

マルチポイント協同編集システムColleagueの実装

田中俊昭 山田満

国際電信電話株式会社 研究所

レイアウトされたマルチメディア文書に対して、複数利用者から同時にリアルタイム協同編集を行う機能の実現を目的として、グループ利用者のためのマルチメディア協同文書編集アプリケーションColleague (Collaborative document editing application for group users environment)を実装した。本システムは、(1)構造化文書を用いて文書構造のアクセス管理を行い利用者の役割分担を可能とする、(2)複数操作権の利用及びレイアウトなどの文書処理の制御により複数利用者での同時編集機能を提供する、などの特徴を有する。本稿では、上記の特徴を実現する詳細なメカニズムについて述べるとともに、利用評価を行い、その有効性を示す。

Implementation of Multipoint Joint Editing System : Colleague

Toshiaki TANAKA and Mitsuru YAMADA

KDD R & D Laboratories

toshi@raku.osi.kddlabs.co.jp

2-1-15 Ohara Kamifukuoka-shi Saitama 356, Japan

The goal of this paper is to realize synchronous joint editing of laid out document containing multimedia. Aiming at the goal, we have implemented collaborative document editing application for group users' environment (Colleague). The system has the following features: First, by assigning the access right to all the document structures, two or more users can play different roles on document production during the conference. Second, by managing multiple tokens and controlling document process such like laying out, users can edit the different part of the shared document simultaneously. In this paper, we discuss the mechanisms to realize the above features, evaluate the Colleague, and show its feasibility.

1. はじめに

近年、コンピュータ技術、ネットワーク技術の発達により、離れたところに存在する複数の利用者が、卓上のパソコンやワークステーションを用いて協同作業を行なうグループウェアの検討がさかに行なわれている¹⁾。この中でも、電子的な文書（以下、単に文書と呼ぶ）の作成をグループ作業の目的としたアプリケーション（AP）は、実時間型、非実時間型の観点から様々な提案がなされている。前者は、遠隔会議システム^{2),3)}などの観点からマルチメディアサービスの有望なアプリケーションと考えられており、また、後者は、電子メールをベースとした協同執筆⁴⁾やワークフロー⁵⁾等のシステムが報告されている。また、標準技術をベースとしたグループウェアも標準化活動のなかで検討が進められている^{6),7)}。

文書をベースとしたグループウェアシステムとしては、Quilt⁴⁾が先駆的であり文書の構造化を用いて利用者レベルに従った簡易なアクセス権を設けた編集機能を非実時間型の電子メールをベースとして実現した。また、GROVE¹⁾では操作イベントに競合が生じた場合のイベントを変換するコンカレンシ制御を用いて同一箇所の同時文書編集を可能にしている。しかしながら、QuiltやGROVEは扱う文書がテキストのみでありマルチメディア文書を扱うまでには至っていない。

筆者らは、これまで、構造化されたマルチメディア文書をベースとしたポイント-ポイントでの文書会議システムODAWorkの開発を行った⁸⁾。本稿ではさらにODAWorkを3者以上の複数利用者環境へ拡張したColleagueシステムの実装について報告する。ここで、本稿のポイントは、3者以上の利用者における効率的な協同編集の実現にある。具体的には、（1）文書内での各利用者の役割設定を可能とするための国際標準ODA（開放型文書体系）⁹⁾の文書構造に基づくアクセス権管理メカニズム、（2）グループ内で共通に用いる文書（以下、共有文書とよぶ）の同時編集を可能とするためのODAの文書構造に基づく操作権の管理法、（3）文書処理モデルで規定される編集処理、レイアウト処理、可視化処理に関する協同編集下での効率的な制御方法、を新たに提案する。

本稿の構成は次の通りである。まず2章において本システムの概要をのべ、次に3章において本稿で解決する問題を明かにするとともにその解決

法を提案する。さらに4章において本方式を実装したColleagueシステムについて述べ、最後に5章において実現システムの評価を行う。

2. Colleagueの概要

2.1 システムの設計指針

以下に本システムの設計指針を示す¹⁰⁾。

（1）分散方式による協同文書処理の実現

協同文書処理の実現形態としては、協同文書処理プロセスを一箇所に置き、各端末にその結果を通知することで複数利用者間の協同文書処理を実現する“集中方式”と、各端末がそれぞれ文書処理プロセスを保有する“分散方式”¹¹⁾が存在する。本システムでは、比較的低速度のN-ISDNや10MbpsのLANなどでデータ量の多いマルチメディア文書を共有文書として利用するために、協同編集時に端末間での転送データ量の少ない分散方式を採用する。

