

仮想的につながった物理的メディアを利用した接続制御

綾塚 祐二[†] 曆本 純一[†]

家庭内などでネットワークに接続される機器が増えるにつれ、ユーザが機器どうしの接続を制御・確認することがかえって繁雑になってくる。そうした接続を直感的に制御できるようにするために、ユーザが判別できる同じラベルと計算機が識別できる同じ ID を持ち、空間を越えて互いにつながっているように振る舞う組になったメディア、tranStick を提案する。1 組の tranStick は仮想的なケーブルの端点として扱え、ユーザは同じ印のついた tranStick を接続したい機器に抜き挿しすればよい。それはそのまま、接続に対するフィードバックにもなる。また、組になったメディア間のみで共有される記憶領域へのアクセスも提供できる。tranStick は鍵情報も持ち、これらの接続・共有は安全に行える。

Connection Control by Virtually Connected Physical Media

YUJI AYATSUKA[†] and JUN REKIMOTO[†]

A virtually connected medium called tranStick is described that functions both as a “virtual wire” and as a “memory card” containing a shared space. A user can connect two devices by simply placing one of a pair of tranSticks with the same identifier into each device. The tranSticks provide feedback indicating that the devices are connected; the connection to be closed or changed in the same way it would be if the devices were connected by a physical cable. A user can also access to a shared space on a network as if the space were in the tranStick. Since tranSticks contain long secret keys, the process of finding another tranStick with the same identifier can be encrypted. The tranStick approach differs from other approaches in that it provides feedback from the connection as well as serving as a medium for establishing a connection, and it enables disconnection and switchover to be done intuitively because the operations are reversible.

1. はじめに

Weiser の提唱したユビキタスコンピューティング環境の理想は、そこに多数の計算機が介在していることを意識させずに人々の活動を支援することである¹⁴⁾。しかし昨今の計算機の小型化、ネットワーク技術や無線通信技術の進歩をもってしてもその理想にはいまだ及ばず、ユーザは身の回りの様々な機器を意識的に操作する必要がある。ネットワークに接続された機器の数は今後さらに増大してゆくと思われるが、意識しなくてよいほど環境に溶け込むまでには依然として長い時間が必要であろう。また、十分に溶け込んだとしても、何らかの「目に見える指示やフィードバック」があったほうがユーザにとって分かりやすい部分も存在するであろう。

現在の環境下で、操作する必要があり、煩雑であるものの 1 つが機器間の接続である。ネットワーク化されていない機器（たとえば、プロジェクトなど）ではケーブルを引き回してきてコネクタに挿す必要がある（会議などで話者が交代するときに絡まったりしないように気をつけつつケーブルの端を受け渡しているのはよく見られる光景である）。またネットワーク化された機器でも、探索された機器の中からメニューなどを使って適切なものを選びだす必要がある。機器のリストが長くなるとこれは厄介な作業となり、またそもそもディスプレイを備えていない機器では使えない。

機器間の接続は「意図したとおりの接続ができているか」ということや、あるいは「現在、どの機器とどの機器が接続されているか」ということが容易に分かるほうが望ましい。その点では、ケーブルで機器どうしを直結する、という方式はとても優れている。接続を切断することや切り替えることも容易に行え、かつ、現在の状態が簡単に判別できる。また、プロジェクトの例を考えれば分かるように、ケーブルによる接続は

[†] 株式会社ソニーコンピュータサイエンス研究所インタラクシオンラボラトリー
Interaction Laboratory, Sony Computer Science Laboratories, Inc.



図 1 ラベルづけされたケーブル
Fig. 1 Cables labeled by numbers.

ケーブルの一端自体が一種のトークンになっており、排他的な接続を行っていることを示すのにも分かりやすい方法である。

そこで、ネットワーク化された世界にケーブルのメタファを持ち込むことを考える。上述のケーブルの利点は主に「両端とその対応関係が一瞥して明確であること」からもたらされている。一端からケーブルをたどって他の端を探さなければならない場合もあるが、途中どこを通っているかはあまり重要でないことも多い。特に多数のケーブルが存在する環境で、ケーブルを色分けしてあったり、ケーブルの両端にラベルをつけたりするのは、両端の対応関係をすぐに分かるようにするためである(図 1)。それを応用し、ユーザに分かりやすい「同じラベル(色, 形, 印など)」を持つ 1 組の識別子を挿された機器どうしをネットワーク経由で接続するという方法を提案する。

この、接続を示すための同じラベルを持つ識別子は、両端のプラグだけあり途中のないケーブルのようなものと見なせ、それをより一般化すると空間を越えてお互いに仮想的につながったメディアであるとも見なせる。仮想的につながったメディアという考え方をすることで、物理的な機器どうしの接続だけでなく、共有記憶領域への接続なども同じメタファを用いて表すことができる。この仮想的につながったメディアを tranStick と呼ぶことにする。

以下では、この tranStick を実現するための具体的な説明を述べ、これを用いて試作したいくつかのアプリケーションを紹介する。tranStick として用いるメディアに鍵情報を持たせることにより、確実な相手の認識や安全な通信が行えることも示す。

2. tranStick

tranStick は、実装面から見ると、ユーザと計算機の双方が、組になった他のメディアを容易に識別できるように設定されたメディアである。ユーザのためには同

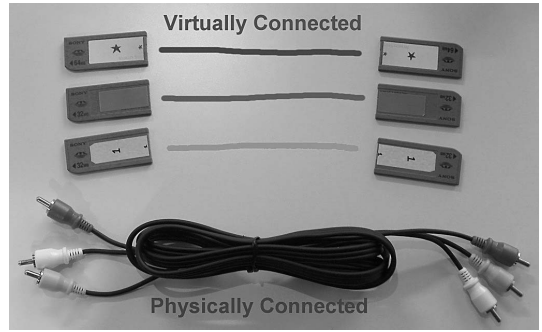


図 2 tranStick と物理的ケーブル
Fig. 2 tranSticks and physical cables.

