

協調作業を促進する CollaboTray のための アプリケーション情報転送制御

桐村健吾[†] 森龍也^{††} 松原重喜^{††}
佐藤慶三^{††} 中島誠^{††} 伊藤哲郎^{††}

個人利用を想定したアプリケーションをそのまま協調作業に利用可能にするプラットフォーム CollaboTray の開発を行っている。個人利用アプリケーションを CollaboTray に載せるだけで、その機能を損なわず、複数のユーザが共有できるマルチユーザ版のアプリケーションとすることができる。従来の遠隔操作ソフトウェアを利用したアプリケーション共有とは異なり、各ユーザはアプリケーションウィンドウのイメージを共有しながらも、イベント情報のアプリケーションへの転送制御により、ユーザそれぞれの計算機上での作業を妨げることなくアプリケーションを操作することができる。本稿では、CollaboTray による遠隔協調作業時のアプリケーション情報（ウィンドウイメージ、情報オブジェクト、イベント）の転送制御の方法について述べる。

Application Information Transfer Control for CollaboTray Boosting Collaborative Work

KENGO KIRIMURA[†] TATSUYA MORI^{††}
SHIGEKI MATSUBARA^{††} KEIZO SATO^{††}
MAKOTO NAKASHIMA^{††} TETSURO ITO^{††}

A novel application platform, called a CollaboTray, is developed for adapting an off-the-shelf single-user application without changing the original source codes to collaborative work. Though typical application sharing software such as VNC allows only one user to operate the application at the same time, a CollaboTray tolerantly allow concurrent use of the application via its clone CollaboTrays, each of which may be on remote computers and is generated by copying the image of the application window. The CollaboTray intervenes between a user and the application, and manages the image/event-information flows among the original and clone CollaboTrays. In this paper, we describe the way for managing the image/event-information flow in transferring them when some application is utilized collaboratively on remote computers.

1. はじめに

計算機による協調作業支援（CSCW）の研究では、個人作業から協調作業へ（あるいはその逆）のスムーズな移行を行えるシステムの開発が注目されている。個人作業と協調作業の間の移行は頻繁に生じ[8]、特に研究室のような環境下で行われる複数人での話し合いの88%が事前に予定されたものでなく、突発的に生じて短時間で終わるといった観測事実[4]から、個人利用を想定したアプリケーションを、そのまま協調作業で共有できることが、CSCW システムの利用可能性を高め、作業効率の向上に重要であると言える。個人利用を想定したアプリケーションを如何にそのまま複数のユーザがリアルタイムに利用できるようにするかが問題となる[3]。

従来のアプリケーション共有の実現方法の多くは、1 台のコンピュータ上でアプリケーションを起動し、ウィンドウのイメージを複数のコンピュータで共有する集中方式[3]を採用し、VNC[7]に代表されるコンピュータの遠隔操作ソフトウェアの利用が有効とされる。サーバとなるコンピュータを、そのスクリーンイメージを共有しながらクライアントとなるコンピュータから操作することができ、個人利用を想定したアプリケーションをそのまま利用する厳密な共有が容易に実現できる。しかしながら、サーバとクライアント双方のユーザがアプリケーションを同時に操作すると、そのイベント情報は混ざり合っってアプリケーションに届けられ、意図しない動作を引き起こす。また、サーバのユーザにとって、共有しているアプリケーションを利用する以外に、情報収集など他の作業を行っている最中に、クライアントのユーザの操作によって、作業を中断せざるを得ない。この欠点は、アプリケーション自体を複製して複数のコンピュータ上で起動する複製方式[5]で回避できるが、あらかじめアプリケーションの複製が必要となり、コンカレントな操作が可能なルーズな共有が可能でも、アプリケーションの整合性の保持が難しい。

これまで我々は、MS Windows OS で利用するアプリケーションをそのまま協調作業用のアプリケーションとすることができるプラットフォーム CollaboTray [1]を提案してきた。ユーザは、アプリケーションを CollaboTray に載せるだけで、その操作感を損なうことなく、クローンを作成して協調作業用のアプリケーションとして利用することができる。集中方式でありながらも、共有されたアプリケーションと個人作業のアプリケーションのコンカレントな操作もできる、緩いアプリケーション共有、が可能となっている。

