

ヒストリグラフを利用したコマンドの再利用の提案と評価

稲葉由倫† 渋谷雄† 辻野嘉宏† 西田知博††

†京都工芸繊維大学

〒606-8585 京都市左京区松ヶ崎御所海道町

e-mail:inaba@hitms.dj.kit.ac.jp, {shibuya, tsujino}@dj.kit.ac.jp

††大阪学院大学

〒564-8511 大阪府吹田市岸部南二丁目 36 番1号

e-mail:nishida@utc.osaka-gu.ac.jp

概要

創造的作業や非定型作業では、試行錯誤がしばしば繰り返される。多くのアプリケーションで見られるアンドゥ機構は、元々、ユーザの操作ミスを容易に回復させるために導入されたものであるが、そのような試行錯誤の支援としても用いられている。一方、試行錯誤の支援方法として、過去に行った作業や、実行されたがアンドゥによって取り消された作業、または、それらの一部の作業の再利用の支援も考えられる。本稿では、ヒストリグラフを用いたアンドゥ機構を拡張した、コマンドの再利用手法を提案する。さらにこの機構を簡単なグラフィックエディタに実装し、評価実験を行い、その有効性を確認した。

A Reuse Method of User Operations Using History Graph Yoshitomo Inaba† Yu Shibuya† Yoshihiro Tsujino† Tomohiro Nishida††

†Kyoto Institute of Technology

Matsugasaki, Sakyo-ku, Kyoto, 606-8585, Japan

e-mail:inaba@hitms.dj.kit.ac.jp, {shibuya, tsujino}@dj.kit.ac.jp

††Osaka Gakuin University

2-36-1 Kishibe-Minami, Suita, Osaka, 564-8511, Japan

e-mail:nishida@utc.osaka-gu.ac.jp

Abstract

In creative works and non-routine works, we have many trial-and-errors. The undo mechanism in most applications is not only to recover any errors and mistakes in user operations easily, but also to support user's trial-and-errors. On the other hand, the reuse of user operations executed (and maybe undone) at past or sequences of them may also be able to support trial-and-errors. In this paper, we propose a reuse method of user operations which is an extension of the undo mechanism based on the history graph. We implement a simple graphic editor with the reuse method and evaluate it.

1 はじめに

創造的な活動においては、しばしば試行錯誤が行われる。例えばなにか文章を書く時、それがそれほど長くない文章であっても度々推敲が行われたり、イラストを描く時などでも修正や訂正が頻繁に行われる。また、試行錯誤は、初心者などによるシステムの探索的学習の際にも行われ、ある操作によりどのような結果が起こるかをユーザが実際に確かめる為に重要である。

このような試行錯誤を支援するため、ヒストリグラフに基づくアンドゥ機構を提案した[2]。しかし、試行錯誤の中には、全体としては捨てられた操作列ではあるが、一部分は生かしたいものがあることも考えられる。

そこで本研究ではこのような非定型作業を効率的に行うための支援方法として、ヒストリグラフを用いたコマンドの再利用を提案し、評価を行った。

以下、2節では代表的なアンドゥ機構とヒストリグラフに基づくアンドゥ機構について述べ、3節では本稿で提案するコマンドの再利用方法について詳説する。さらに、4節では評価実験について、また5節ではまとめを述べる。

2 Undo 機構

2.1 制限線形アンドゥ

制限線形アンドゥ (restricted linear undo) は現在の一般的なアプリケーションで用いられているアンドゥで、ユーザの操作を線形の履歴リストとして保持しており、直前の操作から逆順に取り消すことができる。また、取り消した操作を再び行えるよう、アンドゥされた操作は線形のリドゥリストに順に保持される。ユーザはアンドゥやリドゥを繰り返して行って希望の状態を回復することができる。ただしリドゥリストにはアンドゥされたすべての操作が保持されるので、再実行不可能な操作がリドゥリスト内に残らないよう、制限

線形アンドゥでは新しい操作が行われた時点でリドゥリストを空にしている。

制限線形アンドゥは単純な操作を繰り返すだけで容易に操作を取り消すことができるが、一度の操作で取り消せるのは直前の1つの操作であるので、以前の操作を取り消したい場合には時間がかかる。またリドゥリストに操作が保持されていても新しい操作が行われるとそれらは失われることになるので、過去のすべての状態を回復できるとは限らない。