（2）コマンド方式による遠隔地間の文書操作

遠隔地間での文書操作の実現方法としては、端末の入力デバイスであるマウスやキーボードのイベント信号をローカル端末と同時に遠隔端末にも送出する“イベント方式”と、各入力イベントをCreate（作成）、Delete（削除）、Modify（修正）などの論理的な文書操作コマンドに対応させたPDU（プロトコルデータユニット）として遠隔端末に送出するコマンドベースの方式が存在する（以下、コマンド方式とよぶ）。本システムでは、相互接続性を考慮してITU-Tで検討が進められている文書操作のプロトコルであるDTAM（文書転送及び操作）¹¹⁾で規定されるサービスプリミティブに基づく“コマンド方式”を採用する。

（3）MCU（多地点制御装置）を介した多地点接続の実現

多地点構成の実現方法としては、MCUが中央にある“スター型”及び、各端末がそれぞれ他の端末と直接ポイント-ポイント接続をすることにより多対地構成を実現する“メッシュ型”が考えられる。本システムでは、端末での処理を軽減させることを考慮してMCUを介した多地点構成を実現する。多対地接続を実現するプロトコルは相互接続性を考慮してITU-Tの国際標準であるAGC（オーディオ・グラフィック会議）の多地点通信サービス（MCS）¹²⁾で規定されるサービスプリミティブに基づくこととする。

2.2 システム構成

本システムの構成を図1に示す。本システムは1つのMCUと2者以上の端末から構成される。MCUと端末間はそれぞれTCPコネクションが確立される。各端末はAPとしてODA準拠のワードプロセッサである文書処理部、DTAMの編集操作サービスを提供する編集操作プロトコル部、会議の召集/参加/退席等の管理を行なう会議制御部及び、MCSの多対地通信サービスを提供する多地点プロトコル部から構成される。一方、MCU上には特定のAPが存在せず、多対地制御を行う多地点プロトコル部及び会議制御部から構成される。

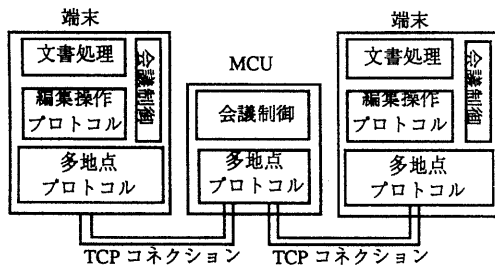


図1 システム構成図 (2端末の場合)

3. 協同編集機能の実現手法

本稿でのポイントとなる協同編集を効率的に行なうには、会議の運用形態、編集機能などの観点から以下の要件を満足する必要がある。

要件1) 文書内の役割設定機能:

利用者Aが1章を、利用者Bが2章を、利用者Cが全体の構成を担当する協同執筆などでは、利用者の役割を設定する機能が必要になる。このため、共有文書内に役割が設定可能なアクセス管理メカニズムを検討する。

要件2) 複数利用者の同時編集機能:

ブレインストーミング等では各利用者がアイデアが浮かんだ時点で共有文書へ反映できるよう、複数利用者が同一の共有文書を同時に編集する機能が必要となる。このため、操作権の管理メカニズム及び、レイアウト処理/可視化処理などの文書処理の制御メカニズムを検討する。

上記の機能を実現する具体的手法を以下に示す。

3.1 文書内のアクセス管理

文書への操作の種類は、Create, Delete, Modifyなど一般的に文書自体に変更を伴う操作、及び

Readのように変更を伴わない操作がある。本稿では、前者の実行可能な権利を一括して"アクセス権"とよび、各利用者に与える。一方、後者のReadする権利は暗黙的に全利用者に付与する。ここで、柔軟な役割設定を可能とするためアクセス権は文書構造である各オブジェクト毎に割当て、複数利用者からの操作を考慮して各オブジェクトにアクセス可能な利用者のリストを設けることとする。すなわちアクセス権とは、"文書内の各オブジェクトに対して、アクセス可能な利用者に与えられる静的な(会議中は不変の)権利"を表す。この際、文書構造のレベルの異なるオブジェクトに割り当てられるアクセス権に関して、運用上、以下の場合を満足するアクセス権の管理法が必要となる。

