

# UnReC : 文字入力に基づいた Undo/Redo

大江 龍人    志築 文太郎    田中 二郎\*

**概要.** テキストエディタの Undo/Redo では、ボタンやショートカットを用いて履歴を一度ずつ操作する手間があり、Undo/Redo 操作後に文章を上書きした際 Undo/Redo が不可能な場合がある。この問題を解決するために、我々は履歴から任意の時点を選択し Undo/Redo を行うことが可能であるインタフェースの有効性を調査している。現在、この目的のために文字入力に基づいた Undo/Redo インタフェースである UnReC を開発している。UnReC を用いることにより、ユーザは過去に入力した文字列をキーとして履歴内を検索し、その時点まで Undo/Redo を行うことが可能である。本論文では UnReC を用いた操作手法と実装を示す。

## 1 序論

テキストエディタにおいて Undo/Redo を行う場合、ユーザは図 1a に示す様に線形の履歴から一度ずつ操作する。この Undo/Redo 操作において目的の時点へ Undo/Redo を行うためには、ユーザは複数回の操作を用いて履歴を探索する必要がある。またこの環境では、履歴から任意の時点を選択することが出来ないため、複数回の操作を用いて履歴を探索する手間がある。さらに、ユーザが Undo/Redo 操作後に文章を上書きした際 Undo/Redo が不可能な場合がある。

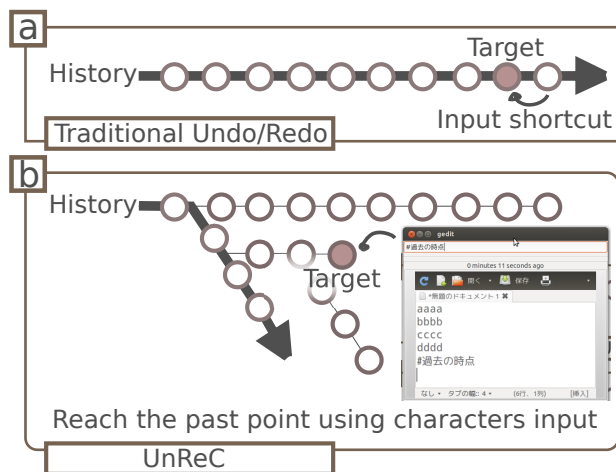


図 1. a) 通常の Undo/Redo , b) UnReC を用いた Undo/Redo .

Copyright is held by the author(s).

\* Tatsuhito Oe, 筑波大学大学院システム情報工学研究科, Buntarou Shizuki and Jiro Tanaka, 筑波大学システム情報系

<sup>1</sup> <http://www.emacswiki.org/emacs/UndoTree>

<sup>2</sup> [http://www.vim.org/scripts/script.php?script\\_id=3304](http://www.vim.org/scripts/script.php?script_id=3304)

履歴から任意の時点を選択する Undo/Redo を可能にした Selective Undo [1] は、選択した時点のみの Undo/Redo を可能にしたが、一般的な線形の履歴に対応していない。また Regional Undo [2] は、一部の範囲のみに Undo/Redo の対象を絞ることを可能にしたが、任意の時点に戻ることが出来ない。任意の時点に戻ることが可能である木構造の履歴に対応したものとして、Emacs の Undo Tree<sup>1</sup> や Vim の Gundo<sup>2</sup> が挙げられる。しかし、これらの環境においてユーザは、ある時点へ到達するために木を探索する必要がある。

我々は履歴から任意の時点を選択し Undo/Redo を行うことが可能であるインタフェースの有効性を調査している。現在、この目的のために文字入力に基づいた Undo/Redo インタフェースである UnReC を開発している。図 1b に示す様に UnReC を用いることにより、ユーザは過去に入力した文字列をキーとして履歴内を検索し、その時点まで Undo/Redo を行うことが可能である。なお、我々は UnReC の名前を Undo/Redo by Characters Input (文字入力に基づいた Undo/Redo) の頭文字から命名した。

## 2 UnReC

UnReC を実際に使用している例を図 2 に示す。図 2 に示す様に、ユーザはテキストエディタと共に UnReC を使用する。UnReC の文字入力フォームに文字列を入力すると、過去の文字入力時点のスクリーンショットが閲覧出来る。そのスクリーンショットをマウスクリックすることにより、その時点まで Undo/Redo することが可能である。

## 3 UnReC の応用

UnReC の応用として「簡易なバージョン管理」と「過去の文章の再利用」が挙げられる。それぞれ以下に説明する。

履歴から「コミット」と入力した時点を検索

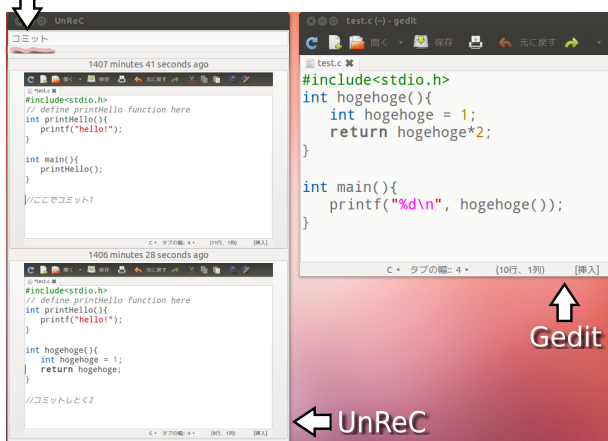


図 2. UnReC の使用例 . この例において、ユーザは「コミット」と文字入力した時点履歴から検索し、そこまで Undo を実行しようとしている .

