

WWWとファイルシステムの双方向リンクによる共同作業支援

古井, 陽之助

九州大学大学院システム情報科学府知能システム学専攻 : 博士後期課程

松永, 勝也

九州大学大学院システム情報科学研究所知能システム学部門

志堂寺, 和則

九州大学大学院システム情報科学研究所知能システム学部門

<https://doi.org/10.15017/1515816>

出版情報 : 九州大学大学院システム情報科学紀要. 8 (1), pp.43-48, 2003-03-26. 九州大学大学院システム情報科学研究所

バージョン :

権利関係 :

WWWとファイルシステムの双方向リンクによる共同作業支援

古井陽之助*・松永勝也**・志堂寺和則**

Collaboration Support by Two-Way Linking between WWW and File System

Yunosuke FURUI, Katsuya MATSUNAGA and Kazunori SHIDOJI

(Received December 13, 2002)

Abstract: This paper describes a new way of computer-mediated collaboration based on two-way linking between discussion and file sharing. We have been developing a system named "Pilgrim" that links a discussion service and a file-sharing service to each other. The system allows users to have discussions on any type of files; it also helps users to find the discussions easily when they browse the files.

Keywords: Collaboration, Discussion base, Two-way linking, Backlink, Awareness

1. はじめに

著者らは、World-Wide Web (WWW, 2.1節を参照)¹⁾ベースの共同作業システム "Pilgrim" を開発してきた。本システムは小～中規模のグループが利用できるクローズドな議論ベースや共有ファイルへのリモートアクセスなどを実現することを意図したシステムであり、具体的には以下のサービスを提供する。

(1) 議論ベース: Pilgrimにアカウントを持つユーザ同士がWWW上でテキストベースのメッセージを非同期に交換できる。これによって各参加者が持つ情報や意見を交換しつつ共同作業を進めることができる。

(2) ファイルアクセス: ユーザはファイルサーバに蓄積されたファイルにWWW経由で読み出し専用アクセスできる。これによって各ファイルは固有のアドレス (URL, 2.1節を参照) を持つ。Pilgrimのアカウント管理機能の保護下で、クローズドなグループのための共有を支援する。

(3) 議論とファイルの双方向リンク: ファイルについて議論する場合、議論ベースの中のメッセージとその議論対象になっているファイルとの間に双方向のリンクを張る。この双方向リンクは、(2)のWWWによるアクセスのみならず、ローカルエリアネットワーク (LAN)内でのSMB (2.2節を参照) によるファイル共有の場合にも提示される。これによって、グループ内でのファイルに関する議論を支援する。ユーザはファイルにコメントを付けたり、また他のユーザのコメントを見つけて読んだり、返信を書いたりすることができる。上に述べた3つのサービスのうち、本論文では(3)についてその仕組みや利点などを論じる。

本論文の以降の構成は、まず2節では本研究の背景となった基盤技術について説明する。3節では本研究が扱う

課題とその解決策を説明する。4節では解決策の具体的な実現方法を説明する。また将来の課題を述べる。5節で本論文をまとめる。

2. 背景

2.1 World-Wide Web

1990年代から発達してきたWorld-Wide Web (WWW)¹⁾は、今日では広域ネットワークでサービスを実現する上で欠かすことのできない、最も重要な情報基盤の一つとなっている。WWWは、その基本的な使い方においては、読み出し専用のファイルアクセスを提供する。WWWを介してアクセスできるファイルなどのリソースは、サーバ名やファイルのパス名などに基づいてそれぞれ固有のアドレスを持つ。このようなアドレスをUniversal Resource Locator (URL)と呼ぶ。

ユーザはWebブラウザと呼ばれるソフトウェアを使って、URLによって指定される適切なWWWサーバの適切なパスなどからHTMLファイルを読み出し、このファイルに基づいて表示されるWebページを閲覧する。これらWebページは、他のWebページと基本的に片方向のハイパーリンクで結び付き合っており、名前の通り世界規模のハイパーテキストを実現する。