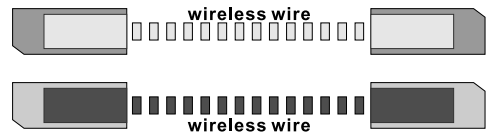


図 3 tranStick の機能 (1): 見えないワイアでつながっている
Fig. 3 Function of tranSticks (1): Connected by wireless wires.

じ色, 形, 印などをつけておく。計算機のためには、計算機が読み取れる形の ID を持たせておく。tranStick が挿される機器は、tranStick が挿されて ID が読み取れる状態で、ユーザのための識別部分がユーザに見えるようになっていたほうが望ましい。ユーザには、組になったメディアが、あたかも空間を越えてつながっているかのようなインタフェースを提供する(図 2)。

空間を越えてつながったメディアというメタファから (1) つながったメディアを介して通信ができる、 (2) つながったメディアとデータを共有している、という 2 種類の基本的な機能が想起される。前者は前節で述べた物理的なケーブルのメタファで表される機能であり、図 3 のように、メディア間に見えないワイアが存在するかのごとく働くことを意味する。この機能を、ここではワイアレス・ワイアと呼ぶ。後者は、複数のメディアが同じ記憶領域につながっている、あるいは同じ記憶領域を内在しているという働きをする機能である(図 4)。前者は同期的なつながりを提供するが、後者の場合は非同期的なつながりを提供することもできる。アプリケーションによっては、これらの 2 種類の機能を同時に用いることも考えられる。

いずれの場合も、tranStick の操作は可逆的である。すなわち、tranStick を挿すことにより接続された 2 つの機器の通信は、tranStick を抜くと切断され、tranStick を挿すことによってアクセス可能になった共有記憶領域は、tranStick を抜くとアクセス不可能になる。あるいは、tranStick を挿し換えることにより切

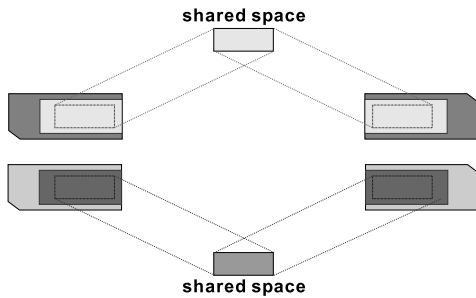


図 4 tranStick の機能 (2): 同じ記憶領域を内在する

Fig. 4 Function of tranSticks (2): Containing shared space.

り替わった接続関係は、その操作をまったく逆に行うことで、元の接続関係に戻る。

tranStick として使うメディアは、最低限、つながっている相手を識別できる ID を持っていれよ。それに加えて認証・暗号化のための鍵情報も持っている、組になった相手を確実に認証することができ、より安全な通信を提供できる。この場合の鍵は、対になった tranStick に同じ鍵が登録されていれよ、ユーザが覚えたり入力したりする必要はないので、長いものにするができる。

2.1 tranStick と他の ID つきオブジェクトとの比較

計算機により識別が可能な ID を付した物理的なオブジェクトを使って仮想的なデータをユーザが触れるようにするというアイデアはこれまでも種々の研究で用いられている。たとえば mediaBlocks¹²⁾ は ID つきの木のブロックに電子的なファイルやスライドなどを関連づけて、それを操作する。Informative Things¹⁾ はフロッピーディスクにデータ本体の代わりにそのデータをネットワーク上から引き出すためのデータを記憶させる。IconStickers⁹⁾ はデータに割り当てられた ID をバーコードとして紙に印刷し、扱う。Want¹³⁾ らは実世界の物に RFID タグをつけて扱うアプリケーションを提案している。Pick-and-Drop⁵⁾ は計算機の画面から ID つきのペンでデータを取り出し、他の画面へと運んでいく手法である。

これらのシステムは、1 個、もしくはひとつかたまりの仮想的なデータを物理的に扱うことに焦点を置いており、機器間の接続などは考えられていなかった。また、ID とそれに現時点で対応しているデータを管理するサーバが必要である。tranStick は機器間の接続を物理的に扱うことに着目し、ID つきのオブジェクトを仮想的なデータに割り当てるのではなく、組になった (同じ ID を持つ) 他のオブジェクトと結びつけて

いる。そのため、アドホックネットワーク環境などの、インターネットに接続されていない、独立した小さなネットワーク内でもインターネット環境と同様に利用が可能である。

2.2 物理的なケーブルとの比較

tranStick は物理的なケーブルの良い点 (端点をはっきりしている) を採り入れつつ、欠点を取り除いている。1 つは、距離に制約されないという点である。物理的なケーブルはあらかじめ決められた距離を越えて接続することはできないが、tranStick の場合は数 cm ほどの距離でも違う建物などでも、海の向こうでも接続ができる。これは、tranStick が Bluetooth やワイアレス USB などの無線メディアと大きく違う点でもある。もう 1 つは、物理的なケーブルと違い、絡まることはないということである。

逆に、物理的なケーブルと比べた場合の tranStick の欠点は、端点の一方 (組になったメディアのうちの片方) をなくしてしまうということである。ケーブルであれば物理的にたどって反対側の端を探すが、tranStick の場合は、誰かが自分の PC に挿したまま持って行ってしまったり、棚などの隙間に落としてしまったりすると、見つからなくなる可能性がある。しかし、接続のために用いている場合は、tranStick を他の組に取り替えても問題なく動く。データの共有を行うために用いている場合はデータ漏洩の危険があるが、これは、通常のメモ리카ードと同じである。

