはシンポジウム⁶⁾ や国際会議⁷⁾ で発表済みであるが, 本論文はこれらの内容を整理するとともに, P2P 導入の効果や比較実験の記述を追加したものである.

comDesk を用いることにより, 利用者は計算機を

[†] 筑波大学 電子・情報工學系
Institute of Information Sciences and Electronics
University of Tsukuba

「流動的な環境」に気軽に参加させることができ、画面転送・遠隔操作に基づくコミュニケーションを自由に行うことができる。画面転送は直感的なインタフェースによって開始することができ、転送状況の確認や調整についても随時行うことができる。また個人が複数の計算機を同時に使用している場合に、comDeskを用いた画面転送に基づく情報交換や遠隔操作によって、複数の計算機を有効に利用できる。

2. P2Pの機構を画面転送・遠隔操作システムに導入する効果

P2Pの定義は広く、実装にも様々な形態がある。Schoderらは「2つもしくはそれ以上の対等なPeerにより構成されるネットワーク上で、適切な情報と通信システムを使って自発的に協調（コラボレート）することを可能とする技術」がP2Pの共通点であると述べている⁴⁾。またMilojicは「非集中化の方式（decentralized manner）により、重要な機能を達成するために分散された資源を使用するシステムやアプリケーションの種類」をP2Pと定義している⁵⁾。実装としては、他のPeerを発見したり必要な情報を保持するために特定のサーバを用いる方式（Hybrid P2P）や、特定のサーバを必要としない方式（Pure P2P）などがある。特に後者の方式に基づくシステムには、Gnutella^{*}やGroove^{**}など、必要な情報をPeer間で保持する機構を備えているものもある。我々はこの「必要な情報をPeer間で保持」する機構と「Peer間の関係を対等にする性質」を画面転送・遠隔操作システムに適用することにより、以下の2つの効果が得られると考えている。

2.1 流動的な環境におけるコミュニケーションの促進

コミュニケーションを行う場を形成するには、場に参加している計算機に関する情報を格納する機構が必要となる。固定的な環境においては、この情報を格納する専用のサーバを配置しておくことが可能である。しかし、流動的な環境においては、場が不定であり固定的な資源を想定できないため、コミュニケーションを行う場を形成する前にサーバを準備する必要があった。

我々は「必要な情報をPeer間で保持する機構」を適用することにより、あらかじめサーバを設置しておくなくても済むようにする。また、利用者がサーバのIPアドレスを調べたり、指定したりする作業から開放

されるようにするため、自律的に他のPeerを発見し設定を行う機構についても適用することにする。

これらの機構を適用することにより、利用者はコミュニケーションを行う場を簡単に形成できる。また、利用者はコミュニケーションを行う場に計算機を簡単に参加・脱退させることが可能となる。これらの利点により、固定的な環境に加えて、流動的な環境におけるコミュニケーションの促進が期待できる。

2.2 多様な利用場面や運用形態への対応

2台の計算機（a, b）が場に参加している状況において、行われる画面転送には、「aの画面をbに転送する」場面と「bの画面をaに転送する」場面の2種類がある。また、計算機（a, b）の利用者を（A, B）とすると、画面転送を開始する操作（転送開始操作）を行う利用者によって、「Aが転送開始操作を行う」場面と「Bが転送開始操作を行う」場面に分類される。このように、画面転送が行われる状況は、転送方向や転送開始操作を行う利用者によって異なる。場に参加する計算機の台数や利用者が増加した場合、考えられる利用場面はより多様になる。

VNC（Virtual Network Computing）⁸⁾や、市販のパッケージソフトウェア^{9),10)}など、画面を送信するサーバと、受信するクライアントが別々のプログラムとして設計されている画面転送ソフトウェアの場合、画面転送の方向が「サーバの画面をクライアントに転送する」方向に限られている。また転送開始操作についても、操作手順および操作を実行するホストが限られている。例えばVNCの場合、画面転送を開始するには（1）起動しているサーバにビューアで接続する（2）Listen Modeにて起動しているビューアに、起動しているサーバからAdd New Client機能により接続するの2つの方法がある。前者は別ホストの画面を要求する際（利用者が操作するホストと転送画面を表示するホストが同一の場合）に使用し、後者は別ホストに自ホストの画面を表示する際（利用者が操作するホストと転送画面を表示するホストが別の場合）に使用する。VNCを利用して多様な利用場面に対応する場合、利用者がサーバやビューアの起動モード等を選択し、操作を行うホストや起動する順序を考慮する必要がある。

P2PにおけるPeer間の対等な関係を画面転送ソフトウェアに適用することにより、転送方向の制限がなくなる。そのため、転送開始操作の手順を簡潔にすることが可能となり、利用者が起動するホストや順序を意識せずに済むようになる。また、転送開始操作を行うホストについての制限を緩和させることが容易となる。これらの利点により、多様な利用場面や運用形態

^{*} <http://gnutella.wego.com/>

^{**} <http://www.groove.net/>

に対応しやすい、自由度の高い画面転送ソフトウェアが実現できる。

3. 画面転送・遠隔操作コミュニケーションツール “comDesk” のインタフェース設計

P2P の機構を導入することにより、画面転送・遠隔操作システムを多様な利用場面や運用形態へ対応させやすくなる。しかし、実際の利便性を高めるには、P2P 機構の導入に加えて、インタフェースの設計が重要となる。