上記のようなアプリケーション共有の実現には、複数ユーザの操作に対応したイベ

[†] 大分大学大学院工学研究科知能情報システム工学専攻

Dept. of Computer Science and Intelligent Systems, Graduate School of Engineering, Oita University

^{††} 大分大学工学部知能法システム工学科

Dept. of Computer Science and Intelligent Systems Faculty of Engineering, Oita University

ント情報を受理して制御する仕組みが必要となる。個々の CollaboTray は、3つのハンドラ（データフロー、アプリケーション、およびイメージハンドラ）からなる。中心となるのは、データフローハンドラで、アプリケーションへのイベント情報の転送を担うアプリケーションハンドラと、ユーザへアプリケーションウィンドウのイメージを描画しイベント情報を受け取るイメージハンドラの間であって、アプリケーション情報（イベント情報、情報オブジェクト、イメージ）のアプリケーションへの転送を制御する。

以下では、緩いアプリケーション共有のための CollaboTray と、データフローハンドラによるアプリケーション情報の転送方法について述べる。ネットワークを介した遠隔協調作業下でのこれらの転送負荷の軽量化方法についても述べ、CollaboTray を用いた遠隔協調作業の例を示し、最後に、今後の課題と展望について述べる。

2. CollaboTray と協調作業

2.1 協調作業と緩いアプリケーション共有

一般に、協調作業は、その環境からみると遠隔協調作業と対面協調作業の2つに分けることができる[6]。前者は、協調する各ユーザが、離れた場所で共通のタスクを行う場合であり、後者は全てのユーザが同じ場所（たとえばテーブル型ディスプレイの回り）に集まり、対面で作業を行う。

ユーザ1がアプリケーション1をコンピュータAで利用しているとして、例えば、遠隔協調作業では、コンピュータBを利用しているユーザ2にアプリケーション1上での情報オブジェクト（文字列や図形）の編集についてアドバイスを求めるとする。通常の作業環境なら、ユーザ1は、情報オブジェクトを、コンピュータB上でアプリケーション1と同等のアプリケーション1'を起動したユーザ2に送ることになる。アプリケーション1のクローンをそのままコンピュータB上でユーザ2がユーザ1と同様に利用できれば、効果的に協調作業を行うことができ、個人作業から協調作業へ容易に移行できる。対面協調作業では、例えば、コンピュータAとBで、それぞれアプリケーション1を利用しているユーザ1とアプリケーション2を利用しているユーザ2が、ユーザ2のコンピュータ2の回りで作業する。この場合も、アプリケーション1とアプリケーション2のクローンをコンピュータ2上で利用し、それぞれのアプリケーション上の情報オブジェクトを扱えるようにすれば、容易に協調作業へと移行することができる。

個々のユーザは、上記のようなそれぞれの協調作業の環境において、共有するアプリケーションのみでなく、それぞれが異なるアプリケーションも利用する。例えば、遠隔協調作業において、ユーザ1は、ユーザ2とアプリケーション1を共有しながら

も、コンピュータA上で、他のアプリケーションをコンカレントに利用する。ユーザ2がアプリケーション1のクローンを利用しようとも、ユーザ1の作業を制限しない、緩いアプリケーション共有が生じる。同様に、対面協調作業においても、共有するアプリケーション以外に個々のユーザが個別のアプリケーションを利用することもある。CollaboTrayは、その上にアプリケーションを載せると、そのクローンを容易に作成することができるだけでなく、オリジナルと複数のクローンを複数ユーザで上記の制限なしに利用することを可能にする。

2.2 CollaboTray のアーキテクチャ

MS Windows OSを対象とした、CollaboTrayのアーキテクチャを図1に、アプリケーションをCollaboTrayに載せた例を図2に示す。CollaboTrayは、図2のように、円形のお盆の形をしたウィンドウであり、その内部には、載せたアプリケーションのウィンドウイメージを表示する。図2の左側の画面上では、ペイントツールとMS Wordを載せ、ペイントツールをのせたクローンCollaboTrayを右側の画面に表示している。載せられたアプリケーションは、個人利用のものであっても、協調作業用のアプリケーションとして、緩いアプリケーション共有を実現できる。そのために、以下の機能を有している。