2.3 適用範囲

我々の主張は、すべての機器間の接続を tranStick を用いて制御すべきというものではない。tranStick はたとえば以下のような状況での制御に向いている。

- 接続対象がユーザの意図によって頻繁に切り替えられる場合 (ノート PC とプロジェクタとの接続のような場合)
- アドホックネットワークや、学会会場での LAN など、一時的なネットワークで、端末どうしの (安全な) 接続を一時的に確立したいとき
- 離れたところにいる家族の端末どうしを接続したいとき (初心者にも分かりやすく、安全で、扱いやすい接続制御が必要である)

逆に向かない例としては、端末と、オフィスの LAN 上のファイルサーバとの接続があげられる。この場合は、接続先はほとんど変わらず、また、サーバの本体はユーザが通常目にするところにはないことも多い。

3. tranStick の実装

tranStick の機能を実現するためには、メディアとそ



図5 tranStick と、それをスロットに挿した状態
Fig.5 tranSticks in slots.

れを機器に挿するためのスロット、そして組になった他のメディアを探索し、そこへの接続を行うソフトウェアなどが必要である。また、組になったメディアを生成するための手順も必要である。この章では、我々の試作した tranStick のシステムを説明する。

3.1 tranStick として使うメディア

tranStick として使うメディアにはメモリスティックを用いた。メモリカードを用いることで、リーダライタとして、PC が内蔵しているものや市販の外付けのものなどが使える。他の種類のメモリカードでもよいが、メモリスティックの場合は図5のようにリーダライタのスロットに挿した状態でラベルなどが比較的可見なものも多く、tranStick には好都合である。USB メモリと呼ばれるものでもよいが、現在のところはメモリカードのほうが小型で場所をとらない。

ID や鍵情報などのための記憶領域は、容量全体と比べるとわずかでよい。残りは、tranStick を共有記憶領域へのアクセスのために使う場合のキャッシュなどとして使うことができる。その場合は tranStick を通常のメモリカードの機能拡張版として使うこともできる。

3.2 tranStick の初期化

tranStick の初期化は、初期化用の機器（PC など）に2枚（もしくはそれ以上）の tranStick を同時に挿しておき、初期化用アプリケーションの「登録」ボタンを押すことで行う。我々の試作では、初期化用アプリケーションは数十ビット長の ID と、1K ビット前後（長さも乱数で決定する）の鍵を生成し、メモリカードに書き込む。1つの機器で組になったすべての tranStick に鍵を書き込むことで、鍵はネットワーク上を流れる必要がない。また、同時に書き込まれるようにしておくことで、意図せず同じ ID や鍵を持つ他の tranStick が生成されることを防止することができる。tranStick は割符のような役割を果たすともいえる。人間のためのラベルづけは現在のところ手作業で行わざるをえない。

ID と鍵を書き込む際に、用途に応じて、ネットワー

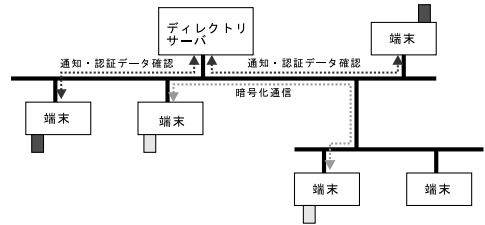


図6 ディレクトリサーバを用いた探索を行う tranStick
Fig.6 tranSticks using a directory server.

ク上の共有記憶領域のアドレスを取得し、そのアドレスや、後述するディレクトリサーバのアドレス、その他のアプリケーションの使うデータなどもあわせ書き込む。メモリカードを用いたことにより記憶領域は潤沢に利用できるため、アプリケーションプログラム自体も書き込んでおけば、tranStick のためのプログラムがまだインストールされていない機器に挿した場合にも最低限の操作で tranStick の機能が利用できる。

3.3 接続先の探索

tranStick をワイアレス・ワイアとして使う場合などには、同じ組に属する（すなわち同じ ID と鍵を持つ）他の tranStick を探す必要がある。ディレクトリサーバを用いる方法とブロードキャスト（もしくはマルチキャスト）を用いる方法があるが、ここではまず、ディレクトリサーバを用いる方法を説明する（図6参照）。ある tranStick に対して利用すべきディレクトリサーバのアドレスは、tranStick 自体に登録されている。

tranStick を検出した機器（機器 A とする）は、機器のアドレスと検出した ID とをディレクトリサーバに伝える。それを受け取ったサーバは、ランダムなチャレンジ文字列を生成し機器 A に返す。機器 A はそのチャレンジ文字列と tranStick に登録されている鍵から MD5 ダイジェストを生成し、サーバに返す。

サーバは、同じ ID を伝えてきた他の機器（機器 B とする）があれば、機器 B に渡した（機器 A に渡したものと違う）チャレンジ文字列と、それに対して機器 B が返してきたダイジェストを機器 A に伝える。機器 A は、そのチャレンジ文字列と tranStick に登録されている鍵から作られるダイジェスト値が、機器 B の生成したダイジェスト値と一致するかどうかを調べ、その結果をサーバに伝える。

そのようにして機器 A, B 双方が相手の生成したダイジェスト値を正しいものと認めると、サーバは双方に相手のアドレスを通知する。アドレスを伝えられた機器は、アプリケーションの接続先としてこのアドレスを設定し（設定の方法は後述）、必要に応じてアプ

