

DirectShare によるコンピュータ作業環境の携帯とデータ共有

千 田 陽 介[†] 佐 藤 裕 一[†] 佐 沢 真 一[†]

磁気ディスクの高性能化, 低価格化に伴い, コンピュータ (PC) 作業に必要なデータは各 PC が持つようになってきている。このため, 出張などの移動時や PC の買い替え時にデータの移動 (コピー) という非生産的な作業を強いられている。この問題は個人のみならず, 他人との共同作業でも発生する。このような問題を解決するため, 筆者らは各 PC がピアツーピアにデータを共有するシステム “DirectShare” を構築した。このシステムでは, ある PC で作業した内容を自動的に他の PC にも反映するためデータの移動やそれに伴う履歴管理作業を必要としない。本稿では DirectShare の概念と技術要素, さらに DirectShare を携帯する構想について述べるものである。

Computer Environment and Data Sharing using DirectShare

YOSUKE SENTA,[†] YUICHI SATO[†] and SHINICHI SAZAWA[†]

In this paper, we present the design of DirectShare, a peer-to-peer communication and collaboration tool. DirectShare provides fast and scalable data-sharing capabilities by taking full advantage of peer-to-peer technology. DirectShare also utilizes auxiliary centralized file-mirroring services. DirectShare makes it possible to maintain the same computer environment on various devices without cumbersome file-copying operations.

1. はじめに

近年, パーソナルコンピュータ (PC) の高性能化・低価格化に伴い生産活動における PC の役割は益々重要になってきている。さらに磁気ディスクの高容量化により, 編集時の文章はもちろん, 過去に作成したデータ, メールのログなど, あらゆるデータをディスク内に保管し, 生産活動の資料にしている。しかし, PC に保存するデータが多くなればなるほど, 職場用, 自宅用, 出張用など PC を複数所有している場合, 各 PC 間でデータを複製し, 整合性を保つ作業が大変になる。これは一人が複数の PC を管理する時だけでなく, 複数の人間がそれぞれの PC でデータを共有する際や, 新しく PC 買い替えた際にも発生する。

このような問題を解決するため, 従来からデータおよびその処理はサーバで集中管理し, クライアントは表示・入力などの軽い処理を行う Sun Ray や Oracle NC などの “Thin Client” という概念がある¹⁾。また NFS や MS-Windows の「ネットワークコンピュータ」(Samba) を用いて, ファイルサーバ上のディスクをあたかもローカル PC のディスクのように振舞わ

せ, データの一元管理を行う方法もある。しかし, このようなクライアント・サーバモデルでは, 原理的にネットワークに繋がないとデータを取り出すことができない。また, 例えばファイアーウォールに守られた LAN 内のファイルサーバに対し, 移動中の PHS 回線や自宅の ADSL 回線からデータを取り出すことも難しい。さらに, 近くの端末にデータを送る際にもサーバ経由となるため, 回線やサーバ性能を十分高くしておく必要がある。

そこで筆者らは, ピアツーピア (P2P) 型でデータを共有するシステム: “DirectShare” を開発した。DirectShare では LAN に繋がっている際にバックグラウンドで必要なデータを各ローカルディスクにため込むため, ネットワークに繋がっていない時でも必要なデータを取り出すことができる。また, サーバを介さない P2P 接続のため, 回線に対する負荷も小さい一方, 各端末がデータを全てローカルディスクに入れるため,

- DHCP や出張先など接続のたびに IP アドレスが変わる端末に対し, どうやって P2P 接続を行うか。
- 勝手なタイミングで LAN に接続し切断する個々の端末の間で, どのようにデータの整合性を保つか。
- 共有データが多過ぎてディスクに入りきれなくなる恐れがないか。

[†] 富士通研究所
Fujitsu Laboratories LTD.

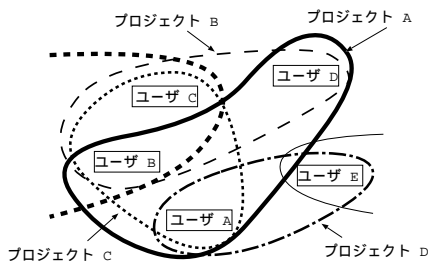


図 1 人の生産活動

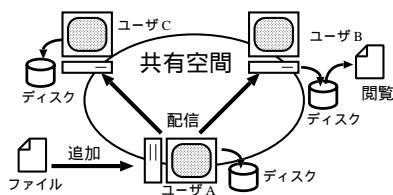


図 2 データの共有

- データを各ローカルディスクに入れることはセキュリティ上問題にならないか。
- などのクライアント・サーバにはなかった P2P 固有の問題がある。本稿では、DirectShare の構想とともに既存技術との違いを述べ、続いてこれらの問題を解く技術的要素について述べていく。

