

2009 年度秋学期 渡辺テーマ研究会
「大学と自宅間でのシームレスなファイル共有環境の検討」

指導教員 渡辺恭人

千葉商科大学 政策情報学部

学籍番号 0740159

酢崎 祐樹

目次

概要

大学と家とのデータのやり取りをより円滑にするプログラムの実装を目指す

1. 背景・目的

1. 1 背景

1. 2 目的

1. 3 報告書の構成

2. 大学と自宅間でのデータのやりとりの現状と問題点

2. 1 大学から自宅へ移動する場合

2. 2 自宅から大学へ移動する場合

2. 3 問題点

2. 4 まとめ

3. 解決法の検討

3. 1 解決へのアプローチ

3. 2 類似サービスのサーベイ

3. 2. 1 rsync, Cwrsync

3. 2. 2 Unison

3. 2. 3 グーグルドキュメント

3. 2. 4 Windows Home Server

3. 2. 5 Dropbox

3. 2. 6 ZumoDrive

3. 3 必要条件と機能

3. 4 オンラインストレージ

3. 5 サービスのイメージとモデル

3. 6 複数の人数でのファイル共有利用

3. 7 複数の人数でのファイル共有環境の提案

4. 設計と実装

4. 1 設計目標

4. 2 必要となる情報

4. 3 システムの構成と要素

4. 4 機能概要

4. 5 機能の実装

4. 5. 1 インターネットでのダウンロード

5. 評価と考察

5. 1 評価

5. 2 考察

5. 3 その他の機能の検討

6. まとめと今後の課題

6. 1 まとめ

6. 2 今後の課題

参考資料

謝辞

1. 背景・目的

1. 1 背景

大学生は、課題やレポートで大学と自宅間でのデータのやり取りが多い。課題が出た時その場で終わらなかったものや、家でやってくることを前提にした課題が出される事がある(図1.1-1)。しかし、編集集中のファイルの持って帰り忘れや、終わった課題ファイルを自宅に置き忘れた、といった問題が起こり、課題の提出不可能に繋がる事がある(図1.1-2参照)。問題の主な原因は既存のオペレーティングシステムでは、複数地点間での作業を想定していないので、遠隔の作業環境とのファイル転送が移動する度に発生する事が原因である。

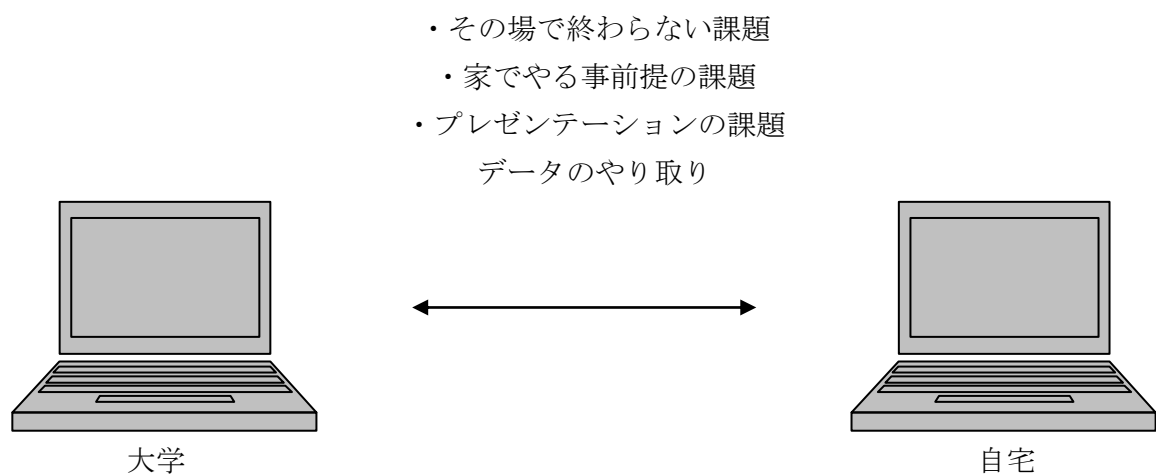


図1.1-1 大学と自宅間のデータのやり取り

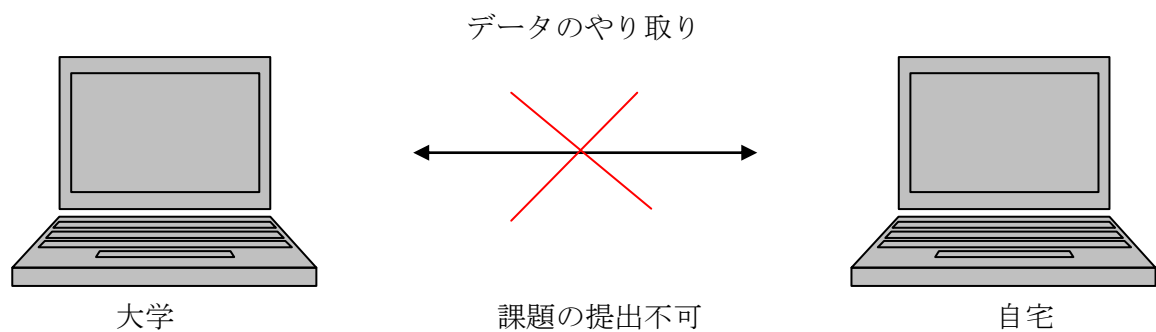


図1.1-2 データのやり取りに問題があった場合

1. 2 目的

本研究では、本学において学生がデータのやり取りにおいて起こる問題を低減させ、シームレスなファイル共有の実現を目的とする。面倒なデータのやり取りを低減させる為に既存のデータのやり取り、行うサービスを調べ、更に加えた方がいい機能や必要性の低い機能を検討していく。また、機能面だけでなくわかりやすいインターフェイスも考慮に入れる。

1. 3 本論文の構成

本論文では、第1章で、本研究に至った背景と、目指す目標について述べ、第2章では、各データのやり取りの現状と問題点に焦点を置いて述べる。また、第3章では、2章で述べた問題を解決するための方法について、議論、検討し、解決方法を考案する。さらに第4章では、3章で見極めた条件や機能を元に、サービス及びシステムの設計と実装について述べる。第5章では、出来上がったシステムの評価と考察する。そして第6章において、研究全体を振り返り、今後の課題を述べ、本報告書を締めくくる。

2. 大学と自宅間でのデータのやりとりの現状と問題点

本章では大学と自宅間、自宅と大学間でのデータのやり取りの現状を挙げ、その問題点と改善方法について述べる。

2. 1 大学から自宅へ移動する場合

大学から自宅へデータを持ち帰るにはUSBメモリがあると比較的スムーズに送れる。USBが無い場合はメールで送信することも可能である。また、設定さえ行っていればWinSCPのサーバにデータを置いて自宅からそのサーバのデータを受け取る事も出来る。

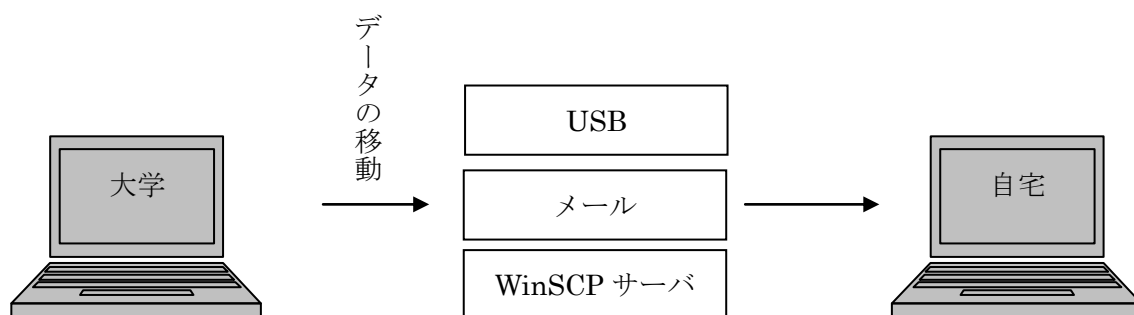


図2.1-1 データ移動の現状

2. 2 自宅から大学へ移動する場合

大学から自宅へ持って行く方法と同じ方法でデータのやり取りが出来る。ただし、メールの場合は大学のHPへ行き、メールシステムのページまで行かなければならない。しかし、この手間はメールシステムのページをショートカットに登録する事で解決する。

2. 3 問題点

現状の方法にはそれぞれにデメリットもある。まず、USBメモリ持っている事が前提である。忘れる、無くす、壊れる、と言った事が起きたら問題が起きる。また、PCにさして読み込みが終わって使えるようになるまでに意外と時間がかかる。メールは学校のメールサーバに送るにしても、自宅に送るにしても一度添付して自分宛に送らなければならない。これはUSBメモリと比べるとひと手間要る作業である。最後にWinSCPは、そもそもデータの送受信の為ではなくWebにデータをアップロードする為の物である為、データの移動操作が手間である。サーバからデスクトップに移動する際も送り先を設定して開かなければならない。

