

修士論文 2002 年度（平成 14 年度）
SMTP を利用した
ファイル共有システムに関する研究

慶應義塾大学大学院政策・メディア研究科
西村 祐貴

平成 15 年 1 月 14 日

SMTP を利用したファイル共有システムに関する研究

本研究では、SMTP を利用したファイル共有システムの設計と実装を行う。

今日、ネットワークを介したさまざまな共同作業において、グループウェア等のファイル共有システムを用いたファイル共有が行われている。共同作業を行う期間は、グループの目的や作業内容等に応じて多様であるが、作業期間が1ヶ月から数ヶ月程度の比較的短い共同作業は、単年度予算プロジェクトにおける報告書執筆や大学における論文指導など多くのケースで一般的である。また近年は、携帯端末やホットスポット等のモバイル利用の普及により、移動中の空き時間を利用した作業の機会が増えており、非オンライン環境で作業するユーザーを想定する必要がある。

このようなグループの特徴は、ファイル共有機能を持つ多くのシステムが想定しているグループ像と異なっている。例えば、ファイル共有機能を持つ多くのグループウェアは、ある程度永続的に存在する組織内において時間をかけて情報(文書)を蓄積し、それを参照することに主眼を置いている。これらのグループでは、膨大な情報を扱うため、高性能・大容量のサーバーを設置し、情報を蓄積する。しかし短期的な共同作業では、情報の蓄積よりも最新の情報を交換することに主眼があり、ファイル共有の場を設けるコストが低いことが優先される。

そこで本研究では、通信に SMTP/POP3 を利用したファイル共有システム Mikan の設計と実装を行った。Mikan では既存のメールサーバーを介してファイルのやり取りを行うため、ユーザーが新たにサーバーを設置・運用する必要がない。更新ファイルを一時的にサーバー上にキャッシュできるため、非オンライン環境で作業しているユーザーが更新時にオンラインになった際に最新版を共有できる。また、履歴の参照や、競合発生の検知、解決等のバージョン管理が可能である。

本研究では評価として、関連研究との比較による定性的評価と、慶應義塾大学大学院政策・メディア研究科特設科目「インターネット時代のセキュリティ管理」において運用実験を行った。その結果、Mikan が必要要件の多くを満たすことを示した。

キーワード：

1:SMTP 2:ファイル共有 3:短期的な共同作業グループ 4:バージョン管理

慶應義塾大学大学院政策・メディア研究科

西村 祐貴

A research of a File Sharing System using SMTP

This research designs and implements a SMTP le sharing system aimed on short-term collaboration within a group.

A short-term collaboration group is a small group that works together for a short period of time formed by specic members. For example, a group writing reports or papers will be formed only for a short working period of a couple of months. Each author may have their own writing environment and working time. An author may even be oine from other members. The group of authors will need to read each other's materials and give comments. In order to transfer le to each other, there must be a method to exchange le between the members.

A characteristic of this kind of a small group would be out of focus for other le sharing systems that are presently available. For example, many groupware software that has a le-sharing capability focuses on preserving many information (les) over a certain amount of time. However, the group in which this research focuses need to form and part frequently, therefore the low cost for the formation of the le sharing group has a high priority.

Mikan, the implementation of this research, uses SMTP/POP3 as the communicating medium and implemented a record keeping structure and contention process alongside with the le sharing mechanism. Because the system uses an existing mail server, the user does not need to manage a server, resulting in low cost. Also, user working oine may obtain the new le when the user renews the le. By managing the version number of the le, contention and version mismatch may be avoided. In this research, Mikan is compared and evaluated qualitatively with other related researches. Also, Mikan was operated in a lecture called "Security Management in the decade of Internet" in Keio University, Graduated School of Media and Governance, showing Mikan ectively supports collaboration works .

Keywords :

1:SMTP 2:File Sharing 3:A Short-Term Collaboration Group 4:Revision Control

Keio University, Graduate School of Media and Governance

Yuuki Nishimura

目次

第1章	はじめに	1
1.1	ネットワークを介した共同作業の発展	1
1.2	共同作業におけるファイル共有システム	2
1.3	本研究の目的・意義	2
1.4	本論文の構成	3
第2章	既存のファイル共有システムの分析	5
2.1	ファイル共有システムの分類	5
2.2	電子メール型	6
2.2.1	メーリングリスト (ML)	6
2.2.2	ESP (Eudora Sharing Protocol)	7
2.2.3	VCoCoA	7
2.3	共有フォルダ/ファイル型	7
2.3.1	Lotus Notes/Domino	8
2.3.2	Concurrent Version System(CVS)	8
2.3.3	オフィスアクセラレータ	9
2.4	P2P グループウェア型	10
2.4.1	Groove	10
2.4.2	Net Peers	11
2.5	既存のファイル共有システムの特徴と問題点	11
第3章	短期的な共同作業を対象としたファイル共有システムの検討	13
3.1	本研究が想定する場面とその分析	13
3.1.1	場面例 1: 報告書や論文の共同執筆	13
3.1.2	場面例 2: 発表資料の共同作成	14
3.2	システムモデルの検討	15
3.2.1	通信モデル	15
3.2.2	ユーザーの識別	16
3.2.3	グループの形成	16
3.3	ファイル共有機能の検討	17
3.3.1	ファイルの作成・更新	17
3.3.2	ファイルの同期	17
3.3.3	競合検知と解決	19
3.3.4	更新履歴の管理	19
3.4	システムの要件定義	19

3.4.1	バージョン管理	20
3.4.2	設置・運用コスト	20
3.4.3	非オンライン環境への対応	21
第 4 章	SMTP を利用したファイル共有システムの設計	22
4.1	システムの概要	22
4.2	作業に沿った動作	22
4.2.1	アカウントの作成	22
4.2.2	プロジェクトの形成	24
4.2.3	ファイルの作成・更新	26
4.2.4	ファイルの同期	27
4.2.5	競合検知と解決	29
4.2.6	更新履歴の管理	31
第 5 章	ファイル共有システム”Mikan”の実装	32
5.1	実装概要	32
5.1.1	実装環境	32
5.1.2	システム構造	32
5.2	システムの機能	33
5.2.1	アカウントの設定	33
5.2.2	プロジェクトの作成	34
5.2.3	ファイルの作成・更新	34
5.2.4	ファイルの同期	35
5.2.5	競合検知と解決	36
5.2.6	更新履歴の管理	36
第 6 章	ファイル共有システムの評価	37
6.1	定性的評価	37
6.2	運用実験	38
6.2.1	実験概要	38
6.2.2	実験結果	39
6.3	評価と考察	44
6.3.1	バージョン管理	44
6.3.2	設置・運用コスト	44
6.3.3	非オンライン環境への対応	45
第 7 章	おわりに	46
7.1	本研究のまとめ	46
7.2	今後の課題と展望	47
7.2.1	想定対象の拡張	47
7.2.2	各機能の拡張	47
謝辞		48

参考文献	50
付録	53
付録 A: 実験用ソフトウェア利用マニュアル	53
付録 B: アンケート選択式集計一覧	62
付録 B: アンケート自由記述回答結果一覧	67
付録 C: アンケート記入紙	70

目次

2.1	Eudora Sharing Protocol(出典は文献 [32])	7
2.2	CoCoA-Editor(出典は文献 [15])	8
2.3	CoCoA と VCoCoA の添削文章の流れ (出典は文献 [15])	8
2.4	CVS を用いたルーチンワーク (出典は文献 [36])	9
2.5	Groove のファイル共有機能	10
2.6	NetPeers: 流動型ファイル共有システム (出典は文献 [38])	11
3.1	職業編集者型モデルとレビュー型モデル	14
3.2	Microsoft PowerPoint 2000: スライド一覧機能	15
3.3	電子メール型におけるファイルロック機構	18
4.1	システム利用の全体像	23
4.2	ユーザー名の表示	24
4.3	X-Mailer ヘッダ	24
4.4	invite メッセージ送信時の処理	25
4.5	invite メッセージ受信時の処理	26
4.6	ファイル情報の表示	27
4.7	ファイル受信時の処理 (全体の流れ)	28
4.8	ファイル送信時の処理 (全体の流れ)	29
4.9	競合発生時の保存ファイル名	30
4.10	版ファイルの保存	31
5.1	Mikan8.0 Lite	33
5.2	アカウント設定画面	34
5.3	プロジェクト設定画面	34
5.4	プロジェクトの切り替え	34
5.5	プロジェクト参加の確認	35
5.6	ファイルの操作と作成・更新	35
5.7	同期が必要なファイルを提示	35
5.8	同期で他メンバーから受け取ったファイルを提示	35
5.9	競合発生	36
5.10	更新履歴の確認	36
6.1	ReadMe.txt ファイルによる簡単なチュートリアル	40
6.2	一つのファイルに対して行われた項目の洗い出し (一部)	41

表 目 次

1.1	本研究が想定するグループの特徴	3
2.1	ファイル共有システムの分類と主なシステム例	6
2.2	既存のファイル共有システムの特徴と問題点	12
2.3	独自メールクライアント型の特徴	12
3.1	バージョン管理に関するシステム要件	20
3.2	設置・運用コストに関するシステム要件	20
3.3	非オンライン環境への対応	21
4.1	アカウント設定項目	23
4.2	プロジェクト設定項目	24
4.3	ファイル管理情報	26
4.4	ファイルの状態表示	27
5.1	本システムの実装環境	32
5.2	本システムの構造	33
6.1	定性的評価：全体との比較	37
6.2	定性的評価：独自メールクライアント型との比較	37
6.3	項目ごとのファイル作成	42

第 1 章

はじめに

本章ではまず、ネットワークを介した共同作業の広がりや共同作業におけるファイル共有の重要性を示し、本研究の目的である SMTP を利用したファイル共有システムを実現することの意義について述べる。次にシステムが対象とするグループの特徴について述べる。

1.1 ネットワークを介した共同作業の発展

今日、人々が電子メールや掲示板、インスタントメッセージ等のインターネット上のアプリケーションを用いて、コミュニケーションを取ることが一般的になった。総務省の平成 13 年度ブロードバンド利用動向調査 [1] によると、パソコンからインターネットを利用している人の利用用途で、その利用率が最も高いのは「電子メールの受送信」で 88.3% であり、「チャットや掲示板の利用」は 34.0% に達している。

ネットワークを介したコミュニケーション手段の獲得は、離れた場所に存在する複数の利用者がネットワークを介してお互いに連絡を取りながら作業をすることを容易にした。企業はオフィス文章や会議の議事録などのネットワーク上での共有によるビジネスの効率化を推進し、大学では研究報告書の共同文書作成が行われている。近年広く普及しているオープンソフトウェア OS の Linux [2] は、世界中の多数の優秀な技術者による大規模な共同作業の成果と言える。

これらの共同作業を支援しているシステムは多種あるが、その基本となっているのは、ユーザー同士が連絡を取り合うためのコミュニケーション機能と、情報や成果物を共有する機能、グループのタスクを管理する機能である。

現在、多くの企業で用いられている、グループウェアの代表的製品に Lotus Notes / Domino [3] があげられる。Lotus Notes は、電子メール、スケジュールリング、文書ライブラリー等の機能を総合的に提供している。IBM によると、Lotus Notes / Domino の 2001 年 11 月現在の発行済みライセンス数は、世界規模で 8,000 万以上、日本でも 1000 万以上に達しており [4]、多くの企業がオフィス文章等のファイル共有を図っている。

オープンソフトウェア開発を支援するサイトである SourceForge [5] は、開発者のために、メーリングリスト、掲示板・フォーラム、タスク管理システム、ファイル保管、バックアップ等の機能を Web ベースで総合的に利用できる環境を無料で提供している。2002 年 12 月現在、約 53,000 のプロジェクトと約 533,000 人のユーザーがこれに参加している。SourceForge

のシステム内部には、広くソフトウェア開発で利用されている CVS(後述) が利用されており、ソフトウェアソースプログラム等の成果物を扱うファイル共有システムとしての役割を果たしている。

1.2 共同作業におけるファイル共有システム

上述したようにネットワークを介した共同作業において、ファイルの共有は重要な位置を占めており、さまざまな用途やニーズを対象としたファイル共有機能を持つシステムが登場し発展を遂げている。

例えば、グループウェアの分野では近年、Lotus Notes/Domino[3]、サイボウズ [6] といった、従来のクライアントサーバー型のグループウェアに対して、Groove[7]、アリエルエアワン [8] 等の P2P 型グループウェアと呼ばれるシステムが登場した。ユーザーがサーバーを設置する必要がなくなり、これまでサーバーの導入や運用で断念していたユーザーや小規模のグループが容易に利用できる環境を提供した。Groove はファイル共有機能は存在し、登録したファイルをすべて複製 (同期) する機能を持っている [9]。

ソフトウェア開発の分野では、ファイルの履歴等を管理するためのバージョン管理システムが進化を遂げてきた。RCS(Revision Control System)[10]、SCCS(Source Code Control System)[11] はファイル単位のロック機構を持っており、これに類似した機構は現在でも、Microsoft Word[12] 等の多くの商用ワードプロセッサで用いられている。RCS/SCCS から発展した CVS(Concurrent Versions System)[13] は、ファイルロックによる排他処理の非効率性の問題を解決するため、リポジトリ (後述) という概念を持ち、多数のファイルをまとめて管理することが可能となった。複数のユーザーによる共同作業をサポートし、グループの成果物を扱うファイル共有システムとしての役割も担っている。CVS は Web サイトのコンテンツ管理や、一部では共同文書作成等にも用いられている。

1.3 本研究の目的・意義

本研究では、短期的な共同作業を行なうグループを対象としたファイル共有システムの設計と実装を行う。本研究では、次のような特徴を持つグループを想定している。

1. ユーザー間でファイル共有を行う必要がある共同作業を行っている
2. 作業期間が短く、特定のメンバーで構成された比較的小規模なグループ
3. 非オンライン環境で作業を行なうメンバーがいる

1.1, 1.2 で述べたように、ネットワークを介したさまざまな共同作業において、グループウェア等のファイル共有システムを用いた、ファイル共有が行われている。

作業期間が1ヶ月～数ヶ月程度の比較的短い共同作業は、単年度予算プロジェクトにおける報告書執筆や、大学における論文指導など多くのケースで一般的である。このような共同作業では、短期的にグループを形成して行われる。

また近年は、ノート PC 等の持ち運びが可能な携帯端末、ホットスポット等のネットワーク接続環境が普及し、組織や大学、自宅に限らず、移動中の空き時間を使った作業の機会が

表 1.1: 本研究が想定するグループの特徴

	本研究	永続的なグループ・組織	ソフトウェア開発
期間	短期間	長期間	短期間～長期間
規模	小規模	小規模～大規模	小規模～大規模
優先事項	場を設けるコスト	膨大なデータ処理	データの整合性

増大しており、オフライン環境で作業し、更新時のみオンラインになるユーザーを想定する必要がある。

このようなグループの特徴は、ファイル共有機能を持つ多くのシステムが想定しているグループ像と異なっている(表 1.1)。

例えば、ロータスノートを始めとする多くのグループウェアは、ある程度永続的に存在する組織内において、時間をかけて情報(文書)を蓄積しそれを参照することに主眼を置いている。これらのグループでは、膨大な情報を扱うため、高性能・大容量のサーバーを設置し、情報を蓄積する。しかし本研究が想定するグループは、短期間の間に流動的にグループ形成を繰り返すため、情報の蓄積よりも最新の情報を交換することに主眼がある。膨大なデータ処理能力よりも、ファイル共有の場を設けるコストが低いことが優先される。

ソフトウェア開発グループではファイル群をコンパイルし、動作させる必要がある。どのバージョンも常にコンパイル可能でなくてはならない。そのため CVS 等のバージョン管理システムでは、ファイル群を 1 箇所で集中管理し、逐次バージョンを残し、競合発生等によるソフトウェアの不具合をできるだけ回避する必要がある。一方、文書作成では、さまざまな人に参照やコメントをしてもらい、最終的に完成させることが目的となるため、必ずしも 1 箇所で集中管理して、逐次バージョンを残す必要はない。ファイルの場を設けるコストが低いことが優先される。

本研究では、このような特徴を持つ共同作業グループを対象としたファイル共有に着目して研究を進める。グループの特徴に基づいたファイル共有システムを設計・実装する。

1.4 本論文の構成

第 2 章では、既存のファイル共有システムについて分析を行なう。既存のシステムを電子メール型、共有フォルダ/ファイル型、P2P グループウェア型の 3 つに分類して、各システムの機能や特徴について述べる。また第 1 章で示した想定するグループの観点から、既存のシステムの特徴と問題点を整理して明らかにする。具体的にはバージョン管理、設置・運用コスト、非オンライン環境への対応の 3 項目で整理する。

第 3 章では、まず本システムが想定する場面例を上げ、分析する。システムモデルの検討項目として、通信モデル、ユーザーの識別、グループの形成、ファイル共有機能の検討項目として、ファイルの作成・更新、ファイルの同期、競合検知と解決、更新履歴の管理の検討を行う。これを基に、第 2 章であげたバージョン管理、設置・運用コスト、非オンライン環境への対応の 3 つの観点から、システムが満たすべき要件を整理する。

第 4 章では第 3 章で示したシステムの要件定義を設計したファイル共有システム Mikan

についてまず全体像を示す。次に作業に沿った流れで、アカウントの作成、プロジェクトの形成、ファイルの作成・更新、ファイルの同期、競合検知と解決、更新履歴の管理について設計を示す。

第5章では設計を基に実装したファイル共有システム Mikan について、実装環境とシステム構造を示し、システムの機能として、アカウントの設定、プロジェクトの作成、ファイルの作成・更新、ファイルの同期、競合検知と解決、更新履歴の管理を説明する。

第6章では、本システムの評価として、関連研究との比較による定性的評価と、慶應義塾大学大学院政策・メディア研究科特設科目「インターネット時代のセキュリティ管理」において運用実験を行い、これらをもとに評価と考察を行う。

第7章では全体の総括と、今後の課題と展望について言及する。

第 2 章

既存のファイル共有システムの分析

本章では、既存のファイル共有システムを 3 つのカテゴリに分類し、それぞれの特徴について言及する。前章で示した短期的な共同作業グループの観点から、既存のファイル共有システムの特徴と問題点を整理する。

2.1 ファイル共有システムの分類

文献 [18] によると、一般にネットワークを介した共同作業の方法として、電子メールのやり取り、共有フォルダ/ファイルへのアクセス、双方向のコンテンツ更新/アクセスの 3 つが知られている。

電子メールのやり取りは、共同作業を行うメンバー間で作業内容をメールとして交換し、直接やり取りする方法である。ユーザーが電子メールを利用できる環境を持っていれば利用できるため、導入コストや運用コストは低い。そのため設置コストを優先事項としているグループで利用される。一方でアクセスする可能性のある全ファイルに関して、ユーザー各人が複製を分類し、保持しなければならないという特徴を持っている。この電子メール型の共同作業では、メーリングリスト (ML) へのファイル添付が広く用いられている他、文書の添削に特化し、バージョン管理機能を持つ VCoCoA 等のシステムが開発されている。

