

グループ共有が可能な Web デスクトップの構築

鈴木啓真^{†1} 兼子正勝^{†2}

本論文では、Web デスクトップとグループウェアの共有機能を融合させた共有型 Web デスクトップ“AppTop”を提案する。AppTopではアプリケーションレベルの共有ではなく、デスクトップレベルの共有を実現している。これにより、従来のグループウェアでは不可能だった自由度の高い、幅広い共有を行うことが可能になった。

Building the Shared Web Desktop

HIROCHIKA SUZUKI^{†1} MASAKATSU KANEKO^{†2}

This research propose a shared web desktop “AppTop” that is a web desktop with sharing function like a groupware. Using AppTop, you can share “desktop” with other people. AppTop is platform-level sharing, not application-level, so AppTop realize freer and more useful sharing services than groupware.

1. はじめに

近年、データや設定情報等をインターネット上に保存する、もしくはアプリケーションを Web ブラウザ上で動作させるクラウド技術が注目されている。クラウド技術のメリットとして大きく二つが挙げられる。

一つ目のメリットは、インターネットにさえ繋がっていればどのコンピュータからでも同じ環境にアクセスでき、データが手に入るという特定のコンピュータへの非依存性である。このメリットを最大限に利用した例として Web デスクトップが挙げられる。Web デスクトップとは、Web アプリケーションの概念をよりクラウド化させたものと言え、OS 自体を Web ブラウザ内で動作させる概念、およびその OS (Web サイト) のことを指し、WebOS と呼ばれることもある。つまり、Web デスクトップではアプリケーションを動作させて使用する、各種データを保存するといったコンピュータ上で行う作業をすべて Web ブラウザ上で実行する。この Web デスクトップを利用することで、どのコンピュータであっても Web ブラウザを立ち上げて Web デスクトップにログインすることで、普段使用しているコンピュータ環境を利用することができる[1]。また、ユーザが利用している様々なサービスをアプリケーションとして Web デスクトップ上で動作させることで、複数利用しているサービスを 1ヶ所へ統合することができるというメリットもある。既存の Web デスクトップサービスの例として Netvibes[2]や AstraNOS[3]がある。それぞれが Web ブラウザ上で動作しているスクリーンショットを図 1 と図 2 に示す。



図 1 Netvibes



図 2 AstraNOS

もう一つのメリットは、Web の特性とも言える共有である。クラウド技術を利用することで、データのやり取り、ドキュメントの共同制作といった共有が気軽に行うことができる。このような共有を積極的に利用している例として、Web システムであるグループウェアが挙げられる。グループウェアとは企業をはじめとした様々な組織内での情報共有やプロジェクトの推進をサポートするシステムである。グループウェアを利用することで、通常組織運営はもちろん、メンバー同士が遠距離で会うことができない組織や、

^{†1} 電気通信大学 情報理工学研究所 総合情報学専攻

^{†2} 電気通信大学 情報理工学研究所 総合情報学専攻

時間の制約が大きくてミーティングが行ない組織など、通常では難しい組織においても運営をサポートすることができる。

しかしながら、このようなグループウェアは複数人での議論、スケジュール管理、TODO 管理、データ共有といったグループ活動に必要な機能のみの提供である上、他のアプリケーションを自由に追加するという拡張性も不十分であり、日常的に利用することは難しい。よって、グループ活動が必要な時にのみグループウェアを使用することになる。これにより新着情報などを見逃すといった問題が発生し、議論の活発化を妨げてしまう。また、Web の活発化に伴い、SNS やニュースサイト、クラウドサービスなどユーザは多くの Web サービスを利用している。そしてユーザは様々なサービスに登録、利用、そして使い分けるといった負担を強いられているのが現状である[4]。その中で更にグループ活動のためだけの Web サービスを利用しなければならないのは、ユーザの負担の点から考えても望ましいとは言えない。

そこで本論文では、汎用性の高い Web デスクトップにグループウェアの共有要素を融合させることで、より親密な共有ができる「共有型 Web デスクトップ “AppTop”」を提案する。

