編集後テキストからのコピー元文章参照手法

猪股　修一[†]　　倉本　到[†]　　辻野　嘉宏[†]　　水口　充[‡]

A Method of Accessing Original Documents from Copied-and-Modified Text

SHUICHI INOMATA[†]　　ITARU KURAMOTO[†]
YOSHIHIRO TSUJINO[†]　　MITSURU MINAKUCHI[‡]

1. はじめに

コピー&ペースト機能は，今やコンピュータ上で作業を行う際に欠かせない機能になっており，テキストエディタや，画像・動画編集ソフトなど，様々なアプリケーションで実装されている．以後，ファイル内のテキストや画像などの要素をオブジェクトと呼ぶ．

コピー&ペーストを行う際には，オブジェクトがコピーされる側のファイルや Web ページ（以下，コピー元ファイル）と，そのオブジェクトをペーストする側のファイル（以下，ペースト先ファイル）が存在する．

コピー&ペースト機能を使っていると，「ペーストを行った後の編集がうまくいかないため，オブジェクトをもう一度ペーストをした時の状態に戻したい」，「コピー元ファイルのコピーした部分以外の内容を参照したい」など，コピー元ファイルに再アクセスする必要が生まれることがある．しかしオブジェクトに対して編集を行った状態では，コピー元ファイルへの再アクセスは難しい．

本研究ではこれらの問題点を解決するため，コピー&ペーストを行ったオブジェクトについて，そのオブジェクトのコピー元ファイルを簡単に調べることができるコピー元情報管理手法を提案する．さらに，コピー元表示機能を持たせた試作アプリケーションについて述べる．

2. コピー元参照における問題点

ファイルの作成にあたりコピー&ペーストを利用する場合は，ペーストしたオブジェクトをそのまま使うのではなく，文章を挿入したり，一部を書き換えたりするなど，編集を加えることが多い．しかし，オブジェクトに対して編集を行ってしまうと，そのオブジェクトのコピー元ファイルへの再アクセスが困難になるという問題点がある．以下に問題となる例を挙げる．

- **事例１**：あるテキストをコピー&ペーストを用いて引用し，新しいテキストを作成していた．しかしペーストを行ったテキストに対する編集がうまくいかず，内容がよく分からない文章になってしまった．そこでもう一度コピーを行った当初の文章を参照したいが，どこからテキストを引用してきたのかを覚えていない．またペーストをした後に編集していたため，テキスト検索でコピー元を探すことは難しい．
- **事例２**：ある事柄に関する調べ物をするという作業をしており，複数の Web ページからテキストをコピー&ペーストして編集し，メモを残しておいた．後日，そのメモの中でさらに詳しく調べたい部分ができたため，その部分をコピーした元の Web ページに再アクセスする必要が生まれたが，事例 1 と同様にペースト後の編集によりテキスト検索はうまくいかない．あらかじめこの作業で使用したコピー元ファイルはブックマークをしておいたが，目的の部分をブックマークリスト内のどのページからコピーしてきたのかを覚えていないため，ひとつずつページを開いて内容を確認しなければならない．

この 2 つの事例のように，ユーザがオブジェクトをペーストし，さらに編集を行った後に，オブジェクトのコピー元ファイル，もしくはコピー&ペーストを行った時点のオブジェクトをもう一度参照しなければならない状況がしばしば存在する．

† 京都工芸繊維大学
Kyoto Institute of Technology
‡ 京都産業大学
Kyoto Sangyo University

コピー元ファイルを参照するには，フルパス名のようなファイルを一意に識別することができる情報が必要である．しかし，ペーストしたオブジェクトのコピー元ファイルに関する情報をユーザが正確に対応付けて記憶しておくことは難しい．

そのため事例 1 では，ユーザはコピー元ファイルを見つけるためにそのテキスト自身をキーとしてファイル検索を行っている．これは，そのコピー元ファイルについて何も覚えていないという場合に発生する．しかし事例 1 のようにペーストをしたテキストに対して編集が加えられている場合は，そもそもどの部分が編集前のテキストであるかの判別が困難であり，検索のための適切なテキストやキーワードを抽出することができない．