WWWに基づいて様々なサービスが提供されているが、最も広く普及したものの一つが議論ベース (discussion base) または電子掲示板 (Bulletin Board System, BBS) である。このサービスを使うと、ユーザはネットワーク上に実装された仮想的な掲示板で、主にテキストからなるメッセージを非同期的に交換し、意見や情報を共有することができる。典型的な議論ベースには、メッセージ内にURLを書く (貼る) と、メッセージ表示の際にはそのURLをクリック可能なハイパーリンクとして表示する機能を持つものが多い。ユーザはこの機能を利用して、メッセージ中にWWWのハイパーリンクを含めることがで

平成14年12月13日受付

* 知能システム学専攻博士後期課程

** 知能システム学部門

きる。

2.2 ファイル共有

今日において、ファイルシステムは計算機システムの基本的な要素である。各種アプリケーションはファイルシステムの存在を前提に作成されている。またファイルに対する様々な操作（作成、移動、削除、名前変更、一覧表示、一覧並べ替え、検索など）を行うためのユーティリティも充実している。これらのソフトウェアを使って、ユーザはファイルシステムに知的活動の過程や成果をファイルとして蓄積していく。

従って、ネットワークを介してファイルシステムを共有することができれば、それはオンラインの共同作業において有用な道具となる。ユーザは、共有資源たるファイル（共有ファイル）に対して、通常各種ソフトウェアをそのまま使って作業することができる。実際に、著者らの研究室においては、日々の研究活動の過程において作成・更新する様々なファイルを蓄積し共有するために共有ファイルサーバを使用している。

ネットワークでファイル共有を実現するための代表的なサーバソフトウェアの一つにSamba²⁾がある。SambaはGNU Public Licence (GPL)に基づいて配布されるオープンソースのソフトウェアであり、LinuxなどのUnix系OS上で動作する。

Sambaが使用するプロトコルServer Message Block (SMB)は、別名Common Internet File System (CIFS)とも呼ばれ、もともとはMicrosoft社製オペレーティングシステム（Windowsなど）で稼動するサーバやクライアントの間で標準的に使われるプロトコルである。SMBはLANにおいて、ローカルファイルにアクセスするのと同じような読み書き可能なファイルアクセスを提供できる。Microsoft WindowsはSMBを標準でサポートするので、Windowsベースのシステム同士は容易にLAN内でファイルやプリンタを共有することができる。

Unix系OSとSambaサーバ、WindowsベースのPCクライアントで、LANでファイル共有をするという利用形態は、手軽さや低コストの魅力もあって広く普及している。

2.3 インターネットショートカット

インターネットショートカットとは、Windowsがデフォルトで定義するファイル形式の一つである。ファイルの中には以下のような書式でURLが記述される。

[InternetShortcut]

URL=http://rainbow.edu/pilgrim/view/fls/doc/

他にもアイコン画像の指定などを書くことができるが、以上の書式のみでもショートカットとしては動作する。このファイルをダブルクリック（Windowsの設定によってはシングルクリック）でオープンすると、Webブラウザが

起動し当該URLをロードする。このファイルの拡張子はurlである。但しこの拡張子は、WindowsエクスプローラなどWindows標準のシェルでは（例えエクスプローラの設定で「登録されているファイルの拡張子を表示しない」を外して全ての拡張子を表示するようにしても）表示されない。

インターネットショートカットを使うと、PCでの作業中にその環境をファイルシステムからWWWへ直接的に遷移させることができる。従ってインターネットショートカットは、ファイルシステムとWWWを統合するシステムを構築するには有用な仕組みである。

2.4 Awareness

Awareness³⁾⁴⁾はオンラインの共同作業において重要な概念の一つである。その定義は「互いの存在や行動を認識し易くすることによってユーザ間のコミュニケーションを支援する技術」である。

オンラインの共同作業では、ユーザ同士は直接対面しているわけではない。互いの存在・行動・態度・気分・興味などは、直接対面であれば容易に気付き得るものだが、オンラインでの非対面による共同作業ではそれに気付くためには計算機の支援が必要である。これを実現するのがawarenessである。