2. デスクトップと共有

コンピュータの GUI 画面である「デスクトップ」は、「机」をコンピュータ上で再現したものだと言える。実際の机の上には作業中の書類があったり、文具や時計といったツールが置かれていたりする。そしてそれらは個人の趣味嗜好に応じて自由に配置されている。この机は個人の机であるため誰かに邪魔されることはない。同様に、通称「デスクトップ」も個人のための GUI であり、ここでの作業は誰かに邪魔されることはない。

一方、机には複数人で共有して使われるものもある。共有の机の例として「ミーティングルームの机」が挙げられる。ミーティングルームの机は参加者全員に囲まれており、その机の上には議論されている内容に関連した書類が置かれていたり、参加者で共有している文具等のツールが置かれていたりしている。そして机の上ではないもののすぐ側にはホワイトボードがあり、議論の進行をサポートしている。このような共有機をコンピュータ上で「デスクトップ」として再現したものが「共有型デスクトップ」である。

共有型デスクトップに似た概念を持つ Web サービスとして、lino[5]が挙げられる。図 3 のように、lino はコルクボードをイメージした Web ページを複数人で共有して、付箋を貼ったり写真を貼ったりすることでコミュニケーションを活発にさせる Web サービスである。しかし、付箋や写真を貼ること以外ではできず、デスクトップのようにアプリケーションを動作させることはできない。アプリケーション

を動作させることができないため、拡張性がなく使用方法が限定的である。よって lino は共有型デスクトップを持ったシステムとは言えない。

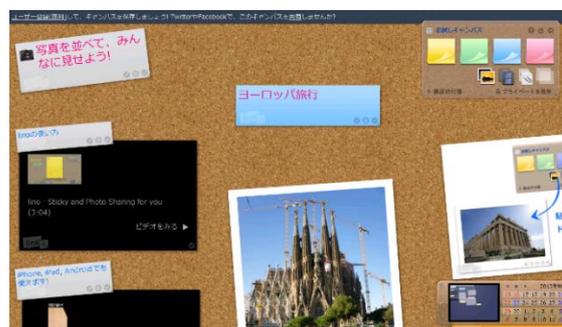


図 3 Lino

また、1 章で説明した既存の Web デスクトップサービスである Netvibes や AstraNOS も共有機能は持っておらず、個人的に利用することを目的としている。よってデスクトップを持ったシステムではあるが、共有型デスクトップを持ったシステムとは言えない。

本論文で提案する「共有型 Web デスクトップ “AppTop”」の新規性は、このような共有型デスクトップを持ったシステムを実現していることである。

3. AppTop

本論文で提案した「共有型 Web デスクトップ “AppTop”」について説明する。共有型 Web デスクトップとは共有型デスクトップを Web デスクトップ上で実装したものである。AppTop は共有型 Web デスクトップであるが、必要に応じてデスクトップをプライベートモードと共有モードに切り替えることができるため、従来の Web デスクトップと共有型 Web デスクトップの両面の特徴を持っているといえる。

プライベートモードは個人利用を目的とした、従来の Web デスクトップと同様の概念である。一方、共有モードは複数人で共有することを目的とした、共有型 Web デスクトップで実装された新たな概念である。共有型デスクトップでは、1 つのデスクトップを複数人で共有することができ、その共有したデスクトップ上では、通常のデスクトップと同様にアプリケーションを動作させたり、データを保存したりすることができる。それらの操作や編集、閲覧は共有しているメンバーによって同時に行うことができる。つまり、スケジュールアプリやメモアプリなどをデスクトップ上に追加することでグループウェアとしての使用ができるのはもちろん、アプリケーションの導入次第で様々な使用方法を期待することができる。

3.1 システム概要

AppTop のシステムは Web サーバとデータベース (DB) サーバ、および利用者のクライアント PC で構成されてい

る。Web サーバ上では PHP を用いたシステムが作動している。このシステムはクライアント PC から来た各種リクエストを処理するいわゆるメインシステムである。DB サーバは、アカウント情報およびユーザの各種データの保存に利用されている。AppTop で利用されるデータはすべて DB サーバに保存される。また、DB サーバへのアクセスは Web サーバ経由で行われ、クライアント PC との直接的なアクセスは行われない。クライアント PC の Web ブラウザ上では JavaScript を用いたシステムが動作している。このシステムは主に Web ページ上の GUI を管理している。JavaScript を用いているため Adobe Flash のように Web ブラウザに特別なプラグインを必要とすることなく利用できる。そのため、汎用性が高いシステムであると言える。