ペーストを行った時点でのオブジェクトを参照する手法には，検索のほかにアプリケーションに実装された Undo 機能を使い，ペースト後の編集を遡る方法がある．しかし，一般に Undo 機能で遡ることができる編集のステップには限界があるため，ペーストを行った時点まで遡ることができない場合がある．また，目的の部分とは関係の無い部分の編集まで遡ってしまう，ちょうど参照したい状態になるまで Undo 操作を繰り返さなければならないため時間がかかってしまう，などの問題がある．

事例 2 では，あらかじめコピー元ファイルのリストを作っておくことで，コピー元ファイルの探索時間を減らし，参照を容易にしようとしている．しかし，作成しているファイルに対してコピー元ファイルが複数存在する場合，ペースト後に編集されたテキストと，コピー元ファイルの内容をおのおの比較して目的のコピー元ファイルを探す必要がある．両者は全体的なテキストの様子が異なるため，一瞥しただけではその同等性を認識することが困難であり，探索に膨大な時間がかかってしまうことが推測される．

このように，ペーストしたオブジェクトに対して編集したことがコピー元ファイルの参照を困難にしていることがわかる．本研究ではこの問題に対して，オブジェクトに編集が加えられた後でも，簡単にコピー元ファイルを参照することができる手法を提案する．

3. 提案手法

本研究では前節で述べた，ペースト後に編集を加えられたオブジェクトからはコピー元ファイルへのアクセスが困難である，という問題を解決するために，オブジェクトに直接コピー元情報の付加を行う手法を提案する．

まず提案手法の概要について説明し，次にオブジェクトがテキストの場合におけるプロトタイプの実装について述べる．

3.1 コピー元ファイル情報の付加

提案手法の概要を**図** 1 に示す．まずコピーが実行された際に，オブジェクトに関するデータだけでなく，コピー元ファイルの URI などのファイルを識別する情報を一緒にクリップボードに送る．次にペーストが実行された際に，オブジェクトに関するデータと共に，この情報をクリップボードから取得し，ペースト先ファイルに記録する．

ここで，ペーストするオブジェクトとコピー元ファイルに関するデータは，システムが互いに対応付けて記録し，その後オブジェクトに対してユーザが編集を行っても，編集後のオブジェクトと，コピー元ファイルの対応付けは保持されるようにする．これにより，オブジェクトに編集が行われた状態でも，ペースト先ファイルのオブジェクトからコピー元ファイル情報を用いて，そのオブジェクトのコピー元を参照することができる．

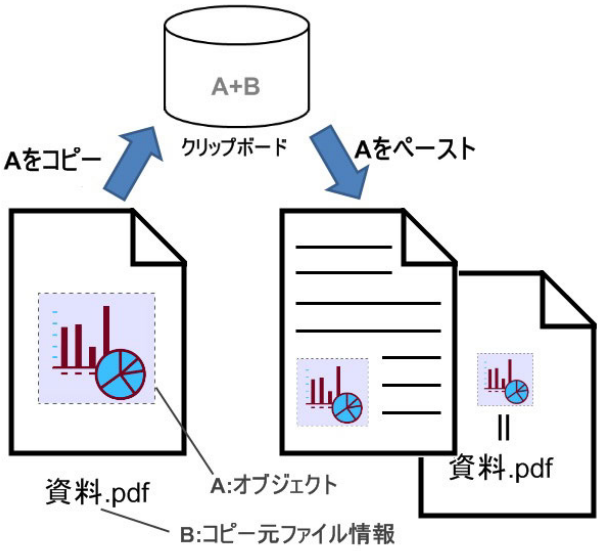

図1 コピー元情報付加の概要

3.2 テキストオブジェクトへの適用

コピー&ペーストの対象となるオブジェクトがテキストの場合，文字単位で自由に分割や挿入が可能であるという特性があるため，ペーストされたテキストと，コピー元情報の対応付けは単純ではない．そこで本手法では XML タグをテキストに付加することで，コピー元ファイルの情報とテキストオブジェクトを対応付

```
<typed>
コピー&ペーストについて
</typed>
<pasted from="http://ja.wikipedia.org/wiki/%E3%82%B3">
コピー・アンド・ペースト（Copy and Paste）とは、文章やデータなどを
    <typed>
    コピー（複写・複製）し、そのコピーしたものを貼り付け（ペースト）
    するという操作を表すコンピュータ用語である。
    </typed>
略称としてコピペを使うことがある．
多くの GUI で統一された操作で利用できる．
</pasted>
```