2. 生産活動におけるデータ共有

一般的な人の生産活動では、図 1 のように複数の人と関わりを持ち、幾つもの生産活動（プロジェクト）を並列に行っている。このプロジェクトは時間とともにメンバが入れ替わり、プロジェクトの達成と共に消滅する。DirectShare ではこの生産活動の単位を「共有空間」という概念でとらえ、円滑に共有空間の管理を行い、各共有空間の参加者の間で簡単にデータの共有や会議を行うことを目標としている。

データの共有とは、例えば図 2 において、ユーザ A が必要なファイルを共有空間内に追加すると、ネットワークを介して自動的に他のユーザ（B, C）の端末に配信され、それを閲覧・編集することができるものである。他のユーザがそのファイルを更新した場合、同様に自動的に他の端末に配信され、各ユーザが持つファイルを常に最新の情報を保つ。この配信はユーザ単位でなく PC 単位で行われるため、一人が複数の PC を持ち、同じ共有空間に所属すると、データのバックアップを自動的に行うことができる。共有されるデータは、図 2 で示したファイルだけでなく、プロジェクトに関連する URL や写真やプレゼンテーション用のイメー

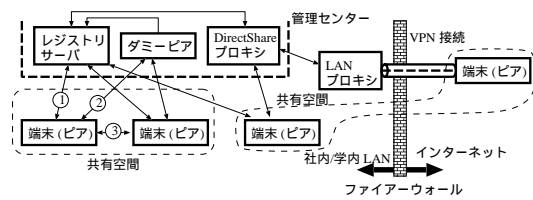


図 3 DirectShare システム全体図

ジ、3次元モデルなど生産活動の内容に応じ様々な形態を取ることができる。

一方、共有空間のメンバ間で会議を開催する際には、配布されたデータを会議資料として遠隔地間で、音声を紹介しながら同時に資料の参照やマーキング、資料の編集などを行う。またこの会議において、共有空間を先生と生徒で共有すれば E-Learning に利用することもできる。

3. DirectShare

3.1 既存技術

このように複数の端末間でデータを共有し、遠隔会議を行うシステムとして、既にいくつかの製品やプロジェクトがある。クライアント・サーバ型では Lotus Sametime などが代表的である²⁾。しかし、先ほど述べたように、クライアントサーバ型はネットワークに接続しなくては利用できない。一方 P2P 接続によるシステムには Groove や gnutella, SOBA などがある。Groove については遠隔会議機能が使いやすく、E-Learning に利用することが難しい³⁾⁴⁾。また gnutella は、サーバを全く使用しないため、端末間の接続に対しスケラビリティがない。SOBA は遠隔会議に重点を置いているため非同期のデータ共有を行うことができない⁵⁾。また現段階ではフレームワークのみで具体的な実装は行われていない。

3.2 概要

このような従来技術を鑑みて、DirectShare では P2P 技術をベースにして、接続の容易性とデータの同期に重点をおき、データ共有と遠隔会議の両立をめざす。

図 3 に DirectShare システムの全体図を示す。図 3 のように DirectShare では、P2P 接続を行う各端末（ピア）の他に、レジストリサーバ、ダミーピア、プロキシサーバの三つのサーバによって構成されている。このうちレジストリサーバは各端末の IP アドレスを集中管理する役割をなすもので、SIP (Session Initiation Protocol) で動作する。各端末は起動時にまずレジストリサーバに接続し、自分がネットワークに接続した

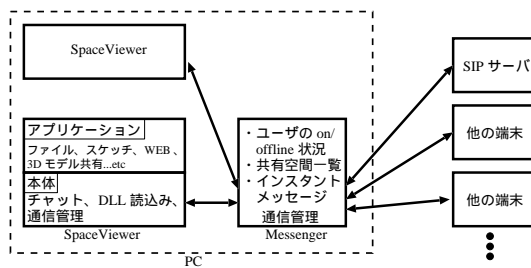


図 4 DirectShare 端末の内部構造

(オンラインになった) ことを自分の IP アドレスと共に伝え、P2P 接続したい端末がオンラインかどうか問い合わせる (1)。このレジストリサーバへの問い合わせと同じくして、ダミーピアにも接続するが (2)、この件に関しては 3.3 節にて詳しく述べるので、ここでは割愛する。レジストリサーバの回答により、自分が接続したい端末の中でオンラインのものを、その IP アドレスと共に知ることができたため、端末に対し P2P 接続を行う (3)。