3.1 制限が少なく直感的な転送開始操作

転送開始操作においては、転送に関わる 2 つのホスト（転送元、転送先）を直感的なインタフェースを用いて指定できるようにする。これにより、利用者は各ホストにおける設定や起動モード等を意識せずに転送開始操作を行うことが可能となる。

また、転送開始操作については、特定のホストに限定せず、全ての参加ホストから行えるようにする。ただし、各利用者の画面情報が第三者によって無条件に取得・転送されることを防ぐため、ホストとその利用者名を対応づけた「利用者単位の管理機構」を導入する。実現方法の詳細については 4.3 で述べる。例えば、ある利用者が複数のホストを参加させている場合、それらのホスト間の転送開始操作は、その利用者が使用している任意のホストから行うことができるようにする。ホスト単位ではなく、利用者単位の管理機構により、多様な利用場面における利便性を向上させることができる。

3.2 転送状況の把握

利用者は、場に参加しているホスト間の通信状況をいつでも把握できるようにする。利用者が取得できる通信状況としては、「どのホストの画面のどの部分が、どこに転送されてどう表示されているか」を知ることができるようにする。これにより、転送元ホストの利用者は、自分の画面がどこに、どのように転送され表示されているかを認識することができる。通信状況は刻々と変化する可能性があるため、実際の状況をリアルタイムに反映する。

3.3 転送後の調整機能

画面転送ソフトウェアには、画面を転送する機構に加えて、転送先の利用者が転送元のホストを操作する機構（遠隔操作機構）が一般的に備わっている。個人で利用する場合は問題になりにくいですが、複数の利用者が参加している場合、画面転送と遠隔操作の状態を、それぞれ制御できることが望ましい。

そこで、転送元ホストの利用者が、画面転送の中断、

遠隔操作の中止、転送先の変更といった操作をいつでも行えるようにする。これらは各画面転送について選択的に行えるようにする。

4. 画面転送・遠隔操作コミュニケーションツール “comDesk” の実装

先の設計に基づき開発した画面転送・遠隔操作システム “comDesk” の機能と実装について述べる。基本的な機能として、転送元ホストの画面情報を転送先ホストにウィンドウ（comDesk Window）として表示する画面転送機能と、comDesk Window にて行われた操作を転送元ホストにて実行する遠隔操作機能を備えている。利用者は、comDesk commander（図 1）を用いることによって通信状況を把握したり、転送開始操作を行ったりすることができる。

4.1 画面転送インタフェース（comDesk commander）

comDesk commander では、参加している計算機をアイコン（ホストアイコン）を用いて表現するとともに、現在の通信状況を視覚的に表現する。図 1 に、ホスト zidane のデスクトップ右上に表示されている comDesk commander のウィンドウを表す。図 1 では、3 台の計算機（phobos, wakaba, zidane）において、comDesk が起動しており、表 1 に示す 3 つの転送が行われていることを示している。ホストアイコンは、各ホストのデスクトップ画像を縮小したものをサムネイルとして表示する。また、画面転送が行われていれば、その状況を表すアイコン（ウィンドウアイコン）を表示する。それぞれの転送は、ウィンドウアイコンと転送元領域表示（黄色またはオレンジ色の矩形）と転送方向表示（黄色またはオレンジ色の矢印）の 3 つによって表現されている。これらの 3 つの情報については、リアルタイムに更新が行なわれる。

操作権状況は、図 2 に示すウィンドウアイコンの枠と内部の色によって表現され、画面更新状況は、転送元領域表示と転送方向表示の色（黄色なら非更新、オレンジ色なら更新中）によって表現される。利用者は、comDesk が動作している任意のホストにおいて、comDesk commander を表示させ、通信状況を確認することができる。