リケーションに通信の開始を促す。

この過程の途中で tranStick が検出されなくなった場合は、それをサーバに伝え、初期状態に戻す。他の機器との通信がすでに開始されている場合は、必要に応じてそれを切断する。

この過程において、tranStick に登録された鍵はネットワーク上を流れることはなく、サーバも、個々の tranStick の持つ鍵を知っている必要はない。サーバを経由してやりとりをしてはいるが、あくまで 2 つの機器が検出した tranStick に関してお互いを認証している。ディレクトリサーバではなくブロードキャストを用いる場合も、同じ ID を持つ tranStick を検出した機器を探すためにブロードキャストを行い、あとは互いの生成したチャレンジ文字列に対する相手の答えを相互に確認する。

ディレクトリサーバは Java で書かれており、Java プログラムの実行できる環境であれば Unix 上でも Windows 上でも動く。クライアントは、現在のところ Windows 上で Visual C++ で書かれたものが動いている。プロトコルは独自のものであるが、HTTP のリクエストと返答に埋め込めるように設計され、ファイアウォールを越えられるようにすることも容易である。

3.4 接続先の設定

アプリケーションが最初から tranStick に対応して作られていて 3.3 節の接続先の探索も行えるようになっていれば、もちろん、問題なく接続先を切り替えて通信を行うことができる。それ以外のアプリケーションでも、外部から接続先の変更を行うか、あるいはプロキシを用いることで接続先を切り替えることができる。

プロキシは tranStick が挿されるそれぞれの機器で動き、アプリケーションは接続先を localhost (127.0.0.1) に設定しておく。プロキシはアプリケーションからリクエストがあるごとに、tranStick を用いて設定された現在の接続先を確認し、リクエストをそこへと中継する。TCP/IP や UDP/IP に対応したこのようなプロキシを、我々は Java で試作した。

通信先と通信元双方が (もしくはアプリケーションによっては通信元のみでもよい) tranStick 対応として作られているかプロキシを用いている場合、tranStick に登録された鍵を用いた暗号化を行うこともできる。次章では実際に暗号化した通信を行うアプリケーションも紹介する。



図 7 tranScreen : 左のノート PC がプロジェクタにつながっている

Fig. 7 tranScreen: the left PC is connected to a projector via the network.

4. アプリケーション例

この章では実際に tranStick を用いたアプリケーション例を紹介する。新たに作成されたアプリケーションもあるが、アプリケーション自体は既存のものをもそのまま無改造で使っているものもある。また、ディレクトリサーバを用いた実装、マルチキャストによる探索を用いた実装の両方が含まれている。

4.1 ワイヤレス・ワイアとしての利用

同じ tranStick が挿されている機器どうしをつなぐアプリケーション例として 3 つを紹介する。tranStick 対応アプリケーションとして作成した画面転送機能つきプレゼンテーションツール、ストリーミング再生のできる音楽プレーヤとプロキシを用いた音源制御、そしてビデオ通話アプリケーションの接続切替えである。これらのアプリケーションは、ディレクトリサーバを用いる方式の接続先探索モジュール (このモジュールを transConnect と呼ぶ。Visual C++ で記述され、PC 上の Windows で動く) を共有して動いている。

4.1.1 tranScreen

このアプリケーションは表示している画面を JPEG 画像として指定したアドレスに送信することのできるプレゼンテーションツールと、送られてきた JPEG 画像を指定された画面位置に表示するサーバからなる (双方とも Java で記述されている)。プロジェクタにつながったケーブルを PC につなぐ代わりに、プロジェクタ (の役目を果たすサーバ) に挿されているのと同じ tranStick を PC に挿すことでその PC 内のプレゼンテーション資料がプロジェクタ画面に映し出される (図 7)。話者が交代する場合も、PC に挿された tranStick を次の話者に渡し、その PC に挿せばよい (もちろん、それぞれの PC は有線/無線で LAN に接続されているという前提である)。取り外して渡し、

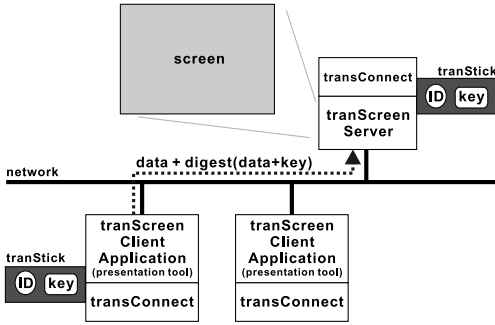


図 8 tranScreen の構成
Fig. 8 Architecture of tranScreen.



図 9 tranSound : スピーカとサーバ
Fig. 9 tranSound: speakers and server.

他のものに挿す」という部分は物理的なケーブルの場合と同じであるが、ケーブルの長さや、絡まったりしないかというようなことをいっさい気にする必要がない。2 台の PC、あるいは 2 台のプロジェクトに同じ tranStick を挿してつなごうとしても何も起こらないのは物理的なケーブルの場合と同じである。PC とプロジェクトのどちらかの tranStick を抜くと、サーバがそれを検知してプロジェクトには “No Input” の画面が現れる。

構成は図 8 のようになっており、送信側は transConnect が設定したアドレスに対して画面データを転送し、その最後に、データと tranStick の持つ鍵から生成されるダイジェスト値を付加する。受信側はそのダイジェスト値が正しいものであることを確認したときのみ、そのデータを表示する。この試作ではデータの暗号化は行っていないが、tranStick の持つ鍵を用いて暗号化することも簡単に行える。今後は VNC⁸⁾ などのプロトコルにも対応することを考えている。