2. 4 まとめ

既存のサービスではそれぞれメリットとデメリットがある。それぞれデータの送受信と

いう目標は達成できるがUSBメモリを除くサービスはデータの送受信をメインとしたサービスではない為、データの送受信をメインにすれば改善点がある。また、USBメモリもデータの送受信をメインとしたサービスではあるが自宅、学校間に限ったものでは無い為、それらを踏まえて、より便利なサービスが検討出来ると考えられる。

3. 解決法の検討

本章では解決方法を検討し、その方法を実現する為に類似のサービスをサーベイしてどこを改善すれば快適なサービスになるかを考える。さらに、本研究の目標に必要な条件と機能について述べる

3. 1 解決へのアプローチ

今回のデータの送受信の問題において重視されるのは手間がかからない事である。手間がかかると後回しにして忘れてしまう為である。それぞれ手間だと思わせない工夫としてわかりやすいインターフェイスと操作性があり、送受信の実行から完了までにかかる時間が少なく快適なサービスでなければならない。また、自宅、学校間であればいつでも利用出来る必要がある。その為にサービスがお金や物(PCやネット環境は除く)が無い状態でも利用出来なければならない。そこで、類似のサービスをサーベイし、そのサービスのどこを改善すれば快適なサービスになるかを考える事で、検討するサービスの参考にする。

3. 2 類似サービスのサーベイ

ここでは、著名な類似サービスについて述べる

3. 2. 1 rsync, Cwrsync

共にファイル転送が出来るフリーソフトのプログラムである。rsyncはUnix専用、CwrsyncはCwrsync Server上でrsyncを稼働させる事でWindowsでもrsyncを使えるようにするプログラムである。1台をサーバとする事で同期行う。サーバや利用方法の設定はコマンドプロンプトで行う。Unix, Windows間のファイル転送も可能である。両方のファイルのリンクが一致することを必要とせず、リンクを横断して、ファイルの相違部分だけを転送する事が出来る。サービスを検討する際、互換性が重要になる事がわかる。また、欠点としては利用できるまでの操作がわかりやすすくない事が上げられる。サポートの充実と直感的な操作が出来る事が理想的である。7日間毎に自動で同期する為の設定ファイルを図3.2.1-1に示す。

```
#!/bin/sh
# This script does personal backups to a rsync backup server. You will end up
# with a 7 day rotating incremental backup. The incrementals will go
# into subdirectories named after the day of the week, and the current
# full backup goes into a directory called "current"
# tridge@linuxcare.com
```



```

# directory to backup
BDIR=/home/$USER

# excludes file - this contains a wildcard pattern per line of files to exclude
EXCLUDES=$HOME/cron/excludes

# the name of the backup machine
BSERVER=owl

# your password on the backup server
export RSYNC_PASSWORD=XXXXXX

#####

BACKUPDIR=`date +%A`
OPTS="--force --ignore-errors --delete-excluded --exclude-from=$EXCLUDES
      --delete --backup --backup-dir=/$BACKUPDIR -a"

export PATH=$PATH:/bin:/usr/bin:/usr/local/bin

# the following line clears the last weeks incremental directory
[ -d $HOME/emptydir ] || mkdir $HOME/emptydir
rsync --delete -a $HOME/emptydir/ $BSERVER::$USER/$BACKUPDIR/
rmdir $HOME/emptydir

# now the actual transfer
rsync $OPTS $BDIR $BSERVER::$USER/current

```

図3. 2. 1-1 7日毎に自動同期する設定ファイル

Rsyncの公式サイトより引用 <http://samba.anu.edu.au/rsync/examples.html>

3. 2. 2 Unison

ファイルの同期をするプログラムである。同期を取りたいパソコンのどちらかをサーバに設定してそこにアクセスする形で実行する。今回は通信方式にsocketを使ったものを試

してみた。サーバはsocketの他にhostとportの設定が必要になるが、アクセス側はhostとportを知ってさえいれば利用できる。システムは比較的シンプルでわかりやすい。接続先を設定した後は再度入力せずに使うことが出来る(図3. 2. 2-1)。また、コマンドプロンプトから同期をする事も出来る。Unisonの同期実行画面を図3. 2. 2-2に示す

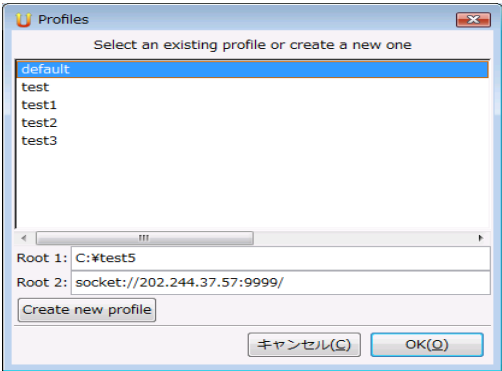


図3. 2. 2-1 アクセス先の登録

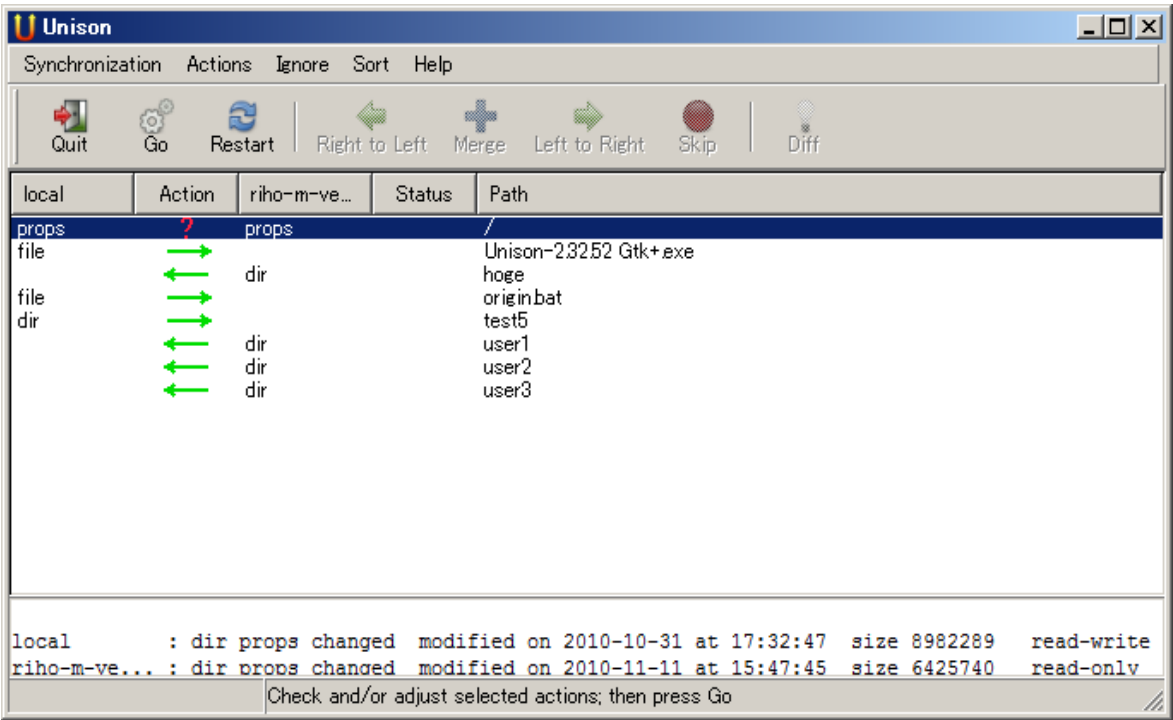


図3. 2. 2-2 Unison同期実行画面

3. 2. 3 Google document

Googleのやっているサービスの一つで、メールアドレス等の登録が必要になる。転送したいファイルをアップロードすれば、メールアドレスとパスワードさえ分かればどこからでもアップロードしたファイルを持ってくる事が出来る。ただし、基本的に文章やHTML転送用に作ってあるようで、jpgファイルや、phpファイルなどは形式を変えて送られてしまう為、利用出来ない。転送しようとする図3. 2. 3-1のようにphpファイルがHTMLなどに変更される。