一方、自分がオンラインになった時点ではオフラインであったが、その後オンラインになった端末に関しては、上述した手順によって相手側から P2P 接続をしてくるため、その時点で相手がオンラインになったことが分かる。このようにレジストリサーバはピアの位置情報を伝えるだけの非常に軽い処理を行っているため、クライアント数の増加に対応することができる。

DirectShare ではレジストリサーバ接続用、P2P 接続用と様々なポートを使っており、そのままではファイアウォールを越えることができない。一般の社内/学内 LAN では、ファイアウォール外からのアクセスに対し、LAN プロキシを構築し、HTTP や SMTP など一般的なプロトコルに関して、VPN でトンネリングして通す。そこで図のように DirectShare 用のプロキシサーバを建て、LAN 外の端末とプロキシ間は HTTP でくるんだ通信を行う。これによりファイアウォール外の端末とも共有空間の共有が可能になる。

図 4 に DirectShare 端末の内部構成を示す。図のように DirectShare は Messenger と SpaceViewer という二つのプロセスから構成されている。このうち、Messenger は共有空間の有無に関わらず常に起動しているもので、保有している共有空間の一覧表示や起動、他のユーザのオン/オフライン状況の表示、インスタントメッセージや共有空間への招待メッセージの送受信などユーザとのインターフェイスを行う。しかし Messenger の一番重要な役割は他の端末との通信を司ることであり、すべての通信は Messenger を介して行

表 1 DirectShare のアプリケーション

名称	機能
アプリ共有	VNC のように画面イメージの形で遠隔地間で任意のアプリケーションを共有する。
WEB	URL を共有、またスクロールやマーキングの共有も可能
Calendar	スケジュール帳を共有
VPS	VPS ⁶⁾ を用いて、遠隔地間で 3D モデルの動作の共有が可能
Sketch	画像イメージを共有、マーキングなどのドロー操作も可能
共有ファイル	ファイルを共有

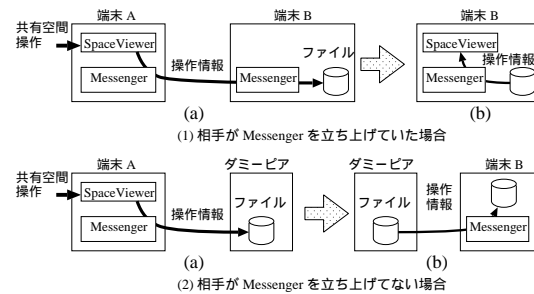


図 5 共有空間の同期

わている。

一方 SpaceViewer は共有空間を表示・編集するもので、共有空間を起動する度に 1 つのプロセスが動作する。共有空間の種類に応じて、共有するアプリケーションを選択できるように、各アプリケーションは DLL の形をしている。そのため、各 DLL を読み込み、Messenger 間との通信を行うとともに、チャットなど基底的処理を行う本体部分と、各アプリケーション用の DLL に分かれている。現在作成された DLL を表 1 に示す。DLL の形状を取っているため、簡単に新しいアプリケーションを拡張することが可能である。

3.3 共有空間の同期

P2P 接続では端末 A が共有空間を操作すると、その情報が直接端末 B に送信され端末 B の共有空間に反映される。この際、端末 A と B が同時に共有空間 (SpaceViewer) を立ち上げていた場合 P2P により直接情報が相手に届くが、端末 B がその共有空間を立ち上げてなかったりオフラインだった場合、そのままでは端末 A の操作が端末 B に反映されず共有空間の不一致が生じてしまう。

そこで DirectShare では操作情報のバッファリングを行っている。図 5 (1) は端末 B が Messenger のみ起動し、SpaceViewer を起動していない時の処理であ

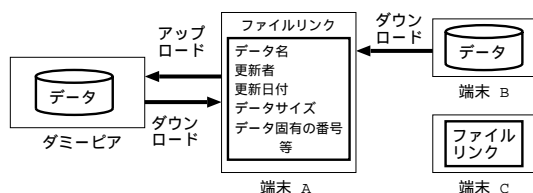


図 6 ファイルリンク

る。図中 (a) において端末 A が共有空間を操作すると、その情報 (パケット) は端末 B に届く。しかし端末 B では SpaceViewer を起動してないため、その情報を解釈して実行することはできず、そのままファイルに書き込む。書き込まれた情報は図中 (b) において端末 B が共有空間を起動した際に読み込まれ、ファイルからパケットを一つずつ取りだし、端末 A の状態に関わらず、それがあたかも今受信されたかのように処理を行い共有空間を更新する。ここでパケットの処理を Messenger でなく SpaceViewer で行うのは、パケットの処理は表 1 で示したアプリケーション依存であるため、Messenger で行うことは拡張性等の理由により難しいためである。この操作により端末 B が持つ共有空間は端末 A と一致することができる。