2.2 選択的アンドゥ

線形アンドゥモデルは単純なアンドゥを提供するものであるが、場合によって大変時間のかかる手段である。そこで、行われた操作の履歴を線形に取り扱うのではなく、過去の任意の操作を取り消せるようにしたものが、選択的アンドゥ (selective undo) [1]である。選択的アンドゥでは、選択した操作の対象がすでに存在しないなどアンドゥが意味をなさない場合にはそのような操作をアンドゥの対象として選択できないようになっている。

過去の任意の操作を取り消すことが可能な選択的アンドゥは強力で、過去にない状態も作り出せるが、操作を示す良い方法が無い。例えば、テキスト表現でリストアップされた実装がなされたものがあるが、それぞれがどの操作に相当するのかわかりにくい。

2.3 ヒストリグラフに基づくアンドゥ

ヒストリグラフに基づくアンドゥでは、アンドゥを「操作の取り消し」ではなく「過去の状態を回復させるための操作」としてとらえている。過去の状態を回復させる方法は、制限線形アンドゥで一つ一つ操作を取り消して行けば状態をさかのぼることは可能だが、時間がかかることが問題である。過去の状態をすべて提示すれば、少ない操作回数で回復できる。そこで過去の作業状態がどのように遷移してきたかをわかりやすく提示する方法として、筆者らはヒストリグラフを提案した[2]。グラフのノードは作業状態を表し、それ

に続くノードとの間を結んだエッジは操作を表している。ノードをクリックするとそのノードが表す作業状態を回復することができる。このアクションをアンドウの操作とし、簡単な操作で過去のいずれの状態も回復できる機構をユーザに提供している。

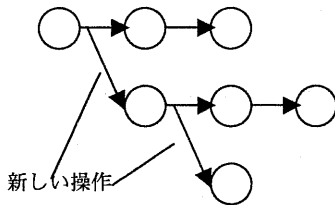


図1 ヒストリグラフにおける状態遷移

ヒストリグラフに基づくアンドウでは、アンドウによりある状態に戻って、そこから新しい状態を表すノードを生成した場合、作業状態とその遷移を表したグラフは枝分かれを生じ、単純には木構造となる(図1参照)。操作が行われて、新しい状態を生じた場合には、木にノードを付加していく。しかし操作を行った結果が過去の状態と同じであれば新しいノードを付加しないで、その過去の状態へのエッジのみを生成する。これは「状態」を基本として考えている為である。即ち操作手順等に関係なく作業状態の構成要素が同じである状態は同一であるとしている。従って構造はグラフとなるのでヒストリグラフと呼ぶ。ヒストリグラフに基づくアンドウでは過去の状態を選択し移行するという操作でアンドウが行えるので、アンドウを行ってもヒストリグラフにノードを追加したりせず、履歴の複雑化を避けている。またヒストリグラフに基づくアンドウはマウスのクリックという簡単な操作で過去の全ての状態に戻れるという利点がある。

3 コマンドの再利用について

3.1 基本的な概念

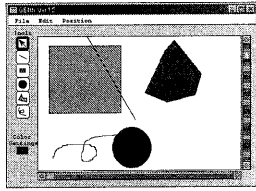
試行錯誤を伴う非定型作業を行う時、アンドウやリドゥを用いることが多いと考えられるが、アンドウによって誤操作が容易に取り消せたとしても、ユーザが誤操作をした後で行った操作がアンドウによって無効になってしまったり、あるいはアンドウされていない操作の一部で再利用したい操作がある場合、ユーザがその操作を同じように再現しなくてはならない。これがもし定型作業として予め内容の決まっている作業ならマクロなどで操作を登録することにより、作業を簡単化できる。確かに、マクロ機能は繰り返しの作業に効果を発揮するが、具体的な作業内容が判明していなければならない。これに対して、アンドウ機構に基づく再利用は操作履歴を用いるので準備を必要としない。しかしながら選択的アンドウではコマンドがテキストで表現されていることが多いためにどのような操作であるかがわかりにくい。次の節ではヒストリグラフを利用したコマンドの再利用について述べる。

