2X-4

垂直分散型 文書サーバにおける 文書管理・操作方式の検討

加藤謹詞　渡辺洋一

ＮＴＴ電気通信研究所

1　はじめに

ｵﾌｨｽにおけるﾊﾟｿｺﾝ,ﾜｰﾌﾟﾛの普及に伴い文書の共同作成のための共用化が必要となってきている。筆者らは、このようなﾆｰｽﾞに答え、大規模文書の共用化のための参照、更新が可能な文書ｻｰﾊﾞの開発を行なっている。

本稿では、大規模文書の共用化を行なう上で特に問題となる、①ﾍﾟｰｼﾞ／章・節管理、②複数ﾕｰｻﾞによる更新の競合、について問題点の整理と検討の結果を述べる。

2　大規模文書共用化の問題点

文書の共用化では、文書の各部分を複数のﾕｰｻﾞが頻繁に更新する運用形態をとると考えられるため、ﾌｧｲﾙｻｰﾊﾞのように一つのﾌｧｲﾙ（一文書）を最小の管理単位とするとﾀﾞｳﾝﾛｰﾄﾞ,ｱｯﾌﾟﾛｰﾄﾞに多くの時間を要してしまう。従って、文書ｻｰﾊﾞでは管理単位をﾍﾟｰｼﾞ／章・節のように更に小さな単位とすることが必要である。

しかし、ﾍﾟｰｼﾞ／章・節を管理単位とすることで、以下のような実現上の問題が生じる。

①ﾍﾟｰｼﾞ単位でｾﾝﾀに保管した後、ﾍﾟｰｼﾞの挿入を行なうとﾍﾟｰｼﾞ番号の振り直しが必要となり、ｵｰﾊﾞﾍｯﾄﾞが大きい。（図１参照）

②１ﾍﾟｰｼﾞの文書の内容を更新し、文字数が１ﾍﾟｰｼﾞを越えた部分をどのように管理するか。（図２参照）

③ﾍﾟｰｼﾞ／章・節の両方で文書を管理するには、どうするか。

④同じﾃﾞｰﾀに対する複数ﾕｰｻﾞによる更新に対しての排他制御は例(ﾃﾞｰﾀﾍﾞｰｽ等）があるが、文書ｻｰﾊﾞでは、ﾍﾟｰｼﾞ／章・節の文書構造の変更に対する排他制御も考慮しなければならない。

3　問題の分析と検討結果

(1) ﾍﾟｰｼﾞ番号の管理方法

各ﾍﾟｰｼﾞに、ﾍﾟｰｼﾞ番号を直接持たせるとﾍﾟｰｼﾞ操作後、各ﾍﾟｰｼﾞに対して、ﾍﾟｰｼﾞの振り直しが必要となる。そこで、各ﾍﾟｰｼﾞには、ﾍﾟｰｼﾞIDと呼ぶ識別子を持たせ、実際のﾍﾟｰｼﾞ番号とは、ﾏｯﾋﾟﾝｸﾞﾃｰﾌﾞﾙにより、対応させるようにする。ﾍﾟｰｼﾞの挿入、削除時にはﾏｯﾋﾟﾝｸﾞﾃｰﾌﾞﾙの書換えで済む。