図2 タグ付けによるコピー元情報の付加

ける．このタグは実際にユーザが編集を行っているファイルそれぞれにひとつずつ存在する XML ファイルに書き込まれる．XML ファイル内のタグ付けの例を**図** 2 に示す．

図 2 に示すように，テキストがペーストされた際には，システムは XML ファイルの対応する箇所にペーストされた文字列を挿入し，その前後にタグを付ける．本論文ではこのタグを pasted タグと呼ぶ．またペーストされた文字列と，ユーザのタイピングによって入力された文字列を区別するため，タイピングによって入力された文字列には typed タグをつける．XML ファイルはユーザがテキストを編集するのに同期して，自動的に内容が変更される．

pasted タグには属性として，コピー元ファイルの位置を URI として持つ．これにより，ペーストされたテキストが編集されても，pasted タグそのものは消えないため，ペーストされた文字列とコピー元ファイルの情報とを結び付け続けることが可能になる．

またユーザがペーストを行った文字列に対して編集を加えた結果，ペーストをした文字列が 1 文字も残っていないという状況が考えられる．このような状況でも「ペーストで貼り付けた文章を編集した」という記憶を持つユーザに，コピー元ファイルの情報を提供できるようにすることが望ましい．そこで，pasted タグ内の文章が編集によって全て消去されても，pasted タグ自体は消さずに残しておく．これにより，ユーザが編集しているテキストからは全て消されてしまったものの，あるファイルからテキストをコピーしてきた，という情報を残しておくことができる．

3.3 プロトタイプ

ユーザがテキストのコピー元ファイル情報を参照することができるテキストエディタのプロトタイプを Windows XP 上に実装した．**図** 3 にそのインタフェースを示す．このプロトタイプは通常のテキストエディタとしてふるまうが，システムはテキスト内のどの部分がペーストされた文字列であるかを，テキストの横にバーを表示することによって示す（図 3 右側）．このバーはコピー元ファイルごとに表示され，このバーの範囲内に，そのバーが示すコピー元ファイルからペーストされた文字列が存在していることを表す．ひとつの行内に複数のコピー元からペーストされたテキストがある場合には，バーが並行して表示される．

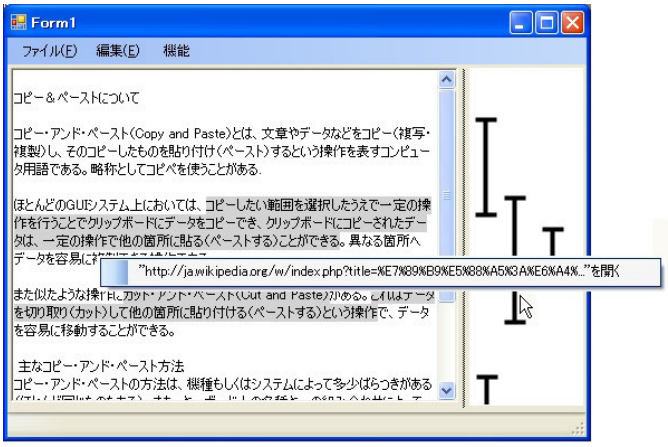

図3 プロトタイプ：コピー元を表示した様子

バーの上にマウスカーソルを乗せると，コピー元ファイル情報をポップアップで表示し，ペーストされたコピー元ファイルに対応するテキストの背景色が灰色に変化する．コピー元ファイルの情報として，コピー元ファイルの URI を表示する．ユーザはこの URI をクリックすることで，コピー元ファイルを参照することができる．

コピー元ファイルの URI は，コピーの際にクリップボードに送られる HTML フォーマットのデータから取得している．そのため，本プロトタイプは，HTML フォーマットのデータをコピー時にクリップボードに送るアプリケーション（Word や Internet Explorer など）からコピーを行った場合のみを対象にしている．