具体例としてはYahoo!メッセージのログイン通知機能が挙げられる。このユーティリティを使うと、メッセージ交換したい相手がつながっているかどうか、いつオンラインにつながったかについてユーザに通知されるようになる。

2.2節で述べたSambaのようなファイル共有システムの場合、自分以外の誰がどのファイルについて何をしているかについて知る手がかりは、ファイル自体の更新状況で知るしかない。このようなシステムの場合は、ファイル共有に関するawarenessが十分に提供されているとは言えない、ということになる。

3. 課題と検討内容

3.1 課題

1節で述べたPilgrimのサービス(1)(2)はWWWベースで実装されており、広域ネットワークに対してサービスすることができる。一方、SMBベースのような共有ファイルシステムは、LANに対してサービスするのに向いている。また、(2)は読み出し専用アクセスを提供するのに対して、共有ファイルサービスでは読み書き可能アクセスを提供できる。従って(1)(2)はファイル共有サービスを実現する上で相補的である。

さて、(1)の議論ベースを使ってメッセージを交換することによって、共有ファイルについて議論する場合について考える。その議論は議論対象であるファイル自身のあり

かとは別のところで行われている。1節で述べた(1)(2)を使って、メッセージ中にファイルのURLを貼れば、メッセージからファイルのありかをハイパーリンクによって参照することはできる。しかし、逆にファイルからそのファイルに関する議論を辿ることができない。

この場合の問題を整理すると以下の通りである。

[問題A] 議論ベースとファイルシステムという2つの作業環境の行き来に関して、前者から後者への遷移はハイパーリンクによって支援されるが、逆は支援されない。まずファイルありきの議論においては、これはユーザビリティ上問題である。

[問題B] 議論ベースだけを見ているファイルの存在に気付くことができるが、ファイルシステムだけを見ていると議論の存在に気付かない恐れがある。これは前節で述べたawarenessに関して問題がある。

3.2 検討

これらの問題に対する解決策として、議論からファイルへのハイパーリンクが存在するとき、ファイルから議論への逆リンクを自動的に生成する機能を考えた。この機能は、1節で述べた本システムのサービス(2)においては、ファイルを表示するWebページ中にハイパーリンクとして逆リンクを挿入することで実現できる。しかし、2.2節で述べたSMBプロトコルでファイルを共有しているような場合には、本来そこにはハイパーリンクを提示するための仕組みが無いので、新たに何らかの工夫が必要である。

この逆リンクを具体的な方法として以下の2通りを検討した。

[解決方法1]

逆リンクをファイルとして生成・表示する。

Windowsエクスプローラなどを使ってユーザがファイル一覧を閲覧するとき、その中に議論からリンクされたファイルがあれば、逆リンクの役割を果たすファイルをもそのファイル一覧に含めるようにする。しかしこのファイルは実体を持たない、共有ファイルサーバが動的に出力する仮想的なファイルである。

ユーザはこの逆リンクファイルを見ることで、当該ファイル一覧の中に議論対象となっているファイルがあることを知り得る。また、そのファイルの中身を閲覧また実行することで、逆リンクを辿って議論を参照したり議論に参加することができる。

[解決方法2]

逆リンクを提示するための専用ソフトウェアを作成する。これを各ユーザのPCにインストールする。

ユーザがファイルを閲覧するとき、閲覧対象のこのファイルが議論からリンクされていれば、この専用ソフトウェアは逆リンクを提示する。提示の仕方としては、ファイルのそばに小さなウィンドウでバブル形式で表示する、画

面の隅にアイコン形式で表示する、ファイル閲覧ソフト(Windowsエクスプローラなど)のメニューバーにメニュー項目またはアイコンを表示する、通知音を鳴らす、などが考えられる。ユーザはこれらのウィンドウやアイコンやメニューに対してクリックなどで反応することによって、リンク元の掲示板文書に到達することができる。