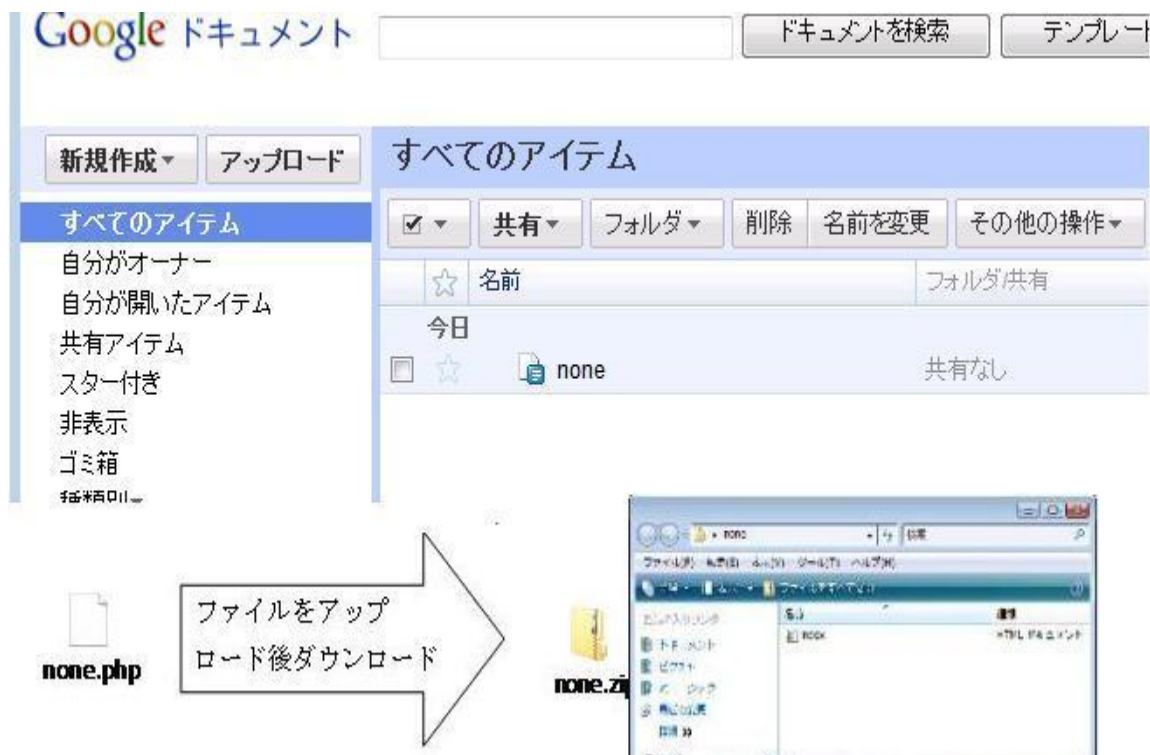


図3. 2. 3-1 phpファイルがアップロードで他の拡張子ファイルに変更される

3. 2. 4 Windows Home Server

Windowで行っている有料製品サービスである。10台までのクライアントPC間でのデータ共有が出来る。接続されたクライアントPCの自動バックアップを常にとっており、万が一トラブルが発生してもバックアップを使ってデータの復元が出来る。また、外部のリモート接続対応パソコンからインターネット経由でアクセスする事で、Windows Home Serverに接続されたクライアントPCを遠隔操作できる。大きな利点は自動バックアップである。PCでもバックアップは取る事も出来る。しかし、バックアップを取る間は仮想メモリを消費する。裏でバックアップ取られて作業は出来るが、動作が悪くなる。Windows Home Serverであればサーバの方でバックアップを取るなのでPCの動作には影響が出ない。その他の共有

機能に関してはフリーソフトで代用できるものが多く、有料である事も考えると、個人よりも企業やサービス対象であると言える。

3. 2. 5 Dropbox

2Gまでなら無料で使えるデータ共有サービスでLinuxやiPhoneにも対応している。利用するには事前にクライアントソフトのダウンロードとアカウント登録としてセットアップ時に名前、メールアドレス、パスワードを入力する必要がある。2台目以降も名前、メールアドレス、パスワードの確認がある。セットアップが完了するとマイドキュメントに「MyDropbox」と言うフォルダが作成され、そこからファイルが同期できる。フォルダにはPhotoとPublicがあるがPublicは広く公開されるフォルダなのでファイルを置かない事が推奨される。消去や上書きを行っても履歴が残るのですぐに元に戻す事が出来る。同期の際はアイコンが「同期中」になる(図3. 2. 5-1)などの工夫がある。別のサーバにファイルを置いている為(図3. 2. 5.-2)、同期したいPC両方がオンラインである必要が無い事も利点である。



図3. 2. 5-1右が「同期中」左が「同期完了」のアイコン



図3. 2. 5-2アップロードしたファイルはDropbox公式HPで確認出来る

3. 2. 6 ZumoDrive

Dropboxと無料での使用限度やダウンロードに必要なアカウント登録、対応機種はほぼ同じである。さらに、機能や操作方法をゲーム方式にする事でなじみやすくする工夫がされている(図3. 2. 6-1)。また、Dropboxが作られたフォルダにあるファイルを読み取るのに対し、ZumoDriveは貸し与えられたデータスペースに保存される(図3. 2. 6-2)。その為、ローカルPCのデータ領域に負荷をかけないですむ。これはオンラインストレージと呼ばれるサービスで、今までのGoogle document、Windows Home Serverにも言えるものである。詳細については3. 4節で述べる。



図3. 2. 6-1ゲーム方式の解説ページ



図3. 2. 6-2ローカルPCのデータ領域以外の領域が作成される

3. 3 必要条件と機能

それぞれのサービスを踏まえ、汎用性、サービスを使える段階に行くまでの手順の明確さ、直感的な操作の実現、事前登録情報の簡略化、利用可能ファイルの拡大、バックアップが取れる事が既存サービスと組み合わせさせた形が学校と家とで行われるデータのやり取りに最適なサービスであると思われる。既存のサービスではどの程度実現されているか表形式で分析してみる(図3. 3-1)。

まとめてみた中から本研究で検討するサービスと他を比較した強みを2つ上げる。一つは学校と自宅間で使う事を前提とした事で余計な機能を除いたシンプルな操作が実現出来る事である。シンプルな操作はサポートの充実と深く関連していると考えられた。それは独自のショートカットやサーバの役割をする物を自動作成もしくは自分で決められた箇所に作成する為である。特に、サーバの役割をする物は場所を移動するとプログラムがうまく動かなくなったりする他、わかりにくい場所に作成されたりする。3. 4節で述べるオンラインストレージであれば用意された物を使う為作成場所の変更が容易であるが、それでも手順を表記されずにデータ共有行うのは難しい。表記された情報と照らし合わせながら作業出来る事が望ましい。

2つ目は事前登録も利用箇所を絞る事でより少なく出来る点である。事前登録は共有を行い際にユーザを特定する為に必要不可欠である。しかし、学校と自宅間に特定すればもっと情報量を少なくしてサービス提供が行えると考えた。

図3. 3-1 サービス評価

サービス	汎用性	手順の 明確さ	シンプ ルな操 作	事前登 録情報 の簡略 化	利用可 能ファ イルの 量	バック アップ が取れ る
Rsync, Cwrsync	△	△	×	△	△	×
Unison	△	△	△	△	△	×
グーグルドキュメント	△	△	△	△	×	×
WindowsHomeServer	△	未検証	未検証	△	△	未検証
Dropbox	○	△	△	△	○	○
ZumoDrive	○	○	○	△	○	×

記号説明 ○：良い △：標準 ×：悪い

3. 4 オンラインストレージ

もう一つ、サービスの必要機能としてオンラインストレージが挙げられる。これは、ユーザに個別のパスワードを与えて事前に用意されたサーバの情報スペースを貸し与える

と言う物である。新サービスでは学生特有の学籍番号をパスワードの部分に割り当てる事でより利用に必要な登録を少なくする事を試みる。

3. 5 サービスのイメージとモデル

データの送受信が出来るプログラムをWebで配信し、それを学校と自宅のPCでダウンロードする。そのプログラムを使う事でデータのやり取りが可能となる(図3. 3-1)。つまりきそうな箇所はダウンロードページに手順を表記する。