- 1) アプリケーションウィンドウの識別とそのイメージの取得
- 2) イベント情報の調整
- 3) アプリケーション情報の転送制御

オリジナルのCollaboTrayは図1のAあるいはBのように、3つのハンドラからなり、アプリケーションハンドラがアプリケーションウィンドウの識別とそのイメージの取得を行い、イベント情報を調整してアプリケーションへ転送する。データフローハンドラは、ウィンドウイメージの表示とイベント情報の取得を行うイメージハンドラとアプリケーションハンドラとの間で、ウィンドウイメージとイベント情報の受け渡しを行う。CollaboTrayのカーネルは、アプリケーションハンドラがアプリケーションAおよびBへイベント情報を転送するタイミングを制御することで複数のアプリケーションのコンカレントな操作を可能にする。

アプリケーションの共有時、アプリケーションハンドラは、CollaboTray B'のように、イメージハンドラとデータフローハンドラのみを複製したクローンを作成する（コンピュータB上のOSとカーネルは省略してある）。オリジナルとクローンCollaboTrayのデータフローハンドラ同士が、アプリケーション情報である、アプリケーションウィンドウのイメージ、イベント情報、および情報オブジェクトの転送を行い、アプリケーションの共有を実現する。各CollaboTrayに備えたローカルクリップボードは、

情報オブジェクトの転送のためのバッファとして機能し、必要に応じてシステムクリップボードへそのデータを転送する。以下、上記1～3の機能について述べ、3についてはその実現方法を次章でくわしく述べる。