4.1.2 tranSound

音楽を MP3 形式で送信するサーバと、ストリーミング再生機能を持つ音楽プレーヤとの接続を制御するアプリケーションである(図 9)。サーバには SLIM DEVICE 社の SLIMP3 サーバを、プレーヤは SLIMP3 を用いている。プレーヤは tranStick で制御されるプロキシをサーバとして音楽を再生するように設定されている。プロキシは transConnect が設定したアドレ

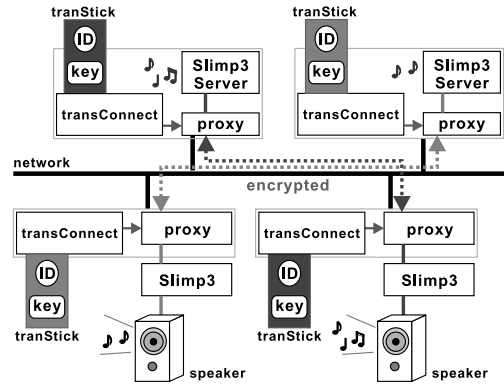


図 10 tranSound の構成
Fig. 10 Architecture of tranSound.

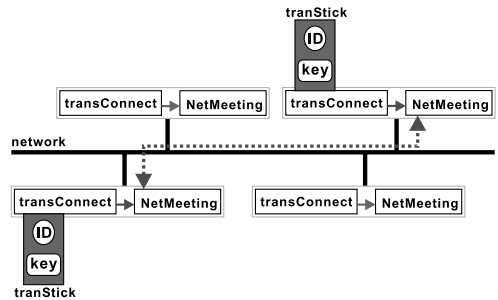


図 11 tranStick で制御された NetMeeting の構成
Fig. 11 NetMeeting Controlled by tranSticks.

スのサーバにアクセスし、データをプレーヤに中継する。この実装ではサーバ側にもプロキシがあり、プロキシ間の通信は tranStick 内の鍵を使って暗号化することもできる。図 10 が構成図である。SLIMP3 サーバやクライアント自体にはなんら変更は加えていない。ユーザは、たとえば、自室のスピーカに挿されていた tranStick を居間へ持っていきそのスピーカに挿すことにより、自室で聴いていた音楽の続きを居間で聴くことができる。また、CD プレーヤに挿されていた tranStick を抜いてラジオに挿すことで、音楽のソースを切り替えることができる。

4.1.3 NetMeeting の制御

NetMeeting は Microsoft 社のビデオ通話アプリケーションである。他のアプリケーションからメッセージを送ることにより接続先を指定することができるので、tranStick で示される接続先が検出された、またはなくなった際に transConnect がメッセージを送り、操作する(図 11)。NetMeeting 自体には手を加えていない。

tranScreen, tranSound の例では接続される機器の間にサーバ・クライアントの関係があったが、この例で

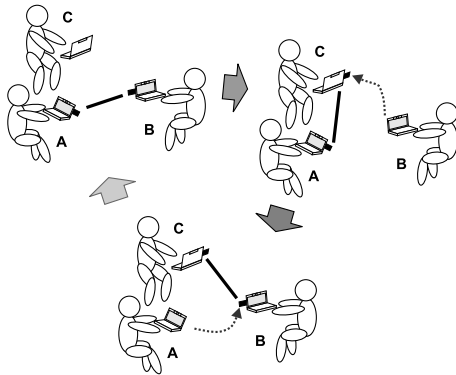


図 12 tranStick で三者間をつなぎ替える
 Fig.12 Changing connection among three.

は接続した端末どうしは対等である．そのため，図 12 のように，初めに端末 A と B に同じ tranStick を挿してつないでいたものを，B に挿されていた tranStick を端末 C に挿し換え A と C をつなぎ，さらに A に挿されていた tranStick を端末 B に挿し換え C と B をつなぎ，ということも可能であり，ここでも，物理的なケーブルと同じメタファが実現できていることが確認できる．このような簡便さは，小型ゲーム機を用いた対戦型のゲームの対戦相手の設定などに有効であろう．

4.2 共有記憶領域へのアクセス

次に，tranStick を用いて共有された記憶領域へのアクセスを提供するアプリケーションを紹介する．サーバ上にある共有記憶領域を利用するものと，Peer-to-Peer でデータを同期するものの 2 種類を試作した．

4.2.1 tranSpace

tranSpace は端末で WebDAV のプロキシとして起動する（これは VisualC++ で記述され，PC の Windows 上で動く）．tranStick が挿されていないときはこのプロキシは働かないが，tranStick を検出すると，http://localhost をターゲットとした Windows の Web フォルダを起動し，tranStick に初期化時に設定された URL へと接続するプロキシとして働く．ファイルを読み書きする際には，tranSpace は tranStick 内の鍵を用いてデータを暗号化・復号する（図 13）．データの読み書きごとに鍵情報を確認するので，tranStick を外すと最早アクセスはできない．それ以外の使い勝手は Web フォルダとまったく同じである．サーバは，Apache 1.3 と mod_dav を用いた WebDAV サーバである．

このアプリケーションでは，ユーザは tranStick を通常のメモリカードと同じように使いつつ，ネットワーク上の共有記憶領域へとアクセスができる．デー

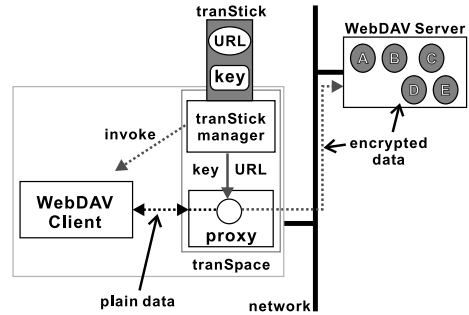


図 13 tranSpace の構成図
 Fig.13 Architecture of tranSpace.

