

蓄積された要素知識群からの伝達を効率的にするインタフェース

Interface for Efficient Transferring From Stored Group of Knowledge Element

福山 悠[†]Yuu FUKUYAMA[†]白石 善明[†]Yoshiaki SHIRAIISHI[†]毛利 公美[‡]Masami MOHRI[‡]

1. はじめに

著者らは知識を共有し、継承し新たな知識を創造する活動を支援するシステムの実現を目標としている。目標としている新たな知識の創造に向けての知識を共有することについて議論し、知識を共有するモデルとそれに基づくシステムを著者らは提案している[1]。本稿では、たくさんの知識の元となるファイル、すなわち蓄積された要素知識群が共有された上での知識の継承について議論する。

今まで経験のない業務を行うときに先達から必要な知識やノウハウを学ぶ場面は多い。以降では、このような知識をもつ者を「知識伝達者」、知識を受け取る者を「知識獲得者」と呼ぶことにする。知識伝達者が知識の伝達する方法には、口頭で伝える方法や必要な情報を含む資料を作成して渡す方法、そして第三者もしくは知識伝達者自身が以前作成した資料を渡す方法などが考えられる。口頭で伝えられる場合を除いて、このような要素知識が含まれる資料は電子ファイルとしてファイルサーバなどで共有されるのが一般的である。時間的制約から知識伝達者は口頭での伝達を望み[2]、さらに可能であれば既に存在するファイルを渡す方法を望む。しかし知識伝達者が抱えている仕事の量によっては、最も時間的制約の緩いと思われる存在する資料の受け渡しさえ行われず、知識の伝達が滞る可能性がある。

本研究では、既に存在するファイルによって知識を伝達する、つまり蓄積された要素知識群からの「知識の再利用」を効率的に行うことに着目する。文献[1]等に示されるような方法でファイルサーバに知識を伝達するために必要なファイルが十分に蓄積されていると仮定し、一般的に多忙と考えられる知識伝達者の負担を軽減し、より効率よく適切なファイルを受け渡すためのアプリケーションのインタフェースを提案する。

2. 知識の継承

2.1 知識の再利用

業務経験者や、業務内容を知っている者（業務把握者）である知識伝達者から直接指導を受けることで、知識獲得者の作業効率はよくなると考えられる。一方で、知識伝達者からすると束縛される時間が多くなることもある。そこで同じ業務が繰り返される部分に関してはマニュアルが作成されたり、または何らかの作成物のある業務ならば、業務経験者が作成あるいは業務把握者が利用した作成物を共有することで知識伝達の効率を向上させるという考えの下で、グループウェアのような知識DBが利用される。

本研究での「知識の再利用」とは、年度毎やプロジェクト毎などで繰り返し行われる業務の際に、業務経験者が作成あるいは業務把握者が利用した資料を繰り返し用いることを指している。

知識の再利用のための記憶媒体の一つにファイルサーバがある。ファイルサーバでは図1のようにグループのメンバーが他のユーザのために知識や情報を含むファイルをサーバに蓄積していく。そして、他のメンバーが必要なファイルを探し出して活用するという形態が一般的である。

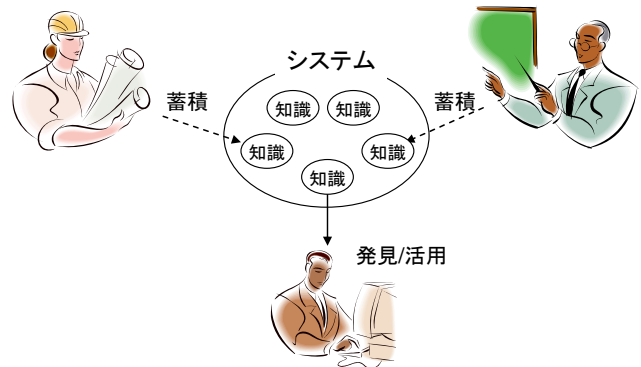


図1 蓄積された要素知識の再利用

2.2 効率的な知識伝達のために

ファイルサーバに蓄積されるファイルは膨大になるため、効率的に知識を再利用するために検索機能が用いられる。より効率的に再利用するために、例えば、共有ファイルの保存時にファイルヘタデータを付加する方法が考えられる。付加されるメタデータの数は多ければ多いほどファイルの検索時に指定できる条件が増え、目的のファイルを発見しやすくなるが、一方ではファイル保存時の負担が増えるという点も見逃せない。発見のしやすさとメタデータを付加することによる負担とのバランスを上手くすることが必要となる。グループウェアの中には、付加するメタデータの種類や付加のしやすさ・自由度などが精査されたものもある。