3.2 ヒストリグラフの利用

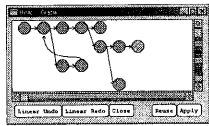
以下では本研究で提案するヒストリグラフを利用したコマンドの再利用方法を実装したグラフィックエディタを例にして、コマンドの再利用方法について述べる。グラフィックエディタにおけるコマンドの再利用では操作対象となるオブジェクトを限定せず、コマンドのみを再利用できる。

グラフィックエディタの構成は、主な作業を行うメインウィンドウ、ヒストリグラフとコマンドの再利用のボタンが配置されたヒストリグラフウィンドウ、グラフ中のノードが表す状態のスナップショットを確認できるプレビューウィンドウである(図2参照)。

また、本グラフィックエディタにおける図形の色の設定方法は、ポップアップウィンドウに備えつけられたスライダでRGBの値を決める(図2(d))。RGBの値は0.00~1.00の間で設定される。設定された値を用いて状



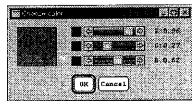
(a)メインウィンドウ



(b)ヒストリグラフウィンドウ



(c)プレビューウィンドウ



(d)色選択ウィンドウ

図2 グラフィックエディタの構成

態が遷移した時には、その値は作業状態と共にノードに保持される。

次に、ヒストリグラフを利用したコマンドの再利用方法を以下に示す。ヒストリグラフではノード間のエッジが操作を表しているのので、ここでは、再利用したい操作を、それを表すエッジを挟む2つのノードを選択することによって指定する(図3)。

通常の作業をしている時(図4(a))、ヒストリグラフは操作が容易なアンドゥをユーザに提供しているので、再利用を行う時にはまずヒストリグラフウィンドウ内の Reuse ボタンを押し、再利用モードへ移行する。このとき再利用可能なエッジの元になっているノードの縁が赤線で強調表示される(図4(b))。これらのうちのあるノードをクリックし始点

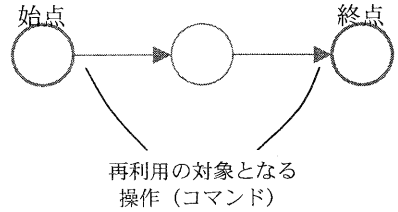
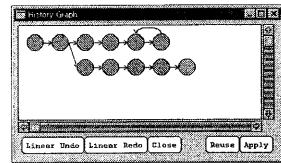
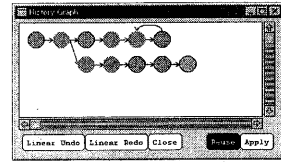


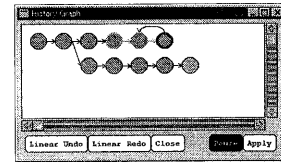
図3 再利用対象の選択例



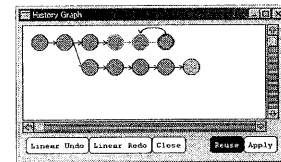
(a)通常時



(b)再利用モードへ移行した時



(c)始点ノードを決定した時



(d)終点ノードを決定した時

図4 再利用したいコマンドを選択する時のヒストリグラフの様子

を決めると、そのノードからどこまで連続して再利用可能かが強調表示される。つまり始点は赤太線、エッジは赤線、最終点は黒太線

で強調される (図 4 (c)). 始点から最終点の間で終点を決定すると, 2つのノードが決定され, その間に経由するエッジにあたるコマンドが適用されることになる (図 4 (d)). 適用結果はプレビューウィンドウにて確認できるようになっている. プレビューが希望の状態であれば Apply ボタンを押し, 作業状態に反映する. 再利用する必要が無ければ再び Reuse ボタンを押し再利用モードから抜ける.

このようなコマンドの再利用には今回のグラフィックエディタに対してだけでなく一般に次のような利点がある.

- ・非定型作業を行う際, 過去に行った操作を繰り返し利用できる
- ・過去の操作列の部分的な利用ができ, 試行錯誤を支援できる
- ・異なるオブジェクトに対して同じ操作を適用できる

以上の利点は試行錯誤をして新しいものを創造したいユーザの作業効率を改善するのに有用であると考えられる.