3.3 インターネットショートカットによる解決

本研究では[解決方法1]を選択し、実際には存在しない仮想的なファイルを提示することにした。また、逆リンクとして提示するファイルには、2.3節で述べたインターネットショートカットを採用した。この方式の長所と短所について検討したので以下に述べる。

[長所a] インターネットショートカットはWindows PCであればサポートされているので、ユーザは新規に専用ソフトウェアをインストールする必要が無く、既存の計算機環境で使用可能である。

[長所b] 見かけ上は只のファイルなので、通常のファイル操作によって操作が可能である。また、ユーザは慣れ親しんだユーザインタフェースを使って逆リンクにアクセスすることができる。例えばWindowsエクスプローラのユーザインタフェースを使って、タイムスタンプや名前などを使ってファイル一覧を並び替えたり、ファイル検索機能を使ったりして、逆リンクを探索することができる。

[長所c] 通常のファイル操作によって複製・保存することもまた可能である。例えば、ディレクトリごと一群のファイルを複製すれば、その中にある仮想的に生成された逆リンクも、元の議論対象のファイルと一緒に複製されることになる。共有ファイルサーバに接続できないLAN外などの環境からでも、予め複製しておいた逆リンクを使って議論を辿ることが可能になる。また、逆リンクを他のファイルと一緒に圧縮形式の書庫ファイルなどに保管しておき、後で必要になったときに取り出して当時の議論を辿るためのきっかけとして使用することもできる。

この仮想ショートカットファイルの名前は、逆リンクの元になるファイルに拡張子.urlを付けたものにする。よって、ファイル一覧を名前順に並べたときには元のファイルと逆リンクとは並んで表示されることになる。Windowsエクスプローラは拡張子.urlは表示しないので、ユーザにとって見た目は元のファイルと同じ名前である。ファイルを名前順で並べれば、元のファイルとその逆リンクは隣同士に表示されることになる。

この仮想ショートカットファイルのタイムスタンプは、元のファイルにリンクしている最新のメッセージと同じにする。よって、ユーザは逆リンクのタイムスタンプを見れば、議論ベースでそのファイルが最後に言及された日時を知ることができる。

逆リンクは、元のファイルに手を加えることなく、それ

とは別のファイルとして提示される。従って、元のファイルの実体がいかなるものであるか、ファイルタイプが何であるかを問わない。

以上のようにして、ファイルに関する議論を支援する仕組みとして議論ベースとファイルシステムの双方向リンク機能を提供する。この双方向リンクによって、議論と共有ファイルを接続・統合し、密なコラボレーションの基礎となる使用環境を構築することを意図した。

[短所a] 元々逆リンクを表示するための仕組みを持たない既存の情報基盤を活用する方式なので、逆リンクをどのようにして表示するか、どのような属性を持たせるかといった点では選択肢は多くない。このような自由度の低さは、**[長所a]**と表裏の関係にある。

[短所b] Windowsエクスプローラでファイル一覧を表示すると、その中にあるディレクトリとファイルは一覧の上と下に分けられてしまう。そのため、議論ベース側でディレクトリへリンクしている場合に、その逆リンクとして現れるインターネットショートカットは、名前順で並べても元のディレクトリの隣には現れない。これは**[短所a]**の自由度の低さに関連するものである。この短所は、当該ディレクトリの中のファイル一覧を表示したときに、“this directory”のような特殊なインターネットショートカットを表示するようにすることで、ある程度解消できると考える。

3.4 利用のシナリオ

この方式を採用した場合の、ユーザによる利用のシナリオは以下ようになる。

1. 1節で述べたサービス(2)を利用して、WWWから共有ファイルシステム上にあるファイルを参照する。このとき表示されたWebページのURLは、参照されたファイルに固有のものである。これに対するコメントとして、議論サービス(1)を使ってメッセージを入力する。このときメッセージ中に上記URLを貼ることによって、コメント先のファイルを参照できるようにする。
2. サービス(1)によってメッセージを参照すると、その中に貼られたURLはハイパーリンクとして表示される。これをクリックすると、コメント先のファイルがサービス(2)によってWebページとして表示される。
3. サービス(2)によってWebページでファイルを参照すると、そのWebページの中にはファイルに対するコメントの一覧が表示される。このコメント一覧の各項目は、対応するコメントへのリンク（即ち逆リンク）である。これをクリックすると、コメントがサービス(1)によってWebページとして表示される。
4. 共有ファイルシステムでファイル一覧を表示すると、ファイルに対するコメントへの逆リンクが、インターネットショートカットとして表示される。これをクリックする