しかしながら、グループウェアは次第に使われなくなるという研究結果[3]もある。業務遂行者がグループウェアを利用するのは業務に取り組み始めたばかりの低いパフォーマンスしか得られない時期であり、一定のパフォーマンスが得られるようになるとグループウェアを利用しなくなるということである。つまり業務遂行者はグループウェアを決まった時期に集中して必要とするため、閲覧されるファイルもある程度共通していると考えられる。そこで、メタデータの使用によりファイルを検索しやすくするシステムによる支援の方向だけでなく、業務経験者が以前に閲覧したファイルを直接的に渡すといった知識DBの活用の方策

[†]名古屋工業大学 Nagoya Institute of Technology

[‡]岐阜大学 Gifu University

により全体としての効率がよくなる場合もあると考えられる。

後者の方策をとる場合、実際に知識伝達者である業務経験者が業務把握者がファイルを渡す場面を考えると、まずファイルサーバから目的のファイルを探し出さなければならない。自分が作成したか、所持しているファイルならば問題ないが、人から教えてもらったファイルの場合、探し出すのに手間取る可能性がある。以前閲覧したファイルなのでどこにあるのかは知っているはずだが、その場所を思い出すのが容易ではないことも多いからである。

そこで共有ファイルだけでなく知識伝達者から知識獲得者へファイルが渡されたという情報、つまりそのファイルが役に立つという「知識」も後々に再利用できれば知識伝達の効率をよくできると考えられる。本研究ではこの伝達情報を「メッセージ」と呼ぶこととし、まずファイルサーバから任意のファイルを送るためのメッセージを知識獲得者に送るためのインタフェースを提案する。

3. インタフェースの設計方針

知識継承を支援するアプリケーションクライアントのインタフェースを開発する目的は、知識伝達者がファイルサーバに蓄積されているファイルを知識獲得者に円滑に渡すことである。ここでの「ファイルを渡す」とはファイルサーバ内のファイルのパスを渡すという意味である。

グループウェアが利用されなくなる主な要因は、業務遂行者である知識獲得者が成熟していくにつれ知識を必要としなくなるということの他に、知識獲得者が検索をし、必要な知識を発見しなければならないという、ちょっとした処理の増加を面倒に思うという心理的な面も考えられる。この知識の継承に伴う処理の負担は、多様な IT スキルを持つ組織においては無視しない方がよいと著者らは考えている。つまり、たとえ理論的には知識の伝達効率がよくなる機能を加えたとしても、ユーザにシステムの利用を敬遠されてしまわないようにする必要があるということである。そこで、インタフェースを設計するにあたっては、既に別の形でなされている知識継承に関わる作業の処理負荷を軽減することを考える。

知識伝達者が図 2 のようなメッセージ付きのファイルを渡すためには電子メールを用いる方法が考えられる。メッセージを渡したい相手に対し、ファイルサーバ内のファイルのパスと簡単な説明などのコメントを添えてメールを送るというものである。しかし、メッセージを送ろうとする度にメールを起動し、アドレス帳から送信相手を探すのは、知識伝達者が多忙であればあるほど面倒であり、メッセージの送信を後回しにしてしまうこともある。この作業の処理時間を減らすためには、伝達したいファイルを見つけたらそのファイルに対し操作を行うことでメッセージの送信ができるようなインタフェースが望ましい。

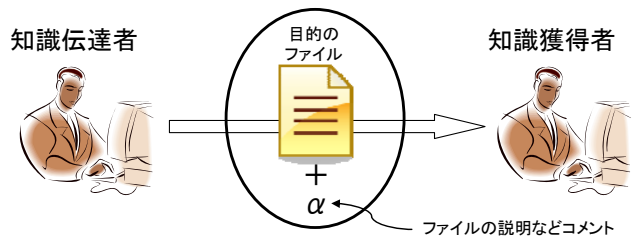


図2 メッセージの概念

4. 提案インタフェース

ファイルサーバには十分な知識の元となるファイルが蓄積されているとする。知識伝達者はファイルサーバと関連付けられたメッセージの配布機能を用い、ファイルと共に伝えたい情報を入力し知識獲得者に送信する。メッセージを受け取った知識獲得者は、メッセージに含まれるファイルパスの情報を用いてファイルサーバから提示された目的のファイルを取得する。以上のようなメッセージの配布機能を有するシステムのイメージを図 3 に示す。