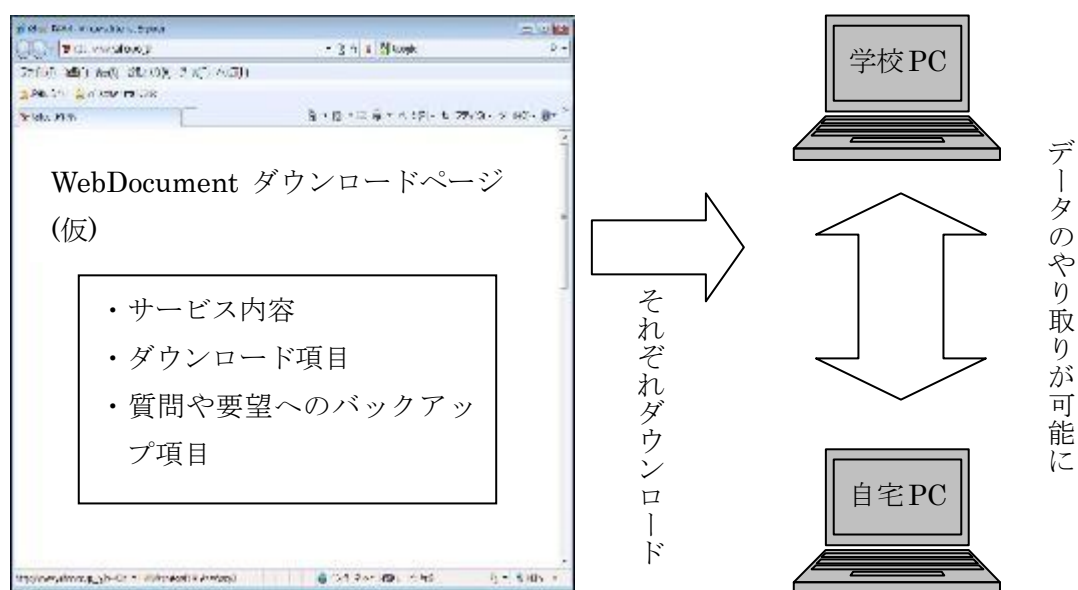


図3. 3-1 サービスイメージ

3. 6 複数の人数でのファイル共有利用

共有環境についてもう一つ、複数の人数で同じファイルを共有する物がある。これは、ファイルを相手に送信しなくても指定した保存場所を知っていれば同じファイルを一緒に編集出来る物である。また、「学生がグループでレポート文書を作成したり、教員が課題文書に赤字でコメントを加えられるなど、いろいろと応用範囲は広いと考えられる」¹。現在はグーグルドキュメントやdropboxで行われている。ファイルではなく、記事の編集で複数の人数で共有し編集出来るものとしてWikipediaが上げられる。Wikipediaからこの複数の人数で共有し編集し合える利点と問題点を説明する。利点は「リアルタイム更新である事」²である。これにより同じ時間に同じファイルを編集した時より良い内容にする事

¹ 木村昌史 大学教育系ネットワークにおける CGM 「情報科学研究」第 17 号 2007 84 ページ

² 樫原真知子、武宗次郎、遠藤友美江、土井亮平 wikipedia の評価 慶応義塾大学文学部図書館・情報学専攻上田修一研究会 2008 5 ページ

が出来る。「問題点として不確実性が挙げられる」³。「リアルタイムに更新が可能である為に、作成中のものや、個人の偏った考えといった十分に検討されていない」⁴部分が「反映されることがある」⁵。十分に検討されていない部分が次々に反映されると、本題からかけ離れてしまう可能性もある。しかし、自由に編集出来る事が発想の邪魔をさせない環境に繋がる為、制限をかけるのは複数で利用出来るメリットを半減させる事となる。編集や更新を円滑に複数人数で行う為には編集元のファイルと誰により、いつ編集されたかという履歴の情報が継承される事が必要になる。これにより、元の形が保持されつつ発想のままに編集を行う事が出来る。

3.7 複数の人数でのファイル共有環境の提案

前述の問題点はユーザ毎に編集する前のファイルと後のファイルを分ける事で解決する。その内容を図で説明する。左の(t)を時間軸として説明する。まず、元のファイルを違うユーザが編集した場合、順次フォルダを作っていく。今回はユーザaを元のファイルの作成者とした。ユーザaのファイルをユーザbが編集すると編集した名前は同じファイルが入ったb1というフォルダが作成される。(①)また、ユーザcが編集するとc1というフォルダが作成される(②)(図 3.6-1)。

³ 榎原真知子、武宗次郎、遠藤友美江、土井亮平 wikipedia の評価 慶応義塾大学文学部図書館・情報学専攻上田修一研究会 2008 7 ページ

⁴ 榎原真知子、武宗次郎、遠藤友美江、土井亮平 wikipedia の評価 慶応義塾大学文学部図書館・情報学専攻上田修一研究会 2008 7 ページ

⁵ 榎原真知子、武宗次郎、遠藤友美江、土井亮平 wikipedia の評価 慶応義塾大学文学部図書館・情報学専攻上田修一研究会 2008 7 ページ

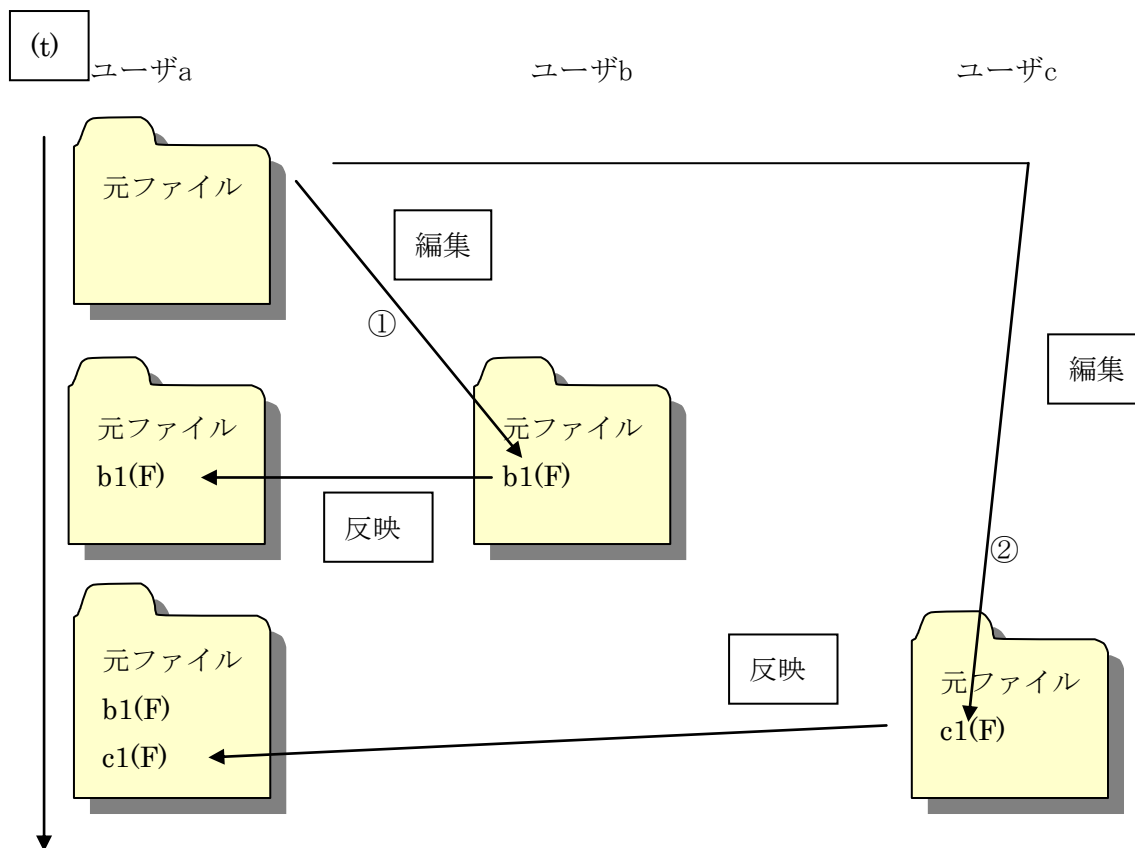


図 3.6-1 複数的人数でのファイル共有の形 1
記号説明 (F):フォルダ (t):時間

ユーザaが元ファイルを変更した場合a1と言うフォルダが作成される。(①)ユーザaが元ファイルに戻り再び編集するとa2というフォルダが作成される。(②)また、ユーザbがユーザaのa2ファイルを編集し(③)、そのファイルをユーザcが編集すると図のようになる。(④)