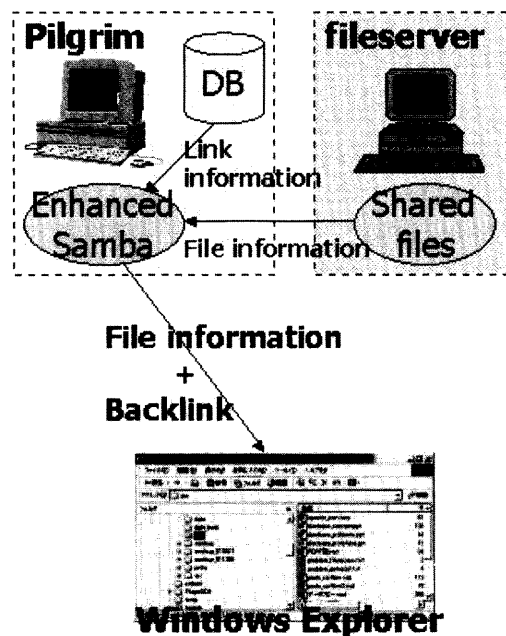


Fig.1 Backlinking.

と、Webブラウザが起動して、上記3.と同じコメント一覧（即ち逆リンク一覧）が表示される。

5. 上記2.-4.によって、WWWの議論ベースとファイルシステムがリンクによって連携するかのよう可以使用できる。
6. 議論ベースの中では、あるテーマに沿った議論における一連のメッセージから、複数のファイルが参照され得る。このような場合、一見関連が無さそうなファイル同士が議論を介して間接的に結び付けられ得る。上記3.-4.によってファイルを参照したり議論の流れを追いかけることで、これらのファイルの関連性を見出す事もできる。

このようにして本システムは、サービス(1)~(3)の連携によってWWWとファイルシステムの双方向リンク（Fig. 1）を実現し、共同作業を支援する。

4. システムデザインと実装

4.1 実行環境

本システムは、RedHat Linux 7.2 + JDK 1.3 + Apache Tomcat 4.0.3 + PostgreSQL 7.1 + Samba 2.2.4[ja]をベースにJava servletsとJSPの組み合わせで作成した。また、クライアントとしては基本的にMicrosoft WindowsベースのPCを想定する。但し他のシステムからでも、WebブラウザやSMBクライアントソフトウェアがあればアクセスは可能である。

WWWベースの議論サービスとSambaベースのファイル共有サービスを組み合わせると、次のような利点がある。即ち、サーバソフトウェアは低いコストで手に入る。クライアントソフトウェアは既に非常に普及しており入

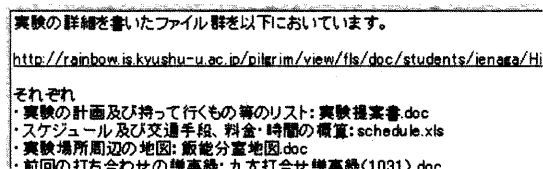


Fig.2 URL pasted in a message.