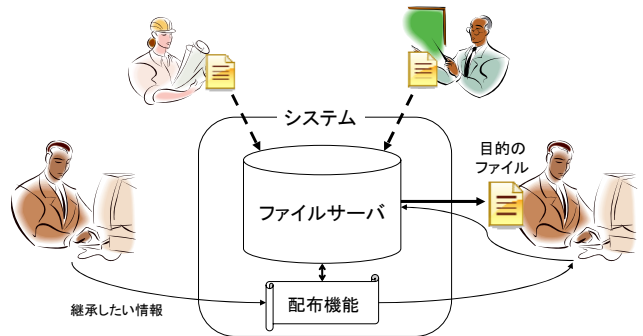


図3 知識伝達を効率的に行うシステムのイメージ

提案インタフェースの実装は、文献[1]のファイル共有システムに対して追加する形で行った。システム全体の構成は図 4 のようになる。

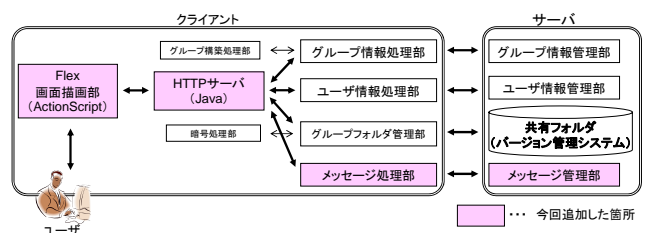


図4 提案インタフェースを含む知識継承を行うシステムの構成

文献[1]では GUI を Swing で作っていたが、利用者の使いやすさの向上を図るために Adobe Flex 3[4]で作直した。文献[1]のシステムで用いているいくつかの Java のライブラリは ActionScript には現在存在しなかったため、クライアント内でまず Java で実装した HTTP サーバを立ち上げ、Flex から Socket 通信を行うことで Java アプリケーションを呼び出し、目的の処理を行うこととした。図 4 の Flex 画

画面描画部と HTTP サーバ、メッセージ処理部、メッセージ管理部で行う処理を表 1 から表 4 に示す。

システムの UI の部分が表 1 にまとめる Flex 画面描画部である。システムウィンドウをはじめ、メニューやメッセージウィンドウなどを描画し、ユーザからの操作を受け付ける。

表 1 Flex 画面描画部の処理

画面描画	ユーザの操作に対応した画面を描画し、HTTP サーバから受け取ったデータを表示する
処理依頼	HTTP サーバにアクセスし、ユーザの操作に対応した処理を依頼する
データ受け取り	処理依頼に対する結果のデータを HTTP サーバから受け取る

GUI を持つ Flex のアプリケーションを起動する前に同一ホスト内で立ち上げておくソフトウェアに、表 2 にまとめる HTTP サーバがある。これは、文献[1]のファイル共有を行うために必要な Java ライブラリと Flex を接続する命令変換機構として機能するものである。

表 2 HTTP サーバの処理

アクセス受付	サーバを立ち上げ、Flex 画面描画部からの処理依頼を待ち受けする
Java モジュールの呼び出し	処理依頼に対応する Java による処理を呼び出す
処理結果の受け渡し	処理依頼に対する結果のデータを Flex 画面描画部へ送る

クライアント側でメッセージに関する処理を行うのが表 3 にまとめるメッセージ処理部である。メッセージの送受信の他、ファイルパスの生成を行う。文献[1]のファイル共有システムにはユーザの PC にインストールされたファイル共有クライアント毎にルートのフォルダが存在し、そのフォルダ以下に共有ファイルがダウンロードされるというものである。ユーザによって同じ共有ファイルであってもファイルパスは異なるので、ファイルパスの生成をクライアント毎に行っている。ただし、ルート以下のパスは全ユーザ共通であるため、メッセージではルート以下のパスの情報を送信し、受信側でそれぞれのファイルパスを生成すればよい。

表 3 メッセージ処理部の処理

メッセージ送信	ユーザによってメッセージウィンドウの送信ボタンが押されたとき、ファイルパスの生成を行った後、メッセージの内容を送信する
メッセージ確認依頼	一定時間おきにサーバに自分宛のメッセージが届いていないかの確認を依頼する
メッセージ受信	メッセージ確認依頼を行い、メッセージが存在した場合受信する
ファイルパス抽出	メッセージ送信時に目的のファイルのパスからルートフォルダ以下のパスを抜き出す
ファイルパス生成	メッセージ受信時にルートフォルダと送られてきたパスを結合する

表 4 にまとめるメッセージ管理部は、クライアント間で送受信されるメッセージを管理するためにサーバに存在する。クライアントから送信されたメッセージを配信先のユーザ毎に保管し、クライアントから要求があればメッセージを送信する。また図 4 のユーザ情報管理部で管理されるユーザ毎の更新履歴の追加を行う。

表 4 メッセージ管理部の処理

アクセス受付	サーバを立ち上げ、クライアントのメッセージ処理部からのアクセスを待ち受ける
メッセージ保管	メッセージを受け取り、ユーザ毎に保管する
メッセージ確認	クライアントからのメッセージ確認依頼に対し、そのユーザにメッセージが届いているか確認する
メッセージ送信	メッセージ確認を行い、メッセージが存在する場合送信する
更新履歴の記録	メッセージを送信したユーザ、受信したユーザそれぞれの更新履歴を追加する

