

ウィンドウをドッキングすることによるマルチタスキング支援

柴田 博仁[†]

大村 賢悟[†]

コンピュータ上でのマルチタスキングを支援するウィンドウ環境として *Docking Window Framework* を提案する。提案システムでは、ドッキングという簡易なユーザインタフェースを介して複数のウィンドウからなるワークスペースをタスクごとに構築し、タスクの切り替えを容易にする。また、タスク内のウィンドウに対して、ウィンドウが重なり合わないようタイル配置し、全てのウィンドウの内容が見える状態にありながらも特定のウィンドウを大きく（または小さく）表示する機能を提供することにより、タスク遂行におけるウィンドウ操作負荷を軽減することを狙いとする。複数のタスクを切り替えながら作業する実験をとおして、提案システムは通常のウィンドウシステムに比べて14~21%効率的に作業できることを示した。

The support of multitasking through window docking

HIROHITO SHIBATA[†]

KENGO OMURA[†]

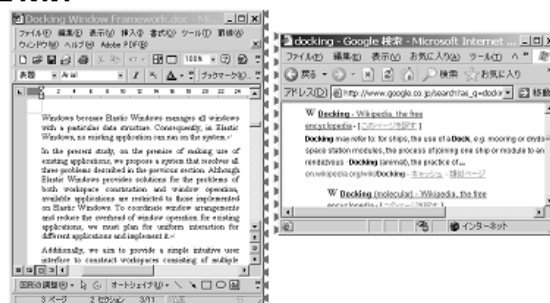
This paper proposes an extended window system called *Docking Window Framework* to support multitasking situations in computers. The proposed system enables to construct workspaces consisting of multiple windows through simple user interface of window docking and to easily switch workspaces. Additionally the system reduces the overhead of window operations within workspaces due to a tiled window approach that does not cause window overlapping and an enlarging (or narrowing) feature to display a specified window as large (or small) as possible remaining all window contents visible. The experiment of a task-switching task showed that subjects performed the task using the proposed system 14% to 21% faster than the case of a traditional window system.

1. はじめに

コンピュータ上で作業を行う場合、1つの文書または1つのアプリケーションで作業が遂行されるわけではない。むしろ、1つの作業の遂行に、複数の文書、複数のアプリケーションを必要とするのが一般的である。論文を書く場合を例にとると、ワードプロセッサの他に、辞書、自分がこれまで書いた論文、他人の論文、図を作成するためのドローツールなどが必要である。そして、これらに対応する複数のウィンドウを作業しやすい位置に配置し、各々に独自のワークスペースを暗黙的に構築して作業を行っているのが実情である¹⁾。

このような業務を現状のウィンドウシステムで行なおうとすると、作業の種類ごとにウィンドウをまとめて作業空間（ワークスペース）を構築し、これらを切り替えながら業務を行うことができないという問題が生じる。現状では、作業を切り替えるたびに、複数のウィンドウをクリックする必要があり、負荷が高い。そして、どのような作業がどれくらいあるのかも把握できない。

Before



After

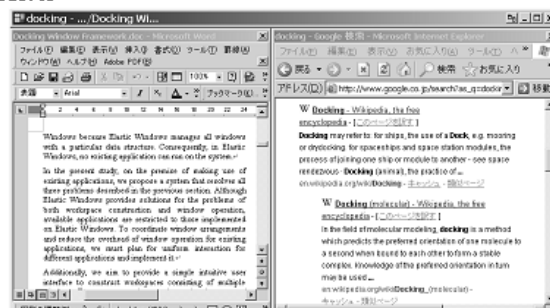


図1 ドッキング前後でのウィンドウの振る舞い

この問題は古くから指摘されており、これまでにさまざまな解決策が示されてきた¹⁾⁻⁵⁾。これら研究では、複数のウィンドウをグループ化し、ウィンドウを同時に前面化したり、閉じたりという操作を可能にする。

[†] 富士ゼロックス株式会社 研究技術開発本部
Research and Technology Group, Fuji Xerox Co., Ltd.