また、利用者は comDesk commander を用いて、以下の操作を行うことができる。

4.1.1 転送開始操作

転送開始操作は、ホストアイコンをドラッグ&ドロップすることによって行なう。正確には、転送元ホストアイコン上でポイントし、ホールドしたままドラッグ

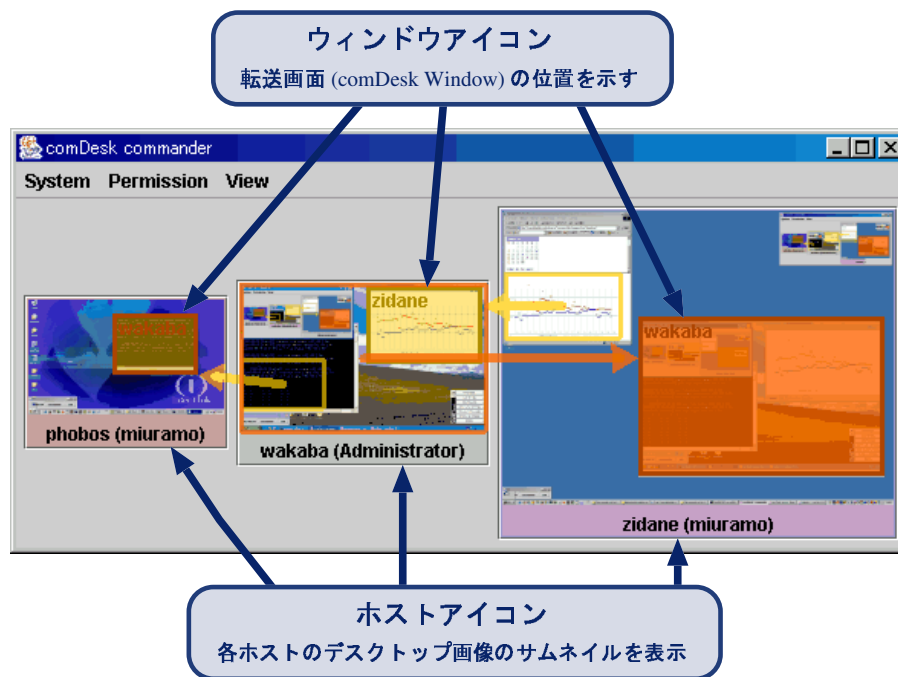


図 1 comDesk commander
Fig. 1 comDesk commander

表 1 図 1 によって表現されている画面転送
Table 1 Transfer Sessions represented in Fig. 1

転送元	転送先	転送領域	操作権状況	画面更新状況
wakaba	phobos	部分	操作可能	非更新
zidane	wakaba	部分	操作権なし	非更新
wakaba	zidane	全体	操作中	更新中



図 2 操作権状況とウィンドウアイコンの表現
Fig. 2 Representation of WindowIcon by control status

して転送先ホストアイコン上でリリース（ドロップ）する。ドラッグ&ドロップによる転送の指示は自ホストを含む、任意の参加ホスト間で行うことができる。転送開始操作によって、画面転送が開始され、転送先ホストに転送元ホストのデスクトップ画像がウィンドウ（comDesk Window）として表示される。

4.1.2 転送画面の操作（移動・再転送・接続の切断）

comDesk では、以下の 4 つの「転送画面を操作する機能」（調整機能）によって、画面の転送状況に関する属性を変更することができる。

移動 は、同一ホスト内での転送画面ウィンドウの位置を変更する機能である。これに付随して、ウィンドウサイズを変更することも可能である。

再転送 は、ホストをまたがった転送画面ウィンドウの移動機能であり、転送先ホストの変更と位置調整を同時に行うことができる。この機能による利点については、5 章で述べる。

接続の切断 は、転送画面ウィンドウを終了し、転送元ホストと転送先ホスト間の接続を切断する。

操作権の調整 は、転送における操作権状況の「操作可能」と「操作権なし」の状態を切り替える。「操作権なし」を指定した場合は、たとえ「操作中」であっても操作ができなくなる。

「移動」ならびに「再転送」の機能は、ウィンドウアイコンをドラッグ&ドロップする操作に割り当てている。この操作は「転送開始操作」におけるホストアイコンを操作する際のメタファと同一である。そのため、利用者は「転送」「移動」「再転送」といった操作を意識することなく直感的に行うことが可能となる。

このような調整機能を備えることにより、転送元ホストの利用者はいつでも転送画面に関する意思を反映することができる。また、転送先ホスト（転送画面が表示されているホスト）の利用者については、転送画

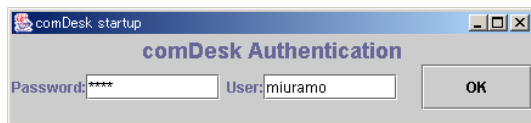


図 3 ログイン画面

Fig.3 Authentication Window

面を自分が使用しているホストに限り再転送することができる。これにより、画面転送・遠隔操作システム全体としての利便性を高めることができる。

4.2 自動構成機構

comDesk の自動構成機構について述べる。comDesk は、起動されると同一サブネット上にある他の comDesk (マスタ) を探す。もし同一サブネット上に他の comDesk (マスタ) があれば、そのホストに自ホストの情報を登録する。もし同一サブネット上に comDesk (マスタ) がなかった場合は、マスタとして動作し、comDesk からの情報を登録する。もしマスタの役目を果たしている comDesk が終了するときには、登録情報とマスタ機能を他の comDesk に移管したのち終了する。もし移管すべき comDesk が存在しない場合には、そのまま終了する。この機構によって、各ホストはいつでも場に参加したり、脱退することが可能となる。

4.3 利用者単位の管理機構

comDesk における利便性を向上させるための「利用者単位の管理機構」について説明する。この管理機構では、ホストを使用している利用者を、ホストが場に参加する時点で特定するもので、転送開始操作を行う際の処理の制限などに利用している。

comDesk を起動すると、最初に利用者名とパスワードを入力するフィールドを備えるログイン画面 (図 3) が表示される。初期状態ではキーボードフォーカスはパスワード入力部に置かれており、利用者名 (User) フィールドには、OS から取得した利用者名が自動的に入力されている。利用者名は通常は変更する必要はないが、必要に応じて変更することもできる。利用者が OK ボタンを押すと、入力された利用者名と、MD5 ハッシュ関数によって変換されたパスワードがマスタに送信され照合される。利用者名が登録されていない場合は登録され、同一の利用者名が既に登録されている場合は、パスワードが一致する場合のみログインが許可される。この仕組みにより、場における利用者とホストとの対応付けを行う。

転送開始操作を行うホストの利用者は、転送元もしくは転送先ホストの利用者と一致する必要がある。こ

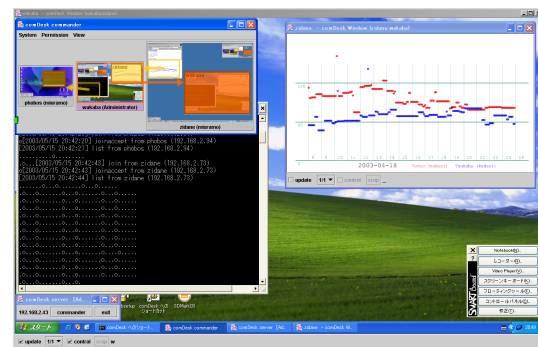


図 4 comDesk Window

Fig.4 comDesk Window

の条件を満たさない転送開始操作は却下される。この条件を満たしていれば、必ずしも転送開始操作を行うホストと転送元ホスト、転送先ホストが一致していなくてもよい。そのため転送開始操作を行う利用者の自由度が増加する。