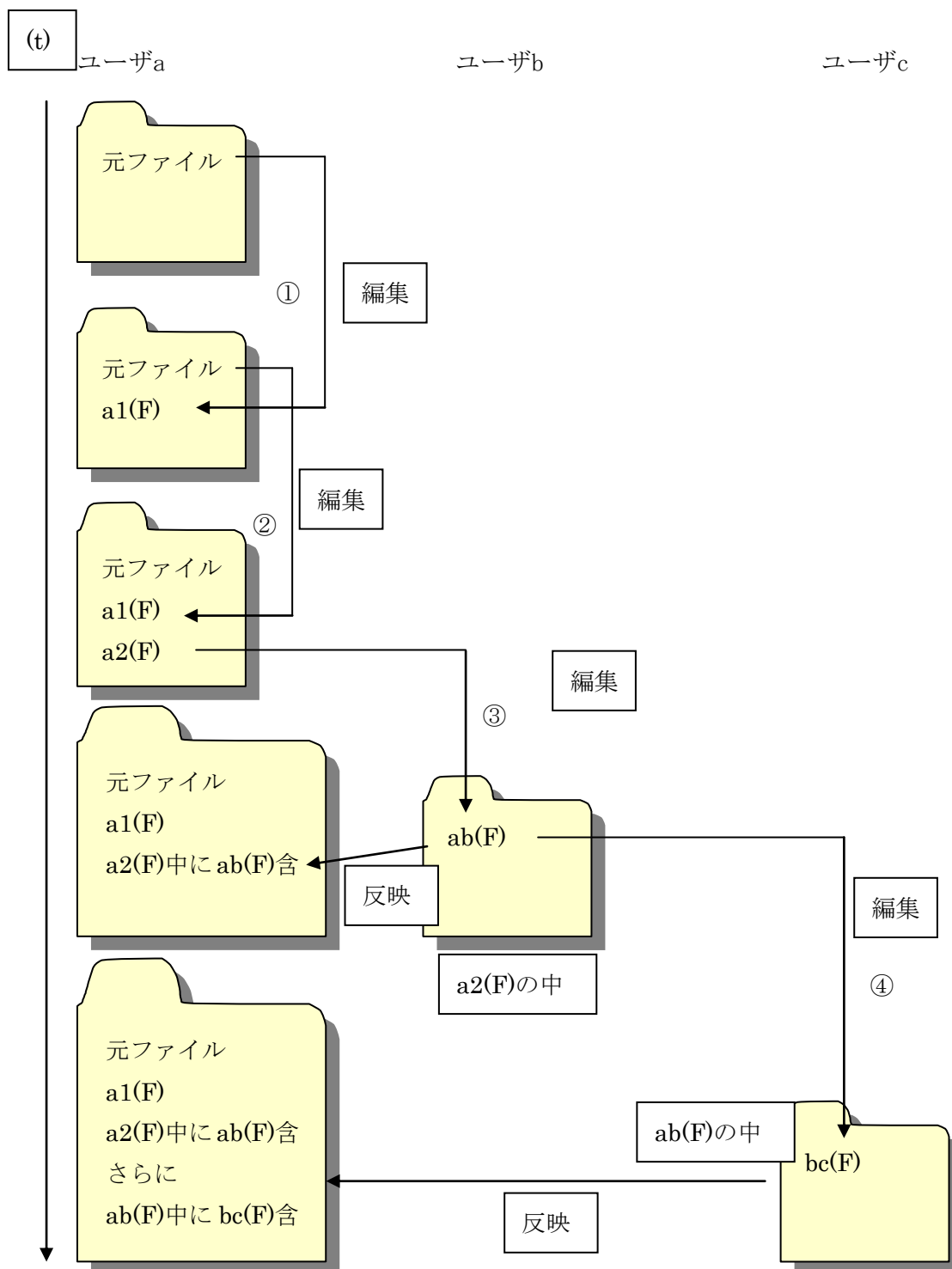


図3. 6-2 複数的人数でのファイル共有の形2

記号説明 (F) : フォルダ (t) : 時間

4. 設計と実装

4. 1 設計目標

誰もがわかりやすく利用出来るフォルダ共有プログラムとして、シンプルに操作出来る様にしなければならない。また、サービスを使うまでの設定が容易で、ファイルのやり取りに使う作業時間が少なくてすむ必要がある。本論文では既存のサービスであるUnisonを利用する事を前提とする。

3. 2 必要となる情報

必要となる情報は以下の情報である

- 学校の学籍番号とPassword
- Unisonを保存した場所
- 共有したいフォルダの入力
- 事前に作られたサーバのHOSTとPORT
- 共有するフォルダのデータが入ったバッチファイル付きのUnison

4. 3 システムの構成と要素

今回、Unisonを選んだのは、コマンドプロンプトが使える事で、簡略化という目標達成に適している為である。コマンドプロンプトのバッチファイルを利用する事でファイルのやり取りにかかる時間を少なく出来る。バッチファイルはコマンドプロンプトでの動作を簡略化し、クリックするだけで出来るようにする物である。簡略化はバッチファイルがファイルのやり取りをする際、上書きするのか、消去するのか、といった項目を聞かない為に実現する。

自宅、大学のどちらからも学籍番号とパスワードを入力する事でバッチファイル付きのUnisonがダウンロードできる。ユーザ毎にダウンロードを分ける事であらかじめサーバ側の共有フォルダの設定は済ませ入力情報を少なく出来る。ダウンロード後に使用PCの共有フォルダの設定とUnisonの保存場所の設定を済ませる事でデータのやり取りが可能になる(図4.3-1)。

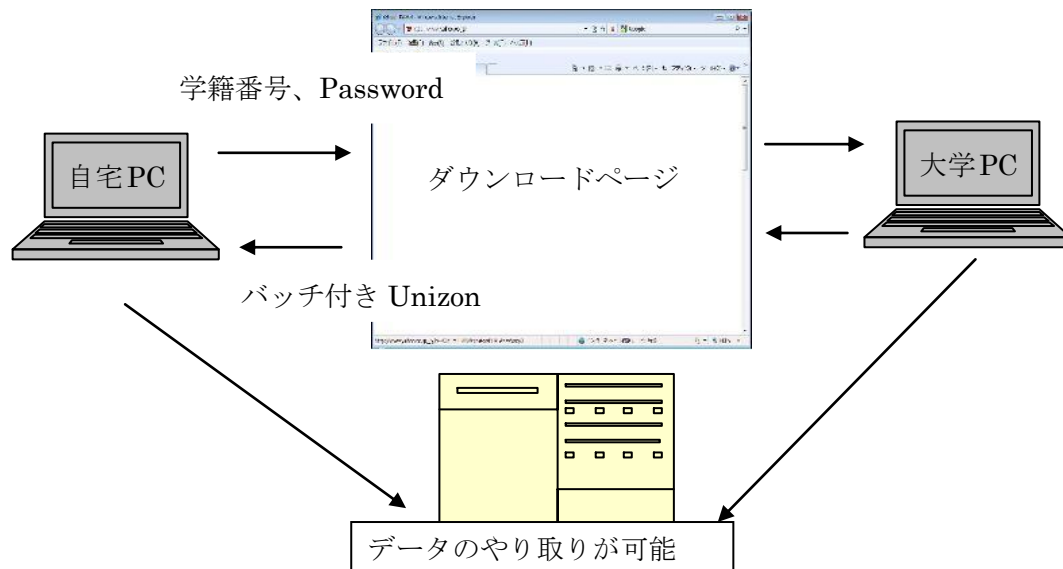


図4. 3-1 システム構成

4. 4 機能の設計

テスト用としてそれぞれユーザA、ユーザB、ユーザCの3ユーザで学籍番号、パスワードの代わりとなるものを割り当てる（図4. 4-1）。

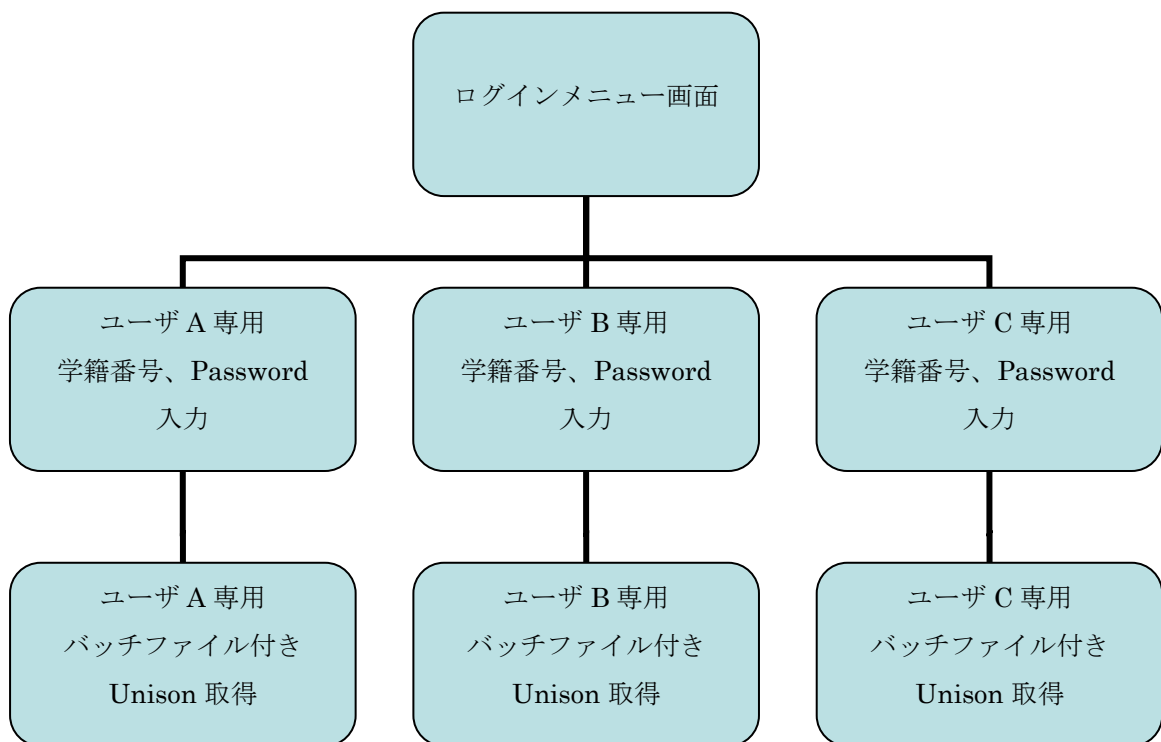


図4. 4-1各ユーザの必要ファイルダウンロード手順

サーバは事前にHOSTとPORTを設定したものを使用する。各ユーザがダウンロードサイトで情報を入力する事で個別の共有フォルダのデータが入ったバッチファイル付きの

Unisonが配布される。また、ユーザのデータはサーバにフォルダごとに管理される（図4.4-2）。



図4.4-2 ユーザデータの管理画面

機能はUnisonに依存したもので、サーバを元にしたフォルダの同期を取るものである。同期を取るにはバッチファイルをダブルクリック（もしくはクリック）する事で可能である（図4.4-3）。

