

共有エディタにおける編集制御とテキストの構造化

5 Q-4

池井 寧 都築功兒 大川善邦
(大阪大学工学部)

1. はじめに

ワークステーションの普及に伴い、グループによる協同作業を
計算機で支援する環境への関心が高まりつつある。

計算機を利用するオフィスワークの1つに文書作成があるが、この文書の作成作業は基本的に協同作業である。文書は通常複数の人間の目を通して、添削、改訂を行った上で初めて完成するからである。また共同執筆の文書では、固より複数の人間がお互いの作成部分の調整を計りながら進められる。しかしながら、計算機上での現在の文書作成の実際部分、つまり文書の入力や変更は、メンバ各個人毎の作業に委ねられている。本来これと並行する作業、即ち内容について他のメンバの見解を求め協議しながら変更する作業を同じ計算機上で行うことは出来ない。通常これらの作業はメンバが面会して紙の上で行うため、最終的な文書化は再び個別に行わねばならず、必ずしも効率的ではない。

協議や変更には、文書の細部を会話と共にリアルタイムで指示したり、そこに文字や記号などの具体的表示を与えることが必要である。従って、単にその文書ファイルのコピーを両者が参照でき、音声の会話が可能だけでは正確な伝達は困難である。

本研究では1つの文書を2台のワークステーションに表示し、同時に2人が編集できる共有エディタを開発している。ここでは共有エディタの基礎的な機能を実現する制御法について述べる。

2. 共有エディタの基本的構成要件と設計仕様の限定化

同一のテキストに対し、複数のサイトで同時に編集操作を行う共有エディタの構成要件としては、①各サイトにおける局所的な編集操作が通常のエディタと同様に時間遅れなく処理される、つまり各サイトのエディタプロセスへの入力には到着順に処理される、②到着順処理により各サイトのテキストは一時的に差異を生ずるが入力終了後一定時間以内に両者は一致する、ことが必要である。

問題を簡単化するため、ここでは考察対象を次の様に限定する。

①編集対象は1次元有限文字列。②編集操作は、1文字単位の挿入と削除、および対象位置を示すキャレットの1文字移動のみ。
③テキストの各文字は一意識別可。④編集入力力は2つのサイトで発生。⑤各サイトでは2つのキャレットを表示。

3. 共有テキストの同時編集の基本的性質

(1) 使用語彙の定義

テキストの順位…テキストの文頭からの座標(値)を順位(数)と呼ぶ。また文頭方向を低位、文尾方向を高位とし、ある文字の低位側の隣接文字を低位文字、高位側隣接文字を高位文字と呼ぶ。

フォーカス(i,j)…テキストの現在の編集対象位置であり、文字列中の隣接2文字、例えばi,jの組で表す。このiとjを各々低位フォーカス、高位フォーカスと呼ぶ。挿入ならば挿入文字はi,jの間に置かれる。また削除の場合は低位フォーカスがその対象となる。

自己フォーカス…当該サイトのフォーカスであり、エディタのウィンドウ内では、キャレット (|) で表される。

共有者フォーカス…同時に編集しているもう一方のサイトのフォーカスであり、キャレット(！)で表される。

入力回数…ある時点のフォーカスに対して入力が複数回行われる時、 n 回目の入力を n 次入力と呼ぶ。更に、2サイトが共通のフォーカスに対して同時に入力するとき、当該サイトで n 次まで入力した後に、共有サイトで入力された m 次までの入力が到着する場合を $n-m$ 次入力と呼ぶ。

(2) 1-0 次入力時のフォーカスの更新

2サイトで1つのテキストを同時に編集する時、両者のフォーカスが離れていれば、編集入力のに到着順が逆転しても同一の結果を得る。従って、各サイトでの入力共有者フォーカスに影響する場合が問題である。自己フォーカスへの編集入力共有者フォーカス(ij)をも変更するのは、同じ(ij)への挿入時、或はi又はjの削除時である。削除時は、被削除文字の両隣接文字から新しい共有者フォーカスが一意に定まるが、(ij)への文字xの挿入時は、次表の2通りが可能である。つまり、共有者フォーカスが(ix)となる場合①と(xj)となる場合②が可能である。但しここで自己フォーカスは、高位方向へ連続して文字挿入が可能となるために必ず(xj)でなければならない(昇順入力条件)。つまり、挿入後のフォーカスは挿入文字の高位側であることが必要である。勿論これは次項以降の共有者フォーカスについても同様である。

	挿入サイトでの表現	共有者サイトでの表現
初期状態	$i \parallel j$	$i \parallel j$
挿入後①	$i \mid x \parallel j$	$i \mid x \parallel j$
挿入後②	$i \quad x \parallel j$	$i \quad x \parallel j$