UnReC を用いることにより、あらかじめメタな文字列を入力し、その時点まで Undo/Redo を行う簡易なバージョン管理が実現可能である . 例えば図 2 において、ユーザはコーディング中に戻りたい時点を実行するために「//ここでコミット 1」や「//コミットしとく 2」等の文字列を入力している . その後 UnReC の文字入力フォームに「コミット」と入力し時点を検索する . 最後に、出現したスクリーンショットをマウスクリックすることにより、コミットした時点まで Undo 可能である .

UnReC の別の応用として、図 3 に示す過去の文章の再利用がある . この例においてユーザはコーディング中に関数を定義し ( 図 3a ) , その後、関数が不要だと考え削除し、数時間コーディングを続けているものとする . ここで改めて関数が必要になった場合、ユーザは UnReC の文字入力フォームに関数名や定義時のコメントを入力し ( 図 3b ) , UnReC 上に表示されるスクリーンショットをクリックしさえすれば、過去の関数の参照及び再利用が可能である ( 図 3c ) .

#### 4 実装

UnReC の実装は保存、検索、Undo/Redo 処理から構成される . 今回、我々は Ubuntu 上の Gedit テキストエディタのプラグインとして UnReC を実装した<sup>3</sup> .

保存 : エディタにおいて 1 行分の文字列が入力されるごとに、システムはその時点における状態を保存する . 保存する内容は、「検索用の 1 行文字列」、

<sup>3</sup> svn checkout [https://subversion.assembla.com/svn/oe\\_public\\_projects/Gedit\\_UnReC](https://subversion.assembla.com/svn/oe_public_projects/Gedit_UnReC)

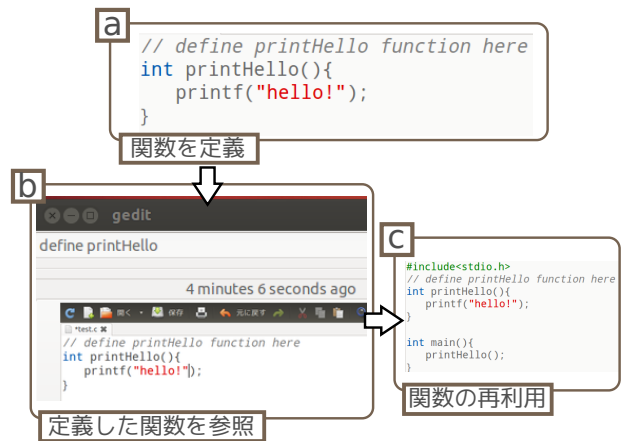


図 3. 過去の文章の再利用例 . a) コーディング中に関数を定義する, b) 必要な際に定義した関数を参照する, c) 過去の関数を現在のコードに再利用する .

「全文章」、「スクリーンショット」から構成される . この保存処理は、通常の Undo/Redo では削除される文章も全て保存するため、任意の時点における Undo/Redo が可能になる .

検索 : UnReC の入力フォームに文字列が入力されると、システムはその文字列が検索用の 1 行文字列に含まれているか否かの判定を行う . 含まれている場合、保存したスクリーンショットを表示する . この際スクリーンショットの順序は、1 行文字列と入力フォームの類似度順にて表示される . ユーザがスクリーンショットを選択すると、システムは Undo/Redo 処理を実行する .

Undo/Redo 処理 : 現在の全文章を全て削除し、過去の全文章を貼り付けることにより、Undo/Redo を行う .

#### 5 結論と今後の発展

本論文では、履歴から任意の時点を選択し Undo/Redo を行うことが可能であるインターフェースである UnReC を示し、さらに UnReC を用いた操作手法と実装を示した .

今後は UnReC を様々なユーザに使用してもらい、実際に使用したユーザの意見を反映させることにより、UnReC をより実用的なシステムとする .

#### 参考文献

[1] T. Berlage. A Selective Undo Mechanism for Graphical User Interfaces Based on Command Objects. *ACM Trans. Comput.-Hum. Interact.*, 1(3):269–294, 1994.  
 [2] Y. Kawasaki and T. Igarashi. Regional Undo for Spreadsheets. In *Proc. UIST 2004 Demonstration Abstract*, pp. 1–2. ACM, 2004.