

## 直接操作技法を用いる共同作業環境でのアクセス権限管理手法

田中友久<sup>†</sup> 渡邊光洋<sup>†</sup> 有賀健一<sup>†</sup>  
藤原隆平<sup>†</sup> 中本幸一<sup>†</sup>

ID を持つペン操作により情報を端末間で移動させるといった、複数のユーザで共同作業を行える実世界指向インタフェースの操作技法が提案されている。こうした共同作業環境における情報共有のためのアクセス権限管理手法として、パーミッションによるウィンドウ保護、縮退操作、委譲操作の概念を提案する。また、提案手法の応用例として実装した、ファイル共有に関するアプリケーションを紹介する。

### Access Control Methods with a Direct Manipulation Technique for Collaborative Environments

TOMOHIISA TANAKA,<sup>†</sup> MITSUHIRO WATANABE,<sup>†</sup> KENICHI ARIGA,<sup>†</sup>  
RYUHEI FUJIWARA<sup>†</sup> and YUKIKAZU NAKAMOTO<sup>†</sup>

Direct manipulation techniques based on the real-world oriented interface, like transferring information between computers by an identifiable pen, have been proposed and enable us to collaborate. This paper proposes a concept of window protection by permission, degeneration and delegation, as access control methods to share information for the collaborative environment. We also show application of file sharing, which is implemented as an example of our proposal methods.

#### 1. はじめに

複数のユーザが複数のコンピュータで容易に共同作業を行える実世界指向インタフェースの直接操作技法がいくつか提案されている。Pick-and-Drop<sup>1)</sup> では ID をもつペンによる操作で、コンピュータ間で情報を容易に移動することを実現している。図 1 のように、複数のユーザがそれぞれ個人用端末の情報を情報キiosk 端末に提示、受け渡したり (図 1 左)、相手の端末を直接触れて情報を授受しながら (図 1 右)、共同作業を進めることができる。

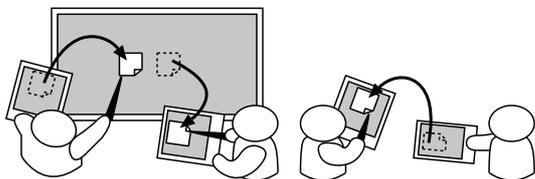


図 1 共同作業の例。複数のユーザがそれぞれ個人用端末の情報を受け渡す。左は情報キiosk 端末を介しての、右は相手の端末を直接触れるインタラクション

このように実現された共同作業環境では、コンピュータはネットワークで接続されているにも関わらず、操作対象のコンピュータのネットワークアドレス、コンピュータ名などの記号的な概念をユーザは意識する必要がない。ユーザはただ操作したいコンピュータの画面のオブジェクトをポインティングデバイスで選択するだけで、複数のユーザが複数のコンピュータを協動的かつ簡単な方法で取り扱うことができる。現在、IEEE 802.11b や Bluetooth などの無線技術と、各種 PDA や Tablet PC などのペン操作型コンピュータが一般的になりつつあり、上記の提案が実用化されるための基盤は整ったと言える。

ところが、提案されている共同作業環境ではコンピュータをすべてのユーザで共有することを前提にしているものが多く、セキュリティやプライバシーについてはあまり議論されていない。例えば、情報キiosk 端末に情報を提示することと、その情報自体のコピーや改変を許可することは別である。また、個人用端末 (PC や PDA など) に情報を提示して相手に見せたとしても、その端末の操作を許可したことにはならない。このように、ただコンピュータを共有するだけでは、そこにある情報もすべて共有されることになり、セキュリティやプライバシーの面で実用上問題がある。

<sup>†</sup> NEC ネットワークス ネットワークス開発研究所  
Development Laboratories, NEC Networks

実際には、複数のユーザが1つのコンピュータ(の画面)で共同作業を行う場合、誰がどの情報をどのように利用できるのかを定義することになるだろう。例えば、あるユーザが自分のコンピュータの特定の情報だけを、別の特定のユーザにアクセスさせたい場合は、自分のコンピュータに対して「どのユーザにどのようなアクセス権をもたせるのか」を事前に設定するなどの必要がある。

従来からアクセス権限の制御方法として、OSではアクセスコントロールリスト(ACL)による制御などが知られている。この方法では基本的に、特定のユーザに許可する(あるいは許可しない)操作を制御対象(リソース)についてそれぞれ設定することで制御を実現する。一般的なマルチユーザOS(UNIXやWindowsなど)は、ユーザをユーザ名(やユーザID)などの識別子で管理し、それぞれのユーザ名に適切なアクセス権限を対応付けることでACLを設定している。