共有フォルダ/ファイルへのアクセスは、特定のサーバー上の共有レポジトリにあるファイル群に、メンバーがアクセスする方法である。この共有フォルダ/ファイル型の共同作業を支援するシステムとして、永続的な組織で情報を蓄積し、膨大なデータを扱うことに主眼を置く、Lotus Notes/Domino を始めとする多くの C/S 型のグループウェアや、ソフトウェア開発で広く用いられている CVS 等のバージョン管理システム等があげられる。

双方向のコンテンツ更新/アクセスは、メンバーがサーバー上にある同じ文書群を同時に直接編集することにより、コンテンツ上で共同作業を行う方法である。編集している文書が常に最新版の状態であるという特徴がある。更新するまでの編集作業量が少なく、最新版を共有または公開することを優先事項としているグループに適している。Wiki[18] 等が有名であるが、本研究が想定する共同作業グループの特徴という観点から、共有フォルダ/ファイル型とほぼ同義であり、統合して扱うことにする。

また最近では、P2P グループウェアと呼ばれるシステムが登場している。多くの C/S 型のグループウェアでは、ユーザーや組織によっては、サーバー導入や運用コストが障害とな

表 2.1: ファイル共有システムの分類と主なシステム例

電子メール型	共有フォルダ/ファイル型	P2P グループウェア型
<ul style="list-style-type: none"> ・ ML ・ ESP (Eudora Sharing Protocol) ・ VCoCoA 	<ul style="list-style-type: none"> ・ Lotsu Notes/Domino ・ CVS (Concurrent Version System) ・ オフィスアクセラレータ 	<ul style="list-style-type: none"> ・ Groove ・ Net Peers

る。それに対して、P2P グループウェアは、ユーザーが新たにサーバーを設置・運用することなく利用できるという特徴を持っている。代表的な P2P グループウェアの groove や、会議やミーティングの場でアドホックなファイル共有を実現する Net Peers 等があげられる。

これを基に、既存のファイル共有システムを表 2.1 の 3 つのカテゴリに分類し、それぞれの特徴について言及する。

2.2 電子メール型

電子メール型は、ユーザーが電子メールを利用できる環境を持っていれば利用できる、場を設けるコストが低いという特徴がある。電子メール型は、ファイル管理機能の有無の観点から、メーリングリスト (ML) と、独自メールクライアント型に分類できる。

ここでは広く用いられている ML と、独自のクライアントを用いるシステムとしてメールクライアント Eudora が持つファイル共有機能 ESP と、文書の添削に特化し、バージョン管理機能を持つ VCoCoA について言及する。

2.2.1 メーリングリスト (ML)

かつて RFC822[19] で定められた電子メールはテキストメールと呼ばれ、本文に ASCII テキストしか扱えないものであったが、RFC2045[20], 2046[21], 2047[22], 2048[23], 2049[24] で定められた MIME により、画像や音声等のバイナリデータを添付することが可能となった。

電子メールにデータを添付する際には、BASE64 と呼ばれる 8 ビット 3 文字を 6 ビット 4 文字に変換する符号方式を用いており、これによりデータ量が元と比較して約 33% 増加するが、その手軽さから広く一般に用いられている。

ML 宛にデータを添付して送信することにより、ML のメンバー間でファイル共有を図ることができる。電子メールを利用できる環境を持っていれば容易に利用できるという特徴がある。

以前は、ML を利用するには、サービスを運用している組織のシステム管理者に、ML の作成やメンバーの追加・削除等の管理を依頼するか、そのようなサービスを利用できない場合には、fml[25], Majordomo[26] 等の ML 管理システムを独自に用意して運用する必要があった。しかし最近では、eGroups[27], FreeML[28], QuickML[29] といった、ユーザーが容易に ML を作成、管理できる無料サービスが広まり、サーバー管理技術を持たないユーザーでも手軽に ML を利用することが可能となった。



図 2.1: Eudora Sharing Protocol(出典は文献 [32])

2.2.2 ESP (Eudora Sharing Protocol)

ESP(Eudora Sharing Protocol) は、メールクライアント Eudora[30] が持つ、ファイル共有機能である [31]. 通信に SMTP を利用し、Eudora を利用しているメンバー間でファイルを共有することが可能である (図 2.1).

「他のメンバが同時に更新する可能性のあるファイルを ESP で使用しないでください」[31] 等の制限はあるが、例えば、遠く離れた家族の間でデジタル写真画像などを共有するなど、一旦共有されたファイルが更新される機会が少ない用途に利用できる。

2.2.3 VCoCoA

CoCoA[33] は、挿入、削除などの添削記号を SGML(Standard Generalized Markup Language)[34] に基づく CCML(Communicative Correction Markup Language)[35] と呼ばれる独自の通信添削用マークアップ言語で表現し、電子メールでの受送信による非同期の添削を実現している (図 2.2).

VCoCoA[15] は、CCML を拡張してバージョン管理が可能となった CoCoA である。添削者が逐次添削していく直列添削型だけでなく、複数の添削者が平行して添削を行う並列添削に対して、文書のバージョン管理とマージ機能を提供している (図 2.3). バージョン管理は CCML を拡張し、バージョン情報をファイルフォーマットに埋め込むことで実現している。

2.3 共有フォルダ/ファイル型

共有フォルダ/ファイル型は C/S 型のシステムで、サーバー上でファイルを管理し、膨大なデータを処理、分類する機能や高機能のバージョン管理機能等を兼ね備えているという特徴がある。

ここでは、代表的なグループウェアでレプリカ機能を持つ Lotus Notes/Domuno, 高機能のバージョン管理機構を持つ CVS, Web ベースで透過的に作業可能なオフィスアクセラレータについて述べる。

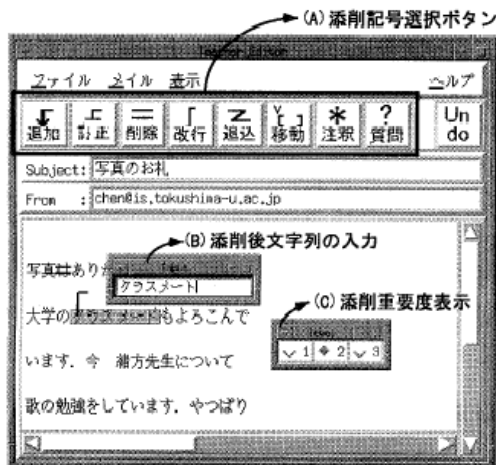


図 2.2: CoCoA-Editor(出典は文献 [15])

2.3.1 Lotus Notes/Domino

Lotus Notes/Domino(以下, ノーツ)では, 共有文書型データベースと呼ばれるデータベースシステムで, 文書管理を行うことができる [3]。ノーツはインデックスとビュー機能を持ち, 作成された日時, 作成者, カテゴリといった条件でデータベースにある情報を一覧表示することができる。

またノーツはサーバーにあるデータベースの一部をクライアント側に持つ, レプリカ機能を有している。ネットワークに接続していなくても, クライアント側で利用ことができ, 再度ネットワークに接続した際には, サーバー側のデータベースに変更点が反映される仕組みになっている。

2.3.2 Concurrent Version System(CVS)

CVS(Concurrent Version System)は, ネットワークを介して複数人が共同作業をする際に, 対象コンテンツのバージョン管理を行うためのシステムであり, オープンソフトウェア開発等で広く用いられている。CVSはコンテンツのバージョンを一元的に管理するため, 任意のバージョンのコンテンツを取り出したり, 以前の状態を参照することができる。

複数人のユーザーで利用するには, サーバー上に共有リポジトリを用意し, 各ユーザーが作業コピーを自分の作業スペースに取り寄せて (checkout), そのコピーに対して作業を行う (図 2.4)。コピーに対して行なった作業やファイルの追加 (add), 削除 (remove) を共有リポジトリに反映させるためには (commit) する必要がある。作業を再開するには, 共有リポジトリの内容を作業コピーに反映させる (update)。日常作業は, [update]->[ファイルの編集/add/remove]->[commit] を繰り返して進める。

複数のユーザーが平行して同じファイルの同じ部分に対して異なった形で編集し, それを commit しようとする競合が発生する。そこで CVS は競合通知と, 競合した結果のファイルをユーザーに提示する。ユーザーはそれを基にマージ作業を行い, 競合を解決する仕組みになっている。

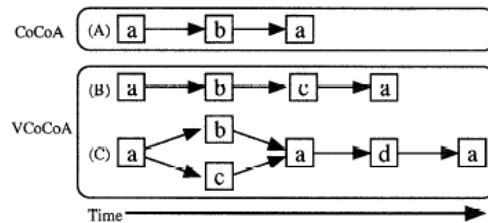


図 2.3: CoCoA と VCoCoA の添削文章の流れ (出典は文献 [15])

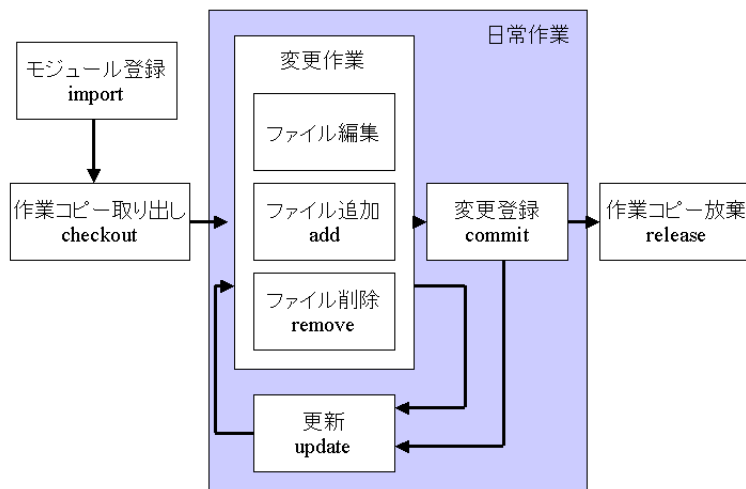


図 2.4: CVS を用いたルーチンワーク (出典は文献 [36])

ソフトウェア開発グループではファイル群をコンパイルし動作させる必要があるため、どのバージョンも常にコンパイル可能でなくてはならない。そこでこのようなファイル群を1箇所で集中管理し、逐次バージョンを残し、競合発生等によるソフトウェアの不具合をできるだけ回避する仕組みが不可欠となる。

その他

Microsoft Visual SourceSafe は、共同ソフトウェア開発を行なう際、ソースコードやデータのリソース管理、差分管理など GUI で行なえる。Microsoft 社製品のバイナリファイルを扱えるという特徴がある。

Web 上での共同作業を支援するバージョンングツールとして WebDAV[37] があげられる。これは現行の HTTP1.1 転送プロトコルを拡張することで、Web サーバーにファイルを格納出来る仕組みを持っている。

2.3.3 オフィスアクセラレータ

オフィスアクセラレータは、「Web ページ上の文書を任意の OLE プロトコル対応ツールで協同作成できる情報共有システム」[14] である。WWW サーバーと連動したサーバー上に、作業分担を定義したテンプレート文書を公開し、そのテンプレート文書に、各ユーザーが個人個人の編集用アプリケーションを用いて書き込むことができる。

Web ブラウザを機能拡張した文書ブラウザを用意することで、Microsoft 社の複合文書機能 ActiveX/OLE に対応させ、文書ブラウザとサーバー間の通信を HTTP(HyperText Transfer Protocol) にプロトコルをカプセル化することでこれを実現している。

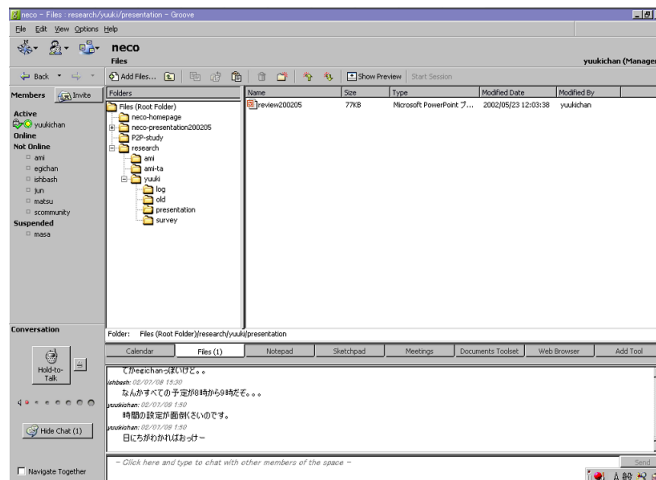


図 2.5: Groove のファイル共有機能

2.4 P2P グループウェア型

共有フォルダ/ファイル型に代表される多くのファイル共有システムは、ユーザーや組織によっては、サーバーの設置や運用等のコストがネックとなる。P2P グループウェア型はそれらのユーザーや組織を対象とし、ユーザーが新たにサーバーの設置・運用を行わなくても利用できる等の特徴を持っている。

ここでは代表的な P2P グループウェアである Groove、アドホックなファイル共有を実現する Net Peers について言及する。

2.4.1 Groove

Groove は、ロータスノーツの生みの親と言われる Ray Ozzie が、新たに作り出したソフトウェアである [9]。P2P 型グループウェアと呼ばれ、従来の C/S 型と異なり、サーバを設置することなく利用できる。一方で、例えば企業などで利用する際にクライアントを管理したい等のサーバーを必要とする要望があれば、サーバーを追加できる柔軟性を備えている。

NAT(Network Address Translator) 等の存在でクライアント間が直接通信できない場合は、Groove Networks 社 [7] のサーバーが Relay している。グループ単位でシェアスペースと呼ばれる作業空間を作成することができる。シェアスペースではチャット、IM 等のコミュニケーション手段から、スケジュール管理、ファイル共有 (図 2.5) 等の多くの機能を、統合されたインターフェースで利用することが可能である。ファイル共有機能は、ユーザーがファイルを登録すると、シェアスペース内の他ユーザーにそのファイルを複製 (同期) する仕組みになっている。共有しているファイルは、一つの大きなファイルという形で格納されており、Groove のインターフェイス経由でないと取り出せない。

また Groove Networks 社は、Microsoft との提携を表明しており、Microsoft PowerPoint を用いたリアルタイムなプレゼンテーションや、Microsoft Word 上でリアルタイムに複数人が編集可能になるといった、Microsoft 社の製品と連動するための環境整備が進められている。



図 2.6: NetPeers: 流動型ファイル共有システム (出典は文献 [38])

2.4.2 Net Peers

Net Peers は、慶應義塾大学環境情報学部齊藤研究室で開発されている、会議やミーティングの場でアドホックにグループを形成し利用できる P2P 型グループウェアである。クリップボード共有機能、ディスプレイ共有機能、流動型ファイル共有システム等を実装している [38]。Jxta プロトコルに準拠し、サーバーレスでネットワークを構築できる。

流動型ファイル共有システムは、ファイルが一箇所に留まらず、常にピアからピアに移動する仕組みを持ち、これによりファイル提供者が常にネットワーク上にいなくても、ファイル入手の可能性を高めている。

2.5 既存のファイル共有システムの特徴と問題点

上述した既存のファイル共有システムについて、表 2.2 にその特徴と問題点を整理する。バージョン管理は、競合の検知・解決の 2 つの機能の有無を表わしている。設置・運用コストは、サーバーの設置・運用やグループ形成が容易等のコストを表わしている。また非オンライン環境への対応は、グループ内にオフラインで作業し更新時にのみオンラインになるユーザーがオンライン時に最新版のファイルを共有可能かを表わしている。

電子メール型 (ML) は、ユーザーがデータのやり取りを既存のメールサーバーを介して行うため、導入・運用コストの低さが利点となる。またデータを手元で保持する必要があるため、非オンライン環境での作業が可能となるが、扱うデータが膨大になると、その分類や保持にコストがかかる。このような特性から、短期的にグループを形成して共同作業を行なう場合に適していると考えられる。

独自メールクライアント型はこの ML の利点を活かしたファイル管理機能を提供している。ESP は電子メールを通信手段としたファイル共有を実現しているが、競合の検知や解決を行う機能は無く、複数のメンバーが同時に更新する可能性のあるファイルを利用できない等の制限がある。VCoCoA は、導入・運用コストが低い電子メールの特性に加えて、テキスト文書が対象だが、複数ユーザーの同時添削を可能とするバージョン管理機能と、それを自動マージする機能を保持している。一方でテキストデータを独自の CCML 言語形式

表 2.2: 既存のファイル共有システムの特徴と問題点

	バージョン管理	設置・運用コスト	非オンライン環境への対応
電子メール型 (ML)	無	低い	作業できる
共有フォルダ / ファイル型	可能	高い	作業できる
P2P グループウェア型	競合検知は可能	低い	ファイル交換時に制限あり

表 2.3: 独自メールクライアント型の特徴

	競合の検知	任意のファイル形式	競合の解決
ESP	無	扱える	無
VCoCoA	可能	独自フォーマットのみ	自動マージ

に直して、専用エディタでブラウズするという方式を取っており、バージョン情報もその中に埋め込まれているため、任意のファイル形式のバージョン管理を行うことはできない。独自メールクライアント型の特徴を表 2.3 にまとめる。

共有フォルダ/ファイルへのアクセス方式では、膨大な文書データを扱うことが可能となる。そのため長期間に及ぶソフトウェア開発や、継続的に過去のファイルデータを蓄積して共有する場合に適している。CVS は履歴管理の他、競合検知、またテキストファイルでは、`diff` コマンドを用いて差分を表示し競合解決を支援する機能を持っている。ロータスノーツや CVS では必要な作業コピーをクライアント側に持つことができるため、非オンライン環境での作業が可能である。一方で本研究が想定する共同執筆等のグループにおいては、サーバーの設置・運用、サーバ管理技術等のコストの高さが障害となる。

P2P グループウェア型は、膨大なデータを扱うのには向いていないが、ユーザーが新規にサーバーを設置・運用する必要がないため、上述したようなコストの敷居は低い。また一時的なグループ形成で文書作成を行なう場合についても、容易にグループの形成できる環境を提供している。しかし、常時オンライン環境を基本としており、ファイルを交換する際にはユーザーが同時にオンラインである必要がある。非オンライン環境で作業を行なうメンバーが複数いる場合には、更新して交換したはずのファイルが行き届かない等の問題が発生する。また、groove は共有したファイル群を一つの大きなファイルという形で格納しており、Windows 付属の Explorer 等の一般的に用いられているファイルマネージャーを利用することができない。

本研究では、バージョン管理、設置・運用コスト、非オンライン環境への対応の 3 点を満たす、短期的な共同作業グループを対象としたファイル共有システムを設計・実装する。次章では、この 3 点について、システムが満たすべきそれぞれの要件をまとめる。