転送元ホストと転送先ホストの利用者が同じ場合の転送開始操作はすべて承諾される。それ以外の場合、操作者以外の関係者 (転送元もしくは転送先ホストの利用者) の承諾を得るため、転送元もしくは転送先ホストに「転送を承諾するか否か」を問うためのダイアログボックスが表示される。ちなみに、場で共有している大画面ディスプレイ用に、自動的に承諾を行う設定も可能である。

4.4 実装の詳細とその他の機能

comDesk の実装には、Java を用いているため、動作する計算機の OS を選ばない。ホスト間の通信には主に RMI (Remote Method Invocation) を用いている。以下、comDesk で実装されている情報共有機能について述べる。

4.4.1 転送画面表示機能 (comDesk Window)

comDesk Window は、転送元ホストのデスクトップ画面 (転送画面) を転送先ホストに表示し、遠隔操作を可能にするウィンドウである (図 4)。通常は、転送画面は PNG (Portable Network Graphics) 形式に圧縮されて送信される。画像更新頻度は計算機の処理能力によって異なるが、現在の実装において 1024×768 の解像度の画面を 1/2 縮小画面で転送した場合、約 2 秒に 1 回程度である。図 5 に、図 4 が表示されているホスト zidane のデスクトップ全体の画面を示す。

転送画面の表示

ウィンドウ下部の [update] チェックボックスにチェックを入れることによって、連続的に画面を更新することができる。また、リストから選択することによって

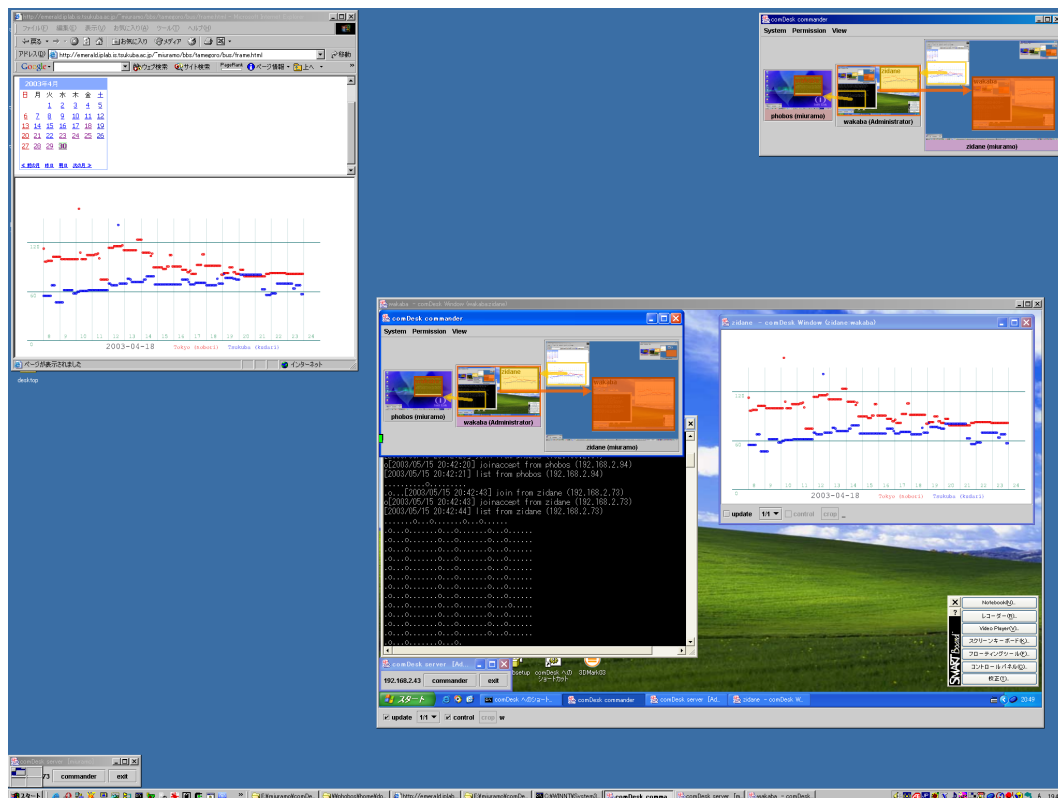


図 5 ホスト zidane のデスクトップ (全体)

Fig. 5 Whole desktop image of host zidane

画面表示の縮小率を変更できる。

部分転送

実際には、デスクトップ画面全体ではなく、一部分の領域を転送すれば十分目的が達成されることが多い。そのため、画面の一部分の領域を転送する機構を実現した。デスクトップ画面の一部のみを転送したい場合は、以下の (1-4) の手順を踏む。(1) 自分自身に転送を行う (2) 表示された comDesk Window においてドラッグ操作によって転送領域を指定する (3) [crop] ボタンを押して部分転送モードにする (4) 再転送機能を用いて目的のホストに転送する。転送元ホストと転送先ホストが同一利用者の場合は、[crop] ボタンを押すことにより部分/全体転送モードを切り替えることができる。部分転送を用いることにより、画面更新にかかる画面作成処理時間および通信量が軽減される。