しかし、実世界指向インタフェースの直接操作技法を用いる共同作業環境にACLを適用するには次のような問題がある。

- あるユーザが別のユーザに自分のアクセス権限の全部または一部を与えようとしても、当事者同士の合意だけではなく、スーパーユーザ(管理者)権限による操作が必要であることが多い。
- 設定されたアクセス権限が意図しているものかどうかを、実際に試して確認することなく、直感的に理解することは難しい。

特にアドホックにユーザが構成される場合は、共同作業中に新しくユーザが加わる度に、コンピュータに対してユーザを登録して、適切なアクセス権限を設定する必要があるため、上記の問題がある場合には作業を円滑に進めることができなくなる。したがって、このような煩雑な設定が必要な環境では、実世界指向インタフェースの長所を活かすことができないと考えられる。

本稿では、特定の個人と強く関係付けられたペン型端末をポインティングデバイスとする共同作業環境において、管理者権限に頼ることなくユーザ同士で選択的に委譲できるアクセス権限の管理手法を提案する。

## 2. 共同作業環境

以降では、デスクトップ画面とペン型端末で構成される共同作業環境を考える。

デスクトップ画面はマルチウィンドウシステムのGUIで構成され、ペン型端末にポインティングされたときに、デスクトップ画面側でそのルートウィンドウ

における座標と共にペン型端末のIDを検出できる必要がある。

ペン型端末はデスクトップ画面のポインティングデバイスであり、ユニークなIDをもつ。ペン型端末は文献1)のペンと同様にピックとドロップの操作に使用できる。異なる点として、個人用の端末であり、その所有者(ユーザ)に専有される点があげられる。

デスクトップ画面は検出したペン型端末のIDからユーザを特定できる。

## 3. アクセス権限の管理手法

本稿の提案手法は、共有されている状態からアクセス権限を減らしたり、禁止する操作を決めていくのではなく、逆に保護されている状態から許可を与えていき、必要な部分だけ共有するという方針に基づくアクセス権限の管理手法である。また、許可を設定するという操作を「どのような許可を与えるのかを指定する操作(縮退操作)」と「誰に与えるのかを指定する操作(委譲操作)」に分割する。縮退操作では許可を設定する前にそれが意図しているものかどうか確認できること、委譲操作では仮想世界(コンピュータの世界)だけではなく、実世界におけるユーザ間のインタラクションも利用することに特徴がある。

以下、保護と許可に関してウィンドウ保護とパーミッションの概念を、許可を与えるという操作に関して縮退操作、委譲操作の概念を説明する(具体的な操作手順は4節で述べる)。

### 3.1 ウィンドウ保護

ユーザは自分のペン型端末でデスクトップ画面にウィンドウを生成することができる。生成直後のウィンドウはそのペン型端末だけで操作可能となり、その他のユーザの操作を無視する(他のユーザとは共有していない)。以下これをウィンドウ保護と記述する。

個人用端末や共用端末のデスクトップ画面がログイン画面のときに、ペン型端末でタップすることでログインして、そのユーザがログアウトするまでその端末を占有する(その他のペン型端末を受け付けない)ことができ、これによりウィンドウ保護が実現できる。

しかし、情報キオスク端末など複数のユーザが1つのデスクトップ画面で同時に作業する端末にウィンドウ保護を適用すると、情報キオスク端末のルートウィンドウを生成したユーザだけが新たにウィンドウを