2. 2で触れたように, 選択的アンドゥモデルが提供している選択的リドゥでは再利用できない操作は選択できなくなっているが, 本研究で提案するコマンドの再利用においても全ての操作が常に再利用可能であるとは限らない. すなわち, システムの状態によってユーザの操作が制限される場合がある. 再利用可能性によって, グラフィックエディタの操作は次のように分類することができる.

図形非選択時: 図形描画

図形選択時 : 図形の移動

図形の変形

図形の色変更

従って本グラフィックエディタでコマンドの再利用が行われる際, 図形選択ツールによって図形オブジェクトが選択されている場合には図形選択時の操作を表すエッジの元となるノードが強調され, 何もオブジェクトが選択されていなければ図形描画の操作を表すエッ

ジの元のノードが強調される.

上に示した分類は, 今回試作の対象としたグラフィックエディタが複雑な機能提供をしていないため簡単なものになっているが, 市販されているアプリケーションは多機能であるため再利用の可能性について細かな分類が必要である.

4 評価実験

4. 1 実験方法

コマンドの再利用は, 特に試行錯誤を行う際に重要な意味を持つので, 次のような3つの条件の下で実験を行った.

条件 a) メインウィンドウのみ使用可能で, ヒストリグラフおよびコマンドの再利用が使用できない. メインウィンドウに備わっている編集機能 (コピー, カット, ペースト, 削除, 制限線形アンドゥ, 制限線形リドゥ) は使用可能である.

条件 b) 条件 a) に加えヒストリグラフが使用可能で, ヒストリグラフに基づくアンドゥが使用できるが, コマンドの再利用は使用できない.

条件 c) 条件 b) に加えヒストリグラフを利用したコマンドの再利用が可能である.

また実験の前に, 条件 c) のシステムを使って被験者にグラフィックエディタの使い方 (基本的な操作, ヒストリグラフの使い方, コマンドの再利用方法) を練習してもらった.

4. 1. 1 実験タスク

コマンドの再利用は試行錯誤を行う際に重要であるため, 以下に示すタスク 1 およびタスク 2 を考えた.

(1) タスク 1

タスク 1 は以前に行われた操作を再利用し, 部分的なアンドゥ・リドゥを行うタスクである. 過去の操作を再現する際のコマンドの再利用の有効性を見る. 実験に用いた絵を図 5

に示す。またタスクは、以下の順に行われる。

1. 左側の木に色を付ける
(葉の部分=黄緑色, 幹=茶色)
2. 家に色を付ける
(屋根=ピンク色, 窓=水色)
3. 右側の木の配色を左側の木と同様に
する [条件cの時, コマンドの再利用を
用いて]
4. 屋根の色を赤色にする
5. 煙突の色をこげ茶色にする
6. 屋根の色だけを前の色に戻す [条件c
の時, コマンドの再利用を用いて]

(2) タスク 2

タスク 2は過去の操作を繰り返し利用する
タスクで, 過去にユーザが行った操作列を繰
り返し使う時にコマンドの再利用が効率的で
あるかを見る。実験に用いた絵を図6に示す。
またタスクは、以下の順に行われる。

1. 3行目にある3つの四角形の色を変え
る (左=灰色, 真中=黄緑色, 右=ピ
ンク色)
2. 4行目にある3つの四角形の色を3行
目と同様にする [条件cの時, コマンド
の再利用を用いて]
3. 2行目にある3つの四角形の色を3行
目と同様にする [条件cの時, コマンド
の再利用を用いて]

被験者にはタスクの内容全てを見せず, タ
スクを小さな作業単位 (ステップ) に分割し,
それらを紙片に印刷したものを渡す。被験者
は1つのステップを終えてから次のステップ
に進むことができる。タスクはどちらも予め
与えられた絵の図形オブジェクトに色を付け
るものであるが, タスク中の色の指定に関し
ては RGB の値ではなく色名により行った。
また条件cではタスク中の特定のステップで
コマンドの再利用を使用するよう指示した。