3.2 デスクトップ

AppTop がブラウザ上で動作しているスクリーンショットを図 4 に示す。これが AppTop のデスクトップ画面である。そして、デスクトップ上には様々なアプリケーションのウィンドウが存在している。AppTop ではこのようなアプリケーションを「アプリ」と呼んでいる。図 4 の例では、デスクトップ上にニュースアプリ、付箋アプリ、天気予報アプリなどが並んでいる。そして、デスクトップは作業スペースであるという設計思想を重視し、デスクトップ上からアプリ以外のものは可能な限り排除した。これにより、アプリを自由にそして最大限に設置できるようにしている。そして、AppTop は 1 アカウントあたり複数のデスクトップを所有することができたため、目的に応じてデスクトップを切り替えて使用することができる。この機能を「ページ機能」と呼んでいる。

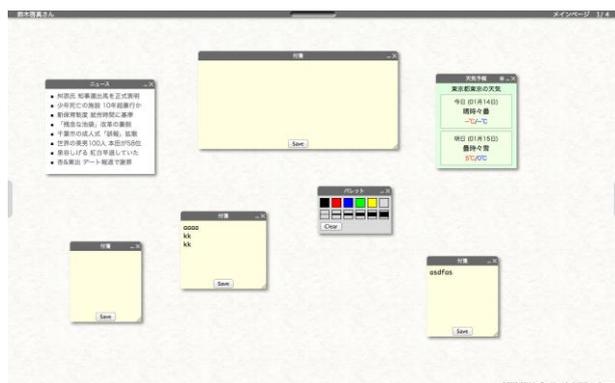


図 4 AppTop のデスクトップ

また、デスクトップから排除されたメニューや設定画面など各種ツールはすべて「コントロールパネル」に設置している。AppTop は主にデスクトップとコントロールパネルの 2 つのウィンドウで成り立っている。コントロールパネルはデスクトップの上部バーを下にドラッグすることで、図 5 のように引き出すことができる。同様にコントロールパネルをドラッグして上に引き上げることで収納すること

ができる。ドラッグして引き出す方式を採用した理由は、ページ遷移を発生させないためである。デスクトップから排除された各種ツールはすべてコントロールパネルに設置されているため、コントロールパネルへのアクセスの手軽さは必須条件であり、ページ遷移を発生させないことで、ユーザにとっての紛らわしさを軽減している。コントロールパネルでは、利用可能なアプリ一覧取得およびアプリの追加、デスクトップおよび共有の管理、アカウント情報の確認および変更、ログアウトを行うことができる。コントロールパネル上のコンテンツは HTTP 非同期通信である Ajax を用いて、必要になった際に読み込まれる。そのため、Web ページの初回の読み込み時に、コンテンツ量が多いコントロールパネルを読み込む必要がなく、読み込み時間の短縮に寄与している。



図 5 コントロールパネルはドラッグして引き出す

3.3 共有機能

共有型 Web デスクトップとして最も重要な機能である共有機能について説明する。共有機能を使うことで 1 枚のデスクトップを複数人で共有することができる。共有機能は先ほど説明したページ機能を用いて実装されており、複数所有するデスクトップから任意のものを選択して共有モードに切り替え使用することができる。デスクトップのデフォルトは個人利用を想定したプライベートモードになっている。

共有している様子を図 6 に示した。ユーザ A はページ機能を使用しており、デスクトップを 5 枚所有している。まず、共有したいメンバーにユーザ D を指定し、手前から 2 枚目のデスクトップを共有モードに設定した。共有の招待を受けたユーザ D は招待を受け入れ、その共有されたデスクトップを自分の 1 枚目に設定した。これにより、ユーザ A の 2 枚目とユーザ D の 1 枚目は 1 つのデスクトップを二人で共有している状態となる。1 つのデスクトップを共有しているため、全く同じものが表示され、編集も共有メンバー全員に反映される。さらにユーザ A は 4 枚目も共有モードにしている。4 枚目の共有は先ほどとは異なるメンバーで設定している。AppTop ではこのように共有モード