第 3 章

短期的な共同作業を対象としたファイル共有システムの検討

本章では、前章でまとめた既存システムの特徴と問題点を基に、短期的な共同作業を対象としたファイル共有システムを設計するにあたってのフレームワークを検討する。

3.1 本研究が想定する場面とその分析

本研究が想定する短期的な共同作業の場面例を上げ、分析を行う。想定する場面例として、報告書や論文の共同執筆、発表資料の共同作成をあげる。

3.1.1 場面例 1: 報告書や論文の共同執筆

文書作成においても、ファイル共有は重要な位置を占めている。例えば次のようなレビュー型の文書作成があげられる。

1. 報告書や論文を共同執筆しており、各担当者が各担当箇所を執筆しつつも、他章への参照やコメントを試みる文書作成
2. 個人で論文や書籍の執筆を行っているが、その執筆物を指導者や仕事仲間等と共有して、コメントを求める文書作成

商業出版における共同執筆や大学におけるオムニバス型の研究計画書作成等で広く用いられているモデルとして職業編集者型モデルがあげられる。これは編集者が各執筆者間の仲介役を果たし、全体の統合を図っていくモデルである (図 3.1 左)。

1 は、企業研究所等で、前章を踏まえた上で次の章を展開する一貫した流れがあるタイプの報告書作成等で用いられているモデルである。このモデルでは、編集作業をする担当者はいらぬものの、執筆者がお互いの担当部分に関してコメントし合う (図 3.1 右)。2 の、個人が自らが執筆した論文等の文書を他人が赤入れし、それを反映していく行為もこのモデルに近い。

職業編集者型モデルは、編集者と著者の間でファイルのやり取りを行い、編集者がファイルの全体的な管理を行っている。一方、レビュー型モデルでは、著者間でもファイルのやり取りを行うため、ファイル共有を適切な方法で実現することが重要となる。

また非オンライン環境で作業し、更新時にのみオンラインになるといった、非オンライン環境で作業する時間が多いメンバーがいる場合が想定されるが、比較的短期間の共同作業であるため、更新時に最新版を共有できることが重要となる。

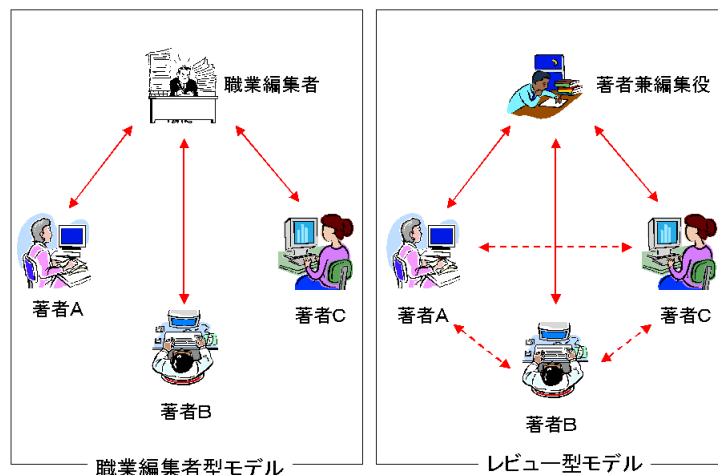


図 3.1: 職業編集者型モデルとレビュー型モデル

3.1.2 場面例 2: 発表資料の共同作成

近年、プレゼンテーション用のソフトウェアが普及し、企業における商品説明や大学における講義、講演、発表等で PC で作成したスライドが用いられる機会が増えている。

これらの資料はスライド単位で作成が可能であるため、一部の組織や、大学のグループワークでは、複数人のユーザーで発表資料を作成する試みが行われている。

各スライドの内容ごとにスライド単位で分担し、それらを持ち寄って、スライドの順番を変更したり加工しながら、一つのファイルにまとめていく (図 3.2)。

発表資料の作成は一般的に短期間であり、比較的頻繁にファイルの更新が行われる。共同執筆同様、非オンライン環境で作業し、更新時にのみオンラインになるユーザーがいた場合、そのユーザーが更新時に最新版を共有できることが重要となる。

以上の想定場面とその分析を踏まえて、システムモデルとファイル共有機能について検討し、ファイル共有システムの要件定義を行う。

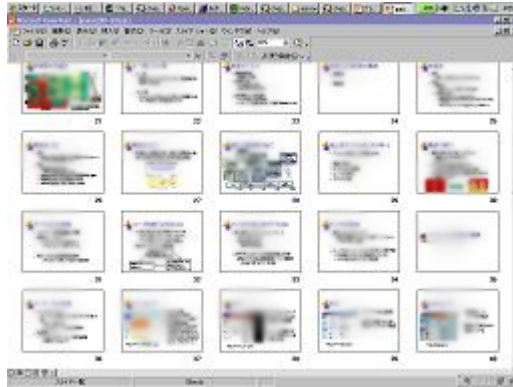


図 3.2: Microsoft PowerPoint 2000: スライド一覧機能

3.2 システムモデルの検討

システムモデルの検討として、通信モデル、ユーザーの識別、グループの形成の3点について検討を行う。

3.2.1 通信モデル

2.5 で示したように、既存のファイル共有システムには、次の3つのモデルがある。

1. 電子メール型
2. 共有フォルダ/ファイル型
3. P2P グループウェア型

共有フォルダ/ファイル型は、膨大な情報を扱うため、高性能のサーバーを設置し情報を蓄積する。しかし本研究が想定するグループは、短期間の間に流動的にグループ形成を繰り返すため、情報の蓄積よりも情報交換に主眼がある。膨大なデータ処理能力よりも、情報交換の場を設けるコストが低いことが優先される。

P2P グループウェア型では、ユーザー同士がファイルを交換する際に、同時にオンラインになる必要がある。非オンライン環境で作業を行なうメンバーが複数いる場合には、更新したファイルが行き届かなかった等の問題が発生する。

電子メール型では、既存のメールサーバーに一時的にファイルをキャッシュすることで、通常の作業を非オンラインで行い、更新時にのみオンラインになるユーザーでも更新時に最新版を共有することが可能となる。また、組織のイントラネット内で、NAT等の存在でクライアント間が直接通信できない場合に既存のメールサーバーをリレーすることで、通信が行えるという利点もある。

そこでユーザーが新たにサーバーを設置・運用することなく、また既存にサービスに一時的に差分をキャッシュできる電子メール型をベースにシステム的设计・実装を行う。

3.2.2 ユーザーの識別

通信に電子メールを用いるため、ユーザーの識別にメールアドレスを用いることが可能である。メールアドレスを利用するにあたって、普段利用しているメールアドレスを利用する方法と、専用のメールアドレスを利用する方法が考えられる。

普段利用するメールアドレスを利用する場合、受信する際に本システムのメールを本システムで、その他のメールを各ユーザーが利用しているメールクライアントを用いて受信できることが望ましい。近年、Becky! Internet Mail[39] や Microsoft Outlook Express[40] といった広く利用されているメールクライアントでは、受信時にサーバー上のメールのヘッダを読み込み、取り込むメールと取り込まないメールを判断するためのフィルタリング機能が実装されている。また既読のメールを後日自動的に削除する設定が可能なメールクライアントが多数存在する。

そこで、本システムにサーバー上のメールのヘッダ情報を取得しフィルタリングする機能を実装することにより、フィルタリング可能なメールクライアントを利用しているユーザーが、メールアドレスを併用したいというニーズに対応することが可能である。

専用メールアドレスを利用する場合は、普段利用していないメールアドレスを利用するか、Yahoo!メール等のサービスで無料メールアドレスを取得し、これを専用メールアドレスとして利用する方法がある。

ユーザーによって、使用しているクライアントや、本システムを利用する期間、用途等が異なるため、ユーザー本人がいずれかを選択できることが望ましい。

3.2.3 グループの形成

短期間の間にグループを再形成していくニーズを満たすためには、容易にグループ形成を行える必要がある。

グループを識別する方法として、ユーザーが独自にグループ名を決めて、その名前を識別子として扱う方法と、グループのメールアドレスとして用いられている ML を用いる方法が考えられる。

前者の方法は、ML を作成するコストがかからないが、各ユーザーのクライアントでメンバーリストを管理する必要があるため、グループへの新規メンバー加入の際に、各ユーザーのクライアントが管理しているメンバーリストの更新に遅延が発生した場合、新規メンバーに届かないファイルが出てくる等、ユーザー間で共有しているデータの整合性が取れなくなる可能性がある。

後者の方法は、ML を作成するコストが必要となるが、新規メンバーの追加は容易である。グループでファイルの共有を行う際は、かならず ML のあるサーバーを介して行われ、その際に常に最新のメンバーリストを参照して配信を行うため、メンバー追加によるデータの整合性問題は発生しない。2.2.1 で述べたように、eGroups[27]、FreeML[28]、QuickML[29] といった、ユーザーが容易に ML を作成、管理できる無料サービスが広まり、サーバー管理技術を持たないユーザーでも手軽に ML を利用することが可能となった。

そこで本システムでは、グループを識別する方法として、ML アドレスを利用し、ML でメンバー管理を行う仕組みにする。またグループに途中参加するメンバーのために、作業ディレクトリ内のファイル群と各ファイルの情報を送信し、最新版を共有する仕組みを用意す

る必要がある。

3.3 ファイル共有機能の検討

ファイル共有機能の検討として、ファイルの作成・更新、ファイルの同期、競合検知と解決、更新履歴の管理の4点に関して検討を行う。

3.3.1 ファイルの作成・更新

ファイルの作成・更新を行う方法として、専用のエディタやツールを用いて行う方法と、ユーザーが任意のエディタやツールを用いて行う方法がある。前者の利点は、専用の機能を付加することが可能な点である。例えば VCoCoA では、独自の言語 CCML を用いて JIS 規格の添削記号を文章中表示する機能やファイルフォーマットにバージョン情報を付加するなど、協同添削グループに対して高度な機能を提供している。

後者の利点は、より幅広い目的や用途にシステムを利用できる点である。例えば、ソフトウェア開発で広く用いられている CVS は、プログラムのソースファイルに限らず、テキストファイルであれば差分を取ってバージョン管理できるため、プログラムのマニュアル作成の他に、Web サイトのコンテンツ管理や一部では共同文書作成にも用いられている。

本研究で想定するグループでは、文書や資料等の幅広いフォーマットのファイルを扱っており、文書や資料といったファイルを作成・更新する際には、各ユーザーが各自のエディタ、ファイルマネージャーを利用した作業が可能であることが望ましい。

また、作成したファイルを他のメンバーと共有する際に、どのファイルが共有されるのかを当ユーザーが認識した上で同期を実行する、アウェアネスの仕組みが必要である。

3.3.2 ファイルの同期

ファイルが同期された状態とはそのファイルが、全メンバーの更新が反映された最新版の状態であることを指す。ユーザーはファイルの同期を行うことでそのファイルを最新版の状態にすることができる。

ファイルの同期は、電子メール型の通信モデルを前提とした場合は、作成したファイルを他ユーザーと共有するためにはメールの送信を、他ユーザーが作成したファイルを受け取るにはメールの受信を行うことで実現できる。

ファイルの同期で問題となるのは、同じファイルを複数のユーザーが同時に更新した際の競合発生である。競合とは、同じファイルを複数のユーザーが同時に編集し、同期した際に、内容の異なる同じ名前のファイルが複数生成される状態を指す。

競合発生を解決する手段として、ファイルロックをかける等の方法によって、競合の発生そのものを防止する方法と、競合発生時の影響規模を小さくすることによって、競合の解決を容易にする方法の2つが考えられる。

ファイルロックをかける方法は、あるファイルを一人のユーザーが更新している間は、他ユーザーは一切更新を行えなくする方法である。これはロックを強制解除する等の方法を除き、競合発生が起こらない方法である。

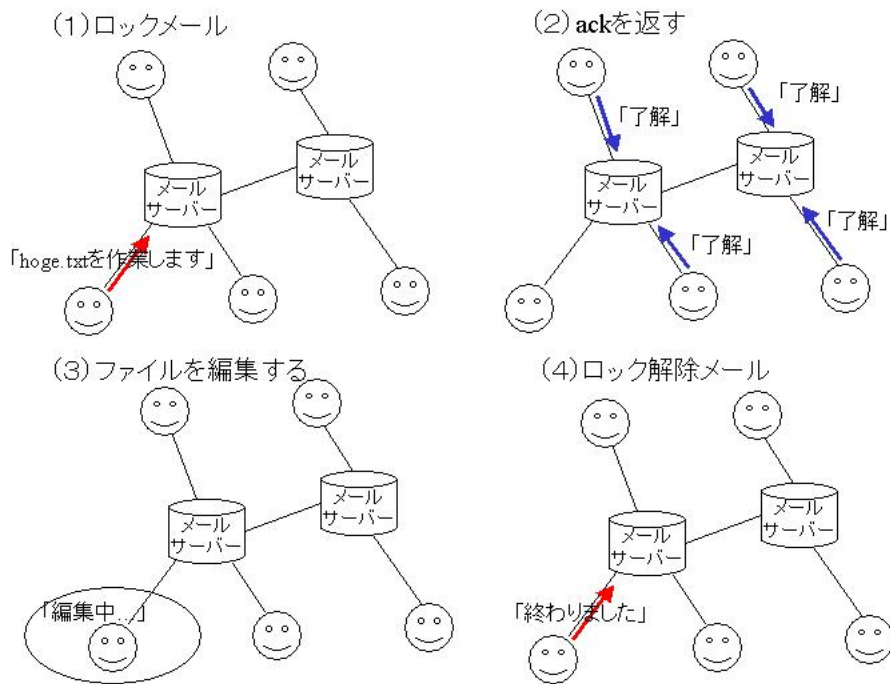


図 3.3: 電子メール型におけるファイルロック機構

電子メール型で完全なロックをかけるためには次のような方法で行う必要がある(図 3.3). ファイルを更新する A さんはまず他メンバーにロックメールを投げる. 他メンバーはロックメールを受け取ると ack を A さんに返す. メンバー全員の ack が返ってきたら, A さんはファイルの更新作業を行い, ロック解除メールと一緒に更新したファイルを送信する. ロックメールを複数のユーザーが同時に投げる可能性があるため, 他メンバーからの ack を待つからファイルの更新作業を行う手間が生じる. 特に非オンライン環境で作業を行っているメンバーがいる場合は ack を待つ時間が長くなるため, この方法は現実的ではない.

そこで競合発生時の問題規模を小さくして解決を容易にする方法を考える. ロックをかけずに, 複数ユーザーが同じファイルを更新可能な機構を持つシステムとして CVS がある. 2.3.2 で触れたように CVS では, 各ユーザーは更新したファイルを共有レポジトリに反映するために Commit コマンドと, 他ユーザーの更新ファイルを受け取るために Update コマンドを用いる. CVS を利用するいくつかのコミュニティでは, Commit する前に Update を実行し, 競合が発生したらそれを解決して Commit するという暗黙のルールがある. 競合の影響範囲を 1 ユーザー内に押さえられるので競合の解決が容易となる. 本システムでは, メールを受送信を同期コマンドとして一つに統合し, Update してから Commit する流れを参考にして, 受信を行ってから送信を行う仕組みにすることで競合問題に対処する. SMTP でメール送信する前に POP3 認証を求め, POP3 before SMTP に対処できる利点もある.

3.3.3 競合検知と解決

先ほど述べたように、競合とは、同じファイルを複数のユーザーが同時に編集し、同期した際に、内容の異なる同じ名前のファイルが複数生成される状態を指す。ファイルの同期によって競合が発生した際、競合を検知し、それをユーザーに通知する。

検知された競合を解決するにあたって次の3つの選択肢が考えられる。

1. 一つのファイルに内容を統合する
2. どちらかのファイルを破棄する
3. どちらかを別名ファイルで保存し、共有する

既存の競合解決として自動マージ機能があるが、扱うファイル形式に依存したり、ファイル内容によっては行単位やパラグラフ単位といった粒度の問題ある。

そこで本研究では、競合検知を重視し、競合解決はファイル形式や内容に依存しない上の3つの選択肢による解決方法を用意する。

また、電子メール型の通信モデルでは他に、同期を実行して受信を行った後に、送信するファイルと同じファイルが送信されてくる場合や、特定のメールが遅延によって後日届いた場合に、競合が発生する可能性が考えられる。

これらの問題に対処するために、各ファイルの現バージョン情報と前バージョン情報を管理し、同期作業でファイルを更新する際に照合して競合検知し、それを解決する手段を用意する必要がある。

3.3.4 更新履歴の管理

更新履歴は、システムが機能上必要とするものと、ユーザーが任意に利用するものがある。ユーザーが更新履歴を任意に利用する用途として次の2つが考えられる。

誰がどのような更新を行ったのかを知る

ファイルを以前の状態に戻す

これを実現するために、同期を実行してファイルの同期が行われる際に、誰がいつどのファイルを更新したかを記録し、過去の版ファイル群を保管する必要がある。

また、先ほど述べたように、同期を実行して受信を行った後に、送信するファイルと同じファイルが送信されてくる場合や、特定のメールが遅延によって後日届いた場合の競合発生に備えて、各ファイルの現バージョン情報と前バージョン情報を管理する必要がある。

3.4 システムの要件定義

上記の議論を、第2章で述べた、バージョン管理、設置・運用コスト、非オンライン環境への対応の観点から整理し、システムの要件定義を行なう。

3.4.1 バージョン管理

バージョン管理に必要な要件は大きく、メンバー加入時のファイルと管理情報の共有、更新ファイル群の同期、競合発生と解決、更新履歴の管理の4点に分けられる(表3.1)。

表 3.1: バージョン管理に関するシステム要件

更新ファイル群の同期	・受信 送信の流れによる競合問題の簡易化
競合発生と解決	・競合発生をユーザーに通知し、競合ファイルは同期不可 ・競合解決をユーザーが容易に実行可能
更新履歴の管理	・更新履歴の作成、過去の版ファイル群を保管 ・各ファイルの現バージョンと前バージョンの履歴管理
メンバー加入時のファイルと管理情報の共有	・新規メンバーが最新版ファイルとファイルの管理情報を取得可能

まず、ユーザー間で更新したファイルをやり取りするため、同期機能が必要となる。同期機能は、メール受信してからメールの送信を行う仕組みにし、競合が発生した場合の影響範囲を押さえ、競合解決を容易にする。

次に競合処理機能が必要となる。競合が発生した際に、それをユーザーに通知し、競合が発生したファイルの同期を中止する。その上でユーザーが容易に競合を解決する手段を提供する。

ユーザーが更新履歴を参照したり、また競合を検知するために、更新履歴を管理する機能が必要となる。

また、途中からプロジェクトに参加する新規ユーザーがいた場合に、すぐに共同作業に参加できるよう最新版ファイルと、競合検知を可能にするために、ファイルの管理情報を取得できる機能が必要となる。

3.4.2 設置・運用コスト

設置・運用コストは大きく、サーバーの設置コスト、ユーザーの利用コストの2点に分けられる(表3.2)

表 3.2: 設置・運用コストに関するシステム要件

サーバーの設置コスト	・新たにサーバーを設置・運用不要 ・NAT を間に挟んだ通信が可能
ユーザーの利用コスト	・環境や用途に応じて、併用メールアドレス、専用メールアドレスの選択可能 ・併用メールアドレスを利用する際のメールのフィルタリングが可能 ・グループ管理やメンバー追加が容易 ・任意のエディタやツールでファイルを作成可能 ・他ユーザーに共有されるファイルに対するアウェアネス

本研究では、短期的な共同作業グループを対象としており、ユーザーが新たにサーバーを設置・運用する必要がなく、また NAT を間に挟んだ通信が可能である必要がある。

ユーザーの利用用途や環境に合わせたアカウント設定が可能であること、またグループ形成やファイルの作成・更新が容易である必要がある。

3.4.3 非オンライン環境への対応

非オンライン環境への対応を行うには、既存のサーバーサービス上でのキャッシュ機能が必要となる (表 3.3)。

表 3.3: 非オンライン環境への対応

キャッシュ機能	既存のサーバーサービスに一時的に差分をキャッシュ可能
---------	----------------------------

非オンライン環境で作業し、更新時にのみオンラインになるユーザーが同期を行った際に最新版のファイルを共有できるよう、既存のサーバーサービスに一時的に差分をキャッシュ可能な必要がある。

第 4 章

SMTP を利用したファイル共有システムの設計

本章では前章でまとめたシステム要件を基に, SMTP/POP3 を利用したファイル共有システム Mikan(みんなでかんたんファイル共有, 以下 Mikan) の設計を行う. まず Mikan の概要としてシステム利用の全体像について述べ, それから作業の流れに沿って, 各機能について設計を述べる.