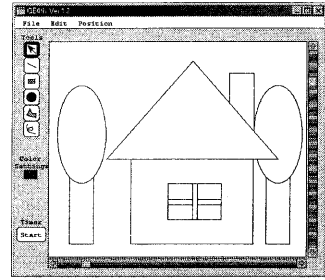


図5 タスク1で用いた絵

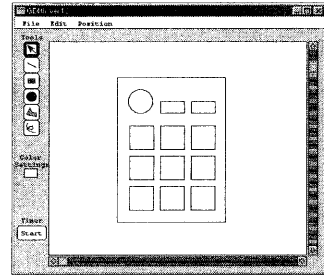


図6 タスク2で用いた絵

4. 1. 2 評価項目

コマンドの再利用の有効性を評価する為に,
ステップの処理時間とエラー率 (タスク達成
の正確さ) を測る。今回の実験タスクでは色
を塗るコマンドを扱っているので, タスク達
成の正確さとは RGB の値で表現された色の
違いとして考える。たとえば, ある図形の色
を元に戻すという作業を考えた場合, 再現さ
れた色が元に戻すべき色と異なっていればこ
れをエラーとし, RGB で表現された色の距離
でその度合を算出した。値は次の式によつて
与えられる。

$$\sqrt{(r_0 - r_1)^2 + (g_0 - g_1)^2 + (b_0 - b_1)^2}$$

ただし,

$$\text{図形の元の色} = (r_0, g_0, b_0)$$

$$\text{再現後の図形の色} = (r_1, g_1, b_1)$$

$$(r, g, b) \text{はそれぞれ} 0.00 \sim 1.00 \text{の値をとる}$$

とする

4. 1. 3 被験者

被験者は理工系の男子大学生及び大学院生
の計 10 名である。日常的にはグラフィック

エディタを使用しておらず、制限線形アンドゥ機構には使われていないが、ヒストリグラフに基づくアンドゥ機構を使用した経験は無い。各被験者は3つの条件全てにおいて実験を行った。

4. 1. 4 アンケート

被験者には全ての実験が終了した後、アンケートに回答してもらった。

4. 2 実験結果

各条件の下でのタスク1及びタスク2においてステップの処理に要した時間を図7及び図9に示す。またエラー率についてもそれぞれ図8及び図10に示しているが、グラフ中の記述に対応する図形オブジェクトを以下に記しておく。

- ・ 図8のグラフでの図形の対応 (図5参照)
 - 図形1 = 右側の木の葉部分
 - 図形2 = 右側の木の幹
 - 図形3 = 屋根の色
- ・ 図10のグラフでの図形の対応 (図6参照)
 - 図形1 = 4行目左の四角形
 - 図形2 = 4行目真中の四角形
 - 図形3 = 4行目右の四角形
 - 図形4 = 2行目左の四角形
 - 図形5 = 2行目真中の四角形
 - 図形6 = 2行目右の四角形

4. 3 考察

まず、タスク1の実験結果について述べる。図7より、コマンドの再利用を使用した条件cの場合、ステップ3と6の処理時間は他の条件のものより短くなっている。その他のステップでは操作は条件に関係なく同じ作業内容であるので処理時間にも差が見られない。ステップ3では条件cについて有意水準5%で有意な差が見られたが、ステップ6では有意な差が見られなかった。これは、ステップ6で与えられた作業を以前の操作の再現によって行った時間と、ヒストリグラフ中から必要な操作を見つけて行った時間が同程度であ