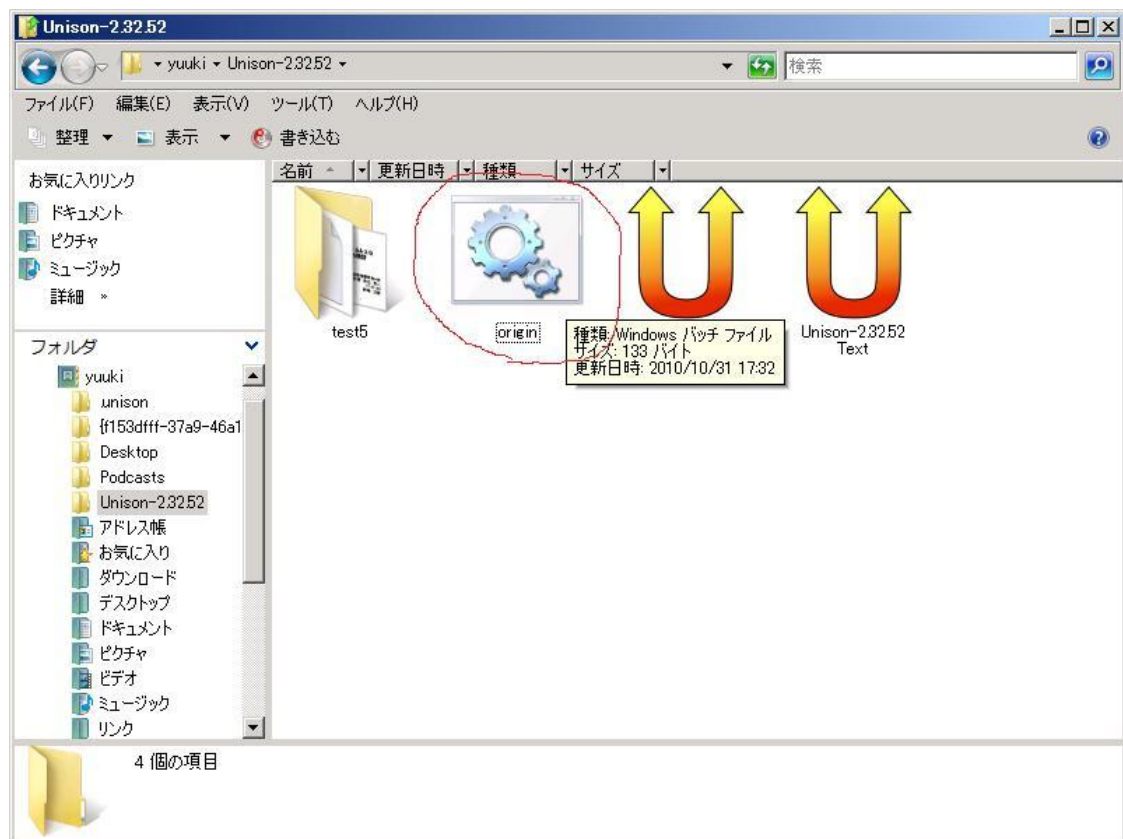


図4.4-3 バッチファイル

“Unisonの置いてある場所” 共有したいファイルの場所

socket://202.244.37.74:9999/user1 -batch

図4. 4-4 バッチファイルの中身(ユーザAの場合)

まず、学校や自宅で起動して同期を取るとサーバにあるデータを指定したフォルダに同期させ表示させる(図4. 4-4を参照)。

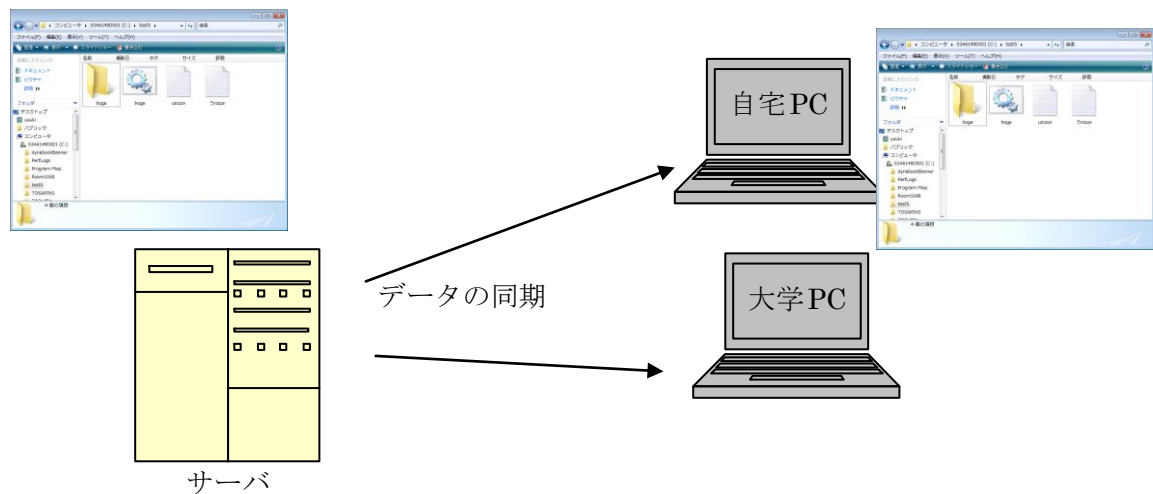


図 4. 4-4 データの指定

消去は一度同期を取った後、表示されたフォルダを消すことで可能である。また、新規にデータを同期させたい場合は指定したフォルダに同期したいデータを入れる事で同期できる(図4. 4-5を参照)。ただし、古いファイルは破棄されるので注意が必要である。また、データを最初に呼び出して同期する以外の新規作成、上書き、消去の3つの動作は複数のファイルで同時に実行が可能である。

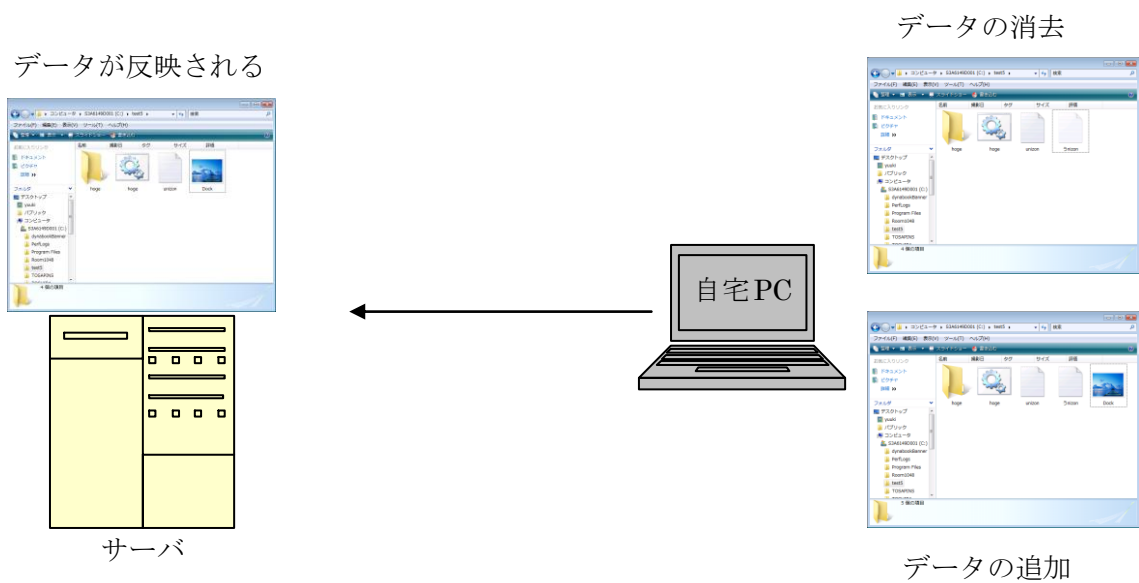


図 4. 4-5 データの消去と追加

例を挙げると、a, b, cのファイルがあったとする。大学で呼び出した後にaを消去し、bを修正する。自宅に帰って共有で呼び出すと修正されたbとcが呼び出される、といった形になる。(図4. 4-6)