---

ペン型端末のIDをユーザIDとして扱い、そのユーザIDからユーザを特定する。  
最初にログインしたユーザなど、情報キオスク端末の管理者となるユーザ。

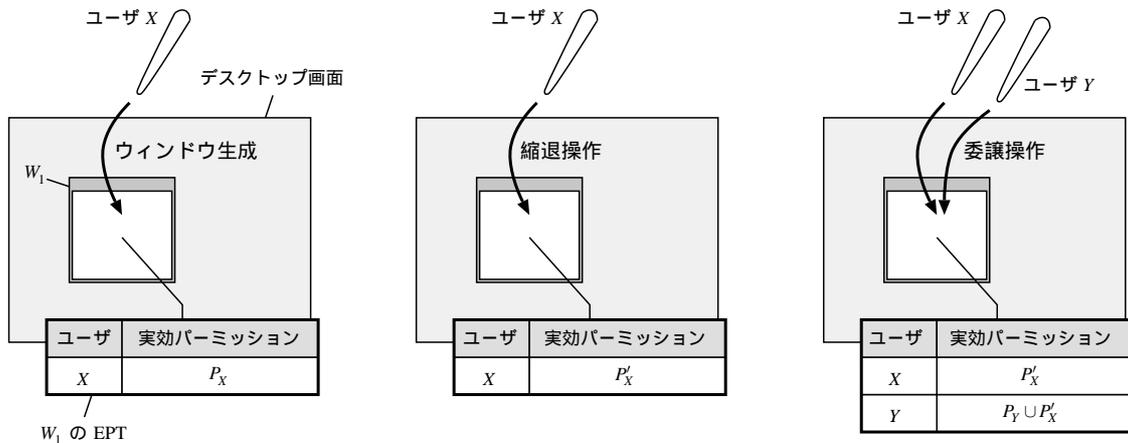


図2 ウィンドウ生成，縮退操作と委譲操作

生成できることになり，共同作業が成り立たない．このような場合は，デスクトップ画面のルートウィンドウを例外的に保護しないこととして，ユーザがそれぞれ自分のペン型端末で情報キオスク端末のルートウィンドウに特定の操作を行って，自分だけが操作できるウィンドウを生成できるようにする．

ウィンドウ保護では，ユーザがウィンドウを破壊するまでデスクトップ画面のリソースが占有される．これが問題になる場合は，ウィンドウへの最後の操作から適当な時間が経過したら，自動的にウィンドウを閉じるなどの手段でリソースを解放すればよい．

### 3.2 パーミッション

ウィンドウ保護（と後述する縮退操作，委譲操作）を実現するため，ユーザに対してパーミッションを定義する．ユーザのパーミッションはデスクトップ画面がそのユーザに許可する操作の集合（デスクトップ画面におけるユーザのアクセス権限）である．さらに実効パーミッションをウィンドウがユーザに許可する操作の集合とし，ウィンドウの属性としてユーザと実効パーミッションの対応表（EPT）を定義する．ウィンドウを生成するとユーザの実効パーミッションはそのユーザのパーミッションで初期化されるものとする．

ユーザがウィンドウを操作する場合，デスクトップ画面はペン型端末のIDと座標を検出して，ユーザとウィンドウをそれぞれ特定し，EPTから求めた実効パーミッションの範囲でユーザの操作が実行される（ユーザは実効パーミッションをもたない操作を実行できない）．また，EPTに存在しないユーザはそのウィ

ンドウに対していっさい操作することができない．

例えば，パーミッション  $P_X$  をもつユーザ  $X$  がウィンドウ  $W_1$  を生成すると，そのEPTは図2（左）のように初期化され， $X$  は  $W_1$  に対して  $P_X$  の範囲で操作が可能になる．

### 3.3 縮退操作

ユーザはウィンドウ毎に自分の実効パーミッション（可能な操作の範囲）を「減らす」操作を実行できる．以後この操作を縮退操作と記述する．縮退操作は不可逆操作である．

生成直後のウィンドウでは，そのウィンドウを生成したユーザが自分のアクセス権限の範囲で自由に操作できる．ユーザがそのウィンドウに縮退操作を実行すると，自らウィンドウに対する操作を制限することになる．このように縮退したアクセス権限は，後述する委譲操作で他のユーザに与えることができる．

例えば，ユーザ  $X$  がペン型端末でウィンドウ  $W_1$  に縮退操作を行うと， $W_1$  のEPTは図2（中央）のように書き換えられ，ウィンドウ  $W_1$  でのユーザ  $X$  の実効パーミッションは  $P'_X$  ( $P'_X \subset P_X$ ) になる．

### 3.4 委譲操作

ユーザはウィンドウ毎に自分の実効パーミッションを他のユーザに与えることができる．以後この操作を委譲操作と記述する．

例えば，ユーザ  $X$  がペン型端末で図2（中央）のウィンドウ  $W_1$  に対して委譲操作を行い，実効パーミッション  $P_Y$  をもつユーザ  $Y$  が  $X$  の実効パーミッション  $P'_X$  を獲得すると， $W_1$  のEPTは図2（右）のようになる．このとき， $Y$  は  $W_1$  に対して  $P_Y \cup P'_X$  の範囲で操作が可能になる（図3）．

