

## バージョン管理システムを利用したグループ内ファイル共有システム

福山 悠<sup>†</sup> 福田 洋治<sup>‡</sup> 毛利 公美<sup>††</sup> 白石 善明<sup>†</sup>名古屋工業大学<sup>†</sup> 愛知教育大学<sup>‡</sup> 岐阜大学<sup>††</sup>

## 1. はじめに

グループのメンバと業務のノウハウを始めとする知識を共有することや、共有した知識を継承し、さらに有益な知識を創造する手段の一つとしてグループでのファイル共有がある。知識の共有、継承、創造を目的とした図 1 に示すようなファイル共有システムには、次のことが必要と考えられる。

- グループの柔軟な構築
- 効率のよいファイルの共有・配布・公開
- 手軽な操作
- 履歴の管理

1 点目および 2 点目は、部署内のみを想定するのではなく、創造的な仕事を部署や所属組織を超えて協業する場合に必要である。3 点目は知識の共有や継承を円滑に遂行するため、4 点目はどのように知識が形式化されていくかの過程をも共有するために必要である。

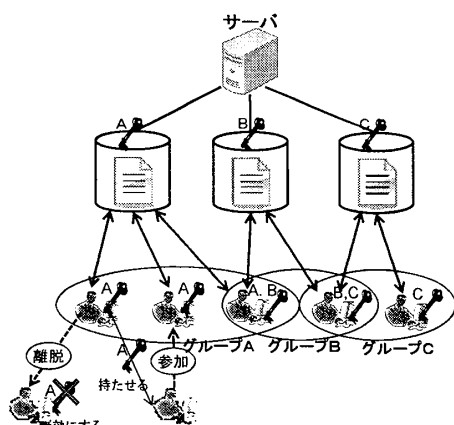


図 1: 柔軟なグループが形成できるファイル共有システム

単なるファイル共有をするだけであれば、電子メールやインスタントメッセンジャー等で共有相手にファイルを送信することで十分かもしれないが、グループのメンバが多くなると非効率になる。ファイルサーバやグループウェアでは組織内で知識を共有、継承することに対応できるものがあるが、組織を超えた運用を効率的に行うには困難を伴う。

そこで、本研究ではインターネット上で組織を超えてソフトウェアを開発する際に広く用いられているバージョン管理システムを利用することを考え、特に subversion[1]とグループファイル共有のクライアントソフトウェアを連携させ、知識の共有、継承を効率的に行うことができるシステムの構築を目標とする。

## 2. バージョン管理システムとファイル管理モデル

オープンソースの開発等に利用されるバージョン管理システムでは、メンバが共有するファイルの削除や編集したファイルの配布を行う場合、その承諾を最終的に判断する責任者（コミッター）が存在する。本稿では、この形態をコミッター存在モデルと呼ぶ。正確な知識を共有、継承するためには、コミッターにより精査された後にファイルの配布が行われるコミッター存在モデルが望ましいという考え方がある。

一方で、コミッターの承認処理が多くなると共有されるべきファイルの配布遅延や、欠損の生じる可能性がある。また、知識の共有、継承をするためには、ファイルの更新過程の中にも暗黙知が存在すると考えられ、多くのファイルが共有されることが望ましいという考え方もある。以上の二点から、できるだけ多くのファイルを生成し、共有するためには、メンバは同じ編集や削除の権限を持つ関係がよいと考えられる。以下ではこの形態をフラット権限モデルと呼ぶ。

知識が形式化されるに至る過程の共有もグループにとって重要であり、あるファイルが完成していく中では、削除されたファイルも何らかの価値を持っていると考えられる。インターネット上で比較的容易に利用でき、すべてのファイルをサーバに保管できるという特徴を有するバージョン管理システムを知識の共有や継承のためのグループファイル共有に導入するためには、コミッター存在モデルからフラット権限モデルへの変更が課題となる。

## 3. 変換機構と提案システム

新規に作成したファイルをメンバと共有するのはフラット権限モデルであっても難しくはない。しかし、共有したファイルを二人以上のメンバで同時に編集し、あるメンバがファイルを更新処理した後に別のメンバが更新処理をすれば、フラット

A Group File Sharing System Using Version Management System

<sup>†</sup>Yuu FUKUYAMA and Yoshiaki SHIRAIISHI • Nagoya Institute of Technology