(3) 1-1 次挿入入力時のフォーカスの更新

両サイトで各 x, y が同時に (i, j) に挿入された場合は、まず共通の結果としての x, y の順列を決定することが必要である。この順列の決定法には、一方のサイトを常に低位側、他方を高位側とするサイト固定順位形と、入力された文字 x, y に依存して順位を決定する文字依存順位形の2つが考えられる。ここでは紙面の都合からサイト固定順位形だけについて述べる。

サイト固定順位形で低位側となるサイトを低位サイト、その文字を x 、高位側となるサイトを高位サイト、その文字を y とする。昇順入力条件を考慮すると可能な結果は次の2通り (a, b) となる。

	低位サイトでの表現	高位サイトでの表現
初期状態	$i \parallel j$	$i \parallel j$
挿入後a	$i x \mid y \mid j$	$i x \mid y \mid j$
挿入後b	$i x y \parallel j$	$i x y \parallel j$

この1-1次入力の結果は、自己の挿入に続いて共有サイトの文字が到着して得られるから、自己の挿入だけを考えた1-0次入力の結果が、その中間段階となっている筈である。従って、1-0次から1-1次への変化の系列を、キャレットが不自然な移動をしない様に組み合わせると、

＜低位サイト＞

初期狀態	1-0次	1-1次	
i j	— i x j	$\swarrow$ i x y j (1-1) $\searrow$ i x y j (1-2)	

<高位サイト>

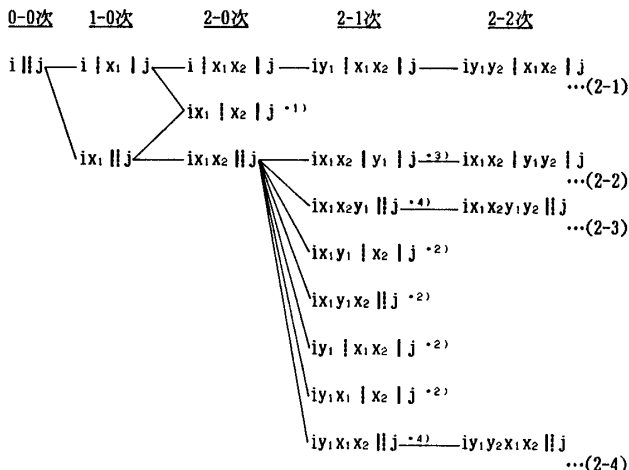
$$i \parallel j \leftarrow \begin{array}{l} i | y | j \\ i y \parallel j \end{array} \quad \begin{array}{l} \text{---} i x | y | j \\ \text{---} i x y \parallel j \end{array} \quad (1-3)$$

$$(1-4)$$

となる。この4通り以外は、共有者キャレットの無意味な移動を伴うのでエディタの制御法として好ましくない。(1-1)は、高位サイトの文字が入力された時点でフォーカスが分離する制御であり、これは(1-3)と対応している。(1-2)と(1-4)はフォーカスが同一のまま保存される制御である。つまり、1-1次入力に対するサイト固定順位形制御には、分離形と保存形の2種類のフォーカス制御が妥当であることが分かる。

(4) 2-2次以上の挿入入力

前項と同様に考えると2-2次の挿入結果は、1-0次、2-0次、2-1次、2-2次の挿入系列でフォーカスが無意味な移動をせずに導かれることが必要である。まず、前項で導入した低位サイト、高位サイトの区別をせずにこの系列を列挙すれば、次のようになる。



ここで2-0次については、挿入順位とキャレットの位置に関して全ての場合を挙げてある。*1)の場合は、共有者の挿入は無いにも関わらずキャレットが移動しており不適当であり、それに続く2-1次以降については検討していない。

2-1次では、順位とキャレット位置の制約を考慮すると7通りの結果が考えられる。 $i x_1 x_2 \parallel j$ に続く2-1次の結果にその総てを記載したが、この中で*2)の4つの場合は挿入文字とキャレット位置に関して一貫性の無い制御を必要とし、不適当である。これに対し*3)は、前項で検討した挿入時にフォーカスが分かれる分離形、*4)は保存形であると見做せる。