4.1 システムの概要

Mikan は, 通信に SMTP/POP3(APOP) を用いたファイル共有システムである. システム利用の全体像を図 4.1 にあげる. 既存のメールサーバーを介して Mikan クライアントがファイルの受送信を行う.

ユーザーは新たにサーバーを設置・運用する必要なく, Mikan クライアントをパソコンにダウンロードし, アカウントの設定, プロジェクトの作成を行うことでファイル共有を行える.

また既存のメールサーバーに一時的に差分がキャッシュされるため, 外出先や移動中等の空き時間にオフラインで作業し, 更新時にのみオンラインになるといった利用にも対応できる.

4.2 作業に沿った動作

作業の流れに沿って, 各作業における Mikan の設計を述べる.

4.2.1 アカウントの作成

アカウントを作成する際に, 表 4.1 の項目を設定する. 設定を行った内容は設定ファイルに保存される. 但しパスワードは, 設定ファイルには保存せず, 起動して最初の同期コマンド実行時に, ユーザーは再度パスワードを入力する.

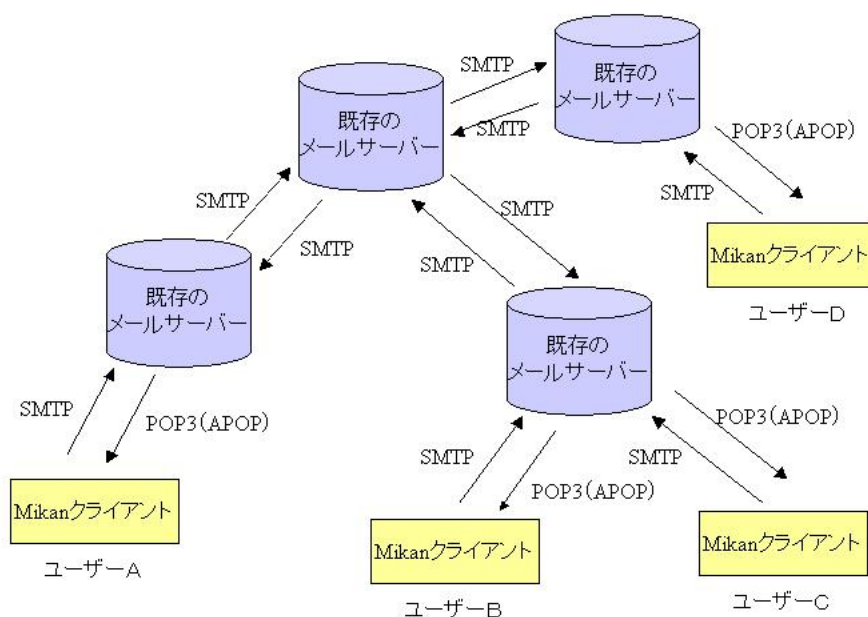


図 4.1: システム利用の全体像

表 4.1: アカウント設定項目

項目	内容
ユーザー名	ユーザーが任意の名前を付ける
メールアドレス	ユーザーを識別する際に用いる. ユーザーは利用するメールアドレスを登録する
POP3 サーバー	利用する ISP の POP サーバーを設定する
SMTP サーバー	利用する ISP の SMTP サーバーを設定する
ユーザー ID	メールを POP で受信する際のユーザー ID を設定する
パスワード	メールを POP で受信する際のパスワードを設定する
認証方式	利用する ISP に応じて POP3 か APOP を選択する
Mikan 以外のメール	Mikan 以外のメールをサーバー上に残すか否かを設定する

ユーザー名は, ユーザーが任意の名前を付ける. ユーザーの使用するメールアドレスによっては, 誰のメールアドレスであるか認識できない場合が考えられる. そこでメールアドレスとセットで, 最終更新者や更新履歴等をユーザーに図 4.2 の形で表示する際に用いる.

メールアドレスは, ユーザーを識別する際に用いる. ユーザーは Mikan で使用するメールアドレスを登録する.

POP3 サーバー, SMTP サーバー, ユーザー ID, パスワードは, 利用する ISP のものをそれぞれ登録する.

| ユーザー名 <メールアドレス>

図 4.2: ユーザー名の表示

表 4.2: プロジェクト設定項目

項目	内容
プロジェクト名	任意のプロジェクト名
ML アドレス	プロジェクトの識別に用いる. 各プロジェクトの ML アドレスを設定する
作業ディレクトリ	プロジェクトの他メンバーと共有するディレクトリを指定する

Mikan 以外のメールをサーバーに「残す」「残さない」を選択することができる. 普段利用しているメールアドレスを併用する際には「残す」を選択する. Mikan はサーバー上のメールを取り込む前に, 各メールのヘッダのみを読み出し, X-Mailer ヘッダ部分を参照して Mikan メールを判断することでフィルタリングを行う (図 4.3). 最後の数字はメッセージの種類を表している. 0 は通常のファイル同期メッセージ, 1 は新規メンバーをプロジェクトに招待する invite メッセージを表す.

| X-Mailer: Mikan 0

図 4.3: X-Mailer ヘッダ

4.2.2 プロジェクトの形成

プロジェクトの作成には, ユーザー本人が新規作成を行う方法と, 既に存在するプロジェクトの他メンバーから invite メッセージを受け取り, 作成を行う方法がある. プロジェクトは複数作成し, 並行して運用できる.

プロジェクトの作成にあたって, 表 4.2.2 の項目を設定する. プロジェクト名は, ユーザーがプロジェクトを切り替える際の識別子として利用する. ML アドレスは, 同期コマンドを実行した際にファイル送信の宛先として用いたり, ファイルを受信した際にプロジェクト毎にファイルを各作業ディレクトリに振り分ける際に用いる. 作業ディレクトリは, 他メンバーと共有するディレクトリを指定する.

プロジェクトの新規作成はこれらの項目を設定して作成する. 設定完了の際に, 記入漏れと指定された作業ディレクトリが存在するかをチェックする. ユーザーが設定を終えると Mikan は, 設定された作業ディレクトリに, ファイル情報を格納するためのファイル (以下 .mikan とする) と, 履歴保存用のディレクトリ (以下 MIKANDIR とする) を作成する. 作業ディレクトリ内のファイルを検索し, 各ファイルの情報を .mikan に格納する. 扱うファ

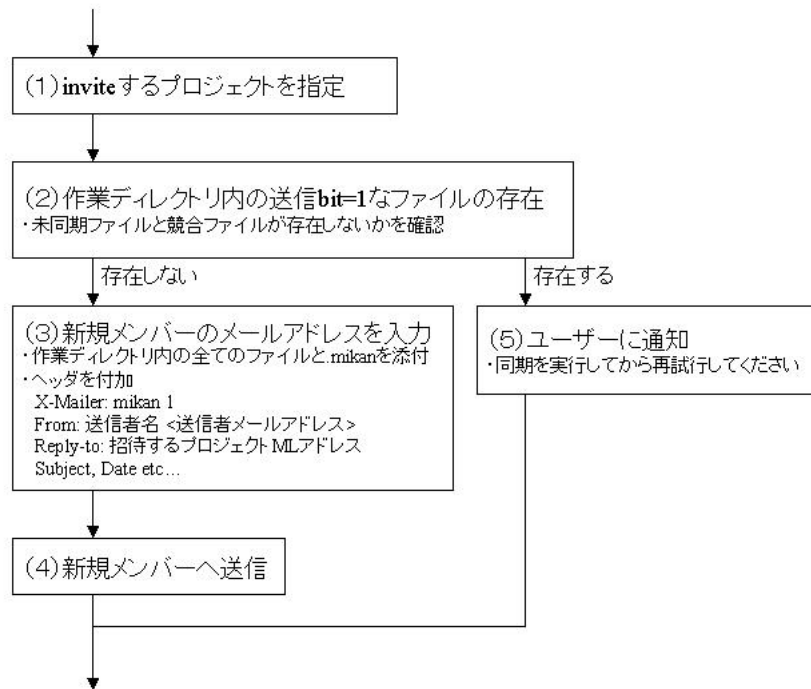


図 4.4: invite メッセージ送信時の処理

イル情報の項目は 4.2.3 で、履歴保存用のディレクトリの詳細に関しては 4.2.6 で後述する。

invite メッセージの送信を行うには、まず招待するプロジェクトを指定する (図 4.4(1))。発信者の作業ディレクトリのファイル情報を確認し (図 4.4(2))、同期されていないファイルが存在した場合は、invite メッセージを送信せず、まず同期作業を行うようユーザーに求める (図 4.4(5))。すべてのファイルが同期されている場合は、新規メンバーのメールアドレスを指定し、Mikan は作業ディレクトリ内のファイルと .mikan を添付し、ヘッダをつけて (図 4.4(3))、新規メンバーに送信する (図 4.5(4))。

このメールが invite メッセージであることを表すために、X-Mailer ヘッダに「Mikan 1」を入れる。From ヘッダには送信者の名前とメールアドレス、Reply-To ヘッダには招待するプロジェクトの ML アドレスをセットする。

invite メッセージを受け取った際の処理を図 4.5 に示す。新規メンバーの Mikan は、X-Mailer ヘッダ部分を参照し invite メッセージであれば、プロジェクト参加の招待があったことをユーザーに伝える (図 4.5(1))。ユーザーが「承認」を選択し、プロジェクト名と作業ディレクトリを設定する (図 4.5(2)) と、Mikan は指定された作業ディレクトリに、添付ファイルを展開し、.mikan 内のファイルスタンプ情報 (後述) を展開した時間に書き換えて、発信者と同じ作業ディレクトリ環境を構築する (図 4.5(3))。その後、サーバー上から invite メッセージを削除する (図 4.5(5))。

プロジェクト参加を保留する場合や、知らないユーザーからのプロジェクト招待を受け取った場合は、プロジェクト参加への招待で「キャンセル」を選択し、次にサーバー上に invite メッセージを残すか否かを選択する (図 4.5(4))。「残す」を選択すると、再度同期コマンドを実行した際に、プロジェクト参加の手続きが行われる。「削除」を選択すると、

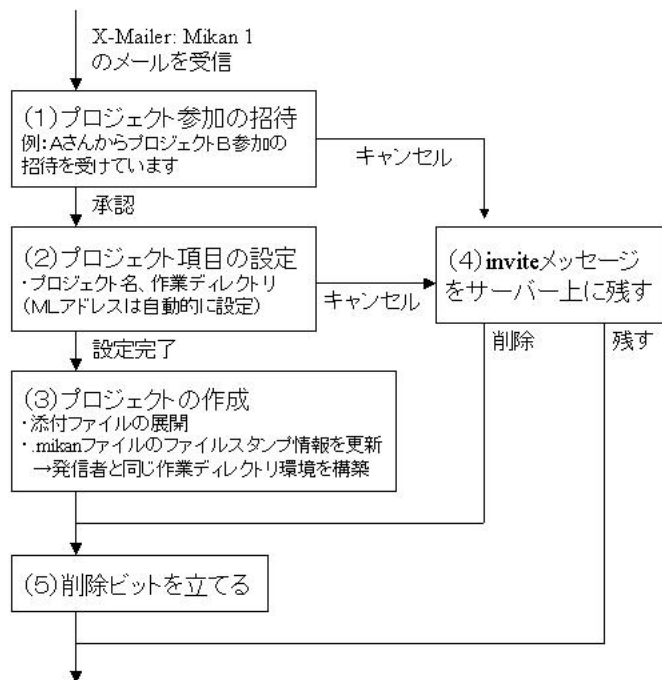


図 4.5: invite メッセージ受信時の処理

サーバー上から invite メッセージを削除する (図 4.5(5)).

4.2.3 ファイルの作成・更新

作業ディレクトリ内のファイル操作は任意のファイルマネジャー、ファイルの作成・更新は任意のエディタやツールを用いる。Mikan は作業ディレクトリ内の各ファイルに関して、表 4.2.3 の情報を管理する。これらの情報は .mikan に格納されている。

Mikan は前同期時のファイルスタンプを記憶し、作業ディレクトリ内の現在のファイルスタンプと比較する。ユーザーが作業ディレクトリ内で、ファイルを作成し保存すると、前同期時と現在のファイルスタンプが一致しなくなるため、送信 bit を 0 から 1 に設定

表 4.3: ファイル管理情報

項目	内容
ファイル名	以下の項目とファイル名をセットで管理
最終更新者	そのファイルに対して最後に更新し同期を行ったユーザー (From ヘッダ)
更新日時	上記のユーザーが同期を行った時間 (Date ヘッダ)
ファイルスタンプ	最後に同期を行った瞬間の各ファイルのスタンプ
送信 bit	0:送信不要, 1:同期が必要なファイル
受信 bit	0:受信済み, 1:受信したファイル

表 4.4: ファイルの状態表示

状態	内容	送信 bit と受信 bit
未同期	当ユーザーが作業しているファイルが、まだ同期していない状態。同期するまで表示	送信=1 受信=0
更新	当ユーザーが作業していないファイルが、他ユーザーによって更新された状態。次回同期時まで表示	送信=0 受信=1
競合	当ユーザーが作業しているファイルが、他ユーザーによって更新された状態。マージするまで表示	送信=1 受信=1
その他	前同期時から更新されていない状態等。何も表示しない	送信=0 受信=0

する。

ユーザーは作業ディレクトリ内のファイル情報を Mikan を介して確認できる。 .mikan に格納している情報を基に、リストビュー形式で表示する (図 4.2.3)。

状態	ファイル名	サイズ	最終更新者	更新日時
----	-------	-----	-------	------

図 4.6: ファイル情報の表示

まず、表示する前に Mikan は作業ディレクトリ内にあるファイル群をチェックし、ファイル名のリストを作成する。そのリストを基に .mikan ファイルの情報と照合を行い、表示するリストビューを作成する。

状態は .mikan ファイルに格納されている送信 bit と受信 bit (表 4.2.3) 情報を基に、各ファイルの現在の状態を表示する。状態には未同期、更新、競合、その他がある (表 4.2.3)。

サイズは、各ファイルの現在のファイルサイズを表示する。ファイルサイズを表示することで、ユーザーが極端に大きなファイルを作業ディレクトリに置き、同期する行為を防ぐ。

最終更新者と更新日時は .mikan の最終更新者と更新日時情報 (表 4.2.3) を表示し、各ファイルを最後に更新したユーザーとそのユーザーが同期を行った日時を知ることができる。

4.2.4 ファイルの同期

作業ディレクトリ内の作業したファイルを他ユーザーの作業ディレクトリに反映したり、他人が更新したファイルを自分の作業ディレクトリに反映するためには同期を行う。同期は、大まかにファイル受信 ファイル送信の流れで行う。ファイルの送信は Base64 形式に変換してメール本体に添付して実現する。

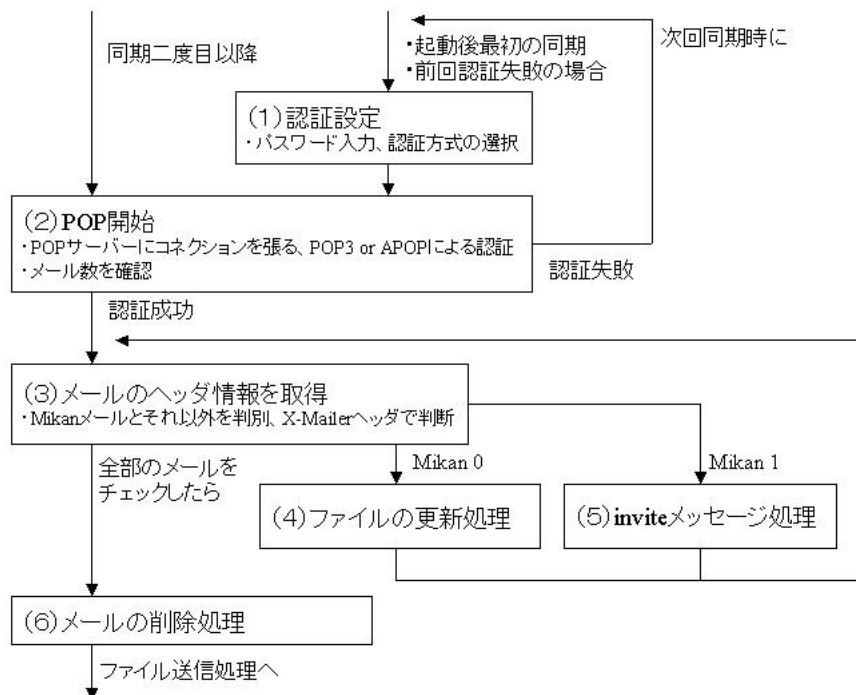


図 4.7: ファイル受信時の処理 (全体の流れ)

ファイル受信

ファイル受信の全体の流れは図 4.7 の通りである。

ファイルを受信する前に、起動して最初の同期実行時や、パスワードが登録されていない場合は、ユーザーにパスワードの入力と認証方式の選択を求める (図 4.7(1))。

Mikan は POP サーバーにコネクションを張り、選択した認証方式に従って認証を行う (図 4.7(2))。認証に失敗した場合は、認証失敗をユーザーに通知し同期を中止する。再度、同期を行った際にパスワードの入力を求める。