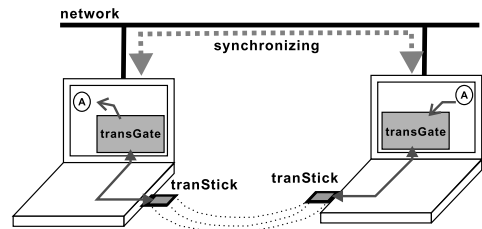


図 14 transGate の概念図
 Fig.14 Concept of transGate.

タは暗号化され，しかもデータを保存するサーバ上にも鍵の情報はないので，安全なデータの共有ができる．そして，メモリカードに直接データを入れて他者に渡す場合と比べて，容量を気にしなくてもよい，後からデータを追加できる，双方向にデータをやりとりできる，などの利点がある．もちろん，「共有」ではなく 1 つの tranStick のみでしかアクセスできないネットワーク上の記憶領域として使うことも可能である．その場合には，tranStick は暗号化の機能を付加された mediaBlock¹²⁾ のように使える．

4.2.2 transGate

tranSpace ではネットワーク上のサーバへのアクセスを行っていたが，このアプリケーションでは tranStick が挿された機器どうしが直接通信し，複数の tranStick 内のデータの同期を行おうとする．すなわち，組になった tranStick のうち 1 枚だけがどれかの機器に挿されている状態では単なる普通のメモリカードと同じであるが，ネットワーク上に対になった他の tranStick が見つかり，お互いに他方になかったファイルを送信しよう．つまりユーザには内部につながったメモリカードのように見える（図 14）．

図 15 の画面では，transGate のブラウザが動いており，ユーザはここにファイルをドラッグ&ドロップできる．対になった tranStick が見つかり，通信が行われる状態（接続状態）のときには左上に “connected



図 15 transGate : 2 台の PC が「同じ」tranStick をアクセスしている

Fig. 15 These two PCs access to the “same” shared space in tranSticks.



図 16 transGate のブラウザ画面 (部分)

Fig. 16 transGate browser.

to ...” の文字が表示され (図 16), 他の機器から転送されてきたファイルのアイコンやサムネイルが画面にフェイドインしてくる。接続状態では, 相手がブラウザ上で移動したファイルの位置も反映され, また相手がブラウザ外へ移動もしくは削除したファイルはフェイドアウトし, あたかもこのブラウザ画面が直接相手へとつながっているような動きをする。

transGate では, マルチキャストを用いて対になった他の tranStick を探索する方式を実装した。同期のためのやりとりは, TCP/IP を用いたプロトコルで行っている。通信開始時の同期で, 名前が同じで内容が違うものが存在した場合には, 双方の名前を変える (重ならない文字列を付加する) ようになっている。

拡張として, tranSpace の機能と transGate の機能を組み合わせ, ネットワークに接続してなくても通常のメモリカードとして使え, 接続していれば容量以上のデータを共有することのできる媒体として使えるようにすることもできる。この場合は, tranStick がネッ

トワーク上の共有記憶領域のキャッシュとして働く。

5. 議 論

5.1 アプリケーションの選択

4 節で紹介したように, tranStick を用いたアプリケーションには様々なものが考えられる。tranStick を挿される機器が単機能なもの (たとえばスピーカ) であれば, 他の機器と接続された際に起動すべきアプリケーションは簡単に決まるが, PC どうしを接続した場合などは自動的に 1 つに決定することは難しい。transConnect では, それぞれの機器においてあらかじめ実行可能な機能を登録してあり, 他の機器と一致する機能を起動するようにしている。一致するものが複数ある場合には, あらかじめ指定してある優先度に基づいて選択される。

他の解決方法はいくつか考えられる。1 つは, 選択可能なリストをユーザに提示し, 選んでもらうということである。2 つ目は, tranStick は接続先の情報だけを提供し, 実際の接続はアプリケーションからユーザが設定する (たとえば, 接続先の名前として “tranStick” を選ぶ, など) というものである。3 つ目として, アプリケーション専用の tranStick を用意し, たとえば NetMeeting 接続用に設定された tranStick を使うと NetMeeting が起動する, という方法である。専用でなくても, tranStick の側に優先度を登録しておくことも考えられる。tranSpace や transGate など共有記憶領域を使うための tranStick は混乱を避けるために専用になっていたほうがよいと思われる。しかし, Peer-to-Peer のやりとりを行う transGate の場合は, 明示的に指定した場合は他のアプリケーションでの使用も可能, としても混乱なく使うこともできるだろう。

5.2 1 組が 3 枚以上の場合

ディレクトリサーバやアプリケーションなどは, 基本的に一対 2 枚の場合で説明し, 実装もその場合のみを考慮したものになっているが, 1 組 3 枚以上の tranStick を用いることも考えられる。もちろん, アプリケーションによっては難しい場合もあるが, 自然に拡張可能な場合も多い。たとえば, tranScreen で 1 つの画面に 2 つ以上の PC の出力をつないだ状態などはどういう動作が妥当かは難しいが, 逆に 1 つの PC の出力を 2 つの画面につなぐのであれば, 双方の画面に PC の出力を表示するのが妥当である。

3 枚以上検知された場合の動作の明確な基準が作れないアプリケーションでは, 可能ならばエラーや警告のメッセージをユーザに見せたほうがよいであろう。

2枚の場合でも、tranScreenのクライアントどうしをつなごうとした場合などには、現在の実装では何も起こらないだけであるが、警告を出すことも考えられる。

1 対多の接続以外にも、たとえば音源、エフェクタ、スピーカのような組合せの3つに同じ tranStick が挿された場合のように、3つ以上の機器の組合せが一意に決まるものであれば、自動的に「音源 エフェクタ スピーカ」のような接続を行うようにすることも考えられる。一意に決まる場合と決まらない場合とをどう区別し、どうユーザに知らせるかは問題である。また、接続している順序に対するフィードバックをどう表現するかも考慮しなければならない。