にしたデスクトップを複数持つことができ、それぞれのデスクトップに対して異なる共有メンバーを設定することができる。

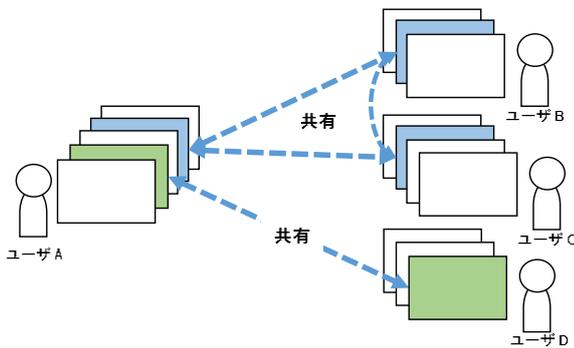


図 6 共有機能のモデル図

共有機能の仕組みについて説明する。AppTop で実装している共有は P2P モデルによる通信ではなく、クライアントサーバモデルによる通信を用いている。共有メンバーの 1 人がデスクトップに対して変更を加えた場合、変更内容のデータは Web サーバに伝えられる。そして Web サーバはその変更内容を恒久的に管理するために DB サーバに保存する。そして他の共有メンバーは再度 Web サーバにアクセスしなおすことで、メンバー内の 1 人の変更点が全員に反映されることとなる。これによりあたかも 1 つのデスクトップを共有しているかのように見せている。すべてのデータがサーバを経由するため、P2P 通信で発生するようなデータの整合性が取れなくなるといった問題が発生しない。

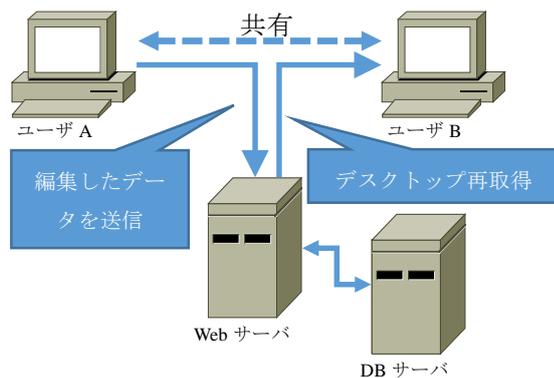


図 7 共有されたデスクトップにおけるデータフロー

3.4 ホワイトボード機能

AppTop の共有型デスクトップは共有機をイメージして作成されている。1 章で述べたように、共有機の側にはホワイトボードなどがありコミュニケーションを活発にしている。そのため、AppTop でもホワイトボード機能を実装することで、より具体的なコミュニケーションが取れるようにした。図 8 はホワイトボード機能を利用した例を示して

いる。AppTop のホワイトボード機能は、デスクトップの背面全体をホワイトボードとして利用できる。そのため、どのような場所にも自由に書き込むことができ、アプリ間の関係性や追記事項などをわかりやすく記述することができる。また、複数人でペンを用いて書き込むことでより直観的なディスカッションを行うことを可能にしている。ホワイトボード機能によって書き込まれた画像情報はすべてサーバに送られ、保存される。

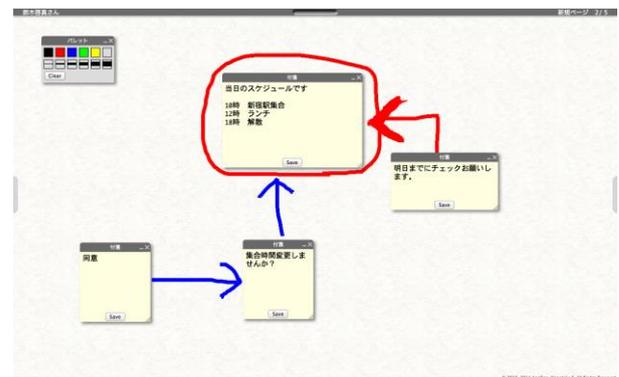


図 8 ホワイトボード機能の利用例

3.5 アプリおよび API

AppTop では AppTop 用の API を公開し、開発者環境を整えることでサードパーティーによるアプリ開発を可能にしている。そこでアプリおよび API について説明する。