しかし、従来研究では、ワークスペースで作業を行うに先立って、どのウィンドウ集合をグループとするのかを指定して事前にワークスペースをデザインしておく必要がある。言い換えるなら、従来のシステムはワークスペース構築後のタスク切り替えを支援するものであり、ワークスペースの構築を容易にすることを主眼においていなかった。ワークスペースは作業中に動的に組みかえられることも少なくないため、ワークスペースを構築する際のユーザインタフェースの改善は運用上大きな効果をもたらすと考える。

さらに、ウィンドウの数が増えるとウィンドウ間の重なりが増え、ウィンドウを切り替えたり、移動したり、サイズを変更するといった操作が増える。われわれが知的財産管理業務でのウィンドウ操作ログを分析した結果では、PCで作業を行っている時間の実に7.4~9.1%がウィンドウ操作に費やされていた⁶⁾。これがPCを用いて本来行いたい知的作業とは全く関係のない操作に費やされている時間であることを考慮すれば、ウィンドウ操作のコストはできるだけ少なく抑えることが望ましい。

以上の点をふまえて、筆者らはワークスペースの構築を可能にし、マルチタスキングでのウィンドウ操作負荷を軽減するウィンドウ環境 *Docking Window Framework (DWF)* を提案し、システムの構築と評価を行ってきた⁷⁾。今回、タスク切り替えを含んだ作業におけるシステムの有用性を検証するための実験を実施したので報告する。

2. システム

本稿で提案するウィンドウ環境は以下の特徴をもつ。システムはMicrosoft .NET Framework 2.0上で動作し、C#で実装されている

- ワークスペース構築のユーザインタフェース
- 複数のウィンドウに対する一括操作
- タイルレイアウトの利用
- ワークスペースの保存と再現

2.1 ワークスペース構築のユーザインタフェース

ワークスペースを構築するための簡単かつ直感的なユーザインタフェースとして、本研究ではパズルのピースをはめ込むようにウィンドウをつなぎ合わせる「ドッキング」と呼ぶインタラクシオンを提案する。

図1にウィンドウのドッキングの前後の振る舞いを示す。ウィンドウを他のウィンドウの近くにドラッグするとウィンドウの両端にギザギザの太い線が表示される(図1上)。これは、両者のウィンドウがドッキング可能であることを示すと同時に、ドッキングの際

にどの位置とどの位置がつなぎ合わされるのかを示す。この状態でウィンドウをドロップするとウィンドウのドッキングが行われる。ドッキング後に全体が長方形になるよう、ドロップされたウィンドウは他方のウィンドウの高さ(縦方向のドッキングの場合には幅)にサイズ変更される(図1下)。ドッキング操作の前後ではウィンドウの位置とサイズが変化するため、その過程がわかるようアニメーション表示を行う。

ドッキングされたウィンドウは、タスクバー上でひとつのアイコンとしてまとめられる。また、ドッキングされたウィンドウ同士は、同時に前面化や移動が可能であり、あたかもひとつのウィンドウであるかのように振る舞い、ワークスペースとして扱うことが可能である。従来のシステムはいずれも、どのようなワークスペースを生成し、そこにどのウィンドウを配置するかを何らかの手段により指定する必要があった。本提案では、ウィンドウをドッキングさせることで1つのウィンドウのように振舞わせ、それをワークスペースと見なす枠組みを提供している。これにより、ワークスペースの構築を宣言することなしに、ウィンドウを並べるといった行為をそのままワークスペースの構築へとつなげることができる。

2.2 複数ウィンドウに対する一括操作

ワークスペース内のウィンドウに対しては一括で操作できることが望ましい。目的を達成するための操作回数が減るだけでなく、複数のウィンドウからなるワークスペースがあたかもひとつのウィンドウであることをユーザに意識させるためにも有効である。DWFでは、結合された複数のウィンドウに対して、以下の一括操作が可能である。

前面化 ワークスペース内のウィンドウあるいはタスクバーでのアイコンをクリックすることにより、ワークスペース内のすべてのウィンドウを同時に前面化することが可能である。これにより、1クリックでワークスペースの切り替えができ、タスク切り替えが容易になる。

移動 ワークスペースのバーあるいは個々のウィンドウをドラッグすると、ワークスペース内の全てのウィンドウが同時に移動する。これにより、ワークスペースの移動や整理が容易になる。

サイズ変更 ワークスペースをひとつのウィンドウのように見せるため、さらには表示領域を有効利用するため、DWFではワークスペースが常に長方形になるようにする。そこで、ワークスペース内のウィンドウをサイズ変更すると、それに応じて他のウィンドウのサイズも変化する。

極小化/極大化 ワークスペース内の全てのウィンドウ内容が見える状態にありながら、それでも指定したウィンドウをできるだけ大きく（または小さく）表示する機能として、極大化/極小化と呼ぶ機能を提供する。特定のウィンドウを極大化（または極小化）することにより、他のウィンドウのサイズや位置も変更される。この変化の過程はアニメーション表示される。