5.3 紛失した場合など

2章で述べたようにディスプレイの接続のためなどに tranStick を用いる場合、物理的なケーブルを使う場合に対する明らかな欠点は、組になった他方を紛失するかもしれない、ということである。物理的につながっていればケーブルをたどって探すことができるが、tranStick の場合は、誰かが自分の PC に挿したまま持って行ってしまったり、棚などの隙間に落としてしまったりすると、見つからなくなる可能性がある。これは課題の1つであるが、tranStick を他の組に取り替えても問題なく動くので、安全性は損なわれない。

tranSpace や transGate の場合には、紛失や盗難により悪意ある者の手に渡った場合にデータ漏洩の危険がある。これは、通常のメモリカードと同じである。また、tranStick を挿した機器が信用できない場合、tranStick 内の鍵などのデータをコピーされてしまうことも考えられるが、これは著作権保護技術などを利用して回避することを考えている。

5.4 我々の他のアプローチとの比較

実世界での物理的な操作や状況に基づいて複数の計算機間の通信を制御するという tranStick の機能は、我々のこれまでの成果である Gaze-Link¹⁷⁾ や FEEL⁷⁾ と同じ系譜に属するものである。それらと比べた際の tranStick の最大の特徴は、接続の指示が同時にそのままフィードバックになっている、すなわち tranStick を挿してあることが接続している状態をそのまま表しているという点である。また、tranStick の操作は可逆的、すなわち接続のための操作を逆に言えば切断ができる。これらは、接続の解除（切断）や切替えも直感的に行えることを意味する。

Gaze-Link も tranStick 同じように物理的に接続先を指定し、接続先を後から見ることもできるが、それはユーザが持ち歩く端末側からのみであり、接続先から接続元を確認することはできない。ユーザの端末上

で接続先を見て切り替えることは簡単であるが、その接続を他の端末に渡すようなことはできない。また、端末に取りつけられたカメラを用いることを前提としているので、据え置き型の機器どうしには適応しにくい。tranStick の場合はメディアに鍵を持たせることにより、暗号化なども容易に行えるという点も Gaze-Link に対する利点である。

機器どうしを近づけることで接続を行う FEEL と比べると tranStick を用いた操作は1段階間的であるが、これも据え置き型の機器どうしでも簡単に適用することができるという点で違いがある。ネットワーク上の共有記憶領域へのアクセスも提供できる点も Gaze-Link や FEEL とのメタファの違いによりもたらされている。これらの技術は排他的なものではなく、用途に応じて使い分けられていくものであろう。

FEEL と関連して、我々は実世界での物理的な操作をもとに安全な通信を提供するというモデルも提案している¹⁶⁾。tranStick も「仮想的につながっている」ことで、このモデルの認証チャネルを実現していると思なせる。

6. 関連研究

ネットワークでつながれた機器間の接続を名前やアイコンでなく、より直接的・直感的に行うための方法は他にもいくつか提案されている。gesturePen¹¹⁾ は接続先を赤外線通信で探す。Touch-and-Connect¹⁸⁾ では、それぞれの機器がボタン1と2を備え、機器AとBを接続したいときは機器Aのボタン1と機器Bのボタン2を設定された時間内に押す。同様な方法として複数の機器の特定のボタンを同時に押す、我々の SyncTap⁶⁾ もある。これらも、機器どうしを接続する時点までを考慮しており、接続していることのフィードバックや、切断、接続の切替えはサポートしていない。

Synchronous Gestures⁴⁾ は2つのタブレットPCを「ぶつける」ことで接続を行う。PCどうしが離れると、接続は解除される。この方法では2つの機器のうち少なくとも1つが持ち運ばなければならない。加えて、接続する機器の接続中の位置関係がきわめて限定される。

ユーザと計算機が識別できる「同じラベル」でつながるべき相手を選ぶこと自体は広く行われている。たとえば、複数のワイアレス・マウスの混信を避けるために送受信双方で同じIDをセットしたり（このIDは通常は目立たないところにある）、ソフトウェア的なラベルとしては IEEE802.11 で使われる ESSID など

がある。しかしこれらはラベル自体を取り外して扱ったり、受け渡したりするようなことは考えられていない。逆に、Microsoft 社の Windows Connect Now では、ある PC の無線 LAN の設定を USB メモリにコピーし、それを他の PC に挿すことにより、既存の設定を容易に他に移行できるようにしている。

仮想的につながった機器という考え方は、計算機とネットワークを利用した遠隔地とのコミュニケーションの研究で用いられてきている。たとえば、InTouch²⁾ や Meeting Pot¹⁵⁾、Peek-A-Drawer¹⁰⁾ などがあげられる。これらはそれぞれ 1 個のアプリケーションであるが、tranStick は、仮想的につながった機器ではなく仮想的につながったメディアであり、その上にアプリケーションを構築する土台を提供している。上述の例でのつながる先を選択するのにも、もちろん利用することができる。

共有記憶領域へのアクセスと似た例としては、ある Web ページへのアクセスを、端末に特定のメディアが存在するときのみに制限する ConnecteD³⁾ という技術がすでに一般で使われている。これは、市販の音楽 CD を鍵として、特典コンテンツへのアクセスを制御するために用いられている。この場合メディアは単に鍵として用いられており、また、ユーザが独自の共有領域を持つようなサービスは今のところ行われていない。