AppTop 上のアプリは、JavaScript および AppTop 用の API を用いて作成されている。よって Windows や Mac で動作しているアプリケーションを AppTop 上でそのまま動作させることはできない。しかしながら、近年様々なアプリケーションが Web アプリケーションに移設されている。例として Google ドキュメント[6]がある。Google ドキュメントでは文章、スプレッドシート、プレゼンテーション作成を Web 上で行うことが可能になっている。そしてこれは JavaScript で実装されている、このような実例からも、JavaScript だけで十分なアプリが作成可能であると考えられる。そして、JavaScript は Web における標準的な言語であるため、既存の Web アプリケーションから AppTop 用のアプリに変換する作業がとても容易というメリットがある。また逆の作業も同様であると言える。よって、汎用性の高いプログラム作成ができ、開発者の負担が軽減できる。

アプリ開発時に利用する API について説明する。API は JavaScript によって記述されており、主に各種イベントハンドラと各アプリのための保存領域を提供している。提供しているイベントハンドラには「アプリが追加されたとき」や「削除されたとき」「設定画面を開くボタンが押されたとき」などがある。アプリ開発者はそれぞれのイベントハンドラに、任意の処理を設定することでアプリを作成していく。API が提供している保存領域は、各アプリが自由に使

用することが可能になっている。保存領域は各アプリそして各ユーザに対して与えられているため、同じアプリであっても利用しているユーザが異なる場合は、保存領域も異なることとなる。各アプリに与えられた保存領域は JSON 形式で任意のデータを扱うことができ、API を利用して読み書きする。この保存領域は API を経由し、最終的に DB サーバに格納されるため、異なる PC からログインした場合であっても同じデータが取得可能である。

各アプリはデスクトップから HTML タグである iframe タグを用いて読み込まれている。iframe タグとは Web ページ上に任意の枠を表示し、その枠の中に別の Web ページを組み込むことができるタグである。つまり、AppTop はデスクトップとアプリが独立した存在になっており、デスクトップ上に配置されているのはアプリを表示するための枠のみとなっている。そして、その枠の中にそれぞれ適当なアプリを表示することで、デスクトップ上にアプリが配置されているように見せている。このようにデスクトップとアプリを独立された理由は3つある。

一つ目は読み込み時間の低減である。iframe タグは Web サーバへ並列した読み込みを行うことが可能である。よってデスクトップと複数個あるアプリが並列して読まれることとなり、読み込み時間が低減される。また、あるアプリに問題が生じていて読み込みに長時間かかってしまう問題が発生したとき、直列してアプリを読み込む場合はその次に読み込まれるアプリに問題がなくても問題のあるアプリの読み込みが完了するまで待つしかない。しかし、並列して読み込む場合は、問題のあるアプリの影響を受けることなく、他の正常のアプリを読み込むことができる。

二つ目は、アプリケーションの開発しやすさである。アプリは AppTop のシステムや他のアプリから独立した存在にあるため、アプリ間の干渉やシステムとの干渉を考慮することなく自由にアプリを開発することができる。

三つ目はセキュリティの高さである。アプリ間の干渉が起きないため、アプリ間でのデータ傍受などを最少にとどめることができる。また、AppTop システムへは API を経由しない限りアクセスできないため、アプリからシステムに対する不正なアクセスも起きにくい。

以上のことから、デスクトップとアプリが独立した構成にした。

3.6 利用可能なアプリ

現在 AppTop 上で利用可能なアプリケーションは「付箋アプリ」「ニュースアプリ」「天気予報アプリ」「電卓アプリ」「ホワイトボードツール」である。AppTop ではサードパーティーによるアプリ開発を可能にしたため、アプリはサードパーティーによって増えていくことを期待しており、本研究ではアプリの開発に重点は置いておらず、必要最低限の開発のみにとどめた。

4. 評価 1

AppTop の利用者としての評価を行った。被験者は電気通信大学の学生、22 歳から 27 歳の男女 7 名、期間は 2013 年 12 月の 1 ヶ月間である。初めに基本的な使い方に慣れるために各自 AppTop を使用してもらった。その後、被験者内でグループを作成し、そのグループ内で共有したデスクトップを自由に利用してもらった。最後に 3 段階評価および自由記述のアンケートに答えてもらった。

まず、Web デスクトップの利用経験について質問した。結果としては、被験者のうち Web デスクトップを利用したことがある人は 2 人であり、残りの 5 人は聞いたことはないが使ったことはないと答えた。現状では Web デスクトップを利用している人はあまり多くないことが分かった。