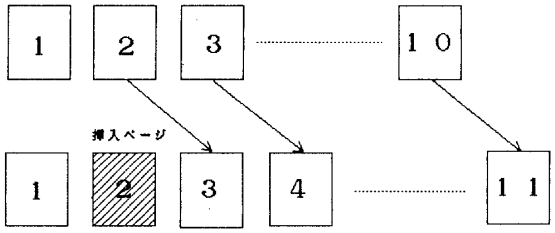

図１　ページ番号の振り直しの例

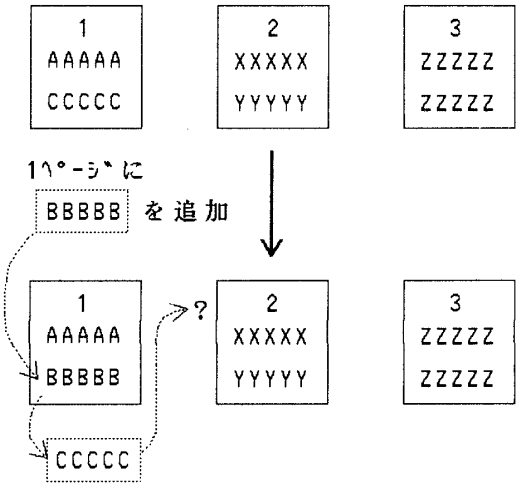

図２　１ページ内データのあふれ

例．既存の1～3ﾍﾟｰｼﾞに対して、2ﾍﾟｰｼﾞ目に新しいﾍﾟｰｼﾞ(ﾍﾟｰｼﾞID=4)を挿入する。

ﾍﾟｰｼﾞID		既存のﾍﾟｰｼﾞ	新しいﾍﾟｰｼﾞ
1	−	1	
2	−	2	→3
3	−	3	→4
4	−		2

(2) 1ﾍﾟｰｼﾞ内ﾃﾞｰﾀのあふれ

文書の更新を行うために文章を書き足していくと１ﾍﾟｰｼﾞに収まらなくなる場合がある。ﾊﾟｿｺﾝ,ﾜｰﾌﾟﾛでは、文書全体を一枚のﾛｰﾙ紙のように連続して管理しているので、基本的に1ﾍﾟｰｼﾞ内でﾃﾞｰﾀがあふれても問題とはならない。

しかし、ｾﾝﾀではﾍﾟｰｼﾞ単位の操作を容易にかつ高速化する目的で、文書を分割して管理するため、ﾛｰﾙ紙のような管理をするには、あふれたﾃﾞｰﾀを次ﾍﾟｰｼﾞへ挿入しなければならない。この挿入操作は、1ﾍﾟｰｼﾞのみで終らない可能性があり、最悪の場合、追加したﾍﾟｰｼﾞ以降の全ﾍﾟｰｼﾞに渡って、挿入の処理が必要となる。

Document Management Method for Document Server.
Kinji Kato, Watanabe Youichi
NTT Electrical Communication Laboratories.

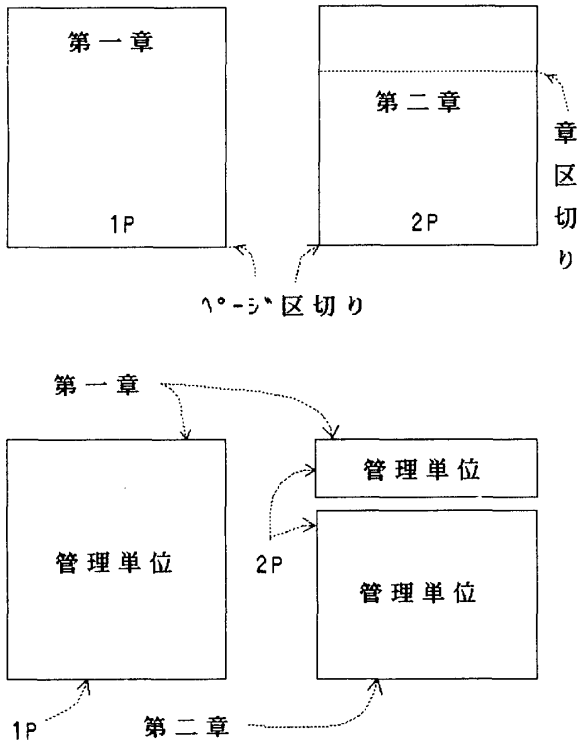

図３　ページ／章・節の管理単位

これらの処理をｾﾝﾀで行なうには、処理量の点から問題がある。そこで、簡易な処理でﾍﾟｰｼﾞあふれを処理するために、ﾍﾟｰｼﾞに収まらない部分を新たに1ﾍﾟｰｼﾞとして追加する方法をとる。この方法では、あふれた部分が1ﾍﾟｰｼﾞとなって追加されてしまうが、特別な処理は何も必要としないので容易に実現可能である。また、追加するﾍﾟｰｼﾞに枝番号を付与すれば、あふれた部分が後続のﾍﾟｰｼﾞに影響しない。

(3) ﾍﾟｰｼﾞ／章・節の両方で管理

ﾍﾟｰｼﾞは紙を意識した文書の物理的な分離概念であり、文書を印刷物として参照する場合等にﾍﾟｰｼﾞの管理が必要である。章・節は人間の視点からの論理的な分離概念である。