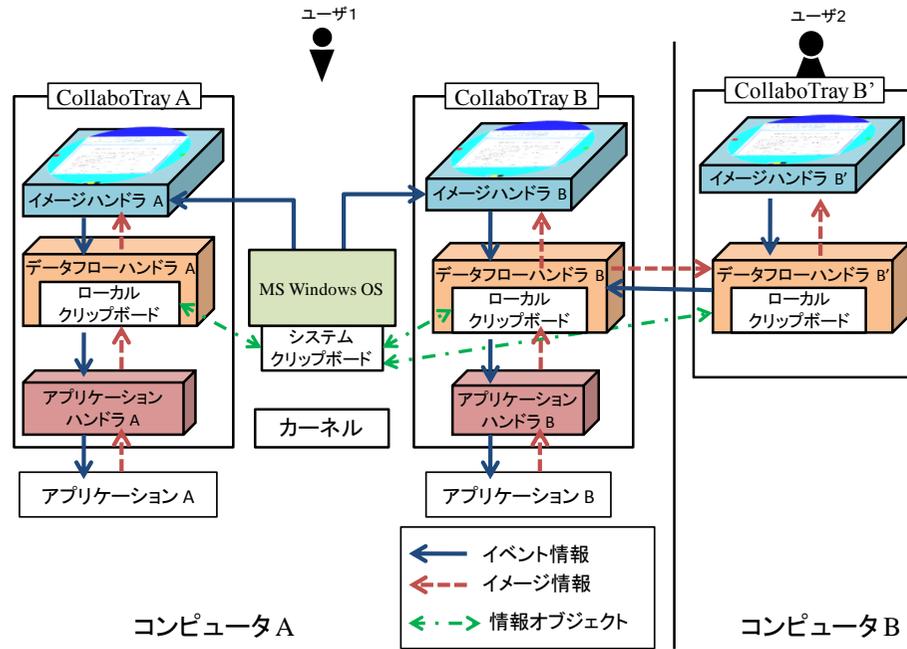


図 1 CollaboTray のアーキテクチャ

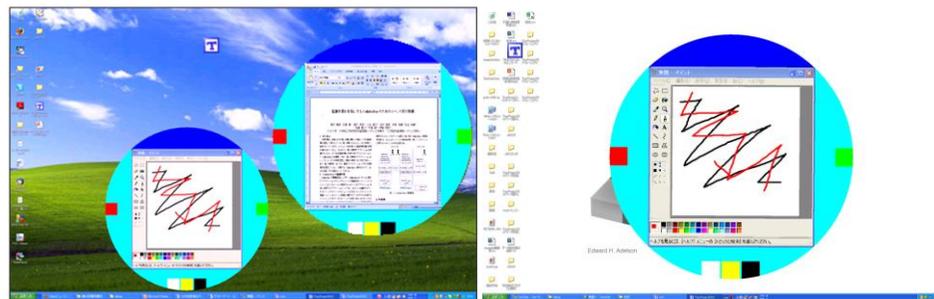


図 2 CollaboTray 利用例

2.2.1 アプリケーションウィンドウの識別とイメージの取得

CollaboTray に載せられたアプリケーションウィンドウの識別には、MS Windows OS の API (Application Program Interface) を利用し、そのイメージを描画用メモリから直接取得する。取得したイメージは、イメージハンドラがスクリーン上に表示し、アプリケーションウィンドウとして利用する。アプリケーションハンドラは、オリジナルのアプリケーションウィンドウを透明化してスクリーン上に配置し、必要に応じて、ユーザの目に触れない程度に可視化して、イベント情報を転送したり、ウィンドウイメージを取得したりする。この手法により、アプリケーション自体を改変することなく、CollaboTray のクローンを作成してアプリケーションを共有する際、アプリケーションを通常と同様に操作できる。

描画用メモリから直接イメージを取得することで、スクリーンに描画されたウィンドウイメージを取得する VNC とは異なり、他のウィンドウの影響を受けずに全体を取得できる。また、アプリケーションを識別しておくことで、メニューやダイアログといったアプリケーションの操作時に動的に生成されるウィンドウのイメージも取得が可能となる。さらに、オリジナルのアプリケーションウィンドウとは独立に、ウィンドウイメージの回転・拡大・縮小が容易となり、テーブル型ディスプレイを利用するような対面協調作業時に、各ユーザが作業をし易くアプリケーションウィンドウの向きを変えられるといった機能も実現できる。

2.2.2 イベント情報の調整

通常、MS Windows OS 上で動作するアプリケーションはユーザの操作に関するイベント情報を OS から受け取る。ユーザが CollaboTray に載せたアプリケーションウィンドウのイメージ上で操作を行うとイメージハンドラにイベント情報が届き、データフローハンドラ、アプリケーションハンドラを経てアプリケーションに転送される。その際、CollaboTray 上でのアプリケーションウィンドウのイメージと実際のアプリケーションウィンドウの位置とが異なるため、アプリケーションハンドラでは、スクリーン上での座標値のずれを調整する。

イベント情報の調整には、メッセージフックと呼ばれる仕組みを利用する。メッセージフックは任意のアプリケーションへのメッセージを取得し、修正・破棄といった操作を行うもので、これにより、アプリケーションが実際に CollaboTray 上にあるかのような情報に修正したイベント情報を作成し、アプリケーションに転送する。イベント発生時にこれらの処理を行うことで、アプリケーションの回転・拡大・縮小など、CollaboTray により付与された機能を使いながら協調作業を行ったとしても、個人作業でアプリケーションを利用しているときのような操作感を実現することができる。

2.2.3 アプリケーション情報の転送制御

アプリケーションの共有時、オリジナル CollaboTray のアプリケーションハンドラに対し、オリジナルとクローン CollaboTray の数だけのイメージハンドラが存在することになる。このとき、複数のユーザがアプリケーションを同時に操作しようとする、ユーザの異なる複数のイベント情報がデータフローハンドラに渡される。これをそのままアプリケーションに転送するとイベントの競合が発生する。これは、個人作業の場合には入力には常に1つであることから競合を想定した設計が為されていないためである。データフローハンドラではイベント情報の競合を解消して一元化を行いアプリケーションハンドラへイベント情報を転送する。

情報オブジェクトの転送は、通常、システムクリップボードを介したコピーアンドペースト操作で行われる。データフローハンドラは、ローカルクリップボードを備え、それを介して、アプリケーションを載せた複数の CollaboTray を接触させた状態で、コピーアンドペースト操作を行うことで情報オブジェクトの転送を行うことができる。

アプリケーションウィンドウのイメージは、オリジナル CollaboTray から、更新のたびにクローン CollaboTray に送信され、アプリケーションの共有時における即時性を確保している。遠隔協調作業における、ネットワークを介したイメージの転送に際しては、その転送量の軽減による即時性の確保を行っている。

3. アプリケーション情報転送制御

遠隔協調作業において、アプリケーション本体を持つオリジナル CollaboTray をサーバ、持たないクローン CollaboTray をクライアントとし、これらのデータフローハンドラ間で、

- イベント情報
 - クリップボードデータ
 - アプリケーションウィンドウのイメージ
- の3種類をやりとりする。

オリジナル CollaboTray からはイメージの転送、クローン CollaboTray からはイベント情報の転送を基本とし、必要に応じてクリップボードデータをどちらからでも転送可能としている。アプリケーションの操作は、オリジナルとクローン CollaboTray 双方で同様に行うことができる。