タップ，ドラッグなどのジェスチャや，アイコンのドロップなどの操作．

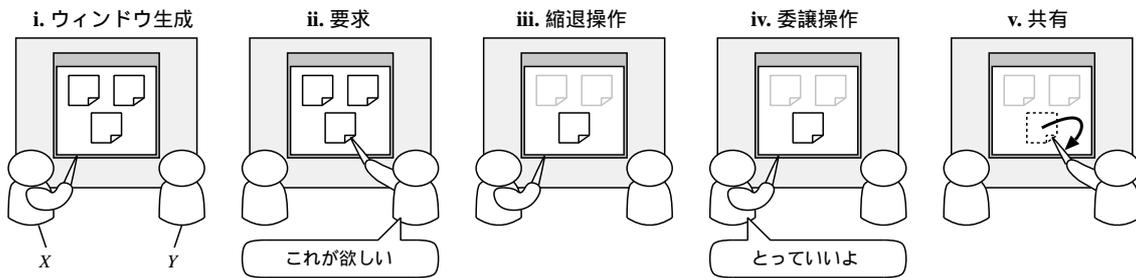


図4 インタラクションの例1. 情報を与えるユーザ X のウィンドウを、情報を受け取るユーザ Y と共有する形態



図5 インタラクションの例2. 情報を受け取るユーザ Y のウィンドウを、情報を与えるユーザ X と共有する形態

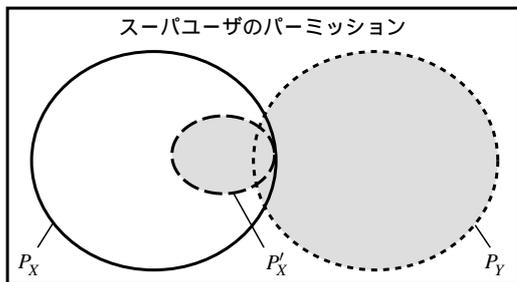


図3 ウィンドウ  $W_1$  でのユーザ Y の実効パーミッション  $P_Y \cup P'_X$  (網掛け部分)

### 3.5 共有のインタラクション

情報を与えるユーザ X と受け取るユーザ Y がウィンドウを共有するためのインタラクションの例をそれぞれ図4と図5に示す。

図4は X のウィンドウを Y と共有する形態である。X は自分が生成したウィンドウを Y に示す (i)。ウィンドウは生成した X だけが操作できる。実世界で Y は X に情報の一部を受け渡すように要求する (ii)。X は要求された情報しかアクセスできないように縮退操作を行い (iii)、Y に権限を与えるように委譲操作を行う (iv) と、Y は要求していた情報を取得することができる (v)。

図5は Y のウィンドウを X と共有する形態である。

X と Y はそれぞれウィンドウを生成、お互いに示しあう (i)。X、Y は自分で生成したウィンドウしか操作できない。実世界で Y は X に情報の一部を開放するように要求し (ii)、要求した情報しかアクセスできないように自分のウィンドウに縮退操作を行い (iii)、X に権限を与えるように委譲操作を行う (iv)。X は要求された情報を自分のウィンドウから Y のウィンドウに移動することができる (v)。

## 4. 実装

提案手法を実装し、操作性を確認するため、ファイル共有に関する応用例を試作、評価した。

### 4.1 デスクトップ画面とペン型端末

デスクトップ画面には液晶ペンタブレット Cintiq C-1500X (G) と一般的な PC を使用した。

ペン型端末には無線 LAN カードを取り付けた PocketGear (MC/PG5000) に液晶ペンタブレット用のスタイラスペンを固定して使用した (図6)。PocketGear には LCD 画面、上下左右方向を入力できる十字ボタンと決定ボタン (ジョグダイヤル) があり、これらを用いてペン型端末を操作するようにした。なお、スタイラスペンには識別用の ID がないので、文献1) 同様にペンに付属のスイッチを固定して、それを ID として利用した。