更に、2-2次では全部で14通りの結果が考えられる。この中で同様の選択をし、xとyの系列が対称となるように配慮すると、2-1次で妥当と考えられる各々に丁度1つずつが対応する。これらの結果の挿入文字の順位は、サイト毎に文字が隣接している点で共通している。サイト毎に高位側、低位側が入れ替わった2種類があるのは、前項の低位サイト、高位サイトの区別がこの場合も可能であることを意味している。つまり、(2-2)は低位サイトがxを入力し、高位サイトがyを入力する前項の割当てと同様であり、(2-1)はその逆の場合である。更にこれらは、共有サイトの挿入と共にフォーカスが分かれる分離形である。勿論、(2-3)、(2-4)が各々の保存形である。

以上より、2-2次以上の次数の入力に対しては各々のサイトの挿入文字を隣接させた順位が妥当な結果であり、1-1次のサイト固定順位形の拡張であること、更にフォーカスの制御では分

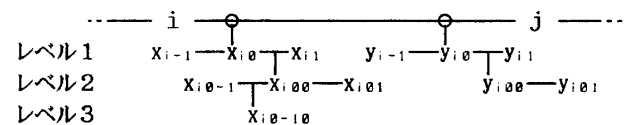
離形、保存形の各々が同様に存在することが容易に推察できる。従って、 $n-m$ 次サイト固定順位形挿入入力において低位サイトで x_n まで、高位サイトで y_m まで挿入した場合は、次に示す結果となると考えられる。

	低位サイトでの表現	高位サイトでの表現
分離形	$i x_1 \cdots x_n y_1 \cdots y_m j$	$i x_1 \cdots x_n y_1 \cdots y_m j$
保存形	$i x_1 \cdots x_n y_1 \cdots y_m \parallel j$	$i x_1 \cdots x_n y_1 \cdots y_m \parallel j$

4. 共有テキストの編集構造モデル

(1) 編集構造モデル

編集入力として、フォーカスの移動及び削除が含まれる場合は、当該サイトで指定したフォーカスが共有サイトでは既に同一の形で存在しない場合が問題となる。当該サイトにおいて既に变化したテキストに対して共有サイトの編集操作が正しく行われ、テキストの順位が互いに等しくなるためには、編集後のテキストの構造が一意に定義され、かつそれが編集要求の到着順に依存せずに構成されることが必要である。このためには、編集後に残る挿入文字の挿入位置が一意に定義されて、その位置がサイト毎に分離していれば良い。前項の結果を考慮すれば、編集前のテキストの各文字の間にサイト別に挿入位置が存在するモデルによってこの要求を満たすことが出来る。即ち、編集前のテキストの文字を基準文字 $\cdots i j \cdots$ とする時、フォーカス(ij)に対する低位サイトの挿入位置をi側、高位サイトのそれをj側とし、サイト毎の挿入文字を分離して順序付ければ良い。下図は挿入時のフォーカスの種類からレベルを分け、添え字に低位フォーカスとレベル内での順位数を用いて表示した例である。



ここで編集終了後のiに続く文字列は、 $i x_{i-1} x_{i0} x_{i0-1} x_{i0+1} x_{i00} x_{i01} x_{i1} y_{i-1} y_{i0} y_{i00} y_{i01} y_{i+1} j$ となる。

(2) フォーカスのタイプと挿入位置

共有サイトから指示されるフォーカスのタイプに対する挿入位置を上述の表記で示す。但し、順位iの基準文字を C_i とする。

フォーカス	低位サイト挿入座標	高位サイト挿入座標
$C_i C_j (i < j)$	$x_{i0}^{*1)}$	$y_{i0}^{*1)}$
$C_i x_{in} (n \leq 0)$	$x_{i(n-1)}$	$-^{*2)}$
$C_i x_{jn} (n \leq 0)$	$x_{j(n-1)}$	$-^{*2)}$
$x_{in} C_j (n \geq 0)$	$x_{i(n+1)}$	$-^{*2)}$
$x_{i1} x_{i(1+1)}$	x_{i10}	$-^{*2)}$
$x_{i1m} x_{i(1+1)}$	$x_{i1(m+1)}$	$-^{*2)}$
$x_{i1} x_{i1m}$	$x_{i1(m-1)}$	$-^{*2)}$

*1) $j=i+1$ 以外の時は複数の座標が可能であるが、ここでは最低位を用いる。*2)これらに対応する高位サイトの挿入要求は省略した。挿入後のフォーカスの位置は、基準文字が含まれる場合について分離形と保存形の区別をすれば良い。

$n-m$ 次入力だけでなく、当該サイトの入力と共有サイトの入力の到着が交互する場合も基準文字の更新を行うことで可能となる。基準文字の更新は、サイトの入力次数を付した編集要求が共有サイトから返送された時点で行えば良い。

5. おわりに

本論文では、共有エディタにおける共有テキストの妥当な処理形態を整理し、これを実現するための構造モデルを示した。