文書の一部を取り出すことを考えると、取り出すために必要な情報として、各ﾍﾟｰｼﾞに何が書いてあるかを示す目次や、章・節の題を参考とする。つまり、ｾﾝﾀ文書の管理単位を、章・節とする方がﾕｰｻﾞの操作に対応しており、ﾏﾝﾏｼﾝｲﾝﾀﾌｪｰｽが優れている。

また、ﾍﾟｰｼﾞ単位での管理はﾃﾞｰﾀの挿入や更新の度に目次を書き換えなければならないが、章・節単位の管理は、題の変更がﾕｰｻﾞ自身の要求によって行なわれるものであるから、ｼｽﾃﾑとして書き換える必要がない。

更に、ﾍﾟｰｼﾞ単位の管理は、上記(3)で述べたように、ﾍﾟｰｼﾞ内ﾃﾞｰﾀのあふれが生ずるが、章・節単位の管理では、論理的な単位となっているため、ﾃﾞｰﾀのあふれは生じない。

上記より、章・節単位の管理は、ﾍﾟｰｼﾞ単位の管理以上に有効であるため、ﾍﾟｰｼﾞ／章・節両方の管理が可能な方法について検討する。

ﾍﾟｰｼﾞ単位で管理するには、文書の管理単位をﾍﾟｰｼﾞ単位で分割しなければならない。同様に章・節単位で管理を行なうには、章・節単位で文書を分割する必要がある。これらを同時に実現する方法として、以下の方法が考えられる。（図３参照）

管理の単位として、ﾍﾟｰｼﾞで分割の後、更に、章・節で分割したものを単位とし、この分割単位にﾍﾟｰｼﾞIDと章・節IDを付与する。同じﾍﾟｰｼﾞIDを持つものは同じﾍﾟｰｼﾞとして管理し、同じ章・節IDを持つものは、同じ章・節として管理する。この方法では、管理情報を特に持つ必要がなく、ﾍﾟｰｼﾞ単位での管理方式に僅かな処理の追加により実現できる。

(4) 複数ﾕｰｻﾞのﾍﾟｰｼﾞ操作に対する排他制御

複数ﾕｰｻﾞが同一文書に対して、目次（ﾍﾟｰｼﾞ情報）の取得後、ﾍﾟｰｼﾞ操作を行なうと、下記のような矛盾が発生する。

	（ﾕｰｻﾞ1）	（ﾕｰｻﾞ2）
(a)	目次取得	
(b)	↓	目次取得
(c)	ﾍﾟｰｼﾞ操作	↓
	目次変更	↓
(d)		ﾍﾟｰｼﾞ操作
		目次変更

（a～dは時刻の推移を表わす。）

ここで、(d)での更新は、実際のｾﾝﾀの管理情報との対応がずれているため矛盾が発生する。この矛盾は、ﾕｰｻﾞ2が更新ではなく、ﾀﾞｳﾝﾛｰﾄﾞであっても発生する。つまり、目的のﾍﾟｰｼﾞを指定し、ﾀﾞｳﾝﾛｰﾄﾞしようとするが目的のﾍﾟｰｼﾞとは異なるﾍﾟｰｼﾞがﾀﾞｳﾝﾛｰﾄﾞされてしまうことになる。これはﾕｰｻﾞの操作として目次取得が単独で必要なのに対し、目次ﾃﾞｰﾀはﾍﾟｰｼﾞ操作と、密接な対応関係があるためである。

上記の矛盾を生じないようにするには、目次取得の操作が更新を前提にして行なわれたものであるか否かを区別し、更新を前提にしたものならば、上記(a)～(c)の間、更新ﾛｯｸをかける必要がある。次表に目次取得より、更新ﾛｯｸが必要な場合の組合せを示す。

先＼後	参照を前提	更新を前提
参照を前提	○	×
更新を前提	×	×

4　まとめ

本稿では、垂直分散型 文書ｻｰﾊﾞの文書管理・操作方式で問題となる、①ﾍﾟｰｼﾞ／章・節管理、②更新の競合に関して述べた。検討の結果、A.ﾍﾟｰｼﾞ番号の容易な管理方法、B.ﾍﾟｰｼﾞ管理におけるあふれﾃﾞｰﾀの処理方法、C.ﾍﾟｰｼﾞ／章・節の両情報での管理方法、D.文書ｻｰﾊﾞ特有のﾍﾟｰｼﾞ操作に対する排他制御の方法、が明らかになった。