<sup>‡</sup>Youji FUKUTA • Aichi University of Education

<sup>††</sup>Masami MOHRI • Gifu University

ト権限モデルでは衝突が生じる。コミッター存在モデルではいずれの更新も反映するか、どれかを承認するといった調停を行うことが前提であり衝突は起こらない。つまり、フラット権限モデルではこのような衝突を回避するように、何らかの機構を導入する必要がある。また、ファイルの削除も実際には保管しているサーバからの削除はせず、メンバのクライアント上だけで削除されたかのように見せるためには同じく何らかの機構を導入する必要がある。そこで、図 2 に示すようなバージョン管理システムでファイルを管理しているサーバに対する既存の命令を発行する前に、メンバのクライアントソフトウェアが命令変換する機構を導入する。

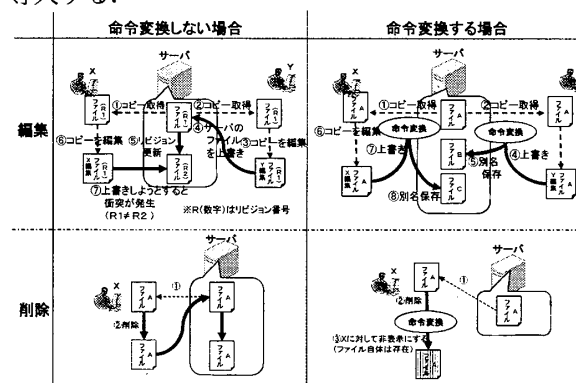


図 2: 命令変換の有無によるフラット権限モデルでの編集/削除の動作

ファイルの編集は、同一ファイル名でサーバに登録しようとするときに、クライアント側にある変換機構が別名で保存をするようユーザに促し、同一ファイル名で上書きしてしまうことのないようにする。ファイルの削除は、サーバに格納されているファイルを削除するのではなく、ユーザの削除命令を受けた変換機構がローカルのフォルダのファイル一覧から表示させないように処理を行う。

この変換機構を導入した提案システムの構成が図 3 のようになる。ユーザのクライアントソフトウェアは、ユーザ認証を行うユーザ情報処理部、所属グループに関連する処理を行うグループ情報処理部、命令変換機構を有するグループフォルダ管理部、サーバに登録し共有するファイルを暗号化する暗号処理部、他のメンバから受け取ったファイルの復号鍵をユーザ情報処理部と連携して取り出す復号鍵処理部、そして、ユーザに GUI を提供する画面管理部から構成される。サーバソフトウェアは、ユーザ情報管理部、グループ情報管理部、およびバージョン管理システムからなる。本研究ではバージョン管理システムに subversion を利用している。

クライアントとサーバは Java 言語をベースに、

クライアントは GUI を Swing で、subversion のサーバとは SVNKit [2] ライブラリを用いて連携するように実装している。

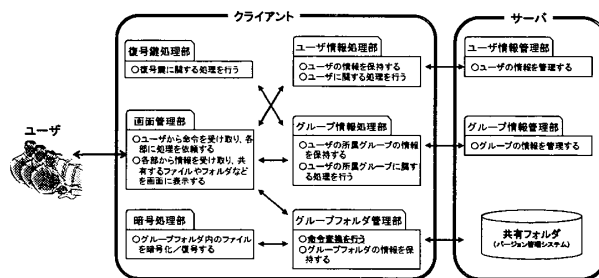


図 3: 提案システムの構成

表 1: ファイルサーバと提案システムのファイル共有に関する比較

運用形態	ファイルサーバ	提案システム
管理モデル	コミッター存在	フラット権限
ファイルの編集	サーバのファイルが上書きされる可能性がある	別名でアップロードするために元のファイルが無くなることはない
ファイルの削除	いつ、誰が削除したか確認困難、また、削除されたことに気づかないこともある	サーバからは削除されないでファイルが失われることはない
グループの範囲	サーバのあるネットワーク内だけで構築できる	組織を超えて構築できる

#### 4. 評価とまとめ

ファイル共有の一般的な方法である Windows や Linux でファイルサーバを構築した場合と提案システムのそれぞれの特徴を表 1 に示す。提案システムでは知識の共有、継承を行うために、ファイルの更新過程がトレースでき、そしてファイルを削除できないように変換機構を導入することで、フラット権限モデルへの変更が可能となった。また、バージョン管理システムを導入することで、一般的な組織で利用できる http をベースにファイル共有ができ、組織を超えた協業ができる。

今後は、グループ構築が柔軟かつユーザの鍵管理が容易な暗号化ファイル共有方式 [3] の実装を行う予定である。

#### 参考文献

- [1] <http://subversion.tigris.org/>
- [2] <http://svnkit.com/>
- [3] 内田, 福田, 毛利, 白石, “単一の鍵で多重帰属できる繰り返し型(2,2)閾値秘密分散を用いたグループファイル共有”, 電子情報通信学会 2009 総合大会 (2009 年 3 月)