遠隔操作

comDesk Window 上で行ったマウスおよびキーボード操作は画面の転送元ホストにおいて実行される。ちなみに画面取得機能および遠隔操作機能は Java のロボット (java.awt.Robot) を利用して実現している。

4.4.2 URL の転送とページの表示

ブラウザのアンカーオブジェクトや、インターネットショートカットファイルをホストアイコンにドラッグ&ドロップすることにより、URL を転送できる。転送されたホスト側においてブラウザが起動し、URL が示すページを表示する。現在の実装は、Windows 環境においては外部コマンド start に URL を引数として実行し、UNIX 環境においてはシェルスクリプトにて Netscape 等のブラウザを利用して起動している。

4.4.3 ファイルの転送 (リモートコピー)

通常のファイルを comDesk Window にドラッグ&ドロップすることによってファイルの転送が行える。転送ファイルはホスト側の固定ディレクトリにコピーされる。内部の仕組みとしては、ホストアイコンがドロップされたファイルのローカル計算機におけるパス名を取得し、そのファイルをバイナリ形式として読み込む。読み込まれたデータは RMI を介して転送され、転送先ホストのファイルシステムにおける固定ディレクトリに同名のファイルとして書き出される。

5. 利 用 例

具体的な comDesk の利用例を挙げる。

5.1 シナリオ 1: 個人で使用する場合

個人が処理能力の異なる 3 台以上の計算機を所有し、同時に使うことも日常的となっている。このような場面において、計算機間の情報交換に comDesk を使用することが考えられる。URL 転送による Web ページの表示機能によって、任意の計算機のディスプレイを使って Web ページを表示することができる。また、ノートパソコンの使いにくいポインティングデバイスの代わりに、デスクトップパソコンに接続されたマウスを使用するといった使い方がある。同一利用者名でログインしていれば、転送元あるいは転送先ホスト以外の comDesk commander から同等の操作が行える。この機能によって、非力な処理能力の計算機を comDesk commander 専用として利用することもできる。また、再転送機能を用いると、デスクトップに転送していた情報をノートパソコンに再転送して、別の場所に持ち運んで使うといったことが可能になる。

5.2 シナリオ 2: 少人数のグループ、共用ディスプレイが存在する環境にて使用する場合

通常は個人単位で作業しており、偶発的に情報共有や操作支援といったコラボレーションの必要性が発生する少人数のグループにおける使用も考えられる。例えば、コンパイル中の見慣れないエラーや複雑なソフトウェアの操作などを、詳しい人に画面情報として転送することによって、より効率的に問題解決を行うことができる。また、壁面提示型のプラズマディスプレイやプロジェクトなどの共用ディスプレイが準備できる場合には、URL 転送機能によって共用ディスプレイへ Web ページを表示したり、個人のノートパソコンに表示されている画面を提示することによるコラボレーションが可能となる。特に部分転送によって画面の一部をピンポイントに提示できることで、画面ベースの情報交換機能がより有効に作用する。

6. 比較実験

流動的な環境における画面転送・遠隔操作ソフトウェアにおいては、画面転送による情報交換を開始するまでの設定にかかる負荷が少ないほうが望ましい。我々は、流動的な環境を想定し、転送を開始する作業に必要な時間と作業内容について比較するための実験を行った。

比較対象として、VNC (WinVNC version 3.3.6) を用いた。Windows が動作している計算機を用意し、

あらかじめデスクトップに VNC サーバ/ビューアと comDesk へのショートカットを作成しておいた。計算機の IP アドレスは既知とし、パスワード (4 文字) は同一のものを使用した。

実験 1 では、2 台の計算機 (A, B) において、まず A の画面を B に転送 (A to B) し、その後 B の画面を A に転送 (B to A) するタスクを 1 人で行なった。

実験 2 では、3 台の計算機 (A, B, C) において、A を共用ディスプレイ用、B, C を個人利用の計算機とみなし、まず A の画面を B と C に転送 (A to B, C) したのち、B, C の画面をリバースコネクション機能を用いてそれぞれ A に転送 (B, C to A) するタスクを 2 人 (B, C の利用者) で行なった。実験 2 においては、A にあらかじめ VNC サーバと VNC ビューア (Listen Mode)、comDesk (転送要求の自動承諾機能を有効) を起動しておき、利用者が A において操作を行わなくてもタスクを完了できるようにしておいた。

図 6 に、実験 1, 2 における作業の流れを示す。図 6 において、矢印が画面転送方向を、黒丸が操作を行う計算機を示している。それぞれのタスクを 3 回ずつ行い、タスクに要した時間と、左クリック数、キー入力数を計測し、タスクに要した時間が最も早かった回のデータを採用した。

表 2 に、実験 1 における結果を示す。左クリック数とキー入力数については、A, B における数値を分けて (A+B) という形式で示している。(A to B) と (B to A) の合計時間について着目すると、comDesk は VNC の約半分程度に抑えられている。VNC の場合は、(1) サーバを起動 (2) パスワードを設定 (3) ビューアを起動 (4) IP アドレスを入力 (5) パスワードを入力 という 5 つの手順を、各転送方向について行う必要があるため、(A to B) と (B to A) はほぼ同じ作業時間となった。comDesk の場合は最初の転送 (A to B) を行うまでに (1) 起動 (2) パスワードを入力 (3) ドラッグ&ドロップによる転送指示 という 3 つの手順が必要になるが、逆方向の転送 (B to A) 実行時には (1) (2) を省略できるため、合計の作業時間は約半分に抑えることができた。

