

# 協調作業におけるグループ内アプリケーション共有開始手続きの円滑化のためのインタフェースの試作

Developing an Interface for Beginning Application Sharing Smartly in a Group of Cooperative Work

岩田 知\*<sup>1</sup> 大園 忠親\*<sup>1</sup> 新谷 虎松\*<sup>1</sup>  
Satoru Iwata Tadachika Ozono Toramatsu Shintani

名古屋工業大学大学院工学研究科情報工学専攻

Department of Computer Science, Graduate School of Engineering, Nagoya Institute of Technology

Sharing application window among users may improve cooperative works. However, configurations of sharing multiple application windows among multiple PCs are complicated. We implemented a management system for the configuration. The main idea of the system is to introduce an analogy of classrooms, which helps users to understand and make the configurations. The system provides a novel interface for management of groups in a classroom. This paper shows the implementation and evaluation of the system. Specifically, we realized a practical application window sharing system by optimizing streaming of windows.

## 1. はじめに

PCを用いた協調作業において、作業中のネイティブアプリケーションのウィンドウ画面を、参加者間で共有することは有用である。先行研究では、ウィンドウ単位での画面共有に焦点を当て、アプリケーション仮想共有システムを開発した [1]。しかし、先行研究には、共有を開始するまでの操作が煩雑であるという課題があった。アプリケーションを共有するためには、共有するアプリケーションや共有相手の選択を行い、共有可能な環境の構築が必要である。本稿ではこの手続きを共有開始手続きと呼称する。この共有開始手続きが円滑に進まない場合、かえって協調作業を妨げてしまう。

本研究の目的は、円滑な共有開始手続きを可能とするインタフェースを実装することで、協調作業の支援を行うことである。本研究では、あらかじめグループ内で利用する端末の情報をサーバに保管しておくことで、ウィンドウ画面の配信先端末を自動で決定し、共有開始手続きの負担を軽減するインタフェースを開発した。本稿ではアプリケーション仮想共有システムと、本研究で開発したインタフェースについて述べる。

## 2. アプリケーション共有システム

本章では実行例や図を用いながら、本研究で開発したアプリケーション共有システムについて述べる。ここで、アプリケーションを公開する側をホストとし、閲覧する側をゲストとする。本システムは、ホスト側のデスクトップ上に表示されたアプリケーションのウィンドウを、ゲスト側のデスクトップ上に表示する。

図1は本システムの実行例である。図内には、計算機1、計算機2、計算機3の三つのPCがある。また、破線で囲まれているのは各端末のデスクトップ上に表示されたウィンドウを示す。実行例では計算機2および3のデスクトップ上に表示されているアプリケーションのウィンドウ画面を、2枚ずつ計算機1へ配信している。配信された画面はシステムが開いたウィンドウで再生され、リアルタイムで同期している。

本研究では、ホスト側、およびゲスト側で起動するための、

連絡先: 岩田知, 名古屋工業大学工学研究科情報工学専攻, 愛知県名古屋市昭和区御器所町, 052(735)5584, iwamoto@toralab.org



図1: 実行例：アプリケーション共有

本システムを組み込んだアプリケーションの開発に Electron を利用した。Electron は Web 技術を用いたアプリケーション開発に利用できるライブラリである。本システムの P2P 通信や、ウィンドウ画面のストリーミング配信は、WebRTC 技術を応用した。また、ゲスト側は Web アプリケーションとしての開発も行い、iPad などのモバイルデバイスからでもブラウザを用いてアプリケーションを共有することができる。

## 3. 開発したインタフェース

本章では本研究で開発した、共有開始手続きの円滑化を可能とするインタフェースについて述べる。グループ内で利用する PC 端末の情報を事前に設定しておくことで、ウィンドウ画面の配信先を自動で決定し、共有開始手続きを軽減するインタフェースを開発した。図2はグループを管理するためのインタフェースである。グループ作成ボタンでグループを作成できる。作成したグループは、グループ一覧に追加される。グループ内メンバー一覧には、グループ一覧で選択したグループに属する PC 端末の一覧が表示される。アプリケーション共有ボタンで共有するウィンドウ画面を選択する画面が表示され、選択および決定すると、現在選択中のグループに属する PC 端末のデスクトップ上に共有されたウィンドウ画面を表示する。また、グループ内メンバー一覧の中の一対に対してのみウィンドウ画面を配信するといったことも可能である。