手容易である。また、これらのクライアントソフトウェアが提供するユーザインタフェースには、殆どのPCユーザは慣れ親しんでいる。

4.2 サービス

(1) 議論ベース

議論ベースは Apache Tomcat 上の JSP および Java サーブレットアプリケーションとして実装した。ユーザがログインしてメッセージを入力すると (Fig. 2), そのメッセージに関するタイトルや日付などの情報をDBMSに保存し、文書それ自体はLinuxのファイルシステム上にテキストファイルとして保存する。このメッセージの表示要求があると、DBMSから目次情報を取り出して文書を探し、Webページとして表示する。

ユーザがメッセージ中にURLを書くと、このURLをそのままテキストファイルに保存するのみならず、DBMSにもURLとそのメッセージと関連付けて保存する。このメッセージの表示要求があると、その中に書かれているURLはハイパーリンクに変換して表示する。

本システムはWWWベースなので広域ネットワークに対応することはできるが、比較的小〜中規模のクローズドな共同作業を支援することを意図しているので、部門外の人間からはアクセスできないようにすることが望ましい。そこで本システムはアカウント管理機能を提供し、アカウントを持たない人からのアクセスをブロックする。

(2) ファイルアクセス

本システムはまたLinuxのファイルシステムの特典ディレクトリツリーへのアクセスを提供する。これは通常のWebサーバと似ている。共有ファイルサーバによってLANで共有されているディレクトリツリーにこの仕組みを適用すると、3.1節で述べたように相補的な2つのファイルアクセス手段が実現する。これによって閲覧されたファイルが、議論ベースのメッセージからリンクされている場合、本システムはDBMSへの問い合わせによってこれを検出して、リンクしているメッセージへの逆リンクを表示する。

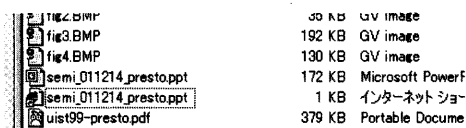


Fig.3 A virtual shortcut file.

(3) 議論とファイルの双方向リンク

このサービスに関して、WWW側での実現は前節で述べた通りなので、本節では共有ファイルサーバ側での実現について述べる。本システムではSambaをそのまま使用するのではなく、著者らが若干の改造を加えたものを使用する。この拡張されたSamba (以下は単にSamba) は逆リンクとなる仮想インターネットショートカットファイル (Fig. 3) を表示する機能を備えている。

拡張の仕方としては、Sambaに直接ショートカット表示機能を追加するのではなく、別に動作するサーバソフトウェアと連携して動作するようにSambaにAPIを埋め込んだ。どのディレクトリにどのようなショートカットファイルを表示するかは、別のサーバにて管理する。これによって次のような利点がある。即ち、Sambaに対する変更を小さく抑えて、Sambaのバージョンアップに対応し易いようにする。データベースが停止してもSamba自体は停止することがないようにする。仮想ファイルに関する制御ポリシーをSambaを停止または変更することなく変更できるようにする。また、Sambaのライセンスによるソースコードの開示を最小限に抑える。

ファイル一覧表示の要求があると、Sambaは各ファイルに対応するURLを作り、このURLを使ってDBMSに問い合わせ、そのURLを含むメッセージが無いかを調べる。もし該当するメッセージがあれば、そのメッセージのURLを含むインターネットショートカットファイルを仮想的に作り出し、ファイル一覧の中に表示する。

このように拡張されたSambaサーバを通して、ユーザは通常の方法で共有ファイルにアクセスすることができる。加えて、もし議論側からファイルが参照される場合、逆向きのリンクがショートカットファイルとして表示される。このショートカットファイルは記憶装置には実際には存在せず仮想的に現れる。このショートカットファイルは、ファイルを参照しているメッセージをリストアップするWebページへのハイパーリンクを含む。ショートカットファイルの名前は、もとのファイルに.urlという拡張子をつけたものである。例えばtest.txtを参照するメッセージ

へのショートカットはtest.txt.urlである。

4.3 日本語 URL

ファイル名が日本語であるような場合、このファイルをURLで指定するようなシステムを構築する際には注意を払う必要がある。実装の詳細に関わる部分であり本論からは少しそれるが、日本語環境で扱うファイルに関係することであり、またユーザインタフェース上の問題でもあるので、日本語URLを扱うための実装上の注意点について以下に述べる。