以下、イベント情報の転送手法と受信後のイベント情報の競合問題の解決方法、クリップボードデータの転送、およびアプリケーションウィンドウのイメージの転送方法について述べる。

3.1 イベント情報の転送

イベント情報を転送する際、誤ったデータの受信は、アプリケーションでの処理に支障をきたす。そのため、イベント情報の転送には、TCP 接続を使い確実に送り届ける。

マウス操作により生じるイベントを例とすると、そのイベント情報は「アプリケーションウィンドウの識別子」、「イベントの種類」、「マウス座標」となる。1つ1つのイベント情報の転送量は少ないが、ドラッグ操作などにより、一度に数百のイベント情報が発生することもある。そのため、「アプリケーションウィンドウの識別子」、「イベントの種類」が同一のものである場合、事前に転送したイベント情報と前者2つが同一のものであることを示す「通信コード」に、「マウス座標」のみを加えて送ることで転送量の軽減を図る。

本来、CollaboTray に載せているアプリケーションは個人作業用に設計されたものであり、同様に MS Windows OS も個人作業を想定して設計されている。このため、複数ユーザが同時に1つのアプリケーションを操作した場合と、異なる複数のアプリケーションを同時に操作した場合にイベント情報の競合が生じる。以下、これら2種類の競合への対処を述べる。

3.1.1 同一のアプリケーションへの同時操作における競合

アプリケーションは1つの入力にしか対応していないため、異なるユーザによる操作が同時に発生した場合、各操作に対応するイベント情報は混じりあって、個々のユーザが意図した通りに順番に処理されない。この問題にはセマフォの考え方によるイベント情報の排他処理を行う。

CollaboTray のデータフローハンドラにイベント情報を格納するキューを用意する。イベント情報には、ユーザを識別するための情報を付加し、キューに既に格納済みのイベント情報のユーザ情報と同じものだけを格納していく。キューより取り出したイベント情報をアプリケーションハンドラを介してアプリケーションに転送する。ユーザ情報が異なるものと判断されたイベント情報は破棄する。これは、先に処理された操作によりアプリケーションの状態が変化し、後から発生した操作が意図しない処理を引き起こすことを防ぐためである。

3.1.2 異なるアプリケーションへの同時操作における競合

MS Windows OS もアプリケーションと同様に1つの入力にしか対応していないため、異なるアプリケーションをほぼ同時に操作した場合、イベント情報の競合が発生する。実際には、1つのアプリケーションへの一連の操作が、他のアプリケーションへの操作に割り込まれて、前者の操作が中断されるという問題を引き起こす。この問題に対処するために、データフローハンドラがそのイベント情報のキューからイベント情報

を取り出し、アプリケーションに転送するタイミングを制御する。

例えば、あるアプリケーションにおけるユーザがマウスを押してから離すまでといった一連の操作をまとめて扱い、他のアプリケーションの操作が行われていない場合には通常通りイベント情報の転送を行い、他のアプリケーションの操作が既に行われている場合には、操作が終わるまでイベント情報の転送を待機させておく。特定のアプリケーションの操作が長時間連続して行われると、その他のアプリケーションへイベント情報が転送されなくなってしまう。このような操作の独占を防ぐため、タイムシェアリング時間を設定し、イベント情報を転送するアプリケーションの優先度を状況に合わせて変化させて対処する。

3.2 クリップボードデータの転送

通常、OS が有するシステムクリップボードを介したコピーアンドペースト操作において、同一のコンピュータ上では、すべてのアプリケーションが共通のシステムクリップボードを利用するため、あるユーザがペースト操作を行う前に別のユーザがコピー操作を行うと、データが上書きされてしまう。また、遠隔協調作業の場合においては、オリジナル CollaboTray があるコンピュータ上のシステムクリップボードにアクセスする必要がある。

この問題への対処として、CollaboTray 毎に専用のローカルクリップボードを持たせて、コピー操作が行われるとそこに一時的に情報オブジェクトを保持し、必要に応じてシステムクリップボードに転送する仕組みにより、協調作業に対応したコピーアンドペーストを実現する。ペースト操作が行われると同時にローカルクリップボードからシステムクリップボードにデータを転送し、通常通りのペースト処理を行わせる。遠隔協調作業においては、ローカルクリップボードのデータを相手の CollaboTray に転送し、同様の処理を行う。CollaboTray 間でのクリップボードデータの転送をフィード機能と呼び、異なるアプリケーションを載せた CollaboTray 同士を接触させた状態で、コピーアンドペースト操作を行うと、通常の操作と同様に情報オブジェクトの転送を行うことができる。