7. ま と め

ユーザに判別しやすい同じラベルと計算機で識別しやすい同じ ID を持ち、空間を越えて互いにつながっているかのごとく振る舞うメディア、tranStick を提案した。tranStick の 1 つの用途は仮想的なケーブル（ワイアレス・ワイア）であり、ネットワークでつながった機器どうしの接続・切断の設定を物理的な操作で直感的に行うことができ、同時に接続関係のフィードバックにもなっている。また、共有記憶領域へのアクセスを提供することもできる。tranStick には鍵情報も登録されており、接続すべき相手の探索や通信自体を安全に行うことができる。今後は、より日常的に使えるアプリケーションを増やし、実際の使用感などを探っていく。

参 考 文 献

- 1) Barrett, R. and Maglio, P.P.: Informative Things: How to attach information to the real world, *UIST '98*, pp.81–88 (Nov. 1998).
- 2) Brave, S., Ishii, H. and Dahley, A.: Tangible

- interfaces for remote collaboration and communication, *Computer Supported Cooperative Work (CSCW'98)*, pp.169–178 (1998).
- 3) Sony Music Entertainment: ConnecteD. <http://www.connected.ne.jp/index.html>
- 4) Hinckley, K.: Synchronous gestures for multiple persons and computers, *UIST 2003*, pp.149–158 (Nov. 2003).
- 5) Rekimoto, J.: Pick-and-drop: A direct manipulation technique for multiple computer environments, *UIST '97*, pp.31–39 (Oct. 1997).
- 6) Rekimoto, J., Ayatsuka, Y. and Kohno, M.: SyncTap: An interaction technique for mobile networking, *Mobile HCI 2003*, pp.104–115 (Sep. 2003).
- 7) Rekimoto, J., Ayatsuka, Y., Kohno, M. and Oba, H.: Proximal interactions: A direct manipulation technique for wireless networking, *INTERACT 2003*, pp.511–518, IFIP (Sep. 2003).
- 8) Richardson, T., Stafford-Fraser, Q., Wood, K.R. and Hopper, A.: Virtual network computing, *IEEE Internet Computing*, Vol.2, No.1, pp.33–38 (1998).
- 9) Siio, I. and Mima, Y.: IconStickers: Converting computer icons into real paper icons, *HCI International '99*, pp.271–275, Lawrence Erlbaum Associates (Aug. 1999).
- 10) Siio, I., Rawan, J. and Mynatt, E.: Peek-A-Drawer: communication by furniture, *Conference Extended Abstracts on Human Factors in Computer Systems (ACM CHI 2002)*, pp.582–583, ACM (Apr. 2002).
- 11) Swindells, C., Inkpen, K.M., Dill, J.C. and Tory, M.: That one there! pointing to establish device identity. *UIST 2002*, pp.151–160 (Oct. 2002).
- 12) Ullmer, B., Ishii, H. and Glas, D.: MediaBlocks: Physical Containers, Transports, and Controls for Online Media, *Proc. SIGGRAPH '98*, pp.379–386 (1998).
- 13) Want, R., Fishkin, K.P., Gujar, A. and Harrison, B.L.: Bridging Physical and Virtual World with Electronic Tags, *Proc. CHI '99*, pp.370–377, ACM (May 1999).
- 14) Weiser, M.: The computer for the twenty-first century, *Scientific American*, pp.94–104 (Sep. 1991).
- 15) 椎尾一郎, 美馬のゆり: Meeting Pot: アンビエント表示によるコミュニケーション支援, インタラクション 2001 論文集, pp.163–164. 情報処理学会 (Mar. 2001).
- 16) 河野通宗, 長健二郎, 綾塚祐二, 暦本純一: ユーザインタフェースを活用したセキュリティモデ

ル, インターネットコンファレンス 2002 論文集, pp.43-51. 日本ソフトウェア科学会 (Nov. 2002).

- 17) 綾塚祐二, 松下伸行, 暦本純一: 実世界指向ユーザインタフェースにおける「見ているものに接続する」というメタファ. 情報処理学会論文誌, Vol.42, No.6, pp.1330-1337 (2001).
- 18) 岩崎陽平, 河口信夫, 稲垣康善: Touch-and-Connect: 無線機器間の接続指示手法, インタラクティブシステムとソフトウェア IX WISS2001, pp.201-202. 日本ソフトウェア科学会, 近代科学社 (Dec. 2001).

(平成 16 年 10 月 18 日受付)

(平成 17 年 2 月 1 日採録)



綾塚 祐二 (正会員)

1970 年生. 1993 年東京大学理学部情報科学科卒業. 1995 年東京大学大学院理学系研究科情報科学専攻修士課程修了. 1998 年同博士課程中退, 同年ソニー株式会社に入社.

1999 年より株式会社ソニーコンピュータサイエンス研究所インタラクショナルラボラトリーに所属. 実世界指向ユーザインタフェースを中心としたユーザインタフェースの研究に従事. 日本ソフトウェア科学会, ACM 各会員.



暦本 純一 (正会員)

1961 年生. 1986 年東京工業大学大学院情報科学研究科修士課程修了. 日本電気, アルバート大学を経て, 1994 年より株式会社ソニーコンピュータサイエンス研究所に勤務.

現在, 同研究所インタラクショナルラボラトリー室長. 理学博士. ヒューマンコンピュータインタラクション全般, 特に実世界指向インタフェース, 拡張現実感, 情報視覚化等に興味を持つ. ACM, 日本ソフトウェア科学会各会員. 1990 年情報処理学会 30 周年記念論文賞, 1998 年 MMCA マルチメディアグランプリ技術賞, 1999 年情報処理学会山下記念研究賞.