表 3 に、実験 2 における結果を示す。結果は、B, C における数値を分けて (B+C) という形式で示している。作業時間の合計を比較すると、実験 1 に比べて comDesk と VNC の差が生じなかった。この理由として、A にはあらかじめ VNC サーバと VNC ビューア (Listen Mode) を起動し設定を済ませていることと、B, C において A に画面を送るために起動した VNC サーバについては、パスワードを設定する作業を行っ

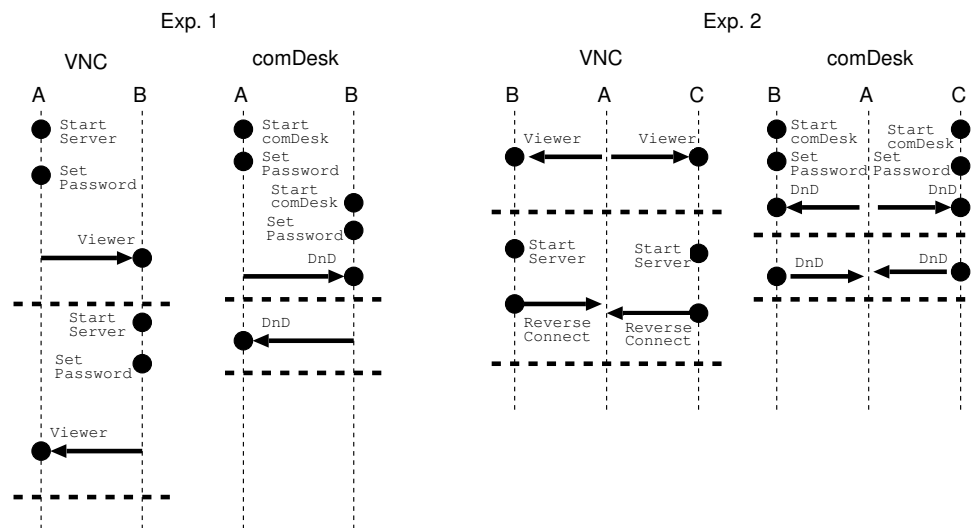


図 6 比較実験における作業の流れ。矢印が画面転送方向、黒丸が操作しているホストを表す。
Fig. 6 Flowcharts of the startup tasks. Arrow represents direction of transferring image. Black-dot represents operating host.

表 2 転送開始操作の比較 (実験 1)

Table 2 Comparison of startup tasks between VNC and comDesk (Exp. 1)

step	Time (sec)		Left Click (times)		Key Input (times)	
	VNC	comDesk	VNC	comDesk	VNC	comDesk
A to B	24	19	6+4	6+7	4+20	4+4
B to A	33	8	5+7	0+0	20+4	0+0
total	57	27	22	13	48	8

表 3 転送開始操作の比較 (実験 2)

Table 3 Comparison of startup tasks between VNC and comDesk (Exp. 2)

step	Time (sec)		Left Click (times)		Key Input (times)	
	VNC	comDesk	VNC	comDesk	VNC	comDesk
A to B,C	13+16	17+13	8+7	7+6	19+19	4+5
B,C to A	12+15	4+6	5+5	2+2	12+13	0+0
total	56	40	25	17	63	9

ていないことが挙げられる。ちなみに、VNC ビューア (Listen Mode) を起動するタスクと、VNC サーバを起動してサーバのパスワードを設定するタスクに要する時間は、それぞれ約 3 秒、約 10 秒であり、サブネット内で最初の comDesk を起動しパスワードを設定するタスクに要する時間は (既存ホスト検出失敗のタイムアウト 3 秒を含め) 約 9 秒であった。

これらの結果から、サーバを起動・設定する回数が少ない場合 (単方向の転送や、あらかじめ決められた転送を開始する場合) にはそれほど差は生じないが、多数のホストにサーバを起動・設定する必要がある場合 (転送方向や操作を行う計算機が定まっておらず、使用する状況が刻々と変化する場合) や、転送開始と切断を頻繁に行うような場合には comDesk が

有利であると考えられる。今回の比較では VNC サーバに設定したパスワードの交換が必要なタスクは含まれていないが、実際には複数の利用者間でパスワードの交換が必要な状況が発生することも予想される。このような状況では複数の利用者と計算機が参加する状況を考慮した comDesk の優位性がより顕著に表れると考えられる。

7. 関連研究

7.1 遠隔操作と情報提示

VNC⁸⁾ は、画面情報を転送するサーバと、受信するビューアが RFB (Remote Frame Buffer) プロトコルを用いて通信する。VNC サーバは画面の転送データ量を減らすための複数の画面処理方式 (エンコーディ

ング) を切り替えることができる。また、VNC サーバから VNC ビューアに接続するためのリバースコネクション機能を備えている。しかし画面転送に基づくコミュニケーションを目的としたものではないため、3 台以上の計算機間での画面転送については考慮されていない。