オープン/クローズ ワークスペースをオープンすることにより、ワークスペース内の複数のウィンドウが同時に開かれ、ウィンドウ位置や表示文書が再現される。また、ワークスペースをクローズすることにより、その中の全てのウィンドウが同時に閉じられる。これによりワークスペースの立ち上げと終了が容易になる。

2.3 タイルレイアウトの利用

現状の多くのウィンドウシステムは、ウィンドウの重なりを許すオーバーラップ・ウィンドウシステムを採用している。しかし、状況によっては、オーバーラップ・ウィンドウシステムよりもウィンドウが重なり合わないタイル・ウィンドウシステムのほうが効率的に作業できることが示されている⁸⁾。そこで、DWFではタイルレイアウト方式を利用する。各ウィンドウのサイズ変更に伴って他のウィンドウの位置やサイズを調整する仕方については、Elastic Windows²⁾で採用されているものと基本的に同じである。

2.4 タスク状態の再現

DWFでは、ワークスペースの状態をファイルに保存し、後で再現することが可能である。これにより、PCを立ち上げるたびにワークスペースを最初から構成し直す手間を省くことができる。

3. 評価実験

複数のタスクを切り替えながら作業する課題において、DWFの有効性を検証するための実験を行った。

3.1 方法

実験デザインと被験者 実験は以下の4種類の作業環境を要因とする1要因被験者内デザインである。

- 紙で作業 (Paper 条件)
- DWFを用いて作業 (DWF 条件)
- 通常のウィンドウシステムで初期状態がタイル配置 (Tile 条件)
- 通常のウィンドウシステムで初期状態がカスケード配置 (Cascade 条件)

被験者は男女同数の18名であり、全員が20代から30代(平均は29.5歳)だった。全員がPC利用暦3年

以上、矯正視力0.7以上であった。

被験者は各々の作業環境で2回ずつ課題を行った。作業条件や文書集合の試行順の効果が相殺されるようカウンターバランスをとった。

材料 実験での課題は、商品リストから条件にマッチする商品の数をカウントする独立な2つのタスクを切り替えながら作業を行うことである。

実験で利用する文書は、デジタルカメラ、プリンタ、液晶ディスプレイなどの商品情報をもとに独自に作成した。1つのタスクで利用する文書は4つあり、いずれも1ページの文書である。4つのうち3つは商品リスト(商品属性は商品名、メーカー、重さ、値段、売れ筋ランキングなど)であり、残り1つは質問文のリストである。商品数は1文書に16個であり、タスク内の3文書では48個になる。1つのタスクでの質問文は全部で6個である。液晶ディスプレイに対する質問文の例を下に示す。

- メーカーがACERのディスプレイは何個か?
- 消費電力が20W未満のディスプレイは何個か?

装置 PCを利用する電子環境において、PCはDell Dimension C512、ディスプレイはNanao製の23インチモニターを利用した。OSはWindows® XPを利用した。

手続き 実験では、2つの商品カウントタスクを切り替えながら作業を行ってもらった。図2に示すように、被験者は最初のタスクの前半3問の質問に答えてから、もう一方のタスクの前半3問に答える。次いで、再び、最初のタスクの後半3問に答えてから、もう一方のタスクの後半3問に答える。タスク切り替えは全部で3回生じることになる。被験者には、以上の作業をできるだけ速かつ正確に行うことを求めた。

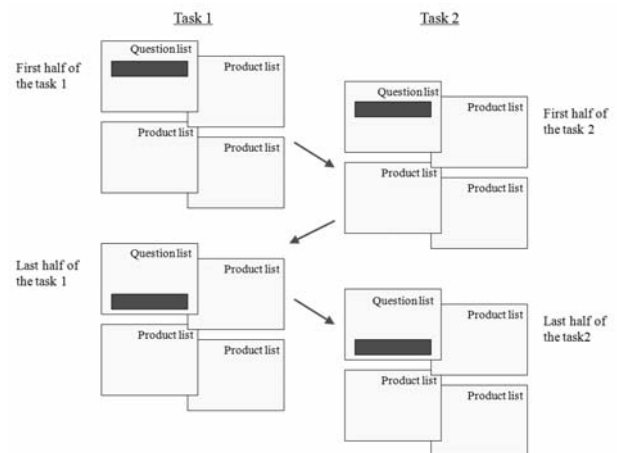


図2 課題におけるタスク遂行の順番

Paper 条件では、A5用紙にモノクロでプリントした紙を利用した。それ以外の電子条件では Adobe®

Reader 9 を利用した。文字サイズが Paper 条件と同じになるよう、事前に表示文書の縮尺を調整した。

3.2 結果

作業条件ごとの作業時間の比較を図3に示す。作業時間に関して1要因分散分析を行ったところ、作業条件の主効果が有意であった [$F(3, 51)=20.35, p<.001$]。LSD法の下位検定を行うと、Paper条件はDWF条件よりも有意に作業時間が短く [$p<.01$]、DWF条件はTile条件よりも有意に作業時間が短く [$p<.01$]、Tile条件はCascade条件よりも作業時間が短い傾向がみられた [$p<.1$]。DWFはタイル配置のウィンドウシステムに比べて14.0%作業時間が短く、カスケード配置のウィンドウシステムに比べて22.0%作業時間が短い。また、紙はDWFよりも高速であり、DWFに比べて14.9%作業時間が短かった。