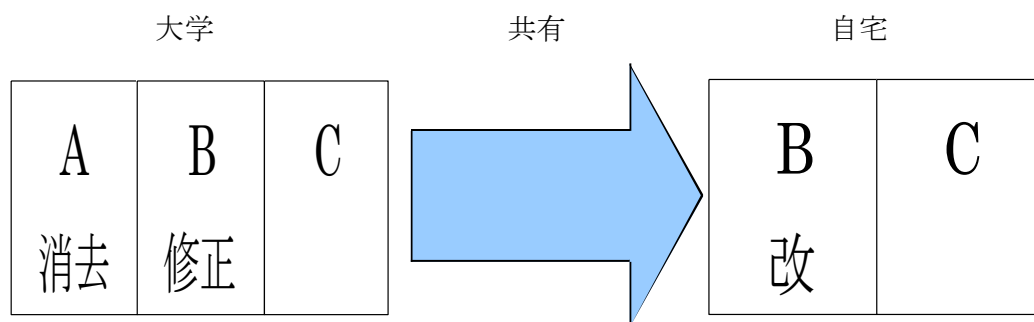


図4. 4-6 データの追加と上書き、消去例

4. 5 機能の実装

本研究では大学と自宅間でのシームレスなファイル共有環境の実現が目標である為、事前入力情報の簡略化、ファイルのやり取りに使う作業時間の減少する環境を実装する事とした。ダウンロードサイトで知っている情報を入力させる事で入力情報を減らし、バッチファイルでやり取りの作業を減少させる。

4. 5. 1 インターネットでのダウンロード

ダウンロードはtest.php、pass○.php、download○.php、dl○.phpで構成される。(○は対応番号)test.php、pass.phpは学籍番号とパスワードを聞くもので、POSTで入力された値が一致した時に次の作業へのリンクを表示する。

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0 Transitional//EN">

<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
<title>TEST</title>
</head>

<body bgcolor="#ccffff">
<br>
※ここは、既存プログラムを改良し、CUCのマイドキュメントと自宅の<br>
指定したフォルダを共有し、ファイルのやり取りを行えるようにする<br>
プログラムを提供するサイトです。<br>
・使用プログラムの:<a href="http://www.cis.upenn.edu/~bcpierce/unison/">本家様
サイト</a><br><br>
学籍番号を入力して下さい<br><br>
<?php
    if(isset($_POST["start"])){
        $stu = $_POST["stu"];

        $stid_data[user1] ="こちらでパスワードの認証をお願いします
http://202.244.37.57/~a740159/pass1.php";
        $stid_data[user2] ="こちらでパスワードの認証をお願いします
http://202.244.37.57/~a740159/pass2.php
";
        $stid_data[user3] ="こちらでパスワードの認証をお願いします
http://202.244.37.57/~a740159/pass3.php
";

        $result = ($stu);

        $message = $stid_data[$result];
    }else{
        $message ="";
```

```

}
?>
<html>
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
    <body>
        <? print $message;?>
        <br>
        <form method="post">
            <input type="text" name="stu" value=""><br/>
            <br>
            <input type="submit" name="start" value="確認">
        </form>

        @使い方<br>

        質問or要望などございましたら、お手数ですが<br>
        a740159@cuc.ac.jpまでお願いします。

    </body>
</html>

```

図4.5.1-1 test.php

```

<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0 Transitional//EN">
<?php
    if(isset($_POST["start"])){
        $pass = $_POST["pass"];

        $rog_data[a1] ="こちらでダウンロード願います
http://202.244.37.57/~a740159/download1.php";

        $result = ($pass);

        $message =$rog_data[$result];
    }else{
        $message ="";
    }
?>

```

```

<html>
    <title>パスワード確認画面</title>
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
    <body bgcolor="#ccffff">
        パスワードを入力してください<br><br>
        <? print $message;?>
        <br>
        <form method="post">
            <input type="text" name="pass" value=""><br/>
            <br>
            <input type="submit" name="start" value="確認">
        </form>
    </body>
</html>

```

図4.5.1-2 Pass1.php

download.phpでダウンロード画面を表示させている。dl1.phpでzipにしたプログラムファイルのpathを通してダウンロード出来る様にしている

```

<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0 Transitional//EN">

<html>
<head>
    <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
    <title>ダウンロードページ</title>
</head>

<body bgcolor="#ccffff">
    こちらでダウンロードお願いします
    <form action="dl1.php" method="post"><input type="submit" value="ダウンロード"></form>

</body>
</html>

```

図4.5.1-3 download1.php

```

<?php

```

```
$file_path = "user1.zip";  
$filesize = filesize($file_path);  
  