7.2 大画面と CSCW

大画面の共有や、複数の計算機を連携させた協調作業環境に関する研究は CSCW (Computer Supported Cooperative Work) 研究として数多く行われている。Co-Lab (Collaboration Laboratory)¹¹⁾ は大画面スクリーンを用いた初期の CSCW 環境であり、発想支援などの協調的作業に利用できる。Prairie¹²⁾ は、基本的に個人で使用する 6 つの画面を組み合わせた大画面表示システムである。用途として、コミュニティの関係図を表したり遠隔地にいる人との共同作業を対象としている。GMD (現在 Fraunhofer-Gesellschaft) の研究プロジェクト i-land¹³⁾ では、協調作業用のデバイスとして壁面リアプロジェクション方式の大画面 (DynaWall)、机型 (InteracTable)、椅子に組み込まれた計算機 (CommChair) を提案している。スタンフォード大学の iRoom²⁾ は、PDA を含む計算機がアドホックに参加する仕組みを備えた大画面会議システムである。EventHeap サーバを介した画面切替えや、遠隔ポインタシステム¹⁴⁾ などを実現している。カーネギーメロン大の Pebbles プロジェクト¹⁵⁾ では PDA を補完的な入出力デバイスと位置付け、計算機の操作に利用する MMUI (Multimachine User Interfaces) という概念を提唱し、プレゼンテーションにおけるスライドショーの切替え操作やスクロールを PDA から行う研究を行っている。これらの研究においては画面と PDA 等のデバイスを連携させる方法に主眼が置かれている。我々は環境に依存せず、集まった参加者同士が画面転送機能によるコミュニケーションをより円滑に利用できるようにすることに主眼を置いている。

7.3 データ転送

暦本による Pick-and-Drop¹⁶⁾ は、ペンを用いた複数計算機間での直接操作手法である。ペンでオブジェクトをつまみあげるといったメタファを用いてオブジェクトの交換や属性の選択を行うことができる。我々はネットワークのみ存在するような環境においても、簡易な設定で画面情報や URL 情報の交換が行えるようにすることを目指している。

7.4 P2P を利用したモバイルグループウェア

「なかよし」¹⁷⁾ では、PHS の内線通信機能を利用したアドホックネットワークを構築し、その上でグループ

ウェアアプリケーションを実現している。端末だけで簡単に協同作業環境を実現するという目的は comDesk と共通であるが、PHS の機能を用いてアドホックネットワークをいかに構築するかという点に主眼が置かれている。我々は、画面転送と遠隔操作に基づくシステムにピアツーピア技術を利用することによって、利用者の操作にかかる負荷を軽減することを目的としている。

8. 今後の課題

comDesk の有用性をより高めるためには、以下の課題を解決することが必要であると考えている。

8.1 安全な認証機構

現在の comDesk の実装では「利用者単位の管理機構」を導入し、転送開始操作時における操作権限の確認を行なっている。しかし、ホストと利用者を結びつけるパスワードの管理や認証は場に参加しているある特定のホストに任されているため、そのホストが信頼できない場合には安全性に問題が生じる。

また、利用者 A の画面が利用者 B に転送されている状況で、利用者 C が利用者 B の画面を要求する場合のように、画面転送のカスケードが生じた場合の権限についても考慮する必要がある。現在は利用者 B のみの判断で転送開始することができてしまうが、安全性を高めるため関連する全ての利用者への確認を行う仕組みを備えることが望ましい。

8.2 通信処理の改善

現在の comDesk の実装では、サブネット内の計算機との通信しか考慮していない。適用領域を広げるためには、ルータを越えたサブネット外の計算機との通信や、ファイアウォールが存在する場合などへの対応をする必要がある。

また、現在はマスタが不正終了してしまった場合、新しいホストが参加できなくなったり、参加中のホスト一覧が取得できなくなってしまう。そのため、マスタ機能を分散化し、通信における信頼性を向上させることが望ましい。

8.3 画面転送方式の改善

現在の comDesk の実装では、画像を PNG 形式に圧縮した状態で送信している。そのため、画像の更新頻度が遅く、遠隔操作を行いにくいという問題がある。そのため、画像の転送方式を改善し、また複数ホストへの画面配送についても考慮することにより、スケーラビリティと有用性を向上させていきたいと考えている。

9. ま と め

P2P 技術を適用した画面送信・遠隔操作システム“comDesk”の設計と実装について述べた。P2P の自動構成技術によって利用者の設定項目を軽減できることに加え、P2P の思想は自由な転送方向との親和性が高いため、多様な利用形態に対応できると考えられる。ドラッグ&ドロップによるインタフェースを採用し、転送開始操作と接続後の転送画面の再転送操作を同一の操作技法によって実現した。本システムは流動的な環境における少人数での会議や共有画面への掲示など、アドホックな画面転送・遠隔操作に基づくコミュニケーションを促進させる効果があると考えられる。

10. 謝 辞

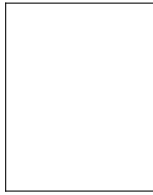
本研究の一部は、文部科学省科学研究費補助金・特定領域研究(2)(課題番号: 15020216)ならびに若手研究(B)(課題番号: 14780183)の支援によるものです。