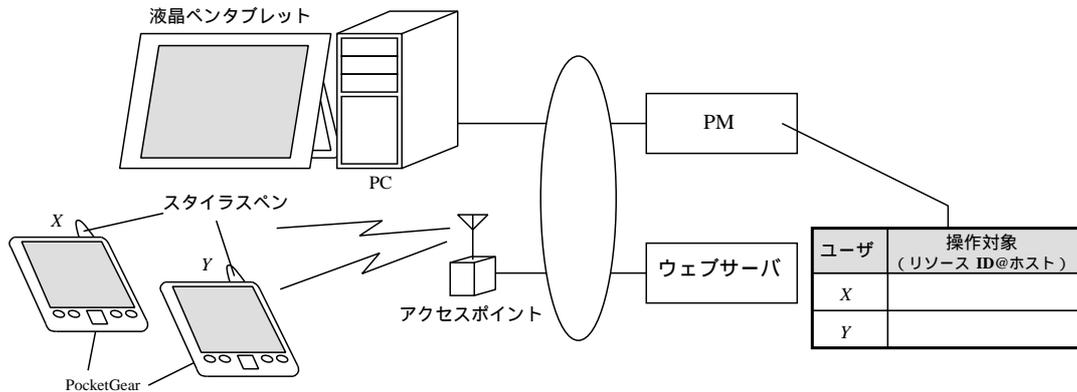


図7 機器とネットワークの構成



図6 試作した2台のペン型端末

ペン型端末およびデスクトップ画面は図7に示した構成のネットワークに接続した。PocketGear側のアプリケーションはiアプリで記述され、PocketGearに付属のi-enablerで動作させた。iアプリはウェブサーバ(CGI)を経由して通信する。

ネットワークではペン型端末が何をポインティングしているのか管理するためのポインティングマネージャ(PM)が動作している。デスクトップ画面はスタイラスペンからのイベントを受け取り、どのペンがどの情報を選択した(ペンダウンした)のかをPMに通知する。一方、ペン型端末は、自分がどこをポインティングしているのかPMに問い合わせ、ピックやドロップなどの操作対象を把握する。

#### 4.2 ピックとドロップ

デスクトップ画面のアイコンをピックする場合は、デスクトップ画面のウィンドウに表示されているアイ

コンをペン型端末でペンダウンする。その状態でピックする(ペン型端末の十字ボタンを下方方向に押す)と、それに関連付けられた情報がペン型端末に転送される。ペン型端末の画面にはピックした情報のアイコンが表示される。

デスクトップ画面へアイコンをドロップする場合は、まずペン型端末自身を操作してドロップしたいアイコンを選択する(ペン型端末の十字ボタンを左右方向に押して選択する)。ペン型端末の画面にドロップしようとするアイコンを選択した状態でデスクトップ画面のウィンドウにペンダウンしてドロップする(ペン型端末の十字ボタンを上方向に押す)と、情報がそのウィンドウに転送される。

このようにピックとドロップの操作を選択できるので、情報を連続してピックする(複数の情報を関連付ける)ことができるし、逆に情報を選んでドロップすることもできる。

#### 4.3 ファイラ

ファイル共有のアプリケーションとして、デスクトップ画面で動作するファイラを実装した。ユーザはペン型端末でデスクトップ画面のルートウィンドウをドラッグして、ファイラのウィンドウを生成できる。デスクトップ画面は各ユーザに独立したファイルシステムを割り当てていて、ファイラは最初そのルートディレクトリ(WindowsのデスクトップやUNIXのホームディレクトリに相当)の内容を表示する。

ファイラのウィンドウにあるアイコンをペン型端末でピックしたり、逆にペン型端末のアイコンをファイラにドロップすることで、デスクトップ画面とペン型端末の間でファイルを転送することができる。

