内容の部分共有を許す文書データベースにおける更新機構

4W－1

河野章博　長島正明　山川正
キヤノン株式会社　情報システム研究所

1　はじめに

近年，DTP の普及などに見られるように文書を電子化して扱うことが多くなってきた [1]．また，データベースの普及により，今まで扱われていた数値や名前などの情報と同様に，文書で表現されている情報もデータベースで管理させる要求があげられている．我々はこの要求に対し，文書として表現されている情報そのものをデータベースに蓄積し，そして最終的にその情報を提供する時点で，表現が文書となるように処理する方法を提案する．今回はこの文書データベースにおける，文書の内容の更新機構について考察した．

2　マルチドキュメントと排他処理

我々は，構造化文書を，文書の単位ではなく，その文書要素の内容を単位としてデータベースに蓄積し，検索，加工などの処理をする OoDIeT(Object Oriented Document Integration Environment & Tools) を開発している．そしてこれらの蓄積された文書要素で構成される情報を，スタイル情報を用いて，最終的に文書の形になるように表現する (図 1)．

ここでいう文書とは，データベースの view とみなせる．OoDIeT では図 1の文書 A と文書 B のように，文書としては全く別，つまり全く別の view であるが，それを構成している文書要素内容を，一貫性を保って共有することができる．このようなデータ利用の形態を，我々はマルチドキュメント環境と呼んでいる．

マルチドキュメント環境では，複数の view が，ある文書要素の内容を共有している．この view で与えられた文書を処理し，データを更新しようとする場合，この共有された文書要素の内容に対しての排他処理が不可欠となる．この排他処理を実現するためには以下の点の考察が必要である．

- lock を行なう単位
- lock をかける機構

Update Mechanism for Shared Document Database with Partial Content Sharing Capability

Akihiro KOHNO, Masaaki NAGASHIMA, Tadashi YAMAKAWA
Information Systems Research Center, Canon Inc.

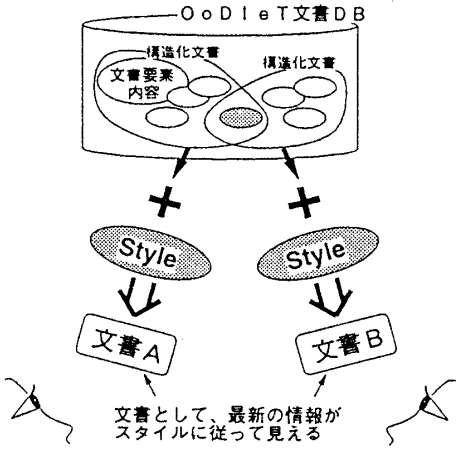

図 1: OoDIeT の処理のイメージ

通常，文書単位で文書を処理する場合は，文書を単位として lock させれば十分であるが，マルチドキュメントの場合に不必要な lock が出てきてしまう．例えば doc1 と doc2 がある文書要素の内容を共有しているとして doc1 に lock をかけると，共有部分にも lock がかかり，ひいては doc2 にも lock がかかってしまうという無用な lock の波及が起こってしまう．この lock の波及を起こさないためには，共有部分に関して通常の文書単位の lock とは別に，その共有内容を単位として lock する必要がある．

また，実際に文書処理を行なうことを考えると，処理対象となっていない文書要素内容の lock は不要である．この処理対象とは，文書中で選択，特定されている文書要素列のことで，これが実際に削除や複写の対象となる．

これらを実現するためには，共有された文書要素内容を切り分けて別の lock の単位にする機構が必要になる．また，文書処理中に処理対象となった文書要素のみに lock をかける機構が必要である．

3　OoDIeT アーキテクチャ

前述の排他処理機構を実現するために我々は OoDIeT で，文書要素と文書要素内容を切り分けて扱う文書モデルを採用した．また，文書情報の実データを文書要素の内容を単位として格納し，そのデータにリンクを張ることによって共有を行