さらに端末 B がオフラインの場合、図 5 (2) のように外部に 24 時間稼働し、代理で情報を受けとる「ダミーピア」を用いて共有空間の同期を行っている。図中 (a) において端末 A が共有空間を操作するとその情報は端末 B の代わりにダミーピアに送られ、保存される。その後、端末 B がオンラインになると、(b) のようにダミーピアから自分宛のデータをローカルディスクのファイルに転送する。その後は先ほどと同様に、SpaceViewer を起動した時点でパケットの処理を行い、共有空間の同期を行う。

図 3 ではダミーピアは管理センターに一つ構築している。しかしすべての端末がこのダミーピアに接続すると負荷が集中してしまう恐れがあるため、各自が自分専用もしくはグループ単位のダミーピアを簡単に構築できるようにしている。自分専用のダミーピアを置くことで、ユーザはダミーピアとの通信が速くなる利点があり、またシステムにとっては管理センターのダミーピアの負荷を下げる事ができる。

3.4 ファイルリンク

これまで述べてきたように DirectShare では P2P で相手にデータを送りつけ相手のディスク領域に保存させる。このため、共有空間を複数持てば持つほどディスクの中にデータが蓄積され、ディスクがパンクしてしまう恐れがある。

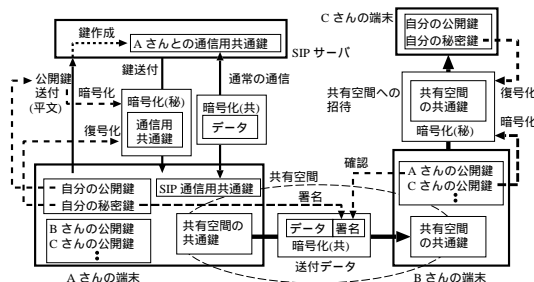


図 7 DirectShare におけるセキュリティ対策

このような問題を解決するため、図 6 のようにデータ本体を持たず更新者やサイズ、識別子 (ID) などの諸元情報のみを保有する「ファイルリンク」機能を構築した。この諸元情報は本体に比べ小さいためディスク容量の節約になる。各端末において、データをリンク化するには、まずダミーピアに対し、他のピアがリンク化した後などの理由により既にデータを持っているかどうか問い合わせ、持っていない場合はデータをアップロードする。その後、保有する元データを削除して諸元情報に入れ換える。リンクを元に戻すには、同じ共有空間を持つ他のオンラインの端末に対し、識別子を元にそのデータを持っているかネットワーク的に近い順に問い合わせていき、もし持っていた場合はその端末からダウンロードして元に戻す。誰も持っていない場合、ダミーピアからデータを取得する。ダミーピアを最後にしているのは、ダミーピアへの負荷をできるだけ下げられるためである。

リンク化を行う時期については 1. ユーザが自分のデータをリンク化する。2. 相手にデータを送りつける際に送信者側が諸元情報のみ渡す。など、簡単な機能については実現している。将来的にはインテリジェンスを持たせ 3. 各端末がディスクの空容量を見て、必要なら最近使われてないデータをリンク化する。ことも考えている。

3.5 セキュリティ

DirectShare はデータが各端末 (PC) にデータが蓄えられるため、PC が盗難にあった際、データを盗まれないようにする必要がある。さらに、通信経路における盗聴やなり済ましなど対策もしなければならない。そのため DirectShare では図 7 のように公開鍵 (RSA 1024bit)、共通鍵 (Rijndael 128bit) の二つの暗号方式を用いてセキュリティ対策を行っている。

鍵を二種類使っているのは公開鍵方式の暗号化は暗号化・復号化に時間がかかるため、音声や共有空間の編集などの通信やディスクへの保存など普段の暗号化は高速な共通鍵方式で処理している。この共通鍵は