本実装の説明ではアイコンとしているが、例えばアプリケーションのウィンドウそのものをピックできてよい。

Mac OSのFinderや、WindowsのExplorerのようなアプリケーション。

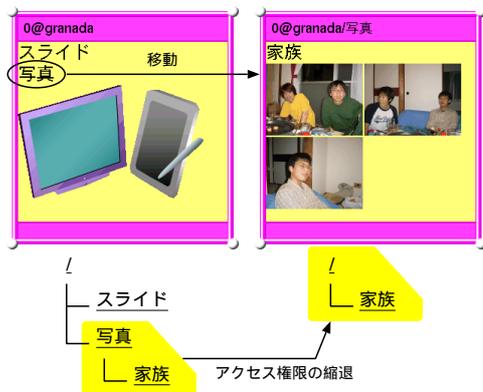


図 8 ディレクトリの移動とアクセス権限の縮退

ファイルのウィンドウはウィンドウ保護の対象になっているので、初期状態ではそれを生成したユーザしか操作できない。

#### 4.4 縮退操作

本実装ではファイルシステムのツリー構造を利用して、子ディレクトリへの移動により縮退操作を行うようにした。ファイルは表示しているディレクトリに含まれる子ディレクトリをリンク（アンカー）で表現し、ユーザがそのリンクをペン型端末でタップすると、ファイルはその子ディレクトリへ移動する（図 8 上）。ただし、親ディレクトリに移動できないようにファイルの機能を制限したので、ファイルは一度子ディレクトリに移動すると、そのディレクトリを新たなルートディレクトリとするサブツリーに閉じ込められる（図 8 下）。子ディレクトリに移動する度にファイルシステムのアクセスできる範囲が狭くなり、アクセス権限が縮退していくことになる。

なお、本実装はあるユーザが縮退操作を実行すると、同じウィンドウを操作可能な全ユーザの実効パーミッションが同様に縮退するという点で、3 節で説明した提案手法とは多少異なる。

#### 4.5 委譲操作

委譲操作におけるウィンドウの状態遷移を図 9 に示す。ユーザが委譲したいファイルのウィンドウをペン型端末でペンダウンしている状態で、ペン型端末の決定ボタンを押すと、ウィンドウは委譲モードになる。ウィンドウは委譲モードを設定したユーザの実効パーミッションを保存しておく（本実装では表示しているディレクトリを変更しないことに相当する）。

委譲モードを設定した（権限を与える）ユーザは、権限を受け取るユーザが委譲モードのウィンドウにタップする行為を実世界で許可することで、権限を受

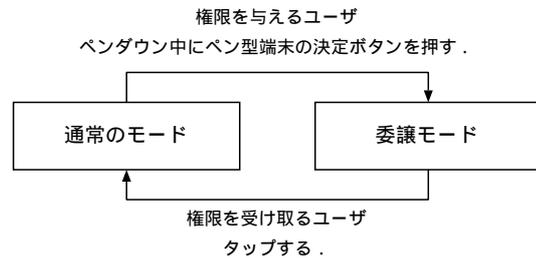


図 9 委譲操作におけるウィンドウの状態遷移

け取るユーザを選択する。そして、その権限を受け取るユーザが委譲モードのウィンドウにタップすると、委譲モードに設定したユーザの実効パーミッションを獲得して、そのウィンドウを操作できるようになる。

委譲操作により獲得した実効パーミッションの有効期間は、そのウィンドウが破壊されるまでとした。

## 5. 応用例

本節では本稿の提案手法の応用例を述べる。

### 5.1 共同作業環境

本稿の提案手法を用いれば、複数のユーザがプレゼンテーション資料などを各自のペン型端末に入れて持ち寄り、会議室などに設置された情報キオスク端末に資料をドロップして表示しながら議論したり、資料（のコピー）の受渡しなどの交渉をユーザ同士で容易に行うことができる。

### 5.2 個人所有 PC の借用

ユーザ A が自分のコンピュータを短時間ユーザ B に使用させたい場合、マルチユーザ OS であればユーザ A がユーザ B のアカウントを新規に（適切なユーザ名、権限で）作成して、ユーザ B にそのアカウントを使用させればよい。ところが、実際には作業が煩雑などの理由で、ユーザ A がログイン中の端末のキーボードやマウスをユーザ B に渡してしまうことが多い。これでは結果的にユーザ A のアカウントをユーザ B に使用させることになり、セキュリティの面で問題がある。

本稿の提案手法を用いれば、ユーザ間の簡単な操作により、ユーザ A が許可した範囲で自分のコンピュータをユーザ B に使用させることができる。

### 5.3 アカウントの切り替え

1 人のユーザが一般アカウントと管理者アカウントなどの複数のアカウントを使い分ける場合、アカウントを切り替える度にログインし直したり、どの画面（ウィンドウ）がどのアカウントのものなのか注意する必要がある。

本稿の提案手法では、1 人のユーザが複数のペン型

端末を使い分けることでアカウントを切り替えることができる。

## 6. 考 察

### 6.1 ユーザ名

Pick-and-Drop<sup>1)</sup>ではコンピュータの名前という記号的概念をユーザが意識することなく、コンピュータ間で情報を転送できる。本稿の提案手法では、特定の個人と関連付けられたペン型端末をポインティングデバイスとしたので、さらに「ユーザ名」という記号的概念を意識することなく、目の前のユーザに触れてもらうなどの操作でコンピュータとユーザの関係を指定できる。