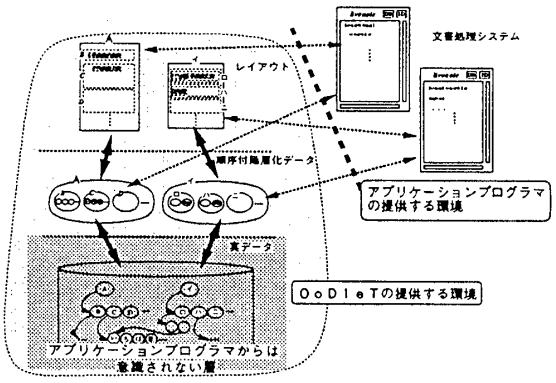

図 2: OoDIET の文書処理モデル

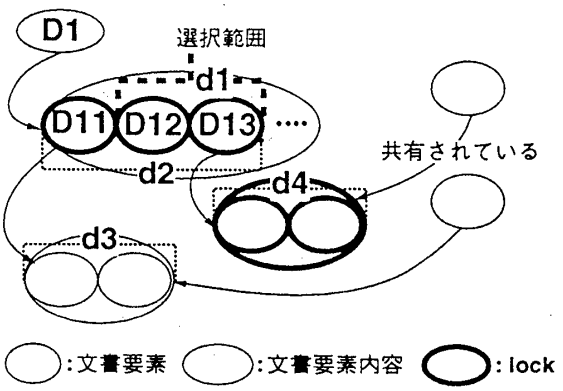

図 3: OoDIET の排他処理モデル

ない，マルチドキュメント環境を提供した．

アプリケーションプログラムにとって，文書は順序付階層化データというモデルとして提供される．文書は，文書要素列と，文書要素の内容として表現される．これはアプリケーションにとってのデータの view であり，共有などの実データにおけるデータの関連は意識されない．アプリケーションは 2 次元レイアウトを提供しているレイアウトモデルをインタフェースとして，順序付階層化データを処理する (図 2)．

前述の排他処理機構を含め、以上のことより，OoDIET アーキテクチャでは以下のことを採用した．

1) マルチドキュメントとその排他処理を実現するために文書要素と文書要素内容を切り分ける．
2) 実データに対して，順序付階層化データという view を提供しアプリケーションから実データを切り離す．
3) lock の単位として，文書固有の文書要素内容群と，共有される文書要素内容の 2 つの単位を用い，これらを自動的に切り分け，排他処理を行なう．
4) 選択された文書要素列を含む内容の lock と，選択された内容全てに波及する lock の 2 つの lock 機構を持たせる．

OoDIET では，文書要素の内容を単位として文書処理を行なうため，図 3に示すように，選択された文書要素列 d1 を含む文書要素 D1 の内容全体の文書要素列 d2 は lock される．この際，OoDIET での共有は文書要素の内容を共有しているので，選択範囲外にある文書要素 D11 が共有された内容 d3 を持っていても，この内容までは lock が波及しない．つまり 1) により，文書要素 D1 の内容文書要素列 d2 に含まれる文書要素 D11 は lock されるが，その内容 d3 は lock されない．

一方，選択された文書要素列 d1 に，共有された内容 d4 を持つ文書要素 D13 が含まれる場合，その内容は全て処理対象とみなされるので lock される．この際に 4) の lock が波及する機構が作用する．

実際のデータベースで排他処理がセグメント単位で行なわれる場合，1 つの文書が登録される際は，その含む全ての文書要素と文書要素内容を同一セグメントに生成する．実際に共有するとき，まず offer という順序付階層化データモデルの提供する演算によって，選択された文書要素内容を共有の対象とする．この際，この文書要素内容と文書要素のセグメントが同一の場合，文書要素内容を別の新たなセグメントに移し共有の対象として提供することによって，3) を実現している．

アプリケーションは，これらの処理を全く意識せずにプログラムできる．これは我々のアーキテクチャが 2) を採用しているからである．

4 まとめ

文書要素を単位としているモデルでは，マルチドキュメント環境を提供することが難しい．マルチドキュメント環境を実現する更新機構の考察より，文書を文書要素の内容を単位として処理する文書モデルを採用しマルチドキュメントをサポートした文書処理システムを実現することができた．

参考文献

[1] 山川 正, 川端 洋一, 田村 秀行,「OA 業界から見た DTP」, 情報処理, Vol. 31, No. 11, pp.1508-1517, 1990