4. 議論

本提案手法では，コピー元情報として URI のみを利用している．提案手法を用いてユーザが実際にコピーを行ったオブジェクトを参照するには，URI を用いてコピー元ファイルにアクセスしたあと，ファイル内からコピーを行ったオブジェクトを探す必要がある．URI だけでなく，ファイル内の具体的なオブジェクトの位置も記録し，提示できるようにすることで，コピーしたオブジェクトを参照する時間をさらに短縮する

ことができると考えられる．

さらに本提案手法では，コピー元ファイルの更新が行われた場合や，位置が変更された場合に，ペーストされたテキストとコピー元ファイルの対応が壊れてしまう．コピー元ファイルに対する操作に対応し，ペースト先ファイルに記録されたコピー元ファイル情報を自動的に更新するようにすることで，システムの信頼性が向上すると考えられる．

また今回実装したテキストエディタでは，コピーの際に HTML フォーマットのデータをクリップボードに送るアプリケーションでしか，コピー元ファイルの URI を参照することができない．アプリケーションの種類によらず提案手法を用いるためには，全てのアプリケーションでコピー元情報をクリップボードに送るようにするか，クリップボード自身にコピー元情報を取得する機能を持たせる必要がある．

5. 関連研究

コンピュータ上で行った作業に関する情報を記録しておき，ユーザの必要に応じてその情報を提示する研究は多数存在する．Time-Machine Computing[1]は，任意の時間にコンピュータの状態を戻し，その時点でのデスクトップ，及びデスクトップ上のファイル内容の再現・比較を可能にしている．Time-Machine Computing のように，ユーザが行った作業のログを記録し，それを提示するシステムを用いれば，「ペースト」というイベントが発生した時点でのファイルを再現することで，特定のオブジェクトがペーストされた時の状態を調べることができるが，複数のログを比較して目的の時点を探し出す必要があり，ログを探索するというオーバヘッドが発生してしまう．

また 1 つのファイルについて，編集に応じてその段階ごとにファイルを保存する，バージョン管理システムに関する研究も行われている[2]．このようなバージョン管理手法を用いても，ペーストを行った状態のファイルを参照することができるが，膨大なバージョンファイル群の中から目的のファイルにアクセスすることは困難である．

また俺デスク[3]のようにファイル間の関連性を用いたファイルアクセス支援の研究がある．特定のファイルに対して，コピー&ペーストというユーザ操作を行ったファイルを抽出することで，そのファイルにペーストしたオブジェクトのコピー元ファイルを知る事が可能である．しかし，複数のファイルからコピーを行いコピー元ファイルが複数存在する場合は，テキストの内容とコピー元ファイルの内容の比較をして，目的のオブジェクトのコピー元を探すオーバヘッドが発生する．

この節で述べてきたように，現存する手法を使ってコピー元ファイルの情報を探す場合は，複数の選択肢の中から目的のものを探す必要がある．提案手法では，ペーストされたオブジェクトからコピー元ファイルが一意に定まることを利用して選択肢をひとつに絞ることで，複数の候補から目的のものを探すというオーバヘッドを無くすことができ，ユーザに対して簡潔にコピー元の情報を提供することができる．

6. まとめ

本論文では，ペーストをした後に編集を加えられたテキストから，コピー元のファイルへアクセスする手法と，テキストエディタとして実装したプロトタイプについて述べた．今後は，コピー元情報として URI だけでなくコピー元ファイル内のオブジェクトの具体的な位置情報を提示することや，この機能を実環境で用いた場合のユーザビリティの評価，テキスト以外のオブジェクトへのコピー元情報の付与などを検討していきたい．

参 考 文 献

1) 暦本純一: Time-Machine Computing: 時間指向ユーザインタフェースの提案，ソフトウェア科学会 WISS1999 論文集，近代科学社(1999).
2) 山本哲男，松下誠，井上克郎: バージョン管理ファイルシステムを用いた保守支援ツールの提案，電子情報通信学会技術研究報告 SS99-23 Vol.99 No.164，pp.65-72 (1999).
3) 大澤亮，高汐一紀，徳田英幸: 俺デスク:ユーザ操作履歴に基づく情報想起支援ツール，情報処理学会第 47 回プログラミング・シンポジウム報告集，pp.15-21(2006).