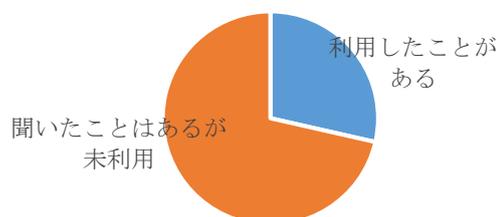


図 9 Web デスクトップの利用経験

次に、「Web デスクトップにおける共有機能は使いやすいか」という質問をした。結果は、5 人が「使いやすい」、2 人が「どちらともいえない」と答えた。「どちらともいえない」と答えた 2 人は、それぞれ「共有メンバーを追加する手順が面倒」「複数人が同時に編集した際の挙動が難しい」という理由によるものだった。

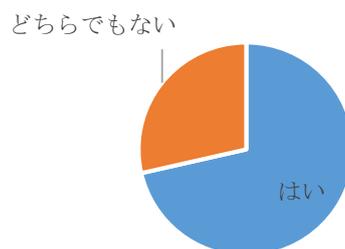


図 10 共有機能は使いやすいか

「Web デスクトップに共有機能を加えることにメリットがあると思いますか」という質問に対しては、6 人が「はい」、1 人は「どちらともいえない」と答えた。「どちらともいえない」と答えた 1 人は、「メリットがあるかどうかはアプリに依存すると思う」という理由によるものだった。

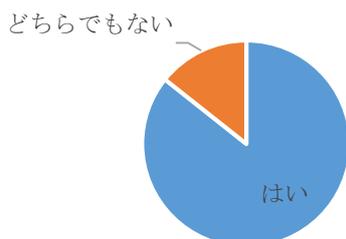


図 11 共有にメリットはあると思うか

「ホワイトボード機能はコミュニティーの活発化に寄与するか」という質問に対しては、7人が「はい」と答えた。「ペンを使って付箋に印をつけるなどコミュニケーションにおいて有効に使いそう」という感想が多くあった。



図 12 ホワイトボード機能は使いやすいか

また、「共有機能をどのように使いたいですか」という質問に対しては多くの人が「研究室で使いたい」と答えた。また「グループ作業時の拠点としたい」という意見もあった。共通していえることは、SNSのような「遊び感覚」というより、プロジェクトを運営する際のコミュニケーションツール、各自の状況管理など仕事において使いたいという点だ。そのほかの意見では「複数人での作業時にリアルタイムに更新してほしい」などがあつた。これらの結果より、AppTopは有用なサービスであるという結果が得られ、Webデスクトップに共有機能を実装することに対する有用性を示すことができた。

5. 評価 2

二つ目の評価として、AppTopのAPIを利用して、アプリ開発をしてもらい開発者としての評価を行ってもらった。被験者は電気通信大学の学生である27歳の男性1人、期間は2013年12月の1ヶ月間である。JavaScriptの学習を始めて半年程度のスキルレベルである。

実際に、いくつかのアプリを自由に開発してもらったが、スムーズにアプリ開発を行うことができた。AppTopのAPIの使い方やAppTopのアプリとして開発することに特別困ることはなかった。通常のWebとして作成していたJavaScriptプログラムを、AppTop用のアプリに変換するこ

とも行っていたが、すぐに変換してAppTop用のアプリにすることができたと述べていた。このことから、AppTopのアプリ開発は手軽に行えることが示された。

6. 考察

6.1 共有機能に関する考察

今回、Webデスクトップにグループウェアの要素を融合させた共有型Webデスクトップに有用性があるという結果が得られた。主な理由としては、Webデスクトップにグループウェアの共有の要素を融合させることで、従来のグループウェアよりも共有するということがより身近になったためであると考えられる。そもそもWebデスクトップには、場所やPC環境に依存せず、様々な使い方ができ、様々なサービスを一ヶ所へ集約できるという十分なメリットがあり、日常的に個人的利用をする価値があるものだと考えられる。そのようなWebデスクトップに共有の要素を融合させ、個人利用であっても共有利用であってもデスクトップを切り替えればすぐさま行き来できるAppTopはユーザにとって身近な存在になるものであると考えられる。