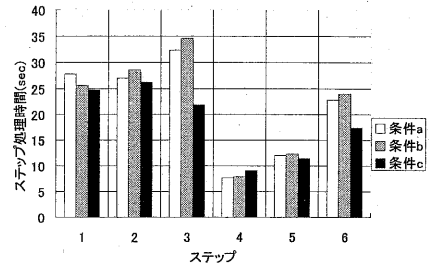


図7 タスク1における条件別タスク処理時間の比較

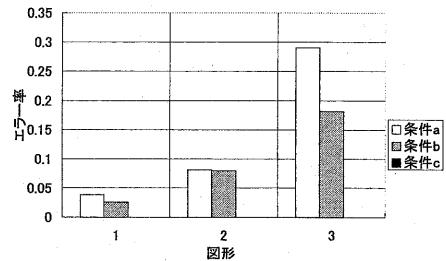


図8 タスク1における条件別エラーの比較

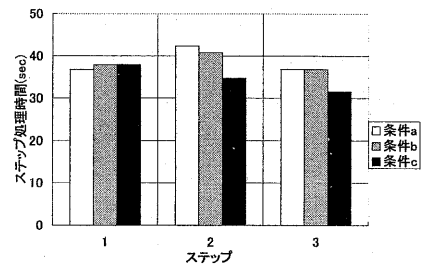


図9 タスク2における条件別タスク処理時間の比較

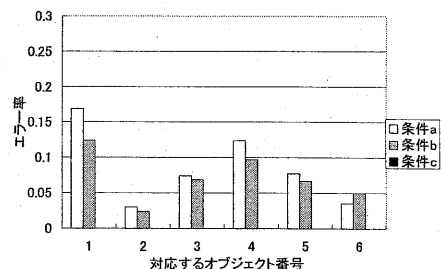


図10 タスク2における条件別エラーの比較

った為と考えられる。また図8より、コマンドの再利用を使用しない場合ではステップ3

及び6でエラーが見られ、操作のやり直しでは状態回復の再現性が低いことがわかった。特にヒストリグラフを使用しない条件aでは部分的な(ステップ6:屋根の色を元に戻す)アンドゥの際のエラーが顕著である。

以上のことから、部分的なアンドゥ・リドゥを行う作業にコマンドの再利用が有効に働いていると考えられる。

次にタスク2の実験結果について述べる。図9より、条件cのステップ2及び3で再利用を用いているので、ステップ処理時間が短くなっている。ただし、ステップ2及び3で条件cについて有意差は見られず、平均処理時間の差も大きくないので今回行ったタスク2のような簡単なタスクでは作業時間の短縮は期待できない。しかしながら、図10を見ると再利用を使用しない場合ではエラーが見られる。よって、操作を繰り返し行うような作業の場合でも、コマンドの再利用を用いると同程度の作業時間で過去の操作を正確に繰り返し適用できることがわかった。

最後にアンケートの結果について述べる。アンケート項目のうち、

- ・コマンドの再利用は便利か
 - ・再利用後の結果は希望だろうか
 - ・コマンドの再利用をまた利用したいか
- の3つについては特に評価が高かったが、
- ・再利用時の操作回数について
 - ・ヒストリグラフの強調表示について

の2項目の評価は低かった。その理由として、今回採用したインタフェースが、再利用したいコマンドを素早く見つけられるような提示法でなかったためと考えられる。また自由記述意見欄でも、ヒストリグラフのノードは全て円形なので希望のコマンドを探しにくいという指摘もあった。

5 まとめ

本研究では作業状態の遷移をグラフ表示し、アンドゥを過去の状態への回復としたユーザ

インタフェースであるヒストリグラフを利用し、コマンドの再利用を提案、評価した。実装したグラフィックエディタでは、始点と終点のノードをマウスのクリックで選び、プレビューを確認して適用するという手法で、コマンドの再利用を提供している。

実験の結果から、コマンドの再利用は部分的なアンドゥ・リドゥにおいて有効にはたらくが、操作を繰り返し行う作業の場合でも、従来法と同程度の時間で操作の100%の再現性を実現できることがわかった。またコマンドの再利用を使ってみて便利だと評価する人が多い反面、再利用したいコマンドを見つけにくいというアンケート結果を得た。

今後、コマンドの再利用で創造的な活動をより効率良く行えるようにインタフェースの改善を行いたい。

6 参考文献

- [1] Thomas Berlage : "A Selective Undo Mechanism for Graphical User Interfaces Based on Command Objects", ACM Transaction on Computer-Human Interaction Vol.1, No. 3, pp. 269-294 (1994)
- [2] 西田知博, 林真志, 辻野嘉宏, 都倉信樹 : "ヒストリグラフを用いたアンドゥ機構の提案と評価", 情報処理学会研究報告, Vol. 99, No. 69, 99-HI-84-12, pp. 67-72 (1998-08)