課題の正解率については、平均ではCascade条件が79.9%で最も正解率が高く、Tile条件が71.5%で最も正解率が低かったが、有意差は見られなかった [$p>.1$]。

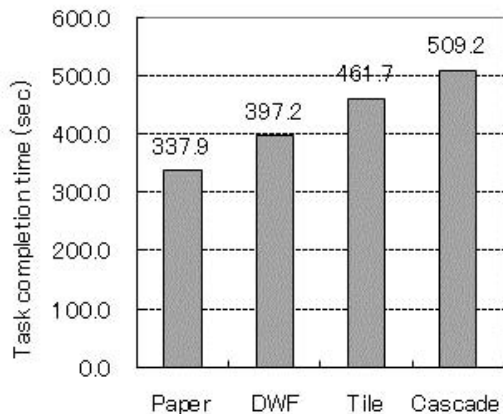


図3 作業時間の比較

3.3 考察

タスク切り替え課題の遂行において、DWFは通常のウィンドウシステムよりも高速である。1クリックでタスク切り替えが容易に行えること、タスク内でも極小化/極大化の機能を利用することで必要な文書の全体を簡単に閲覧できることが課題での高速化につながっていると考えられる。

また、提案システムは紙の作業効率に及ばなかった。提案システムは、1クリックでのタスク切り替え、タイル配置、極小化/極大化など、紙では提供できないマルチタスキングを支援するさまざまな機能を盛り込んでいる。それでも提案システムが紙に及ばない点をふまえると、紙はマルチタスキングを行うための優れたツールであると同時に、紙をお手本としてマルチタスキングを効率的に支援する方法論を獲得できること

が示唆される。

4. おわりに

提案システムは、タスク切り替えを頻繁に伴う作業において14%から22%効率的に作業を行うことができた。これは通常のウィンドウシステムに対する提案システムの優位性を示すものであるが、提案システムは紙の作業効率には及ばなかった。今後は、紙と人間とのインタラクシオンを詳細に分析し、マルチタスキングを効果的に支援する紙の利点をシステムの改良へと反映させていきたい。

登録商標について

- Adobe® Reader は、アドビシステムズ社の米国およびその他の国における登録商標または商標です。
- Windows®は、Microsoft Corporation の米国およびその他の国における登録商標または商標です。
- その他、掲載されている会社名、製品名は各社の登録商標または商標です。

参考文献

- 1) Henderson, J.D.A., and Card, S.K.: Rooms: The use of multiple virtual workspaces to reduce space contention in a window-based graphical user interface, *ACM Transactions on Graphics*, Vol.5, No.3, pp.211-241, (1986).
- 2) Kandogan, E. and Shneiderman, B.: Elastic Windows: Evaluation of multi-window operations, In *Proc. CHI '97*, pp.250-257, (1997).
- 3) Robertson, G., et al.: The Task Gallery: A 3D window manager, In *Proc. CHI '00*, pp.494-501, (2000).
- 4) Robertson, G., Horvitz, E., Czerwinski, M., Baudisch, P., Hutchings, D., Meyers, B., Robins, D., and Smith, G.: Scalable Fabric: Flexible task management, In *Proc. AVI '04*, pp.85-89, (2004).
- 5) Smith, G., et al.: GroupBar: The TaskBar evolved, In *Proc. OZCHI '03*, pp.41-50, (2003).
- 6) 柴田 博仁: 大画面ディスプレイ・多画面ディスプレイの導入による業務効率化の測定. 情報処理学会論文誌, Vol.50, No.3, pp.1204-1213, (2009).
- 7) 柴田 博仁, 大村 賢悟: ワークスペースの構築を可能にするウィンドウシステムの提案と評価. 人工知能学会論文誌, Vol.26, No.1, pp.237-247, (2011).
- 8) Bly, B. and Rosenberg J.: A comparison of tiled and overlapping windows, In *Proc. CHI '86*, pp.101-106, (1986).