3.3 アプリケーションウィンドウのイメージの転送

イメージの転送については通信量の軽減が課題となる。アプリケーションウィンドウのイメージは更新頻度が高く、また、アプリケーションウィンドウのイメージのサイズは、ウィンドウのサイズ、色の数によって非常に大きなサイズとなってしまう、イメージの転送に時間がかかってしまうと描画が遅れ即時性が失われてしまう。

この問題への対処として、更新頻度が高く、直接アプリケーションの処理に影響を及ぼさないことに注目して[2]、UDP 接続による通信によって転送を行う。さらに、RFB プロトコル[7]を参考にした方法として、イメージをブロックに分割した上で減色

処理を加え、更新ブロックのみを転送する。

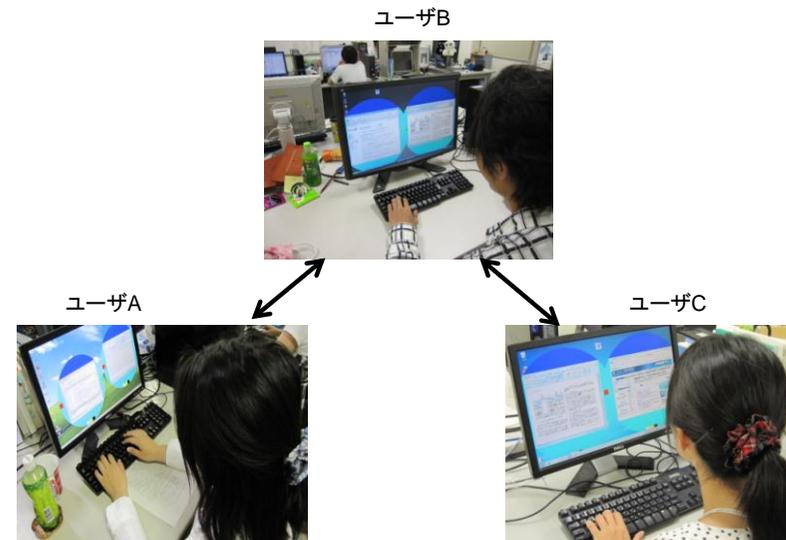


図 3 遠隔協調作業例

4. 協調作業例

CollaboTray を用いた協調作業例の様子を図 3 に示す。この作業では、3 人のユーザが CollaboTray を利用して別のユーザのアプリケーションを共有しながら論文の執筆と添削を行っている。それぞれのユーザは表 1 に示す普段利用している OS とアプリケーションで作業を行っている。CollaboTray では、共有しようとするアプリケーションが、他のコンピュータにインストールされている必要はないためバージョンの異なるアプリケーションを利用していても協調作業を行うことができる。

表 1 各ユーザの使用する OS とアプリケーション

	OS	アプリケーション
ユーザ A	MS Windows XP	MS Word2003
ユーザ B	MS Windows XP	MS Word2007, Fire Fox
ユーザ C	MS Windows 7	Internet Explorer

この作業では、ユーザ B が執筆中の論文のファイルを MS Word 2007 で開き、それ

を CollaboTray に載せてユーザ A と C で共有している。VNC などのスクリーン共有と異なり、ユーザ A が論文の添削を行っている最中に、ユーザ B が FireFox を使ってインターネットで情報を調べるといった別の作業を行うことができている。また、ユーザ C は Internet Explorer を CollaboTray に載せて使い、調べた情報をクローン CollaboTray を介してユーザ B の MS Word 2007 にコピーアンドペーストした。ユーザ B も FireFox 上から、異なる情報オブジェクトをコピーアンドペーストしたが、ユーザ B と C がコピーした情報オブジェクトは、それぞれ正確にペーストされた。