認証に成功した場合は、まずサーバー上のメール数を確認し、次にサーバー上にあるメールのヘッダを取得して、Mikan メールとそれ以外のメールを判別する (図 4.7(3))。X-Mailer ヘッダが「Mikan 0」であれば、作業ディレクトリ内のファイルの更新を行う (図 4.7(4))。

X-Mailer ヘッダが「Mikan 1」であれば、図 4.5 の invite メッセージ受信時の処理に従う (図 4.7(5))。

全てのメールヘッダとその処理を確認した後、サーバー上から不要となったメールを削除する (図 4.7(6))。図 4.7(4) のメールは削除、図 4.7(5) のメールは、プロジェクト作成完了時か拒否時に削除される。Mikan 以外のメールはアカウント設定の「Mikan 以外のメールをサーバー上に残す」(表 4.1) の設定に応じて処理する。

ファイルの更新処理 (図 4.7(4)) については、競合検知と解決で後述する。

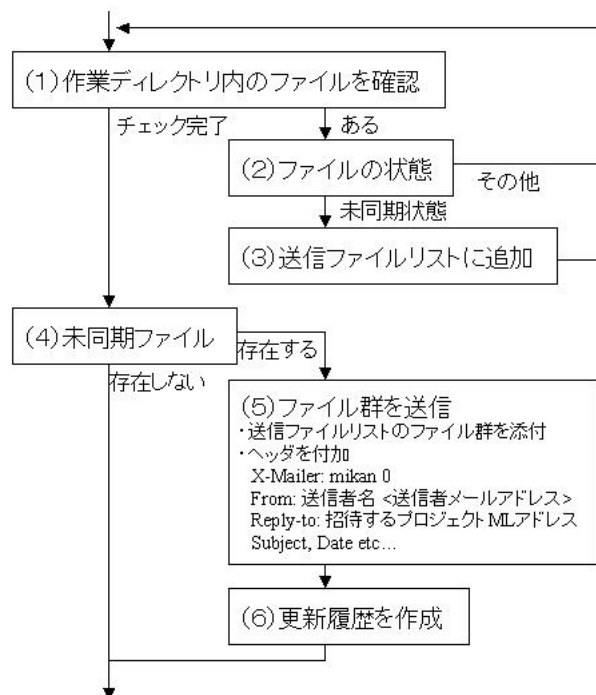


図 4.8: ファイル送信時の処理 (全体の流れ)

ファイル送信

ファイル送信の全体の流れは図 4.8 の通りである。

ファイル群を送信するにあたって、作業ディレクトリ内のファイルの存在を一つずつ確認する (図 4.8(1))。各ファイルの状態を確認し (図 4.8(2))、未同期状態であれば、送信ファイルリストに加える (図 4.8(3))。

作業ディレクトリ内のすべてのファイルを確認し、未同期ファイルが存在しなければ、ファイル送信の処理を終了する (図 4.8(4))。存在した場合は、未同期状態のファイル群を添付し、ヘッダを付加して ML アドレス宛に送信する (図 4.8(5))

このメールがファイル更新メッセージであることを表すために、X-Mailer ヘッダに「Mikan 0」を入れる。From ヘッダには送信者の名前とメールアドレス、Reply-To ヘッダには招待するプロジェクトの ML アドレスをセットする。

最後に送信したファイルに関して更新履歴を作成し、ファイル送信の処理を終了する (図 4.8(6))。更新履歴の作成については 4.2.6 で後述する。

4.2.5 競合検知と解決

競合検知

競合の検知は、ファイルの同期でメールを受信した後、ファイルの更新処理 (図 4.7(4)) 部分で行われる。

ファイルの更新処理は以下の流れで行う。まず、メールをプロジェクトごとに振り分け

を行う。プロジェクトの振り分けは Reply-to ヘッダを見て判断する。Mikan は送信時にプロジェクトの ML アドレスを Reply-to ヘッダに格納している。

次にメールに添付されているファイルをファイル単位で順番に処理する。作業ディレクトリ内に同じファイル名が存在するか否かを確認し、もし存在しなければ、そのまま作業ディレクトリにコピーする。

存在した場合は、そのファイルを現在当ユーザーが作業を行っている未同期状態のファイルであるか否かを確認する。未同期状態であれば競合を検知し、受信したファイルを図 4.9 の形で保存する。競合した未同期ファイルは同期されず、競合発生をユーザーに通知する。

| ファイル名 (1). 拡張子 |

図 4.9: 競合発生時の保存ファイル名

未同期状態でなければ、次に送られてきたファイルの前バージョン番号と、当ユーザーの作業ディレクトリ内のファイルの現バージョン番号を照合し、一致した場合は、上書き保存を行う。

一致しない場合は、MIKANDIR 内の更新履歴を参照して、送られてきたファイルの前バージョンを既に受け取っているか否かを確認し、受け取っているものであれば、競合検知して図 4.9 の形で保存し、ユーザーに競合通知を行う。受け取っていないものであれば上書き保存を行い、後日受け取った際に、更新履歴として保存し、競合が検知されればユーザーに通知する。

競合解決

発生した競合を解決する手段として、競合解決を行うには次の 3 つの選択肢がある。ユーザーはいずれかの対処を選択し、競合を解決してから未同期ファイルを同期する。

1. 一つのファイルに内容を統合する
2. どちらかのファイルを破棄する
3. どちらかを別名ファイルで保存し、共有する

仮に競合したファイル名を `yuuki.txt` 送られてきたファイル名を `yuuki(1).txt` とすると、1 では、まず当ユーザーは `yuuki.txt` に編集した内容を `yuuki(1).txt` にマージし、マージが終わったら競合解決ウィザードで 1 を選択する。Mikan は `yuuki(1).txt` を `yuuki.txt` に上書き保存し `yuuki(1).txt` を削除する。 `.mikan` を書き換え `yuuki.txt` を未同期状態のファイルとして扱う。

2 は、送られてきたファイルと見比べて、自分の編集を破棄する場合に用いる。Mikan は `yuuki(1).txt` を `yuuki.txt` に上書き保存し `yuuki(1).txt` を削除する。1 とは異なり `yuuki.txt` は更新状態のファイルとして扱う。

3 は、まずユーザーに別名のファイル名 (yuuki-hoge.txt 等) を入力させる。Mikan は yuuki.txt を yuuki-hoge.txt として保存し、yuuki(1).txt を yuuki.txt として保存する。yuuki-hoge.txt は未同期状態のファイルとして、yuuki.txt は更新状態のファイルとして扱われる。

4.2.6 更新履歴の管理

Mikan は、同期を行った際に更新履歴を作成する。履歴保存用のディレクトリである MIKANDIR に、過去の版ファイル群と、各ファイルの更新者と更新日時を記録している。

版ファイルは、オリジナルのファイル名の拡張子前に、シリアル番号を付加して保存している (図 4.10)。

| ファイル名 (番号). 拡張子 |

図 4.10: 版ファイルの保存

番号は 1 から始まり、最も大きな番号が与えられているものが最終更新の内容となる。Mikan は番号が付けられたそれぞれのファイルについて、更新者と更新日時を記録する。ユーザーは誰がどのような更新を行ったのかを知ることができる。

第 5 章

ファイル共有システム”Mikan”の実装

本章では, 第 4 章で示した設計を元の実装したファイル共有システム”Mikan”の仕様について述べる.

5.1 実装概要

Mikan の実装環境とシステム構造について述べ, 全体像を示す.

5.1.1 実装環境

実装は, Microsoft Windows オペレーティングシステム環境で, C 言語と Win32 SDK(Software Development Kits) を用いて行なった. コンパイラには Microsoft Visual C++ 6.0 を用いた (表 5.1.1). 本システムは, Microsoft Windows XP/Me/2000/98/NT4.0 環境で動作する.

表 5.1: 本システムの実装環境

OS	Microsoft Windows XP Home Edition Version 2002 Service Pack 1
開発言語	C 言語 + Win32 SDK (Software Development Kits)
コンパイラ	Microsoft Visual C++ 6.0

本システムのメイン画面を図 5.1 に示す. プロジェクト名を切り替えるボックス, 各プロジェクトの作業ディレクトリ内のファイル情報を表示するリストビュー, 作業の経過を表示するステータスバーで構成されている.

5.1.2 システム構造

本システムは大きく分けて, 操作制御機構, 通信制御機構, 設定情報管理機構, ファイル管理機構からなる (表 5.1.2).

操作制御機構は, ユーザーの GUI からの操作, 例えばメニュー操作やプロジェクトの切り替え等を制御し, 他の機構への橋渡しを行なう.



図 5.1: Mikan8.0 Lite

通信制御は、ユーザーが同期あるいは、新規ユーザーへのファイル送信を行なった際に、SMTP/POP3(APOP)を利用して、ファイルの受送信を行なう。

設定情報管理機構は、ユーザーが設定したアカウント情報やプロジェクト情報を、INIファイル等のファイルに登録したり、起動時に読み出す等の設定情報を扱う。

ファイル管理機構は、各プロジェクトにおいて指定された作業ディレクトリ内のファイル群を管理し、未送信ファイル等を判別する。またそれらの情報をリストビュー表示しユーザーに提示する。

表 5.2: 本システムの構造

機構	役割
操作制御	ユーザーのメニュー操作やボックス操作等を制御
通信制御	ファイルの受送信を制御
設定情報管理	アカウント情報やプロジェクト情報を管理
ファイル管理	バージョン管理およびファイル情報の表示

5.2 システムの機能

本システムの機能として、アカウントの設定、プロジェクトの作成、ファイルの作成・更新、ファイルの同期、競合検知と解決、更新履歴の管理について述べる。

5.2.1 アカウントの設定

[設定]-[基本設定]を選択して設定を行う(図 5.2)。

普段利用しているメールアドレスを併用する場合には”「Mikan」以外のメールをサーバー上に残す”をチェックをすることで、POPを行う際にサーバー上に存在するメールヘッダを取得し、フィルタリングを行って Mikan システムのメールのみを取り込むことが可能である。



図 5.2: アカウント設定画面

5.2.2 プロジェクトの作成

プロジェクトの作成は, [プロジェクト]-[新規作成] で行う (図 5.3). 左上のタブを操作することでプロジェクトの切り替えが可能である (図 5.4). プロジェクトを複数作成し, 同時に運用することができる.



図 5.3: プロジェクト設定画面



図 5.4: プロジェクトの切り替え

新規メンバーを追加する際には, [プロジェクト]-[新規メンバーにファイル群を送信] を選択することで, 現状の最新版となるファイルとファイル管理情報を, 新規メンバーに送信することが可能となる. 新規メンバーがこれを受け取ると, プロジェクト参加の確認が行われ (5.5), プロジェクト作成ウィザードに沿ってプロジェクトを作成できる.

5.2.3 ファイルの作成・更新

ファイルの操作は, Windows 付属のエクスプローラ等任意のファイルマネージャーで, ファイルの作成・更新はユーザーが各自のエディタを利用することが可能である (図 5.6). 扱うファイル形式についても特に制限はない.

リストビューの一番右側に「未」が表示されておりユーザーは自分がどのファイルを送信する必要があるのかを把握できる (図 5.7). Mikan はユーザーが更新したファイルのみ



図 5.5: プロジェクト参加の確認

を判断して転送する。

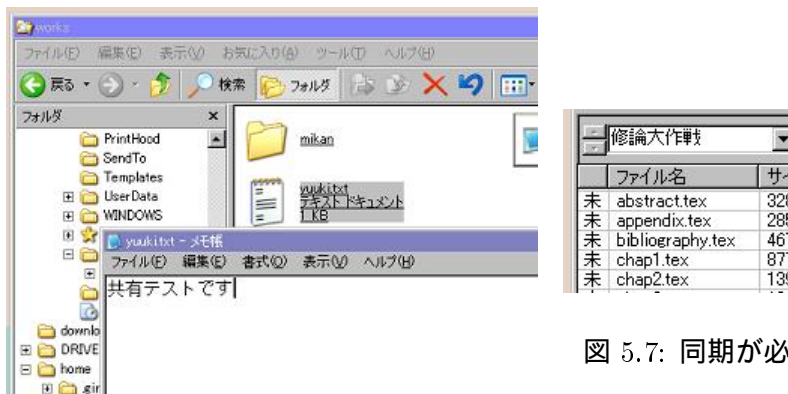


図 5.7: 同期が必要なファイルを提示

図 5.6: ファイルの操作と作成・更新

5.2.4 ファイルの同期

作業ディレクトリ内のファイル群を同期するには [ファイル]-[同期] で行う。また [ファイル]-[定期的に同期を行う] を選択すると、分単位で自動的に同期を行うことが可能である。 [設定]-[同期] で、何分毎に自動的に同期を行うかを設定する。

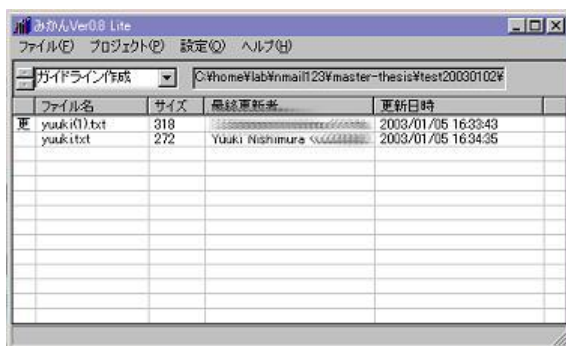
同期を実行してファイルを受信すると、リストビューの一番右側に「更」が表示される (図 5.8)。また [設定]-[同期] では、同期を実行して更新されたファイルがあった時に、ダイアログを出して知らせる設定が可能である。



図 5.8: 同期で他メンバーから受け取ったファイルを提示

5.2.5 競合検知と解決

複数のユーザーが一つのファイルに対して頻繁に更新を行うと競合が発生する。例えば yuuki.txt で競合が発生した際、Mikan は同期して送られてきたファイルを yuuki(1).txt で保存する (図 5.9)。その際、yuuki.txt は同期されず、ユーザーはこれらをマージしてから再度同期を行う。



ファイル名	サイズ	最終更新者	更新日時
yuuki(1).txt	318	XXXXXXXXXXXXXXXXXXXX	2003/01/05 16:33:43
yuuki.txt	272	Yuuki Nishimura <XXXXXXXXXX@XXXXXX>	2003/01/05 16:34:35

図 5.9: 競合発生

5.2.6 更新履歴の管理

各プロジェクトで指定した作業ディレクトリには mikan ディレクトリが作成される。mikan ディレクトリの中には、過去の更新履歴とファイルが残されており、各ファイルの更新履歴の確認や、以前のバージョンに戻すことが可能である (図 5.10)。

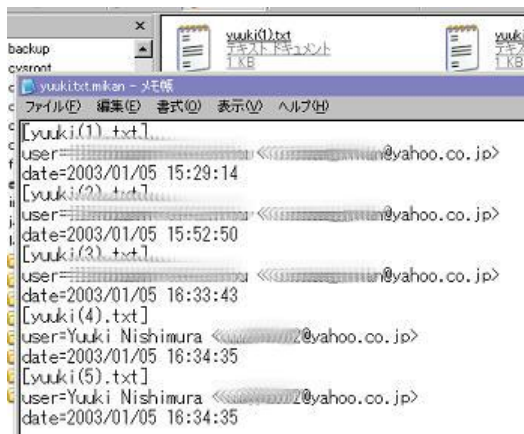


図 5.10: 更新履歴の確認

第 6 章

ファイル共有システムの評価

本章では本システム Mikan が, 3.4 で整理したシステム要件を満たしているか評価する. 評価は, 定性評価として既存のファイル共有システムと, 本システムが満たす要件を比較する. また運用実験を実施して考察を行い, 最後に評価と考察をまとめる.

6.1 定性的評価

本システム Mikan と 2.5 で述べた関連研究を 3.4 で整理したシステム要件を基に比較する. 前述した表 2.2, 表 2.3 に Mikan を追加し, 比較を行った結果を表 6.1, 表 6.2 に示す.

表 6.1: 定性的評価：全体との比較

	バージョン管理	設置・運用コスト	非オンライン環境への対応
電子メール型 (ML)	無	低い	作業できる
共有フォルダ / ファイル型	可能	高い	作業できる
P2P グループウェア型	競合検知は可能	低い	ファイル交換時に制限あり
本研究	可能	低い	作業できる

表 6.2: 定性的評価：独自メールクライアント型との比較

	競合の検知	任意のファイル形式	競合の解決
ESP	無	扱える	無
VCoCoA	可能	独自フォーマットのみ	自動マージのみ
本研究	可能	扱える	3つの選択肢を用意

バージョン管理に関しては, まず本システムはすべての版ファイルと更新履歴を残しており, ユーザーがそれを参照することが可能である. 次にこれを用いて競合検知を行うこ

とが可能である。また、ファイルの同期に受信してから送信するという流れを採用することで、競合問題の影響範囲を押さえ競合解決を容易にしている。ファイルは任意のファイル形式を利用することが可能で、競合解決手段はファイル形式や内容に影響しない解決方法として、3つの選択肢という形で提供している。以上のことから競合検知と解決が可能であると言える。

次に設置・運用コストに関しては、通信モデルにSMTP/POP3(APOP)を利用して、既存のメールサーバーを介してファイルのやり取りを行うため、ユーザーが新規にサーバーの設置・運用することなく利用可能である。ユーザーの識別としてメールアドレスを利用しているが、サーバー上のメールヘッダ情報を取得してフィルタリングを行い、本システムのメールのみを取り込むことが可能である。そのため、ユーザーがフィルタリング可能なメールクライアントを利用していれば、普段利用しているメールアドレスを用いて通信可能である。またグループ形成には無料サービスのMLを利用でき、サーバー管理技術を持たないユーザーでも手軽にグループ管理、メンバー追加を行うことが可能である。以上のことから、設置・運用コストは低いと言える。

最後に非オンライン環境への対応について述べる。本研究では、通信モデルとして電子メール型を選択し、既存のメールサーバーを介して通信を行っている。そのため、メールを取り込むまで、メールサーバー上に更新ファイルをキャッシュすることができ、非オンライン環境で作業を行っているユーザーが更新のためにオンラインになった際に、最新版を共有することが可能である。よって、非オンライン環境への対応は可能であると言える。

6.2 運用実験

本節では、慶應義塾大学大学院政策・メディア研究科特設科目「インターネット時代のセキュリティ管理」において本システムを運用し、その結果を基に考察する。

6.2.1 実験概要

実験の目的、対象、手順について述べる。

実験目的

本実験の目的は、本システムを運用して、各機能が動作しているか、またその使い勝手を知るためのものである。特に、本システムの導入や利用のしやすさ、また本システムの活用法や改善すべき点について考察する。