メッセージを送るときには、知識伝達者は図 4 のグループフォルダ管理部によって取得された共有ファイルのリストの中から知識獲得者に渡したいファイルを選び、図 5 のようにメッセージ送信のためのウィンドウを表示させる。選択ファイル共有するグループのメンバーのリストを図 4 のグループ情報処理部によって取得し、送信相手を選択すると共にコメントを入力し、これらに選択ファイルのパスを加えたものをメッセージとして送信する。

知識獲得者がメッセージを受信した場合、図 5 のウィンドウ下の更新履歴にその旨が表示され、右上の新着メッセージボタンが点滅し、メッセージを受信したことがわかるようになっている。

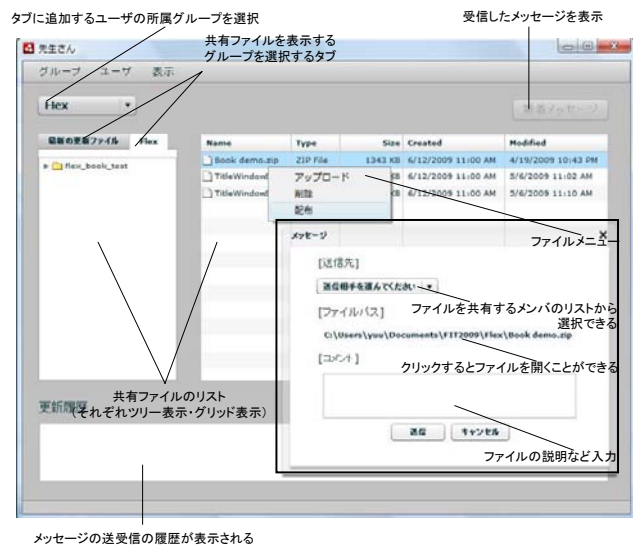


図5 提案インタフェースを実装したアプリケーションクライアントの GUI

提案インタフェースの操作をまとめると次のようになる。

【知識伝達者】

知識伝達者はファイルサーバで共有されているファイルを選択する
メニューから“配布”を選択する
表示されるメッセージウィンドウ上で送信相手を選び、コメントを書く
送信ボタンを押す

【知識獲得者】

新着メッセージボタンを押す
表示されるメッセージウィンドウでコメントを読み、ファイルのリンクをクリックして目的のファイルを観覧する

5. 実験

提案インタフェースの使用により、知識伝達者が知識獲得者に対しメッセージを送ることに対する負担が軽くなることを確認する。

知識伝達者がファイルサーバ内のファイルを知識獲得者に渡すためのメッセージを、(1)電子メールで送った場合と(2)提案インタフェースで送った場合でそれぞれ処理時のクリック数と処理時間を計測した。計測は目的のファイルを発見したところから、送信ボタンを押すところまでとした。またコメントは同一時間で同一内容を生成するものとして、コメントを書き込む時間は含めていない。

実験環境は表5に示すとおりであり、表6が実験結果である。電子メールでメッセージを送る場合に比べて、提案インタフェースは知識伝達者がメッセージを送る処理の負担を軽減できることを確認した。

表5 実験環境

OS	Windows Vista Business
実装環境	Adobe Flex 3(Version3.2.0)
	Java (Version1.5.0_11)
開発環境	Eclipse (Version3.3.2)
メーラ	Mozilla Thunderbird(Version2.0.0.12)

表6 メッセージ伝達に関わる処理負担の比較

	クリック回数	処理時間 [sec]
電子メール	12	53
提案インタフェース	6	12

6. おわりに

本研究では知識を共有し、知識を創造することを支援するための知識の継承に着目した。効率的な知識の継承のためには、ファイルにメタデータを付加することによる知識獲得者が主体となる検索ベースのシステムよりも、知識をもつ者が知識を伝達しやすいシステムの方が継続して利用されやすいと考え、知識伝達者がファイルを知識獲得者に教えることを補助するインタフェースを提案した。

今後の課題として、メッセージの再利用による効果を確認する実証的なデータを継続して採取し、評価することが挙げられる。

参考文献

- [1]福山 悠, 福田 洋治, 毛利 公美, 白石 善明, “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”, 情報処理学会第71回全国大会, 3P-5 (2009).
- [2]犬塚 篤, 中森 義輝, “IT を利用した知識共有への提言”, 電子情報通信学会論文誌, Vol.J86-D-I, No.4, pp.179-187 (2003).
- [3]藤田 幸久, 仲瀬 明彦, 中山 康子, 鳥海 不二夫, 石井 健一郎, “組織における知識継承のモデル化”, 電子情報通信学会論文誌, Vol.J90-D, No.1, pp.52-61 (2007).
- [4]Adobe-Flex 3, <http://www.adobe.com/jp/products/flex/> (2009/7/3 閲覧)