また、プラットフォームレベルの共有を実現することで、様々なグループ活動に対応することができ、グループ活動がより楽しくなるということも考えられる。たとえば、ある企業のグループ活動においてAppTopを利用すれば、カレンダーアプリでスケジュール管理を行い、付箋アプリでアイデアを記録するという通常のグループウェアとしての使い方だけでなく、Twitterアプリを設置することで企業のPR用アカウントをメンバー全員で管理できたり、チャットアプリを設置してその場でディスカッションを始めたりといった更に楽しい使い方ができると考えられる。また、デスクトップにホワイトボード機能を実装したことも大きな影響があつた。実際に被験者に利用してもらった際も、ホワイトボード機能が一番活発に利用されていた。グループ活動には、コンピュータの世界であっても実世界同様、ペンで書いてディスカッションしていくという行為が必要であることがわかった。

ただし、AppTopが提供しているのは共有サービスではなく、共有プラットフォームである。つまり、AppTopが便利なものになるかはアプリに強く依存していると言える。アプリを幅広くたくさん提供できれば、AppTopのグループ活動への使い道は幾万通りにも広がる。しかしながら、現状存在しているアプリだけでは通常のグループウェアよりも劣った存在になってしまう。AppTopの検討すべき点は、どのようにしてアプリを充実させていけばよいかという点であると考えられる。この点を考慮して、AppTopではアプリの作成のしやすさを考慮している。評価の結果から、AppTopアプリの開発は手軽であることが示されたが、これが示していることは通常のWebを作る作業とAppTopのアプリとして作成する作業は同レベルであるということである。

ある。つまり、そもそもの JavaScript でアプリケーションを作成するという行為の大変さについては示せていない。アプリは AppTop の要とも言える要素なので、開発しやすさについては今後も考えていかなければならない点である。

6.2 使いやすさに関する考察

使いやすさに関して高い評価を得ることができたのは、不要なものを極力排除したシンプルな UI によるものが大きいと考えられる。また、ドラッグでアプリを移動させたり、ページ遷移をなくしたりすることで、よりネイティブなアプリケーションに近い使い勝手を実現していることも大きく影響していると考えられる。

ただし、考慮しなければならない点は、評価のアンケート結果でもあったが、現在複数人で編集を行った際にリアルタイムで変更が反映されない点である。つまり、誰かが変更した場合、他のメンバーは手動で更新しないと反映されない仕様になっている。現在、変更をリアルタイムに反映させる実装を検討している。しかし、リアルタイムを実装すると問題になるのは排他制御の問題である。共有人数が少ない場合には問題にならないことが多いと思うが、多くなった場合に複数人で同時に編集をし、それがリアルタイムに反映されたらデスクトップの秩序が保てなくなってしまうことが想定される。アプリの移動一つにとっても、同時に移動させようとしたらアプリの移動の衝突が発生する。リアルタイム実装と同時に、この排他制御問題を検討し、今後の改善に向けて検討していく必要があると考えられる。

7. おわりに

本論文では Web デスクトップにグループウェアの共有の要素を融合させた「共有型 Web デスクトップ“AppTop”」を提案し、評価の結果より共有型 Web デスクトップの有用性を示すことができた。AppTop を用いることで共有するということをより身近に、そして楽しく活用できることがわかった。しかしながら、本システムが今後どのように発展するかについては、本システムで提供できるアプリの種類や数に大きく依存している。近年、Web 上でアプリケーションを作成し、動作させることが容易に、そして盛んになってきているので、AppTop としては、開発者向け環境を整えていき、より AppTop 用のアプリを開発しやすい環境を作ることに重点を置いて改良していく必要があると考えられる。

参考文献

- 1) George Lawton: Moving the OS to the Web, IEEE Computer Society, Vol41, No.3, pp.16-19 (2008)
- 2) Netvibes
<http://www.netvibes.com/>
- 3) AstraNOS
<http://www.astranos.org/>
- 4) Gianni Bosio: A User Perspective on Cloud Computing, Advances in Human-Oriented and Personalized Mechanisms, Technologies

- 5) and Services (CENTRIC), 2010 Third International Conference on, pp.1-4 (2010)
Lino
<http://ja.linoit.com/>
- 6) Google ドキュメント
<https://docs.google.com/>