実験対象

実験は、慶應義塾大学大学院政策・メディア研究科特設科目「インターネット時代のセキュリティ管理」の受講生を対象に行った。

授業は奈良先端技術大学大学院の合同授業で、他にもSoI(School of Internet)[41]の受講者等、遠隔地から参加している受講者が多いことが特徴である。授業時間外は本授業用ML

で議論が活発に行われている。

この授業は、「インターネットに代表される情報通信技術が広く活用でき、市民の生活がより豊かになる可能性があるとともに、上のような問題が同時に発生しているような時代において、セキュリティをどのように捉え、維持していくのか、学生同士が議論することで、自ら考え、行動を起こしていくための契機を提供する」[42]ことを狙いとしている。

授業は、グループワーク形式で行われており、毎回の授業やMLでの議論を基に、受講者全員で一つの信頼形成ガイドラインの作成を行う。

本授業では、学事日程の関係上、2002年12月20日の授業後、2003年1月16日まで授業が行われない。またSoI受講者が積極的に参加可能となるよう、MLでの議論を中心にガイドライン作成を行った。

運用実験はMLでの議論と、ファイル共有手段として本システムを利用し行った。

実験手順

運用実験の手順は次の通りである。

まず2002年12月20日の授業において、ガイドラインを作成するにあたって、ファイルを共有する際に、Mikanを使ってもらえるよう依頼した。

その後、授業用MLで改めて告知し、2003年1月5日に実験用ソフトウェア利用マニュアルを公開した(付録A)。実験参加に承認を得られた受講者を対象に、被験者として実験用のMLをeGroupsに作成し、各被験者にinviteメッセージを送信した。被験者は全部で8名である。

被験者は実験用ソフトウェア利用マニュアルを参照し、Mikanをダウンロードして必要な設定を行った。Mikanを導入する上での疑問点は授業用MLで応答し、2003年1月7日に運用を開始した。

簡単なチュートリアルとして、Mikanの作業ディレクトリにReadMe.txtファイルを置き、被験者が書き込みテストを行えるよう工夫した(図6.1)。

実験用MLにデータを取るためのメールアドレスを登録し、共有ファイルの内容、更新頻度等のログを取り分析した。また、こちらからは特に活用法は提示せず、被験者がMikanをどのように活用するかを観察した。

また2003年1月10日にメールにて個別にアンケート記入紙を送信し、被験者の属性やMikanへの感想・要望等を調査し、7名から返信を得た。

6.2.2 実験結果

まず被験者の特徴について述べ、その後、ログ解析とアンケート調査を基に考察を行う。

被験者の特徴

まず被験者の特徴についてアンケートを基に整理する。アンケート結果は付録B、Cに載せる。

こんにちわ、西村です。

みかんをご利用いただきありがとうございました。

始めは使い方に戸惑ってしまうと思いますので、
まずはこのファイルを利用して、使い方を試してみましょう。

下の落書きスペースに何かメッセージを記入して保存し、
[ファイル(F)]-[同期(R)]を押してみましょう。
記入例：こんにちわ。西村です。よろしく申し上げます。

(中略)

-----以下，落書きスペース-----

こんにちわ。西村です。
もしわからないことがあったらどしどし聞いてください。
よろしく願いいたします。(2003/01/07)

TA の斉藤です。
よろしく願いいたします。
今夜はケーブルTVで「スチュワーズ物語」を観て圧倒されました。
(2003/01/08)

図 6.1: ReadMe.txt ファイルによる簡単なチュートリアル

・履修方式 (問 1-1)

被験者の本授業への履修方式は、SFC からの受講生が 4 人、SOI からの受講生が 3 人であった。遠隔地からの参加者がいるため、ML と本システムを活用して、ネットワークを介した作業を行った。

・デジタルメディアの利用 (問 1-2～問 1-6)

被験者は日常的に電子メールを活用している。チャット (Web 等) やメッセージングについては被験者によって異なる。

・非オンライン環境からの利用 (問 1-7～問 1-9)

被験者は日常的に常時接続環境を利用している。一方で、外出時にはノートパソコンを用いて作業を行ったり、外出先からインターネットに接続して作業を行っている。

ログ解析

次に実験用 ML に流れたメールログを基に、本システムの利用や活用についてまとめる。

2003 年 1 月 8 日 0 時 34 分から 2003 年 1 月 13 日 16 時 10 分までの間に全体で 99 回の同期が行われ、ファイル単位では延べ 130 回の更新が行われた。扱われたファイル形式はテキストファイルが中心であるが、一部 Microsoft PowerPoint のファイルが用いられた。

大まかな共同作業の流れとしては、まず、一つのファイルに対して受講者が文章を追加していくことで、課題の内容として必要な項目を洗い出した。

そのファイルの一部を図 6.2 に示す。

受講者が項目を追加し、授業の TA (Teacher Assistant) が「#」を用いてコメントをつけている。これを参考に受講者たちが必要な項目を追加したり、現在上がっている項目の整理を行った。

<p>顧客との契約</p> <ul style="list-style-type: none">- 申し込み方法 (梶原案) 「名前」「住所」「電話番号」「生年月日」「銀行口座」 「メールアドレス」「携帯番号」の情報をクレジット会社へ送る #TA: 本人であることを証明する情報は公開鍵証明機関に送ればよいのでは?- 顧客の認証 (梶原案) 顧客が携帯電話の支払をきちんとしているかどうかを確認する ドコモの顧客データは、現段階では外部に公開できない。要法改正。 #TA: これは認証ではなく、審査しているのでは?
--

図 6.2: 一つのファイルに対して行われた項目の洗い出し (一部)

その後、大まかな項目ごとにファイルが作成され(表 6.2)、引き続き更新作業が行われた。

表 6.3: 項目ごとのファイル作成

0. 進め方について.txt
1. 登場人物のまとめ.txt
2. 現在の論点集.txt
3. 機能と主体のマッピング案.txt
4. 消費者のクレジット加入申し込み.txt
5. 小売店のクレジット加盟手続き.txt
6. クレジット決済の手続き.txt
7. トラブル時の保障.txt
8. 金融機能の主体の信用形成.txt
9. 認証機能の主体の信用形成.txt
10. その他の主体の信用形成.txt
発表項目と担当者.txt

アンケート

本システムの利用について、アンケートを基に整理する。

・本システムの導入について(問 2-1~問 2-3)
導入や利用のしやすさは被験者によって異なった。今回の実験では説明不足だったこともあり、本システムを導入する前のフリーメールアドレス取得の際に、POP 転送サービス設定でつまづく被験者が何人かいた。

・本システムの活用について(問 2-4~問 2-8)
今回の実験では、本システムを利用することで、作業の進捗状況把握や更新作業がスムーズであったことがわかる。また全ての被験者が電子メールによるデータ交換に比べて便利であったと答えた。

・プロジェクトへの参加について(問 2-9, 問 2-10)
被験者のファイルの更新頻度は被験者によって異なるが、全ての被験者が他人の更新されたファイルを参照していた。

・本システムへの要望と問題点 (問3～問6)

アンケートの自由記述回答結果を付録Cに載せる。

まず本システムを使って良かった点について述べる。多くあげられていたのは、容易にファイル共有が可能である点である。

CVS サーバーを立てるなどの手間がいらず、気軽に簡単な方法でファイルの同期がとれることはすばらしい。

これまでは手作業で行っていたバージョン管理・ファイルのアップロードなどの作業が自動化されており、軽コストで多くの人々と常時最新のファイル共有が行えた。

サーバーを立てなくてよい

また、他の被験者のファイル参照やPCを用いた協調作業に関する意見もあった。

更新後のファイルをリアルタイムで参照できる点がよかったと思う。

また使う際の操作も比較的簡単で、PCを用いた協調作業を容易にできる点がすばらしい。

次に本システムが改善すべき点について述べる。初心者にもわかりやすいマニュアル作り、他OSへの対応、ファイルの削除権限に関する点が多くあげられた。

初心者にも理解しやすいようなインターフェースや、マニュアル作成に力を入れていただければ幸いです。

OSをWindows以外のものでも使えるようにしてほしい(自分はLinuxを使用し、Windowsは借り物で作業を実施したため)。

不要なファイルをupしたときの削除機能または削除希望通知機能が欲しい

6.3 評価と考察

6.1と6.2を踏まえて、バージョン管理、設置・運用コスト、非オンライン環境への対応の3点から、評価と考察を述べる。

6.3.1 バージョン管理

本システムでは、各利用者のローカルなPCの作業ディレクトリ内にMikanディレクトリを構築し、その中にすべての版ファイルと更新履歴を残しており、ユーザーが過去の版ファイルや更新の過程を参照することが可能である。また、作成したファイルを同期させる際には、一旦メールの受信を行って最新の更新状況を確認してから送信を行うという流れにすることで、更新されたファイルを他のメンバーに渡す以前に競合を検知することを可能にした。これによって競合の解決が容易となった。

運用実験では、ログ解析からも多くの更新が行われており、一つのファイルに対して複数のユーザーが更新を行っていた。以上のことから、本システムは履歴管理と競合検知の要件を満たしていると言える。

一方で競合解決手段については、ファイル形式や内容に影響されない解決手段を実現しているが、これはファイル単位の管理であり、具体的な解決プロセス、特に編集内容のマージについては自動で行うことはせず利用者に委ねている。この点については、ファイル形式や内容に応じたマージ手段を提供する等の検討が必要であると考えられる。

6.3.2 設置・運用コスト

本システムでは、通信にSMTP/POP3(APOP)を利用し、既存のメールサーバーを介してファイル共有を行っている。これによりユーザーが新たにサーバーを設置・運用することなく、Mikanクライアントを各自のPCにインストールすることで、ファイル共有を実現できる。NATを間に挟んだ通信も可能である。本システムでは、サーバー上のメールヘッダを取得して、フィルタリングを行い、本システムのメールだけを取り込むことが可能である。これにより、フィルタリング機能を持つメールクライアントを利用しているユーザーは、普段利用しているメールアドレスを利用することが可能である。またグループ形成には無料サービスのMLを利用でき、サーバー管理技術を持たないユーザーでも手軽にグループ管理、メンバー追加を行うことが可能である。アンケート結果でも、サーバーを設置する必要がなく、容易にファイル共有が行える点が記述されている。

ファイルは、ユーザーが任意のエディタやツールを用いて作成・更新可能であり、任意のファイル形式を扱うことができる。運用実験では、テキストファイルの他にMicrosoft PowerPointのpptファイルが共有された。アンケート結果においても、本システムの活用に関して、作業の進捗把握や更新作業がスムーズであり、すべての被験者が電子メールによるデータ交換に比べて便利であったと答えている(問2-8)。以上のことから、本システムが3.4.2で示した要件をほぼ満たしていると言える。

一方で、運用実験でも本システムの改善すべき点として、初心者にも使いやすいインターフェースや不要なファイルを同期したときの削除機能の実現等が、被験者からのアンケート結果で述べられており、これらの点について検討が必要であると考えられる。

6.3.3 非オンライン環境への対応

本システムでは、通信モデルとして電子メール型を選択し、既存のメールサーバーを介して通信を行っている。そのため、メールサーバー上に更新ファイルがキャッシュ可能で、非オンライン環境で作業を行っているユーザーが更新のためにオンラインになった際に、最新版を共有することが可能である。以上より、非オンライン環境への対応は可能であると言える。

一方で、運用実験のアンケート自由記述回答では、複数のPCを利用しているユーザーがおり、このようなユーザーがそれぞれのPCから、本システムを利用して同じプロジェクトに参加するといったニーズが考えられる。この点について検討を行う必要がある。

第 7 章

おわりに

本章では全体を総括し、今後の課題と展望について言及する。

7.1 本研究のまとめ

本研究では、次のような特徴を持つグループを対象に、SMTP を利用したファイル共有システムを設計・実装した。

ユーザー間でファイル共有を行う必要がある共同作業を行っている

作業期間が短く、特定のメンバーで構成された比較的小規模なグループ

非オンライン環境で作業を行うメンバーがいる

このような特徴を満たす共同作業グループは、単年度予算プロジェクトにおける報告書執筆や、大学における論文指導、発表資料の共同作成を始めとして数多くあり、グループの特徴から、バージョン管理、設置・運用コスト、非オンライン環境への対応の 3 点を満たすファイル共有システムが必要となる。

そこで本研究では、電子メールプロトコル SMTP/POP3(APOP) を利用したファイル共有システム Mikan の設計と実装を行った。

Mikan は、既存のメールサーバーを介してファイルのやり取りを行うため、ユーザーが新たにサーバーを設置・運用する必要がなく、また既存のサーバーサービスに更新ファイルがキャッシュできるため、非オンライン環境で作業しているユーザーが更新時にオンラインになった際に最新版を共有できる。

Mikan ではファイルを同期する際に、まずメールの受信をおこなってから競合が起きているかを確認し、その後ファイルを送信するという方法を採用することで、競合発生の影響範囲を抑え、競合解決を容易にしている。また Mikan は、すべての版ファイルと更新履歴を残しており、ユーザーが各ファイルの更新履歴を参照することが可能である。

本研究では評価として、定性的評価と慶應義塾大学大学院政策・メディア研究科特設科目「インターネット時代のセキュリティ管理」において運用実験を行った。その結果、Mikan が必要要件の多くを満たすことを示した。

7.2 今後の課題と展望

本節では、以上を踏まえて今後の課題と展望について述べる。

7.2.1 想定対象の拡張

本研究では、1ヶ月から数ヶ月という短期的な共同作業を行うグループを対象に研究を進めたが、次の要件を満たすことでより多くのグループを対象とすることが可能となる。

複数のPCを利用するユーザーがそれぞれのPCから同じアカウントで同じプロジェクトに参加可能

MLを利用しないグループ形成が可能

ユーザーの中には、例えば企業や自宅といった場所に応じて、利用するPCが異なる場合が考えられる。そこで、同じメールアドレスを利用して複数のPCから同じプロジェクト参加可能な仕組みを用意する必要がある。

また日にち単位のさらに短期間な共同作業を行う際には、ML作成が高いコストになるケースが考えられる。このようなケースではMLを作成しなくてもメンバーリストを作成し、ファイル共有が可能な仕組みを用意する必要がある。

7.2.2 各機能の拡張

運用実験を通じて、新たに必要なシステム機能を設計・実装する必要がある。

まず、ファイル内容のマージ手段の提供があげられる。現在のMikanは編集内容のマージをユーザーに委ねている。ファイル形式や内容に応じたマージ手段を提供していくことで、競合解決をより容易にする必要がある。

次に、不要なファイルを同期したときの削除機能の実現があげられる。本システムでは、送信して他人に共有されたファイルを削除する機能がなく、運用実験では誤ってファイルを同期した被験者がいた。そこで削除機能を用意し、削除が行われた際に、そのことを他ユーザーに通知するアウェアネス機能が必要となる。

また、アンケート結果でも述べられていたが、ユーザーインターフェースを更に改良し、初心者にも理解しやすいインターフェースを提供するために、運用実験と評価を繰り返し行う必要がある。

謝辞

本論文を執筆するにあたって、多くの方々にお世話になりました。

まず、本論文の主査である慶應義塾大学環境情報学部教授 村井純氏に感謝いたします。村井氏は、貴重なアドバイスを頂き、また受け持っておられる授業を運用実験の場として提供して頂きました。有難うございました。

本論文の副査である慶應義塾大学大学院政策・メディア研究科委員長 徳田英幸氏に感謝いたします。徳田氏には、授業「先端研究 A」において励ましの言葉を頂き、また副査を快く引き受けて頂きました。有難うございました。

本論文の副査である慶應義塾大学看護医療学部専任講師 宮川祥子氏に感謝いたします。宮川氏には、論をまとめる上での多くのアイデアの他、研究の進め方について多くのアドバイスを頂き、絶えず励ましの言葉をかけていただきました。本当に有難うございました。

次に、MAUI プロジェクトの諸先生方に感謝いたします。慶應義塾大学環境情報学部助教授 楠本博之氏、同学部助教授 中村修氏には、研究で行き詰った時に、親身になって相談に乗って頂きました。慶應義塾大学環境情報学部専任講師 南政樹氏には日頃のプロジェクト活動で、大変お世話になりました。また、MAUI の先輩方にはミーティングでの確かなアドバイスを頂きました。特に慶應義塾大学大学院政策・メディア研究科博士課程 小川晃通氏には、システムに関して貴重なアイデアを頂きました。本当に有難うございました。

村井研究会で PIE 時代から指導をくださった、国際大学グローバルコミュニケーションセンター助手・研究員 石橋啓一郎氏に感謝いたします。石橋氏には、大学と都内という離れた地理的条件の中、IRC にて数多くの議論とそしてアドバイスを頂きました。また本システムの試験運用に快く応じていただきました。本当に有難うございました。

慶應義塾大学大学院理工学研究科開放環境科学専攻後期博士課程 江木啓訓氏に感謝いたします。江木氏には日頃の研究活動において大変お世話になり、また、本研究のアンケート記入紙の作成に時間を割いて頂きました。本当に有難うございました。

慶應義塾大学大学院政策・メディア研究科教授 金子郁容氏に感謝いたします。金子氏には、Netcom プロジェクトへの聴講参加をお許しいただき、異なった分野からの鋭い指摘とアドバイスを頂きました。有難うございました。

慶應義塾大学大学院政策・メディア研究科博士課程 齊藤賢爾氏に感謝いたします。運用にあたって多くのアドバイスを頂きました。

ご指導いただいた内山映子氏、鳥谷部康晴氏、廣石透氏、林亮氏、また小島元氏に感謝いたします。内山映子氏には、研究で行き詰った時に、親身になって相談に乗って頂きました。

鳥谷部氏と小島元氏には私の拙い英語概要を美しい英文に直していただきました。廣石氏と林氏には、たびたび励ましのお言葉を頂き、また差し入れを頂きました。有難うございました。

国際大学グローバルコミュニケーションセンター主任研究員 土屋大洋氏, 主任研究員 上村圭介氏, 主任研究員 浅野眞氏に感謝いたします。土屋氏, 上村氏, 浅野氏は, お忙しい中インタビュー調査に快く応じていただきました。本当に有難うございました。

本研究の運用実験に参加して下さった, 慶應義塾大学大学院政策・メディア研究科特設科目「インターネット時代のセキュリティ管理」受講者の皆様に感謝いたします。本システムをご利用いただき, アンケート記入をして頂きました。有難うございました。

大学院棟 20 で共に苦闘し励ましあった豊野剛氏, 菅沢延彦氏, 田原裕市郎氏, 若山史郎氏, で共に苦闘し励ましあった海崎良氏, 三川莊子氏, 小川浩司氏, 石原知洋氏と, 普段の研究を共にした neco の皆に楽しい時間を過ごせた事を感謝します。提出前夜, 暖かいトン汁とおいしいお稲荷を用意してくれた rg の皆に感謝します。そして最後に, お正月に帰省できなかったにも関わらず, 暖かく私の研究活動を見守ってくれた両親に感謝いたします。本当に有難う。