ファイル名に日本語の文字コードが含まれることがありうる。本システムのWebサーバ側を介してファイルにアクセスする場合、そのURLにはファイル名が含まれるので、ファイル名に日本語の文字コードが使われていればそれがURLにも含まれなければならない。しかし、URLのエンコーディングの仕方はWebブラウザごとに異なるので、どのWebブラウザからも同じようにファイルにアクセスできるようにするためには、Webサーバがハイパーリンクを表示する際にそのリンク先URLを予め特定のエンコーディングでエンコードしておく必要がある。

しかしこの結果として、URLが大変長くなる。これは、システムの記憶容量をいくらか侵食するとかデータベースのカラム幅が長くなる必要があるといったシステム的な問題とは別に、まず長いというだけで覚え難いというユーザインタフェース上の問題を引き起こす。エンコーディングされるので更に覚え難くなるし、またURLを見ただけではどのファイルか判りにくく、ファイル名からURLを推測するのも難しくなる。このような問題を解決するためには、各種WebブラウザのURLエンコーディングが規格化される必要がある。

特定のWebブラウザに限定して考えると、この問題は一部緩和される。本システムではWebサーバ側でUTF-8を使ってエンコードするようにしており、一方本論文執筆時点において最も普及しているWebブラウザであるMicrosoft Internet Explorerのデフォルト設定ではUTF-8を使う。従って、Internet Explorerを使用する限りにおいては日本語の文字コードがそのまま混じったURLによる本システムへのアクセスは成功するので、ファイル名に基づいてそのURLを組み立てるのは比較的容易となる。

4.4 現状と課題

実装した本システムをPCサーバにて稼動し、動作することを確認した。このサーバマシンのCPUはPentiumPro(200MHz)だった。

現在の実装では、ファイル一覧表示を実行するたびに、

各ファイルごとにDBMSへの問い合わせを行うため、これがパフォーマンス上のボトルネックになっている。一回のファイル一覧表示において、実際には数回~5回程度の要求をSambaは受け取るので、DBMSへの問い合わせは更に多くなる。その結果、ファイル一覧を一回表示するのに30秒もかかることがある。現在のこのようなパフォーマンスでは実用上問題があるが、これはDBMSへの問い合わせをキャッシュすることによって解決できると考える。

アカウント管理機構は、現在PilgrimとSambaでは別々になっている。Sambaの使用をLAN内にとどめる限り、現在のところこれは深刻な問題ではないが、将来的には解決すべき問題である。

もう一つの課題は、ファイルの移動によるリンク切れである。メッセージからファイルへのリンクはDBMSではURL文字列によって管理されているので、ファイルが移動したり名前が変わったりすると、そのファイルの新しいURLによるDBMSへの問い合わせは失敗し、逆リンクは表示されなくなる。一方、メッセージ中に貼られているファイルへのハイパーリンクはそのまま残っているから、これは単に「切れたリンク」になる。Sambaにおいてファイル操作の履歴をDBMSに反映するようにすれば、この問題は解決できるだろう。但しこれは、Sambaに対して更に複雑な処理を要求することになるので、上記パフォーマンスに関する課題より更に先の課題ということになる。

5. ま と め

共同作業支援システム“Pilgrim”に実装した、議論ベースとファイルとの双方向リンク機構について述べた。この機構は、議論ベースのメッセージからファイルへのリンクがある場合、そのファイルの近くにインターネットショートカットファイルを表示することによって、ファイルから議論ベースへの逆リンクを実現する。これによってWWWとファイルとの間で双方向リンクによる連携を実現し、共同作業を支援する。

コンセプトにおいては、ユーザの従来の使用環境との互換性や、共同作業におけるawarenessに対して配慮したものである。ただしパフォーマンスなどにおいては今後に課題が残った。

参 考 文 献

- 1) W3C, *About The World Wide Web*,
<http://www.w3.org/WWW/>
- 2) 日本Sambaユーザ会, *Sambaとは?*,
<http://www.samba.gr.jp/doc/whatsamba.html>
- 3) 石井: *グループウェアのデザイン*, 共立出版, 1994.
- 4) 松下; 岡田: *コラボレーションとコミュニケーション*, 共立出版, 1995.