ユーザ名の入力というタスクだけを考えると、ICカードや生体情報などを利用した個人認証でも目的は達成できる。しかし、アクセス権限の指定のようにリソースとユーザを同時に指定するタスクでは、ペン型端末での操作はより直接的である。

### 6.2 縮退操作

UNIXではrootユーザの権限で動作するプロセスが、場面に応じてsetuidシステムコールを使用して、より少ない権限のユーザとして実行することができる。また、chrootシステムコールによって、プロセスは自分のルートディレクトリを変更し、アクセス可能なファイルシステムの領域を限定することができる。

本稿で提案した縮退操作は、このような操作の類似概念である。縮退操作後に自ら操作すれば、委譲しようとしているアクセス権限が意図通り設定されているかどうかを容易に確認できる。

縮退操作の実現例としては、本実装の他にも、ウィンドウの実効パーミッションをチェックボックス付き項目として一覧表示して、それらのチェックの一部を外すことで、読み出し（ピック）や書き込み（ドロップ）などを禁止する方式も考えられる。特にファイラでは、ディレクトリ毎にパーミッションを設定して、ファイルの子ディレクトリに移すと、そのファイルに対する操作がその子ディレクトリに設定されたパーミッションに限定されるようにすることも考えられる。

また、それぞれのウィンドウでどのようにアクセス権限が縮退されているかをユーザに提示する方法については今後も検討が必要である。

### 6.3 委譲操作

本実装では委譲されたアクセス権限の有効期間を

<sup>1)</sup>例えば実効パーミッションをウィンドウのプロパティとして表示する方法。

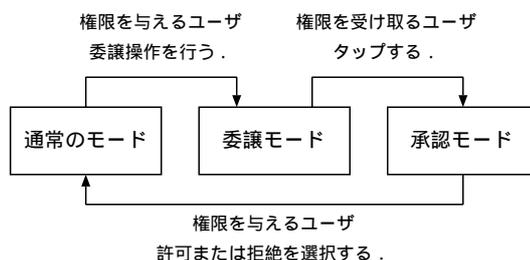


図10 権限を与えるユーザが最後に許可または拒絶する方式の委譲操作におけるウィンドウの状態遷移の例

表1 ペンとペン型端末のアフォーダンス

行動	ペン	ペン型端末
放置		×
代替		×
共用		×

ウィンドウの存在期間と同一にしたが、ユーザのパーミッションとして永続的あるいは期間限定で委譲したり、委譲したユーザが後でそれをキャンセルできる形態が適している応用もある。また3人以上のユーザ構成では、アクセス権限を受け取ったユーザが、それをまた別のユーザに与えることができるため、再度委譲できる回数を制限する仕組みが必要である。

本実装の委譲操作では、アクセス権限を誰に委譲するかを選択する際に実世界での許可を伴うようにしたが、それではセキュリティ的に不十分な場合もある。より安全な委譲操作の実現例としては、図10のように権限を与えるユーザが最後に許可または拒絶できる方式も考えられる。

### 6.4 ペンとペン型端末の比較

ペンとペン型端末ではその特性として表1に示したようなアフォーダンスの違いがあると考えている。例えば、ペンは置き忘れたり（放置）、別のペンでも用が足りたり（代替）、貸し借り（共用）しやすいが、ペン型端末（個人用端末）はその逆である。また、ペンがドラッグや文献<sup>1)</sup>のピックにより保持できる情報は1つ（または1つのグループ）で、それが保持されている期間も短い。一方、ペン型端末は複数の情報をそれぞれ任意の期間保持できるので、ユーザはペン型端末を自分の専有物として受け入れやすくなる。

このような特性から、ポインティングデバイスとユーザを関係付ける場合は、ペンよりもペン型端末の方が適していると思われる。

### 6.5 プライバシー

共同作業環境での情報のプライバシー状態を秘密（private）と公開（public）に分類すると、ペン型端末

で秘密情報、情報キオスク端末など共用目的のデスクトップ画面で公開情報を扱うことで、プライバシー管理を実現できる。しかし、情報の存在は公開するが所有や操作に関しては禁止するなどのように、実際の共同作業では秘密と公開の中間にあたるプライバシー状態も必要だろう。

提案手法ではデスクトップ画面への情報提示と、共有のためのアクセス権限管理を区別しており、ユーザが折衝して公開の範囲を広げられるので、より柔軟なプライバシー状態を扱うことができる。