参 考 文 献

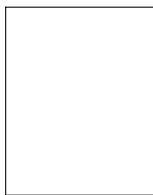
- 1) Brad Johanson, Armando Fox, and Terry Winograd. The Interactive Workspaces Project: Experiences with Ubiquitous Computing Rooms. *IEEE Pervasive Computing Magazine*, Vol. 1, No. 2, pp. 71–78, 2002.
- 2) Armando Fox, Brad Johanson, Pat Hanrahan, and Terry Winograd. Integrating Information Appliances into an Interactive Workspace. *IEEE Computer Graphics & Applications*, Vol. 20, No. 3, pp. 54–65, 2000.
- 3) Brad Johanson and Armando Fox. The Event Heap: A Coordination Infrastructure for Interactive Workspaces. In *4th IEEE Workshop on Mobile Computing Systems and Applications (WMCSA 2002)*, June 2002.
- 4) Detlef Schoder and Kai Fischbach. Peer-to-Peer Prospects. *Communication of the ACM*, Vol. 46, No. 2, pp. 27–29, February 2003.
- 5) Dejan S. Milojicic, Vana Kalogeraki, Rajan Lukose, Kiran Nagaraja, Jim Pruyne, Bruno Richard, Sami Rollins, and Zhichen Xu. Peer-to-Peer Computing. Technical Report HPL-2002-57, HP Laboratories Palo Alto, March 2002.
<http://www.hpl.hp.com/techreports/2002/HPL-2002-57.html>.
- 6) 三浦元喜, 志築文太郎, 田中二郎. 画面転送・遠隔操作に基づくコミュニケーションツールの設計. インタラクシオン 2003 論文集, pp. 263–270, February 2003.
- 7) Motoki Miura, Buntarou Shizuki, and Jiro Tanaka. A Framework for Transferring Desktop Images and Remote Operations in Multiple Computer Environments. In *Proceedings of 10th International Conference on Human - Computer Interaction*, Vol. 2, pp. 198–202, June 2003.
- 8) Tristan Richardson, Quentin Stafford-Fraser, Kenneth R. Wood, and Andy Hopper. Virtual Network Computing. *IEEE Internet Computing*, Vol. 2, No. 1, pp. 33–38, 1998.
- 9) Symantec Corporation. pcAnywhere.
http://www.symantec.com/region/jp/products/pca_enterprise/.
- 10) 3AM Laboratories PL. RemotelyAnywhere.
<http://remotelyanywhere.com/>.
- 11) Mark Stefik, Daniel G. Bobrow, Greg Foster, Stan Lanning, and Deborah Tatar. WYSIWIS Revised: Early Experiences with Multiuser Interfaces. *ACM Transactions on Office Information Systems*, Vol. 5, No. 2, pp. 147–167, April 1987.
- 12) Kishore Swaminathan and Steve Sato. Interaction design for large displays. *Interactions (ACM Magazine)*, Vol. 4, No. 1, pp. 15–24, January 1997.
- 13) Norbert A. Streitz, Jörg Geißler, Torsten Holmer, Shinichi Konomi, Christian Müller-Tomfelde, Wolfgang Reischl, Petra Rexroth, Peter Seitz, and Ralf Steinmetz. i-LAND: An interactive Landscape for Creativity and Innovation. In *Proceedings of the CHI 99*, pp. 120–127, May 1999.
- 14) Brad Johanson, Greg Hutchins, Terry Winograd, and Maureen Stone. PointRight: Experience with Flexible Input Redirection in Interactive Workspaces. In *Proceedings of UIST 2002*, pp. 227–234, 2002.
- 15) Brad A. Myers. Using Handhelds and PCs Together. *Communication of the ACM*, Vol. 44, No. 11, pp. 34–41, November 2001.
- 16) Jun Rekimoto. Pick-and-Drop: A Direct Manipulation Technique for Multiple Computer Environments. In *Proceedings of UIST'97*, pp. 31–39, October 1997.
- 17) 倉島顕尚, 前野和俊, 市村重博, 田頭繁, 武次將徳, 永田善紀. 集まったその場での協調作業を支援するモバイルグループウェアシステム「なかよし」. 情報処理学会論文誌, Vol. 40, No. 5, pp. 2487–2496, 1999.

(平成 2003 年 5 月 15 日受付)

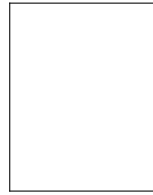
(平成 2003 年 11 月 5 日採録)

**三浦 元喜**（正会員）

1974 年生。1997 年筑波大学第三学群情報学類卒。2001 年筑波大学博士課程工学研究科修了。博士（工学）。現在、筑波大学電子・情報工学系助手。筑波大学先端学際領域研究（TARA）センター勤務。Web インタフェース、視覚化、グループウェアに興味を持つ。日本ソフトウェア科学会、ACM 各会員。

**志築文太郎**（正会員）

1971 年生。1994 年東工大理学部情報科学科卒。2000 年同大学大学院情報理工学研究科数理・計算科学専攻博士課程単位取得退学。博士（理学）。現在、筑波大学電子・情報工学系講師。ヒューマンインタフェースに関する研究に興味を持つ。日本ソフトウェア科学会、ACM、IEEE Computer Society、電子情報通信学会、ヒューマンインタフェース学会各会員。

**田中 二郎**（正会員）

1975 年東京大学理学部卒業。1977 年同大学院理学系研究科修士課程修了。1984 年米国ユタ大学計算機科学科博士課程修了、Ph.D. in Computer Science。1984 年から（財）新世代コンピュータ技術開発機構で第五世代コンピュータ核言語の研究開発に従事。1993 年より筑波大学に勤務。現在、電子・情報工学系教授。プログラミング言語やヒューマンインタフェースに興味をもっている。ACM 日本支部で CACM 日本語版編集長。ヒューマンインタフェース学会評議員。日本ソフトウェア科学会学会誌編集委員。IEEE Computer Society、電子情報通信学会、計測自動制御学会、人工知能学会各会員。