readfile($file_path);  
?>
```

图4.5.1-4 dl1.php

5. 評価と考察

5. 1 評価

機能の動作確認

初めにダウンロード出来るか試してみる。<http://202.244.37.57/~a740159/test.php>へアクセスすると以下のような画面が表示される。(図5. 1-1)

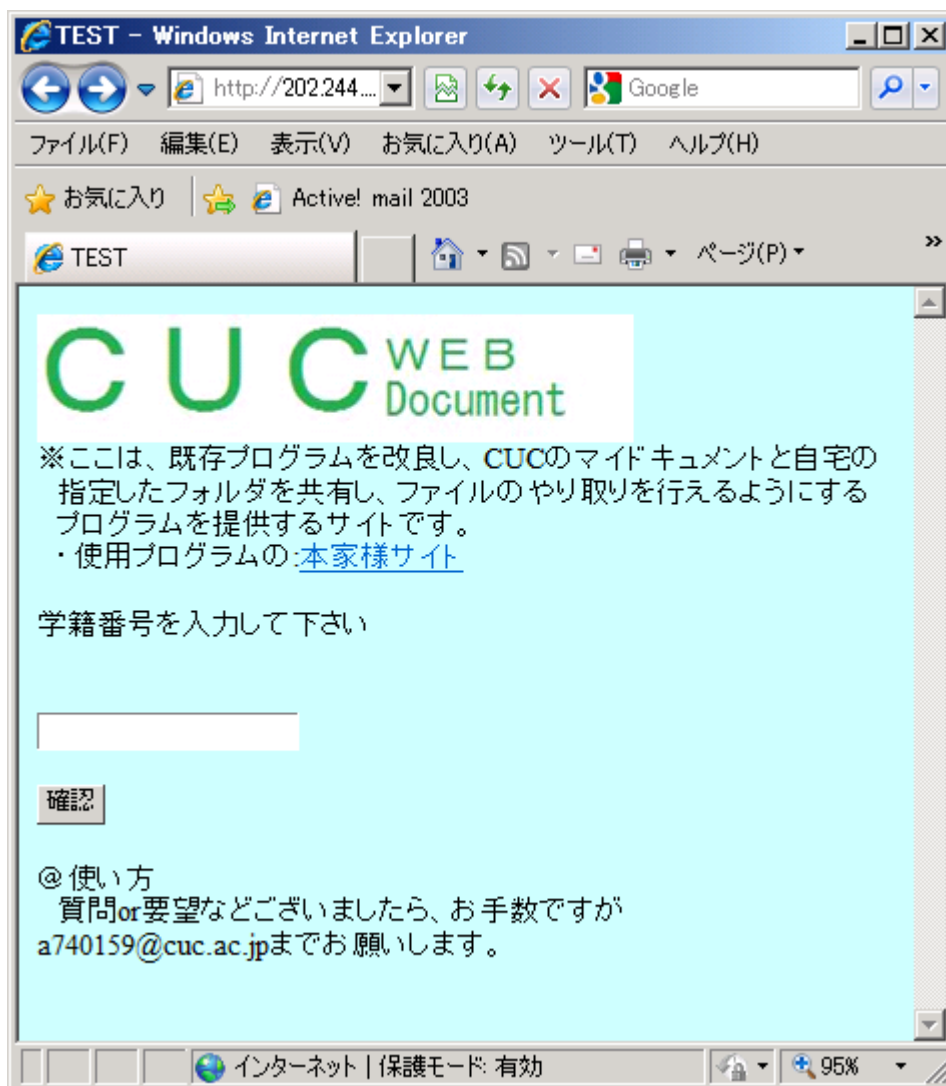


図5. 5-1 学籍番号確認画面

学籍番号を入力し、手順を踏むと次のような画面が表示される(図5. 5-2)

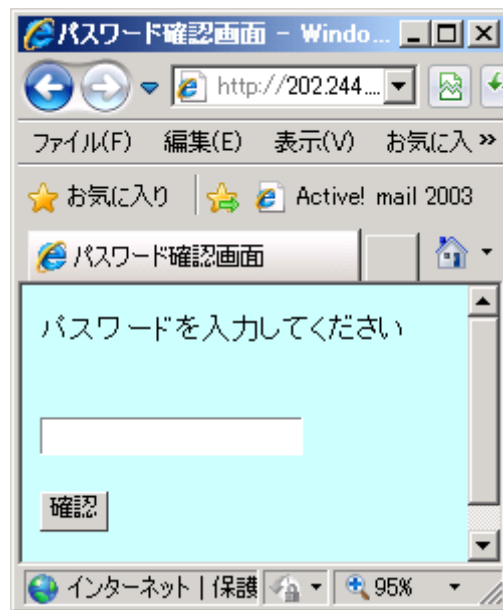


図5. 5-2 パスワード確認画面

パスワードを入力して、手順を踏むと以下のような画面が表示され、ファイルのダウンロードが出来る(図5. 5-3)。ファイルを設定後にファイル共有する事が出来た。

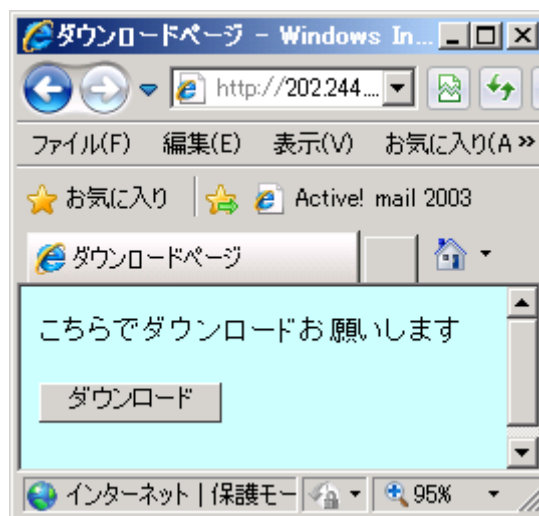


図5. 5-3ダウンロード画面

5. 2 考察

本研究の目的は大学と自宅間でのシームレスなファイル共有環境の実現である。目的完了とまではいかなかったが、事前入力情報の簡略化、ファイルのやり取りに使う作業時間の減少の基盤になる形は実装出来た。ダウンロードからサービス利用までには慣れていれば1分程度で可能である。また、1回の作業時間はCUCWebMailの場合インターネットを開いていない状態からお気に入りのショートカットを経由して添付の設定を含めると9秒程度、WinSCPでデスクトップにショートカットを設定して行った場合3秒程度、今回のシス

テムはデスクトップに設定して行った場合2秒程度と、作業時間の減少は実現している。ここまでは既存サービスでも実現されているが、Unisonで実現している事がポイントとなる。それは、他のサービスがクラウド化して機能の追加が製作者任せなのに対しUnisonはコマンドプロンプトやプログラムを組む事でカスタマイズ出来る事である。カスタマイズの具体的な内容は次項で述べる。

5. 3 その他の機能の検討

以下の機能を実装する事でよりシームレスなファイル共有環境になる

●複数で利用できる環境を実現

(複数の人数で利用する事の利点と問題点については3. 6、3. 7を参照)個人ユーザの設定とは別に複数で利用出来るフォルダの設定をバッチファイルに入れる事で実装出来る。新しい複数の人数で利用できるファイル共有環境は同期を行う毎に決められた名前のフォルダが作成される様に設定する事で実装出来る。

●Unisonの日本語ファイルを実行出来る環境の実現

Unisonでは日本語ファイルが利用出来ない。コマンドプロンプトで一時的にファイルを違う文字列に変換して送り、送り先で元に戻す様に設定する事で実装出来る。

●自動更新とバックアップ

自動更新にする事でフォルダに放り込むだけで良くなり、よりシームレスな環境となる。バックアップを常に取り取る事でファイルの誤った消去や上書きにも対応出来る。実装はタスクスケジューラを利用する。同期実行のバッチファイルを定期的に行うように設定すれば自動更新となる。さらに更新や新しくファイルを作ると他のフォルダにも保存される様に設定すればバックアップを実装出来る。

6. まとめと今後の課題

6. 1 まとめ

本研究では大学と自宅間でのシームレスなファイル共有環境の実現を目指した。まず、大学と自宅間のデータのやり取りの現状を把握し、目的を整理した。次に解決方法を検討する為に既存のサービス (rsync, Cwrsync, Unison, Google document, Windows Home Server, Dropbox, ZumoDrive) をサーベイし、検討するサービスと他のサービスを比較して強みと言える機能を、シンプルな操作である事と、事前登録情報を簡略化する事の2つであるとした。それらを踏まえてサービスのイメージとモデルを考え、システムの検討と設計を行った。また、その他のファイル共有環境として、複数人数でファイルの共有を挙げ、メリットとデメリットを述べた。そして、デメリットの解決方法としてファイル共有環境の形を提案した。システム製作では必要な情報をまとめ、Unisonを利用して実装を目指した。事前入力情報の簡略化を行うダウンロードサイトと、ファイルのやり取りに使う作業時間の減少させる為のバッチファイルを作成した。機能の動作を確認し、バッチファイルで共有するフォルダとUnisonの保存場所を設定する事ですぐに共有環境を利用出来る様になった事を確認した。最後にダウンロードまでにかかる時間とサービス利用までにかかる設定の時間を計測し、目標をどの程度達成出来たか評価した。

6. 2 今後の課題

より改善が必要である。事前入力情報の簡略化についてはダウンロードサイトの学籍番号とパスワードの入力画面を一つにまとめられる。まとめる事でダウンロードまでのサイトの移動及び手間が少なくなる。共有するフォルダやUnisonの場所の設定をわかりやすくする必要がある。それぞれの設定項目をフォームで取り上げて、選択するだけで設定出来る様にする。これにより、設定の難しさを解消し、共有までにかかる時間を減らす事が出来る。また、5. 3で挙げたその他の機能として、複数で利用出来る環境を実現、Unisonの日本語ファイルを実行出来る環境の実現、自動更新とバックアップを実装すると、よりシームレスな環境を作る事が出来る。

参考資料

[1]グーグルドキュメント

<https://docs.google.com/>

[2]rsync ウェブページ

<http://www.infoscience.co.jp/technical/rsync/index.html>

[3]Unison ウェブページ

<http://www.cis.upenn.edu/~bcpierce/unison/>

[4]WindowsHomeServer ホームページ

<http://www.microsoft.com/japan/windows/products/winfamily/windowshomeserver/default.aspx>

[5]Dropbox のセットアップと使い方

<http://bizmakoto.jp/bizid/articles/0809/12/news103.html>

[6]ZumoDrive ウェブページ

<https://www.zumodrive.com/>

[7]第 22 回 ZumoDrive か Dropbox か。選択肢増えたクラウドストレージ

<http://bizmakoto.jp/bizid/articles/0902/08/news006.html>

[8]OCN インターネット用語辞書「オンラインストレージ」

<http://www.ocn.ne.jp/support/words/a-line/83I8393838983C839383X83g838C815B83W83T815B83r83X.html>

[9] 木村昌史 大学教育系ネットワークにおける CGM 「情報科学研究」第 17 号 2007 84 ページ

http://bus.nihon-u.ac.jp/laboratory/pdf/17_08.pdf

[10] 檜原真知子、武宗次郎、遠藤友美江、土井亮平 wikipedia の評価 慶応義塾大学文学部図書館・情報学専攻上田修一研究会 2008 5,7 ページ

<http://www.slis.keio.ac.jp/~ueda/semi/2007Wikipedia.pdf>

[11]西沢直木 「PHP による Web アプリケーションスーパーサンプル 第 2 版」2006 ソフトバンク クリエイティブ株式会社

[12]西村祐貴 「SMTP を利用したファイル共有システムに関する研究」 慶応義塾大学大学院政策・メディア研究科 2002

<http://www.sfc.wide.ad.jp/thesis/2002/master/yuuki/yuuki.pdf>

謝辞

本研究を行うに当たり、ご指導を頂いた渡辺 恭人准教授に感謝いたします。論文の構成・参考語句及びサービス・プログラミング・執筆に至るまで、親身にご指導いただき、感謝いたします。佐藤 孝祐氏にはプログラミング・参考文献・細かい疑問に至るまで答えていただき、感謝いたします。親身にご指導いただいたおかげで実力以上の卒業論文を執筆する事が出来ました。渡辺ゼミでは、プログラミングやデータベース等未経験からのスタートになりましたが、的確な指導をいただき、世界観及び作成の幅を広げる事が出来ました。先輩方の論文や発表を参考にさせていただき興味を持つテーマを思いつき、本研究のきっかけとなりました。ここまでの過程に様々な疑問や問題がありましたが、その都度、渡辺 恭人准教授、佐藤 孝祐氏に助言していただき、ここまで研究を進める事が出来ました。最後に、私の論文に関わって下さった方全員にもう一度感謝を述べさせて頂き謝辞とさせていただきます。ありがとうございました。