2003 年 1 月 14 日 7:00 大学院棟 20 にて

参考文献

- [1] 総務省, “ブロードバンド利用動向調査”, 総務省から三菱総合研究所への委託により実施.
- [2] The Linux Home Page at Linux Online, <http://www.linux.org/>
- [3] IBM Lotus Software, <http://www.lotus.com/>
- [4] Lotus Software - ノーツ/ドミノ R5.0,
http://www-6.ibm.com/jp/domino07/lotus/home.nsf/Content/DP1_Notes_Domino_R50_top
- [5] SourceForge.net, <http://sourceforge.net/>
- [6] サイボウズ Oce 4, <http://cybozu.co.jp/products/cb4/>
- [7] Groove Networks, Inc, Desktop Collaboration Software, <http://www.groove.net/>
- [8] アリエル・ネットワーク (Ariel Networks HomePage), <http://www.arielnetworks.com/>
- [9] 吉田究, “Groove スターターガイド”, オーム社開発局, (2001)
- [10] RCS GNU Project Free Software Foundation (FSF),
<http://www.gnu.org/software/rcs/rcs.html>
- [11] cvshome.org, <http://www.cvshome.org/cyclic/cyclic-pages/sccs.html>
- [12] Microsoft Oce Home Page, <http://www.microsoft.com/oce/>
- [13] cvshome.org, <http://www.cvshome.org/>
- [14] 田淵仁浩, 安部豊子, 中島一彰, 鮎川健一郎, 伊藤文子, 前野和俊, “オフィスアクセラレータ: Web ページ上の文書を任意のパソコンソフトで協同作業できるシステムの提案と実装”, 情報処理学会論文誌 Vol.40 No.11, (1999)
- [15] 緒方広明, 葉田善章, 矢野米雄, “VCoCoA: VCCML を用いた非同期型協同添削支援システム”, 情報処理学会論文誌 Vol.40 No.11, (1999)
- [16] 堀雅和, 加藤康記, 村越広享, 山見太郎, 島津明, 落水浩一郎, “電子メールをベースにした討議型共同文章作成支援システムの設計”, 情報処理学会第 38 回グループウェアとネットワークサービス研究会, (2001)
- [17] 谷川由紀子, 鈴木栄幸, 加藤浩, “情報共有技術を用いた学級新聞共同作成支援システム - 設計思想と評価”, 情報処理学会論文誌 Vol.40 No.11, (1999)

- [18] Bo Leuf, Ward Cunningham, “The Wiki Way: Quick Collaboration on the Web”, Pearson Education, Inc (2001)
- [19] RFC822: Standard for the Format of ARPA Internet Text Messages, <http://www.faqs.org/rfcs/rfc822.html>
- [20] RFC2045: Multipurpose Internet Mail Extensions (MIME) Part One, <http://www.faqs.org/rfcs/rfc2045.html>
- [21] RFC2046: Multipurpose Internet Mail Extensions (MIME) Part Two, <http://www.faqs.org/rfcs/rfc2046.html>
- [22] RFC2047: Multipurpose Internet Mail Extensions (MIME) Part Three, <http://www.faqs.org/rfcs/rfc2047.html>
- [23] RFC2048: Multipurpose Internet Mail Extensions (MIME) Part Four, <http://www.faqs.org/rfcs/rfc2048.html>
- [24] RFC2049: Multipurpose Internet Mail Extensions (MIME) Part Five, <http://www.faqs.org/rfcs/rfc2049.html>
- [25] fml project, <http://www.fml.org/>
- [26] Majordomo, <http://www.greatcircle.com/majordomo/Welcome.html>
- [27] Yahoo! eグループ, <http://www.egroups.co.jp/>
- [28] FreeML, <http://www.freeml.com/>
- [29] QuickML, <http://quickml.com/>
- [30] Eudora, <http://eudora-jp.com/>
- [31] Eudora 5.1-J for Windows ユーザーズマニュアル, <ftp://download.eudora-jp.com/pub/eudora/manual/ewman51.pdf>
- [32] Eudora Sharing Protocol, <http://eudora-jp.com/techsupport/tutorials/win/6.html>
- [33] Ogata, H. and Yano, Y., “CoCoA: Communicative Correction Assisting System for Composition Studies, Proc., ICCE 97, (1997)
- [34] Overview of SGML Resources, <http://www.w3.org/MarkUp/SGML/>
- [35] Ogata, H., Yano, Y. and Wakita, R., “CCML: Exchanging Mark-up Documents in a Networked Writing Classroom”, Computer Assisted Language Learning: An International Journal, Vol.11, No.2, (1999)
- [36] 大月美佳, “入門 CVS”, 秀和システム, (2001)
- [37] WebDAV REsources, <http://www.webdav.org/>

- [38] Net Peers, <http://www.slab.sfc.keio.ac.jp/fukui/itsystem/netpeers.html>
- [39] Becky! Internet Mail, <http://www.rimarts.co.jp/index-j.html>
- [40] Microsoft Outlook Express, <http://www.microsoft.com/windows/ie/default.asp>
- [41] WIDE University - School of Internet, <http://www.soi.wide.ad.jp/>
- [42] インターネット時代のセキュリティ管理 - シラバス,
http://www.soi.wide.ad.jp/class/20020020/materials_for_student/01/soi20020020-plan.pdf

付録

付録として、実験用ソフトウェア利用マニュアル、アンケート選択式集計一覧、アンケート自由記入回答結果一覧、アンケート記入紙を載せる。

付録 A: 実験用ソフトウェア利用マニュアル

(2003/01/10 現在)

みかん version 0.8 Lite for Windows XP/Me/2000/98/NT4.0

みかんは「みんなでかんたんファイル共有」の実現をめざしています。

変更履歴

2003/01/08 FAQ(Frequently Asked Question)のコーナーを設置

2003/01/07 みかん Ver0.8.2 Lite リリース

Ver0.8.1 に含まれるメモリ管理のバグを修正

2003/01/06 みかん Ver0.8.1 Lite リリース

Ver0.8 に含まれる特定条件下で送信できないバグを修正

2003/01/05 みかん Ver0.8 Lite リリース



Contents

1. ソフトウェアの概要
2. 利用するにあたって

3. インストール方法
4. 利用方法
5. FAQ(Frequently Asked Question)
6. 今後の予定とわかっている不具合
7. 連絡先と再配布について

1. ソフトウェアの概要

みかんは、通信に SMTP/POP3 を利用したファイル共有システムです。既存のメールサーバーを介してファイルのやり取りを行ないますので、ユーザーが新たにサーバーを設置・運用する必要はありません。

ユーザーは共有したい任意のフォルダ(ディレクトリ)を指定し、そのフォルダに成果物(ファイル)を置くだけで、グループ全体に共有することができます。

グループでファイル共有をしたいけど、グループの存在期間が短期間なのでサーバー設置・運用はちょっと、といった場合や、メールと同じようにオフラインで作業して送るときだけオンラインといった使い方ができますので、メンバーがオフラインで作業する時間が多い場合に力を発揮します。

2. 利用するにあたって

本ソフトウェアは、各ユーザーを識別するアカウントとしてメールアドレスを利用しています。メールアドレスは普段利用しているメールアドレスを利用する方法と、専用のメールアドレスを利用する方法があります。

・メールアドレスを併用する場合

みかんは、みかんから来たメールだけを判断して取り込むことができます。みかん以外のメールはサーバー上に残すことができるので、他のメールを間違えて取り込んでしまう心配はありません。

一方で、普段利用しているメールクライアント側でみかんのメールを吸い込まないように工夫する必要があります。例えば、Becky!や Outlook Express を利用されている方は、フィルタリングの設定が行なえますので、ヘッダ:「X-Mailer」、マッチする文字:「Mikan」と設定することで、みかんメールをサーバー上に残すことが可能です(確認済)

他の多くのメーラーでも例えば既読のメールを1日後に削除する等の設定をすることによって併用は可能となります。

・専用メールアドレスを利用する場合

普段利用していないメールアドレスがあればそれを専用メールアドレスとして利用できます。また Yahoo!等で無料のメールアドレスが取得できますので、これを利用する手が

あります。

Yahoo!メール：<http://mail.yahoo.co.jp/>

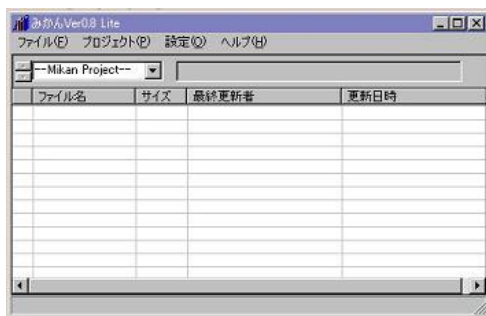
3. インストール方法

・ダウンロード

ソフトウェアは以下よりダウンロードできます。

<http://www.sfc.wide.ad.jp/~yuuki/mikan/>

適当なディレクトリで展開して、mikan.exe を実行すると次のようなアプリケーションが立ち上がります。



・アカウントの設定

まず [設定 (O)]-[アカウント設定 (A)] を選択して、アカウントの設定を行ないます。なお、セキュリティの関係上、パスワードは設定ファイルに保存されません。みかんを立ち上げて最初の同期を行なう際に、パスワードダイアログが表示され、パスワードを打ち込む仕組みになっています。



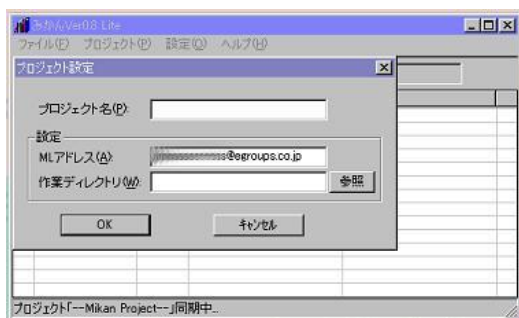
名前 (N):	任意の名前付けます. メールの From に付く名前です.
メールアドレス (A):	アカウントとして利用するメールアドレスを設定します
POP3 サーバー (R):	POP サーバーを設定します
SMTP サーバー (S):	SMTP サーバーを設定します
ユーザ ID(U):	ユーザー ID を設定します.(メールを受信するための ID、メールアドレスの@の前)
パスワード (P):	パスワードを設定します. 保存はできません.
認証方式 (H):	利用するプロバイダに応じて POP3 か APOP を選択します
「みかん」以外のメール	チェックすると「みかん」以外のメールをサーバー上に残します (メールアドレス併用の際はこちらを選択してください), チェックをはずすと全てのメールを取り込みます. 「みかん」以外のメールは削除されます

・プロジェクトに参加する

[ファイル (F)]-[同期 (R)] を選択することで, ファイルを受け取ったり送ったりすることができます. しばらくすると invite メッセージが届きます.



「はい」を選択してプロジェクトを作成してください. すると次のようなプロジェクト設定画面が現われます. 任意のプロジェクトと共有するフォルダ (ディレクトリ) を指定してください. 指定したフォルダ内のファイルは全て共有の対象となりますので, とりあえずはからっぽなフォルダを指定することをお勧めします.



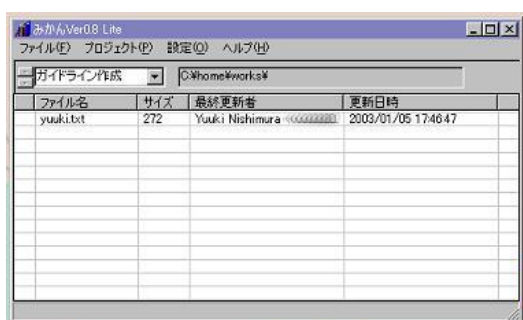
プロジェクト設定を終えると, プロジェクトが作成されています. 左上のタブを操作す

プロジェクト名 (P):	任意の名前をつけます. 例: ガイドライン作成
ML アドレス (A):	みかんはML アドレスで, プロジェクトの識別を行なっています. 今回はいじらないでください
作業ディレクトリ (W):	共有したいフォルダ (ディレクトリ) を指定します. 参照を利用すると指定しやすいです.

ることでプロジェクトを切り替えることができます. (例: ガイドライン作成)



あとは指定した作業ディレクトリに共有したいファイルを置き, 同期ボタンを押すだけでグループ内でファイル共有を行なうことができます.



4. 利用方法

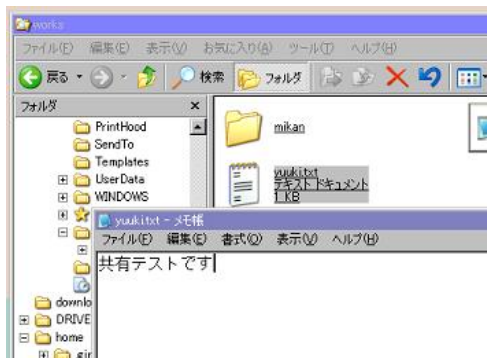
・通常作業

通常の作業時に特に制限はありません. ファイルの操作はエクスプローラ等任意のファイルマネージャ, 文章作成は好きなエディタを利用できます. ファイルはテキスト, バイナリ問わず扱えますが, グループの他メンバーの迷惑にならないよう, サイズが極端に大きなファイルは控えるようにしましょう.

[ファイル (F)]-[同期 (R)] で他のメンバーの更新物の受け取り, 自分が作業したファイルの送信が行なえます. 送信されるのは作業して書き換えたファイルのみです.

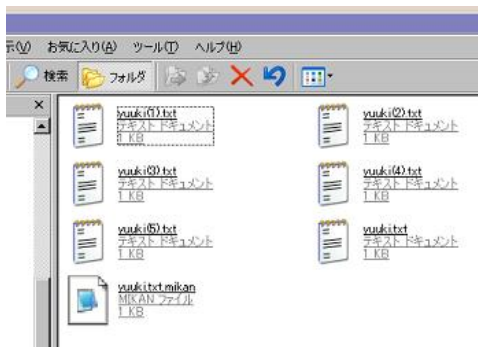
[ファイル (F)]-[定期的に同期を行なう (A)] を選択すると, 一定時間ごとに同期を行なうことができます. 時間は [設定 (O)]-[同期設定 (R)] で指定できます (デフォルトは 10 分になっています).

なお, [定期的に同期を行なう] を選択している際には, システムの関係上, いくつかのメニューが利用できなくなりますが, チェックをはずすことによって再び利用することができます.

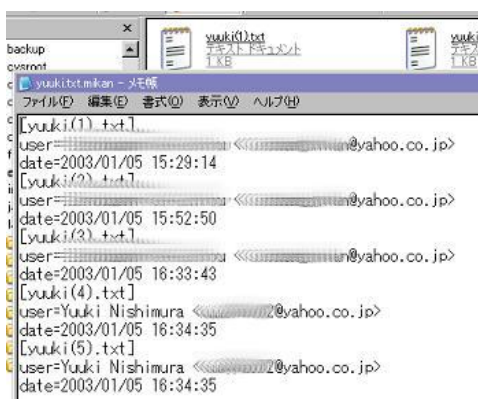


- ・過去の履歴を参照したい

作業フォルダ (ディレクトリ) には.mikan というファイルと, mikan というフォルダが作成されています. (それぞれ重要なものなので削除しないようにお願いします.) mikan フォルダの中には, 過去の履歴が残されています.



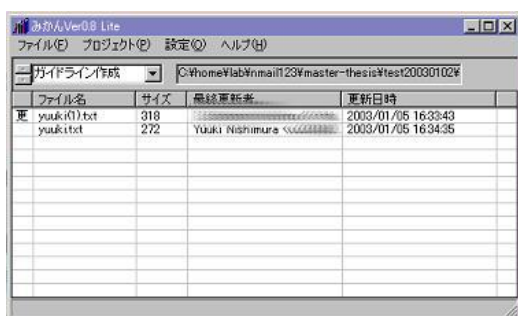
yuuki.txt を例にあげると, yuuki(1).txt が最も古く, yuuki(5).txt が最も新しいファイルになります. また yuuki.txt.mikan には各ファイルの更新者と更新日時が書かれています.



これを眺めることで, ファイルを誰がいつ更新したのか把握することができます.

・複数人で一つのファイルをいじる場合

複数のユーザーが一つのファイルに対して頻繁に更新を行なうと、競合が発生します。競合とは例えば A さんがファイル `yuuki.txt` をいじている時に、B さんがいじた新しい `yuuki.txt` を受け取るケースです。みかんではその場合、B さんのファイルを `yuuki(1).txt` で保存します。



ファイル名	サイズ	最終更新者	更新日時
yuuki(1).txt	318	Yuuki Nishimura	2003/01/05 16:33:43
yuuki.txt	272	Yuuki Nishimura	2003/01/05 16:34:36

競合が発生した場合、A さんがいじた `yuuki.txt` はすぐには共有されません。競合が発生した場合は次のような対処を行なってください。まず [定期的に同期を行なう] にしている場合はチェックをはずします

(1) 同じ名前のファイルで更新を続けたい場合自分が書き換えた `yuuki.txt` の内容を `yuuki(1).txt` に反映して `yuuki.txt` として保存し、`yuuki(1).txt` を削除します。その後、同期を行います。

(2) 別名のファイルで更新を続ける場合 `yuuki.txt` を `yuuki-hogehoge.txt` など別名に置き換え、`yuuki.txt` を削除します。その後、同期を行います。`yuuki(1).txt` も新しい `yuuki.txt` が来次第削除します。

・その他の注意点

今回は、基本的に [プロジェクト (P)]-[設定変更 (O)] で、作業フォルダ (ディレクトリ) の変更は行なわないでください。ファイルの整合性がとれなくなる可能性があります。どうしても変更したい場合は、次のように作業してください。

元作業フォルダ内のすべてのファイルと `mikan` フォルダを新しいフォルダにコピーします。`.mikan` ファイルをメモ帳などで開き、`rcv=0` を全て `rcv=1` に書き換えてください。これを行なわないとみかんは全てのファイルが更新されたファイルであると判断し、全てのファイルを送信してしまいます。

5.FAQ(Frequently Asked Question)

Q1. アカウント設定のユーザ ID というのは何？

ISP からメールを受信するための ID とパスワードのことです。例えば私のヤフーのアカウント (`*****@yahoo.co.jp`) を例に出しますと次のようになります。

名前： Yuuki Nishimura

メールアドレス：*****@yahoo.co.jp
POP サーバー： pop.mail.yahoo.co.jp
SMTP サーバー： smtp.mail.yahoo.co.jp
ユーザー ID: *****
パスワード：ヤフーのパスワード
認証方式：pop3 を選択

Q2. Yahoo!メールを取得したのですが、同期すると認証エラーが出ます。

無料メールアドレスのサービスには大きく分けて、(1)Web メールと呼ばれるものと(2)POP3 を利用するものがあります。

(1)の Web メールとは、メールの内容確認等の操作を全て Web 上で行うもので、基本的に Becky!や Outlook Express といったお手持ちのメールクライアントでメールを扱うことができないサービスです。

Yahoo!メールはデフォルトではこの設定になっています。(今お使いのメールクライアントでは Yahoo のメールを読めないと思います) そこで設定を変更して(2)POP3 サービスを利用できるようにします。

設定変更は次のように行います。

- 1, <http://mail.yahoo.co.jp/>にアクセスしログインする
 - 2, 左側のメニューに「オプション」がありますのでそこをクリックします
 - 3, メール管理という欄に「POP アクセスと転送」がありますので、そこをクリックします
 - 4, あとは Web ページの指示に従って POP アクセスの設定を行ってください
- 広告メールの受信については Yahoo!メールのアカウントを指定されてかまいません。みかんではこれらのメールを無視あるいは読まずに削除できます。

6. 今後の予定とわかっている不具合

みかん Ver0.9 Lite にてサポートする内容

- ・履歴を参照するためのビューアを用意し、人単位や日時単位でソートして見られるようにする。
- ・競合発生時の処理をウィザードを介して操作できるようにする。
- ・複数のプロジェクトを同時に同期させる巡回機能を用意する。
- ・処理メッセージの充実 (更新時のダイアログ表示やタスクバーでの表示等)

長期にわたって実現していきたい事項

- ・ML に依存しないグループ形成とアウェアネス (誰が参加しているのかがわかる) の実現
- ・メールベースの特徴を活かしメッセージ機能の充実 (新しいメールシステムモデルの創造)

現在わかっている不具合

- ・ APOP 認証に失敗する (Ver0.8.2)
- ・ 正しいパスワードを入力しても同期できない (Ver0.8.1) Ver0.8.2 で解決 (2003/01/07)
- ・ 一定の条件下で未送信ファイルが送信されない (Ver0.8) Ver0.8.1 で解決 (2003/01/06)
- ・ 本人の更新が履歴に残らない (上記と関連するバグ)
- ・ ステータスバーの処理内容表示が残ったままになる.