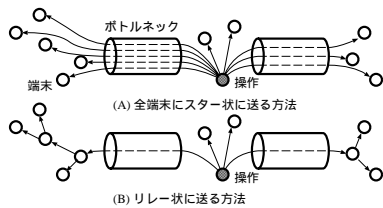


図 8 共有空間の編集情報を分配

共有空間毎に異なるため、第三者を新しく共有空間に招待する時に共通鍵を安全に伝える必要があり、そこに公開鍵方式を利用している。また、公開鍵は済ましを防ぐためパケットの署名付けにも用いている。

特に機密レベルの高い共有空間では、コピーや印刷などすべて禁止して、利用者といえどもデータを DirectShare の外に持ち出すことはできないようにしている。しかし、この方法でも例えばカメラを使ってモニタ画面を撮影することを防ぐことはできない。さらに、ネットワークに繋がってなくても共有空間内の情報を参照できるように、鍵は PC 本体の中に特殊な形でしまっているが、これは利便性を高める反面、十分な実力と悪意を持った盗難者によって解読される恐れもある。このような問題を解決するため高機密な共有空間では、ユーザ情報を薄くモニタ画面に表示して撮影を牽制する、指紋等の認証機能を入れ鍵の一部にするなどの対策を考えている。

3.6 データ配信

複数の端末が同一の共有空間に属している場合、その空間への編集情報は各端末に伝達される。その際、図 8 (A) のように全端末にスター状に送るより (B) のように適当な端末をハブ化してリレー的に送る方が効率が良い。特に図のように途中でボトルネックがある場合 (B) の方式は有効となる。これはセールスマン巡回問題の一種と考えることができる。しかし一般的な巡回問題は全端末 (全都市) の接続状況を把握し最適解を求めるものに対し、本問題はハブ化している端末も、誰かの端末のため途中でサインオフして消えることや、誰かが新しくサインオンすることなどダイナミックに接続状況が変わる。そのため、時間をかけて最適解を求める従来の巡回問題の解法は不向きとなり、高速に準最適解を高速に求める必要がある。

その解法を概念を図 9 に示す。端末の接続は必ず順次行われるため、最初はまず二つの端末 A, B のみ接続している。この場合端末 A にとって B はハブ端末と考えることができる。そこに新たな端末 C が加わると、既存の端末 A, B は C とのネットワーク上の距離を測定し互いに伝え合う。ここでネットワーク上の

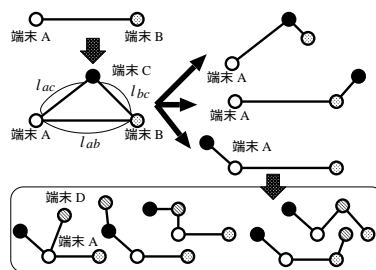


図 9 ハブ端末の検出



図 10 パケットのループ

距離とは、回線速度による指標値で ping の戻り値や、適当な大きさのパケットを転送した際の時間などから算出する。AB, AC 間の距離をそれぞれ l_{ab} , l_{ac} , BC 間の距離を l_{bc} とすると、 $l_{ac} > l_{ab} > l_{bc}$ の時には端末 C はハブ端末 B の配下に、 $l_{ab} > l_{ac} > l_{bc}$ の時には端末 C を新たなハブ端末として端末 B は C の配下に、それ以外の時は B も C も孤立したハブ端末とする。さらに新たな端末 D が現れた場合、同様に既存のハブ端末の中から近い順に距離を比較し、1. 既存のハブ端末の配下に入れる。2. 既存のハブ端末と入れ換える。かどうか判断し、全てのハブ端末に対し条件を満たさなかった場合 3. 新しいハブ端末にする。またハブ端末がオフラインになった場合、配下の端末がすべて新しく入り直したとして同様の処理を行う。

このように、各端末は全端末がどのハブ端末に属しているかというルーティング情報を持つ。全端末にパケットを送付する際は、各ハブ端末に対し宛先とともにパケットを送付する。パケットを受けとったハブ端末側でも同様のルーティング情報を持っているため、その情報に基づき自分宛以外のパケットは各ハブ端末に転送する。ここでルーティング情報は個々の端末が勝手に作るため必ずしも一致しない。そのため図 10 のようにパケットがループしてしまう可能性がある。そこでパケットは転送される度にどの端末を通ってきたかという履歴を追加して、転送時に重複してないか確認することでループを阻止している。

一方図 8 におけるボトルネック自体の高速化も望まれている。このようなボトルネックの原因は一般に RTT (Round Trip Time) が大きい、パケットロス率が高いなどの回線品質によるものが多い。一般的な TCP 通信ではパケット毎に ACK を返す構造