5. 関連研究

多くの研究で遠隔操作ソフトウェア、例えば VNC や NetMeeting¹、がアプリケーション共有に用いられるのは、ユーザが通常利用するコンピュータのスクリーンのイメージそのままを取得してクライアント側と共有するため、アプリケーションの変更が不要で、そのまま協調作業に利用する点において適しているためである。サーバ側のスクリーンのイメージをクライアント側のスクリーンに 1 つのウィンドウとして表示し、サーバ側とクライアント側で、操作権の切り替えを行いながら、作業を行える。しかしながら、サーバ側とクライアント側で、特定のアプリケーションを集中して操作する場合に注目した厳密なアプリケーション共有であり、クライアント側での共有アプリケーションの操作が、サーバ側での他のアプリケーションの操作を中断させることへの対処はなかった。MS Windows OS のように 1 人のユーザの操作を想定した OS では、一度に操作できるアプリケーションは 1 つに限られる。複数のコンピュータでアプリケーションを共有する状況では、クライアント側でアプリケーションの操作が行われると、サーバ側の OS 上での操作権は共有アプリケーションに移り、他のアプリケーションの操作ができなくなる。これには、CollaboTray のように、サーバ側でのアプリケーションへのイベントの転送を制御させることが必要となる。

複製方式によるアプリケーション共有でも、協調作業用のアプリケーションを作成するのではなく、通常利用するアプリケーションを利用する試みがなされている。それらは、アプリケーションに共有のための機能を付加するツールキットを用意し、アプリケーションの複製を利用することに注目している。たとえば、[5] では、MS Word や MS PowerPoint といったアプリケーションを協調作業用のアプリケーション CoWord や CoPowerPoint へと変更するツールキットを提案している。しかしながら、共有しようとするアプリケーションに固有の API を利用した変更があらかじめ必要で、このような API の提供のないアプリケーションへの対応が難しい。CollaboTray では、MS Windows OS のみの API を利用した仕組みにより、個人利用のアプリケーションを

そのまま複数ユーザで利用できるようにしている。したがって、MS Windows OS 上で直接起動するアプリケーション (Java アプリケーションのように仮想マシン上で動くものを除く) は、そのウィンドウを CollaboTray に載せるだけで、複数のコンピュータでの共有が可能となる。

6. おわりに

本稿では、個人利用を想定したアプリケーションをそのまま協調作業用として利用できる CollaboTray のアーキテクチャと遠隔協調作業におけるアプリケーション情報の転送方法について述べた。代表的なアプリケーション共有の方法である、集中方式と複製方式の利点を有し、簡単にアプリケーションを協調作業に用いることができ、かつ他のユーザとのコンカレントな操作を実現できる。

今後は、CollaboTray を用いて遠隔協調作業ができるユーザを特定するためのインタフェースデザインとともに、緩いアプリケーション共有による協調作業の効果について精査する。

参考文献

- 1) Abe, Y. et al.: Tolerant Sharing of a Single-user Application among Multiple Users in collaborative work. In Companion Proceedings of the ACM Conference on Computer-Supported Cooperative Work (CSCW'10). ACM, pp.555-556 (2010).
- 2) Alwis, B., et al.: GT/SD: Performance and simplicity in a groupware toolkit. In Proceedings of the 1st ACM SIGCHI Symposium on Engineering Interactive Computing Systems (EICS'09), ACM, pp.295-274 (2009).
- 3) Begole, J., et al.: Leveraging JAVA applets: Towards collaboration transparency in JAVA. IEEE Internet Comput., Vol.1, No.2, pp.57-64 (1997).
- 4) Gutwin, C., et al.: Supporting informal collaboration in shared workspace groupware. Journal of Universal Computer Science, Vol.14, No.9, pp.1411-1434 (2008).
- 5) Sun, C. et al.: Transparent adaptation of single-user applications for multi-user real-time collaboration. ACM Trans. Comput.-Hum. Interact. Vol.13, No.4, pp.531-582 (2006).
- 6) Tuddenham, P.: Distributed tabletops: territoriality and orientation in distributed collaboration. In Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI'07), ACM, pp.2237-2242 (2007).
- 7) Richardson, T. et al.: Virtual network computing. IEEE Internet Computing Vol.2, No.1, pp.33-38 (1998).
- 8) Scott, D. et al.: System guidelines for co-located, collaborative work on a tabletop display. In Proceedings of European Conference Computer-Supported Cooperative Work (ECSCW'03), Kluwer Academic Publishers, pp.159-178 (2003).

1 <http://www.microsoft.com/downloads/details.aspx?FamilyID=26c9da7c-f778-4422-a6f4-efb8abba021e>