7. 連絡先と再配布について

このソフトウェアについての質問やバグ報告等は yuuki@sfc.wide.ad.jp 宛にお願いします.

このソフトウェアパッケージは, 株式会社軟式様の nMail.dll を利用させていただいております. nMail.dll の著作権は株式会社軟式様 (URL: <http://www.nanshiki.co.jp>) が保持しています.

本ソフトウェアは正式リリース版ではないので, 二次配布は行なわないでください. またご自身の責任でご利用ください.

付録 B: アンケート選択式集計一覧

問 1. あなた自身についておうかがいします

問 1-1 あなたの履修方式はどちらですか

a:SFC での受講生	****	4 人
b:SOI での受講生	***	3 人

問 1-2 あなたは電子メールを一日に何度くらいチェックしますか

a:ほぼ一日中	**	2 人
b:一日に 10 回程度	*	1 人
c:一日に 3 , 4 回	****	4 人
d:一日 1 回程度		0 人
e:一日 1 回未満		0 人

問 1-3 あなたは電子メールを一日に何通くらい書きますか

a:10 通以上	*	1 人
b: 5 ~ 10 通	***	3 人
c:1 ~ 5 通	***	3 人
d:ほとんど書かない		0 人

問 1-4 電子メールを使ってグループワークなどの共同作業を行う事がありますか

a:日常的に行っている	*	1 人
b:2,3ヶ月に 1 回程度行っている	*	1 人
c:これまでに何度か行ったことがある	****	4 人
d:まったく行わない	*	1 人

問 1-5 チャット (Web 等) を利用することがありますか

a:日常的に利用している	*	1人
b:時々利用する	**	2人
c:これまでに何度か利用したことがある	**	2人
d:まったく利用しない	**	2人

問 1-6 メッセンジャー (MSN,ICQ 等) を利用することがありますか

a:日常的に利用している	**	2人
b:時々利用する	*	1人
c:これまでに何度か利用したことがある	**	2人
d:まったく利用しない	**	2人

問 1-7 外出時にノートパソコンで作業をすることがありますか

a:日常的に行っている	*	1人
b:時々行う	***	3人
c:これまでに何度か行ったことがある	*	1人
d:まったく行わない	**	2人

問 1-8 外出先からインターネットに接続して作業をすることがありますか

a:日常的に行っている	**	2人
b:時々行う	**	2人
c:これまでに何度か行ったことがある	***	3人
d:まったく行わない		0人

問 1-9 常時接続環境でインターネットを利用することが多いですか

a:日常的に利用している	*****	7人
b:時々利用する		0人
c:これまでに何度か利用したことがある		0人
d:まったく利用しない		0人

問 2. 今回利用したシステムについておたずねします

問 2-1 実験ツールは利用しやすかった

a: とてもあてはまる	***	3人
b: どちらかといえばあてはまる		0人
c: どちらともいえない	****	4人
d: あまりあてはまらない		0人
e: 全くあてはまらない		0人

問 2-2 実験ツールの導入に違和感があった

a: とてもあてはまる	*	1人
b: どちらかといえばあてはまる		0人
c: どちらともいえない	**	2人
d: あまりあてはまらない	***	3人
e: 全くあてはまらない	*	1人

問 2-3 実験ツールを使うことに違和感があった

a: とてもあてはまる	*	1人
b: どちらかといえばあてはまる	*	1人
c: どちらともいえない		0人
d: あまりあてはまらない	****	4人
e: 全くあてはまらない	*	1人

問 2-4 実験ツールを利用する事で、作業の進捗状況がよくわかった

a: とてもあてはまる	***	3人
b: どちらかといえばあてはまる	***	3人
c: どちらともいえない	*	1人
d: あまりあてはまらない		0人
e: 全くあてはまらない		0人

2-5 実験ツールを利用する事で、他のユーザと協調して作業できた

a:とてもあてはまる	***	3人
b:どちらかといえばあてはまる	**	2人
c:どちらともいえない	**	2人
d:あまりあてはまらない		0人
e:全くあてはまらない		0人

問 2-6 実験ツールは自分の作業スタイルに適していた

a:とてもあてはまる	*	1人
b:どちらかといえばあてはまる	****	4人
c:どちらともいえない	**	2人
d:あまりあてはまらない		0人
e:全くあてはまらない		0人

問 2-7 実験ツールによって更新作業がスムーズであった

a:とてもあてはまる	***	3人
b:どちらかといえばあてはまる	***	3人
c:どちらともいえない	*	1人
d:あまりあてはまらない		0人
e:全くあてはまらない		0人

問 2-8 実験ツールは電子メールによるデータの交換に比べて便利だと感じた

a:とてもあてはまる	***	3人
b:どちらかといえばあてはまる	****	4人
c:どちらともいえない		0人
d:あまりあてはまらない		0人
e:全くあてはまらない		0人

問 2-9 私は、今回の実験でよく自分のファイルを更新した

a:とてもあてはまる	*	1人
b:どちらかといえばあてはまる	**	2人
c:どちらともいえない	***	3人
d:あまりあてはまらない	*	1人
e:全くあてはまらない		0人

問 2-10 私は、よく他人の更新されたファイルを参照した

a:とてもあてはまる	***	3人
b:どちらかといえばあてはまる	**	2人
c:どちらともいえない	**	2人
d:あまりあてはまらない		0人
e:全くあてはまらない		0人

問 2-11 私は、この実験の趣旨を理解していた

a:とてもあてはまる	*	1人
b:どちらかといえばあてはまる	****	4人
c:どちらともいえない	*	1人
d:あまりあてはまらない		0人
e:全くあてはまらない	*	1人

問 2-12 私は、ぜひ今後もこのシステムを使ってみたい

a:とてもあてはまる	**	2人
b:どちらかといえばあてはまる	**	2人
c:どちらともいえない	**	2人
d:あまりあてはまらない	*	1人
e:全くあてはまらない		0人

付録 C: アンケート自由記述回答結果一覧

問 3. 今回利用したシステムについて、使ってよかった点をご自由にお書きください。

更新後のファイルをリアルタイムで参照できる点がよかったと思う。

思ったより簡単にセットアップできたこと。

CVS サーバーを立てるなどの手間がいらず、気軽に簡単な方法でファイルの同期がとれることはすばらしい。

また使う際の操作も比較的簡単で、PC を用いた協調作業を容易にできる点がすばらしい。

これまでは手作業で行っていたバージョン管理・ファイルのアップロードなどの作業が自動化されており、軽コストで多くの人々と常時最新のファイルの共有が行えた。

最終更新者名が表示される

更新があったファイルがどれなのかわかる

サーバを立てなくてよい

問 4. 今回利用したシステムについて、改善すべき点をご自由にお書きください。

OS を Windows 以外のものでも使えるようにしてほしい (自分は Linux を使用し、Windows は借り物で作業を実施したため)。

また、誤ってファイルを送信しないよう、「***のファイルを更新してよろしいですか？」等の確認画面があるとさらに使いやすいと思う。

意図しないファイルをディレクトリに置いて更新してしまうとファイルを回収できない点が改良できれば良いと思います。

概念が簡単に分かる図つきのマニュアルがあったほうが良い。

共有されたファイルのファイル名一覧リストをクリックすると、Windows の関連づけに従って、ファイルを開いて欲しい

APOP への正式対応

最小化でアイコンがタスクバーに入って動作する、更新があったら右下に小さなウィンドウを開いてお知らせする (メッセージャーのように)、といった動作をして、mikan の存在がよりユーザーから自然に隠されるほうがよい。

初心者にも理解しやすいようなインターフェースや、マニュアル作成に力を入れていただければ幸いです。

UP したいファイルと UP したくないファイルを接続前にチェックする欄があれば便利です。吉田さんのように Virtual PC であるためにほかのユーザに気を使われることも少なくなると思います。

不要なファイルを up したときの削除機能または削除希望通知機能が欲しい
ほかの OS の人も使える方式はないのでしょうか？

問 5. 今回の実験について、その他感想やコメント、ご意見等があればご自由にお書きください

メールによるやりとりの場合、すぐに返事が欲しい場合等にどうしても不具合が生じ、また、多人数の場合、1つのファイルを更新するのにもまとめるのが難しい。今回は、その点でファイルを共有することができ、円滑なグループワークに資することができたものとする。

mikan を使いこなすことに時間を取られてしまったような気もしました。専用のディレクトリの意味づけなど、マニュアルの最初のあたりにはっきり書いてあると良いと思います。sfc の方にはそんな方はいらっしゃらないとは思いますが一般人には少しとつきにくいような気もしました。

特になし。

実用性のあるツールだったと思います。今後は本システムを改良し、SOI 受講生の方が更に積極的にグループワークに参加できるよう努めていきたいと思っています。本当にありがとうございました。

自宅で Ver0.8Lite 版を使っています。一度は必ずパスワードが違うというメッセージが出るのでずっとログイン? 状態で使っています。

新しいバージョンもダウンロードしたのですが、接続したままなら現バージョンで困らないのと、exe ファイルだけを上書きするだけでよいのかどうかわからなかったため、そのまま使っています。

このアプリはレジストリ情報等を書き込むのでしょうか？外出先で使用するには、出先の PC 環境を変更したくないのでレジストリ情報等を書き込むのであれば、自分のマシンを持っていくほかはないですね。

できることなら、FD で持ち歩けるほうが嬉しいです。その場合は、FD に過去の履歴文書を残すかどうかのオプションも欲しいです。出先でも最新の情報を見て、自宅で、もう一度履歴をダウンロードできれば嬉しいです。

今回利用し始めるときに難しいと感じたのは、Yahoo の手続きと、デリバーという (Yahoo 独自?) 言葉の意味がわからず、デリバーを認めないとみかんを使えないという Yahoo のシステムを理解できていなかった事が原因です。

みかんの使い方ではパスワードが何に対するパスワードなのかその他ヘルプファイルが充実すると問題はないと思います。

ファイルを共有して作業をする時に他の環境の人も読めるようにメンバーすべてが配慮できるかどうか大切なことだと感じました。

StarSuite の org 版のように Win 上でも Unix 上でも共通で読み書きできるフリーソフトが一般的になればみかんのようなファイル共有ソフトとプレゼンテーションソフトも同時に用いて共同作業ができれば便利になると思います。

ただ、現在普及している統合ソフトには KJ 法のような表現には適していないので、フリーでソースも公開され、商用・非商用を問わず自由に複製・改変・拡張・再配布可能な IdeaFragment <http://member.nifty.ne.jp/nekomimi/> のようなものもあれば(これも、もちろん、いろんな OS の上で動けば嬉しい) 共同作業がしやすいと思います。

付録D: アンケート記入紙

2003年1月10日

ファイル共有システム"みかん"に関する調査」ご協力をお願い

慶應義塾大学 大学院 政策・メディア研究科
修士課程2年 西村 祐貴 (yuuki@sfc.wide.ad.jp)

今回の「インターネット時代のセキュリティ管理」の授業では、信用形成ガイドラインの作成にあたって、ファイル共有システム"みかん"を利用させていただきました。

"みかん"は、既存のメールサーバーを利用してファイルのやり取りを行なっているので、ユーザーが自分でサーバーを用意する必要がありません。例えば、グループの活動期間が短くてサーバーを用意するのは面倒だ、という場合に役立ちます。またオフラインで作業しているメンバーを考慮して開発を進めています。

今回の調査では、実際にご利用いただいて"みかん"に関して、ご意見・ご感想をお聞きしたく、ご利用いただいた皆様を対象として調査を実施いたします。何卒ご協力の程、よろしくお願い致します。

この調査の結果は修士論文「SMTPを利用したファイル共有システムに関する研究」としてまとめさせていただくとともに、今後"みかん"を改良していく上での参考とさせていただきます。論文に関しては、慶應義塾大学 SFC メディアセンター内書架にて閲覧可能となりますので、そちらをご覧ください。

< 回答内容に関するプライバシーについて >

回答の有無や回答内容は成績その他に一切影響いたしません。回答いただいた内容は修士論文の執筆と本システムを改良していく上での参考とさせていただきます。その他の用途には利用いたしません。何卒調査にご協力の程、お願い申し上げます。

実験システム評価シート

問 1. あなた自身についておうかがいします。
選択肢の中からあてはまる項目を選び、
[] の回答枠にアルファベットで記入してください。

1-1 あなたの履修方式はどちらですか []

- a: SFC での受講生
- b: SOI での受講生

1-2 あなたは電子メールを一日に何度くらいチェックしますか []

- a: ほぼ一日中
- b: 一日に 10 回程度
- c: 一日に 3 , 4 回
- d: 一日 1 回程度
- e: 一日 1 回未満

1-3 あなたは電子メールを一日に何通くらい書きますか []

- a: 10 通以上
- b: 5 ~ 10 通
- c: 1 ~ 5 通
- d: ほとんど書かない

1-4 電子メールを使ってグループワークなどの
共同作業を行う事がありますか []

- a: 日常的に行っている
- b: 2 , 3 ヶ月に 1 回程度行っている
- c: これまでに何度か行ったことがある
- d: まったく行わない

1-5 チャット (Web 等) を利用することがありますか []

- a: 日常的に利用している
- b: 時々利用する
- c: これまでに何度か利用したことがある
- d: まったく利用しない

1-6 メッセンジャー (MSN, ICQ 等) を利用することがありますか []

- a: 日常的に利用している
- b: 時々利用する
- c: これまでに何度か利用したことがある
- d: まったく利用しない

1-7 外出時にノートパソコンで作業をすることがありますか []

- a: 日常的に行っている
- b: 時々行う
- c: これまでに何度か行ったことがある
- d: まったく行わない

1-8 外出先からインターネットに接続して作業をすることがありますか []

- a: 日常的に行っている
- b: 時々行う
- c: これまでに何度か行ったことがある
- d: まったく行わない

1-9 常時接続環境でインターネットを利用することが多いですか []

- a: 日常的に利用している
- b: 時々利用する
- c: これまでに何度か利用したことがある
- d: まったく利用しない

問 2. 今回利用したシステムについておたずねします
選択肢の中からあてはまる項目を選び、
[] の回答枠にアルファベットで記入してください。

2-1 実験ツールは利用しやすかった []

- a: とてもあてはまる
- b: どちらかといえばあてはまる
- c: どちらともいえない
- d: あまりあてはまらない
- e: 全くあてはまらない

2-2 実験ツールの導入に違和感があった []

- a: とてもあてはまる
- b: どちらかといえばあてはまる
- c: どちらともいえない
- d: あまりあてはまらない
- e: 全くあてはまらない

2-3 実験ツールを使うことに違和感があった []

- a: とてもあてはまる
- b: どちらかといえばあてはまる
- c: どちらともいえない
- d: あまりあてはまらない
- e: 全くあてはまらない

2-4 実験ツールを利用する事で、
作業の進捗状況がよくわかった []

- a: とてもあてはまる
- b: どちらかといえばあてはまる
- c: どちらともいえない
- d: あまりあてはまらない
- e: 全くあてはまらない

2-5 実験ツールを利用する事で、
他のユーザと協調して作業できた []

- a: とてもあてはまる
- b: どちらかといえばあてはまる
- c: どちらともいえない
- d: あまりあてはまらない
- e: 全くあてはまらない

2-6 実験ツールは自分の作業スタイルに適していた []

- a: とてもあてはまる
- b: どちらかといえばあてはまる
- c: どちらともいえない
- d: あまりあてはまらない
- e: 全くあてはまらない

2-7 実験ツールによって更新作業がスムーズであった []

- a: とてもあてはまる
- b: どちらかといえばあてはまる
- c: どちらともいえない
- d: あまりあてはまらない
- e: 全くあてはまらない

2-8 実験ツールは電子メールによるデータの交換に比べて
便利だと感じた []

- a: とてもあてはまる
- b: どちらかといえばあてはまる
- c: どちらともいえない

- d: あまりあてはまらない
- e: 全くあてはまらない

2-9 私は、今回の実験でよく自分のファイルを更新した []

- a: とてもあてはまる
- b: どちらかといえばあてはまる
- c: どちらともいえない
- d: あまりあてはまらない
- e: 全くあてはまらない

2-10 私は、よく他人の更新されたファイルを参照した []

- a: とてもあてはまる
- b: どちらかといえばあてはまる
- c: どちらともいえない
- d: あまりあてはまらない
- e: 全くあてはまらない

2-11 私は、この実験の趣旨を理解していた []

- a: とてもあてはまる
- b: どちらかといえばあてはまる
- c: どちらともいえない
- d: あまりあてはまらない
- e: 全くあてはまらない

2-12 私は、ぜひ今後もこのシステムを使ってみたい []

- a: とてもあてはまる
- b: どちらかといえばあてはまる
- c: どちらともいえない
- d: あまりあてはまらない
- e: 全くあてはまらない

問3. 今回利用したシステムについて、
使ってよかった点をご自由にお書きください。

問 4. 今回利用したシステムについて、
改善すべき点をご自由にお書きください。

問 5. 今回の実験について、その他感想やコメント、
ご意見等があればご自由にお書きください

=====

ご協力ありがとうございました。