図 2: グループ管理インタフェース

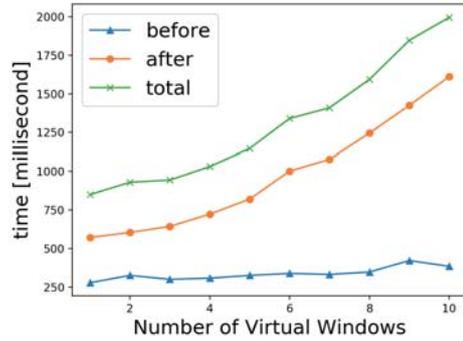


図 4: 全体時間

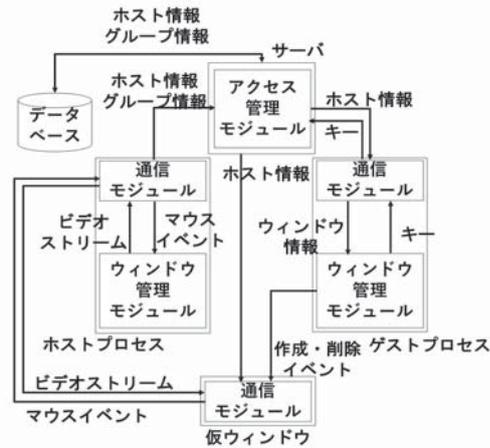


図 3: システム構成図

#### 4. システム構成図

本章では本システムの実装手法について述べる。図3は本システムの構成図を示す。本システムは大きくホストプロセス、ゲストプロセス、サーバ、仮ウィンドウの四つに分けることができる。ホストプロセスは通信モジュールおよびウィンドウ管理モジュールを持っている。ウィンドウ管理モジュールの役割は、ウィンドウのビデオストリームの取得およびマウスイベントをポストすることである。通信モジュールはサーバや仮ウィンドウとの通信を行う。サーバには、ホスト側のデスクトップ上に表示されているウィンドウや、それらを一般公開するかどうかなどの情報を送信する。サーバでは、ホストプロセスおよびゲストプロセス間の通信を行なっている。ホストプロセスから送られたホスト情報はデータベースに格納される。また、サーバ内にはアクセス管理モジュールがある。アクセス管理モジュールでは、各ホストが一般公開するウィンドウ群と、事前に登録したグループに関する情報を管理している。ゲストプロセスは通信モジュールおよびウィンドウ管理モジュールを内包する。通信モジュールはサーバとの通信を行う。サーバから受け取ったホスト情報から、ホスト名やウィンドウ情報などを抜き出し、ホスト選択リストや仮ウィンドウ選択リストに表示している。ウィンドウ管理モジュールでは通信モジュールから受け取ったウィンドウ情報から仮ウィンドウ作成イベントを発生させ、仮ウィンドウを表示する。

#### 5. 評価実験

本章では評価実験について述べる。評価実験では、実際にゲスト側 PC のデスクトップ上にウィンドウが表示されるまでの経過時間を調査した。経過時間は、ゲスト側 PC のデスクトップをビデオカメラで撮影し、アプリケーション共有の操作を完了してから、ウィンドウが表示されるまでのフレーム数をビデオカメラのFPSで割ることにより求めた。ビデオカメラはFPSが60のものを用いた。また、共有したウィンドウ画面のサイズは960x1,920とし、同時に共有するウィンドウ数は1から10枚とし、各枚数の場合を同じ手法で測定した。

図4は、ビデオカメラで撮影した結果から取得した各時間である。横軸はゲスト側 PC のデスクトップ上に表示されたウィンドウ数、縦軸は計測した時間を示す。before が共有されたウィンドウのフレームが表示されるまでの時間、after がフレームが表示されてから共有されたウィンドウの画面が表示されるまでの時間である。また、total は before と after の和である。本システムはウィンドウのフレームが表示されてからP2P通信を始め、デコードなどのWebRTCに関連する処理を行うようになっている。実験結果より、WebRTCに関連する処理が共有されたウィンドウ数に依存することがわかった。共有されたウィンドウ数が増えるほど経過時間は増すが、10枚でもおよそ2,000msであり、実用的であると考えられる。

#### 6. おわりに

本稿では協調作業支援で利用されるアプリケーション共有システムにおける、共有開始手続きの円滑化が可能なインタフェースについて述べた。協調作業を行うグループを管理するためのインタフェースを実装した。グループ内のPC端末をアプリケーション共有先の対象とすることで、共有開始手続きを軽減する。評価実験では本インタフェースを用いた場合に、実際に共有が開始されるまでに必要な時間を計測することで、実用性について評価した。要する時間は最大でもおよそ2,000msであり、実用的であると判断した。

#### 参考文献

[1] Satoru Iwata, Tadachika Ozono and Toramatsu Shintani.: Any-Application Window Sharing Mechanism based on WebRTC, 2017 6th IIAI International Congress on Advanced Applied Informatics, vol.00, pp.808-813(2017).