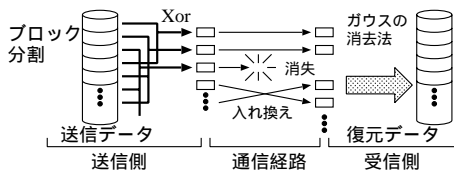


図 11 UDP + FEC による高速転送

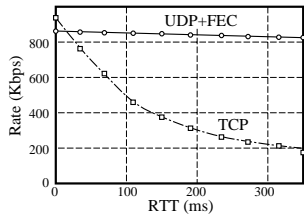


図 12 TCP と UDP+FEC の転送速度の比較 (回線速度 1Mbps)

上このような低品質の回線では十分な速度を得られない。そこで ACK を用いない UDP でパケットを連続的に送付し、パケットロスを FEC (Forward Error Correction) で修復する方法により、高速にデータ転送を行っている。同様の方法に Luby らが提唱している Tornado code があるが⁷⁾、これは繰り返し演算を行って FEC を実現している性質上ロジックが複雑でメモリ消費量が大きいという欠点があり、P2P 接続のため各端末に組み込むには適さない。

そこで、筆者らは単純に図 11 のようにデータを細かなブロックに分割し、適当なブロックの組合せに対し XOR 演算を行ったデータを、組合せ情報とともに送信し、受信側で組合せ情報に対しガウスの消去法を適用することで元データを復号する FEC を開発した。送付データを数 % 余分に送ることで、どのパケットが消失しても受信できたパケットから元データを修復することができる。図 12 に RTT による TCP と本方式の差を示す。RTT の発生は、帯域制御ソフト: Dummynet⁸⁾ を用いた。この図より RTT が大きくなればなるほど TCP に比べ本方式が優位に立つことが分かる。また実際に日米間 (RTT: 約 200ms) で測定したところ最大で TCP の 7 倍という結果が得られた。

4. DirectShare の携帯と今後

現在、DirectShare は MS-Windows2000 および Xp の PC アプリケーションの形を取っており、端末は PC に固定されている。このため、例えば出張先から従来と同じ環境を使いたい場合、出張先で専用の PC を借りて、DirectShare をインストールするか、自前のノー

ト PC を持ち込むしかない。

そこでコンピュータ環境である共有空間を気楽に持ち運ぶため、通信手段、ストレージ、認証機能などを備えた USB デバイス (PeerToken) を開発している。これは、例えば出張先で適当な PC を借り、PeerToken を挿すだけで、指紋等による本人確認完了後 DirectShare が立ち上がるというもの。共有空間のデータは PeerToken 内部のストレージに保存するため、借りた PC には痕跡を残さない。さらに携帯性を良くするため小さな形態にする予定なので、それに伴いストレージの容量が少ないことが予想される。この点に関しては 3.4 で述べたファイルリンク機能を用いて必要最小限のデータのみ保存して対応することを考えている。

現在、PC は至る所に存在する。この PeerToken が実現すると、各人は重くてかさばる PC 単体は持ち歩かずに、適当な PC に PeerToken を挿すことで、どこでも自分の作業環境を起動することが可能になる。

5. おわりに

P2P コミュニケーションシステム DirectShare について、その構想と技術要素について述べた。DirectShare では、個人のコンピュータ環境に必要なのはコンピュータではなく、付随するデータと人のネットワークであるという考えから、それを共有空間という概念で捉え、簡単に端末間で共有空間のミラーリングを行い、データ共有と遠隔会議を行うことができる。今後はさらに共有空間をコンピュータから切り離し、PeerToken という小さなデバイスに入れて、ユビキタス環境を実現するつもりである。

現在 DirectShare は社内適用による評価を行っている段階であり、10 ~ 20 拠点による遠隔地間での会議やデザインレビューなどに用いられている。

参考文献

- 1) J.Golick, "Network computing in the new thin-client age", netWorker: The Craft of Network Computing 3(2), pp30-40 (1999)
- 2) <http://www.lotus.com/>
- 3) <http://www.groove.net/>
- 4) Andy Oram, "PEER-TO-PEER Harnessing the Power of Disruptive Technologies", O'Reilly & Associates, INC (2001).
- 5) <http://www.soba-project.org/>
- 6) <http://salesgroup.fujitsu.com/plm/vps/>
- 7) M.G.Luby et al., "Practical Loss-Resilient Codes", Annual ACM Symp. on Theory of Computing, 150-159 (1997)
- 8) http://info.iet.unipi.it/~lugi/ip_dummynet/