#### 6.6 ポインティングの検出

液晶ペンタブレットを用いる場合、どのスタイラスペンがどの座標をペンダウンしているのかを、デスクトップ画面側で検出している。しかし、どのデスクトップ画面のどの座標をペンダウンしているのかを、環境側のカメラや、ポインティングするペン型端末自身が検出する方式を使用することもできる。

本実装ではウィンドウシステムのポインタが画面に1つしかなく、複数のスタイラスペンを同時にペンダウンしても、先に検出されたペンがポインタを独占するため、ユーザは交互にデスクトップ画面を操作しなければならない。1つのデスクトップ画面で複数のユーザが同時にペンダウンする必要がある場合は、ペン型端末側や環境側がポインティングを検出する方式を使用すればよい。

### 7. 関連研究

Augmented Surfaces<sup>2)</sup>では作業空間内のノート型コンピュータ、プロジェクトで拡張された机や壁、マーカで識別可能な実世界オブジェクトなどの間で、オブジェクトを空間的に連続してドラッグする技法(Hyperdragging)を提案している。また、HyperPalette<sup>3)</sup>ではPDAをプロジェクトで拡張された机に対する入力デバイスとして利用し、PDAと机の間でオブジェクトを移動させる技法(Scoop-and-Spread)を提案している。これらの研究では個人用コンピュータ(ノートPCやPDA)とユーザの関連付け、ユーザのアクセス権限の管理については考慮されていない。

Wearable Key<sup>4)</sup>ではユーザが触れることでコンピュータを動的に個人化するActive Personalizationという概念を提案している。コンピュータがユーザを識別する点は同じであるが、提案手法は環境に配置された複数のコンピュータをユーザが利用するというよりはむしろ、

複数のユーザがコンピュータで協調的なインタラクションを行う点に主眼を置いている。

EMMIE<sup>5)</sup>はARを利用して作業空間を3次元に拡張し、3次元ポインティングデバイスでアイコンなどの仮想オブジェクトを空間にドラッグすることで、異なるコンピュータ間でのDrag-and-Dropを実現している。各オブジェクトはプライバシー状態をもち、ユーザはそれを公開か秘密に設定、確認することができる(privacy lampsとvampire mirrors)。公開情報は全ユーザで共有し、秘密情報は各自のHMDだけに表示することでプライバシー保護を実現する。しかし、公開か秘密という粗い制御だけでは不十分な場合もある。本稿の提案手法とは、ユーザ毎に異なるアクセス権限を設定できる点が異なる。

### 8. まとめと今後の課題

ペン型端末をポインティングデバイスとする共同作業環境と、アクセス権限の管理手法としてウィンドウへの縮退操作、委譲操作の概念を提案した。また、ファイル共有に関する応用例を実装し、自分が限定した範囲の情報を特定のユーザと共有する作業を行い、提案手法の操作性を確認した。

今後は、より具体的で有用なアプリケーションとその操作性の評価について考察したい。また、縮退操作、委譲操作のより自然な実現方法、ウィンドウの属性であるEPTの提示方法を検討する予定である。

### 参考文献

- 1) Jun Rekimoto. Pick-and-Drop: A Direct Manipulation Technique for Multiple Computer Environments. In *Proceedings of UIST '97*, pp. 31-39 (1997).
- 2) Jun Rekimoto and Masanori Saitoh. Augmented Surfaces: A spatially continuous workspace for hybrid computing environments. In *Proceedings of ACM CHI '99*, pp. 378-385 (1999).
- 3) 綾塚 祐二, 松下 伸行, 暦本 純一, "HyperPalette: PDAを利用する複合計算機環境" in *WISS '99, インタラクティブシステムとソフトウェア VII*, 近代科学社, pp. 109-118 (1999).
- 4) 松下 伸行, 綾塚 祐二, 暦本 純一, "Wearable Key: 触ることで個人化されるユビキタスコンピュータ環境" in *WISS2000, インタラクティブシステムとソフトウェア VIII*, 近代科学社, pp.125-130 (2000).
- 5) Andreas Butz, Tobias Höllerer, Steven Feiner, Blair MacIntyre, and Clifford Beshers. Enveloping users and computers in a collaborative 3D augmented reality. In *Proceedings of IWAR '99 (International Workshop on Augmented Reality)*, pp. 35-44 (1999).

天井などに設置され、デスクトップ画面とポインティングデバイスを撮影するカメラ。