- ・ 2章全体は利用者Bの担当で、2章に属する2.1節は利用者Aが担当する。
- ・ 2章のアクセス権を保有することにより、2章の下位に属するすべてのオブジェクトのアクセス権を暗黙的に保持する。

前者の要求を満足させるために文書構造においては下位レベルのアクセス権を優先することとした。また、後者の要求を満足させるために省略時値(Default)規則を適用し、あるオブジェクトのアクセス権が省略された場合には、その上位のアクセス権をInherit(継承)することとした。これを実現するため、各オブジェクトが保有するアクセスリストの値として任意の数の"参加者識別子(ParticipantID)"、事前に指定された"グループ識別子(GroupID)"、すべての利用者がアクセス可能な"all"、及び省略時値"Default"の4種類を指定可能とした。例えば、図2のように項1 (ObjectID=3_0_1_0)は、アクセスリストが省略(Default)されているために、その上位オブジェクト節2 (ObjectID=3_0_1)のアクセスリスト(利用者Cのみがアクセス可能)を継承する。また、節1 (ObjectID=3_1_0)は、allが指定されているが、その従属オブジェクトの項1 (ObjectID=3_1_0_0)のアクセスリストがAのみであるので、上位オブジェクトの属性に関係なく上記項1は、Aのみがアクセス可能となる。このように構造上のレベルが異なるオブジェクトに対しても、個別にアクセス権の指定が可能となるので、種々の形態の協同編集が可能となる。但し、ObjectIDはオブジェクト識別子を表し、図2のよ

うに文書の構造最上位から該当するオブジェクトに至るまで各階層で一意に付与されるオブジェクトの番号を順次に並べたものである。例えば、図2中の2章に属する節1は、論理ルート(3)、章2(1)、節1(0)を結合した3_1_0がObjectIDとなる。

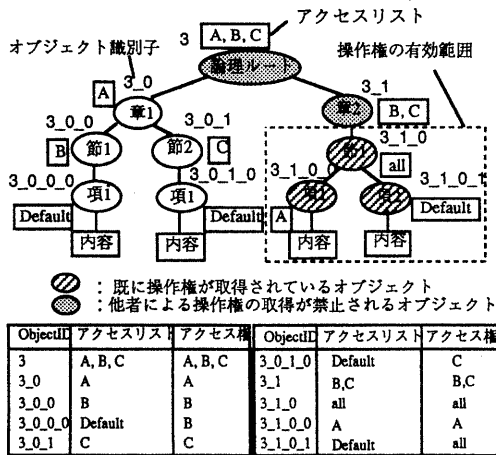


図2 アクセス権と操作権の例

3.2 操作権の管理

協同編集の一貫性を保証しながら実時間型システムを実現するには、複数の利用者が文書内の同一オブジェクトに対して時分割で操作する必要がある。この際、ある時点においてはアクセスリストに登録されたなかで一人の利用者だけがそのオブジェクトに対してアクセス可能となりえる。本稿では任意の時点であるオブジェクトに対して唯一の利用者が取得できるアクセス可能な権利を"操作権"とよぶ。即ち、操作権は文書内の各オブジェクトに対応でき、ある時点におけるアクセスリストのなかの一つの実体(Instance)であると言える。本システムでは、簡単化のため、各利用者は1度に1つの操作権だけを取得できることとし、各利用者は操作したい箇所(文書中のオブジェクト)を指定して操作権の取得要求を行う。ここで、各利用者からの操作権の取得要求は、MCSプロトコルとの整合性を考慮してMCUが一括管理するのが望ましいが、以下の問題が生じる。

・複数の利用者からの操作権の取得要求に対して要求された操作権の適用範囲が競合する可能性がある。ここで操作権の競合を判断するためには、共有文書の文書構造を理解する、すなわち、文書処理APをMCU上へ実装する必要がある。しかし

ながら、MCUは図1に示すように特定のAPを保有せず汎用的な多対地制御のみを行うので操作権の競合管理が困難となる。この問題点を解決するために、以下の方式を提案する。

各操作権をオブジェクトに割当られるIDを用いて識別する。MCUでは各端末からの操作権の取得要求に対して取得されているオブジェクトと取得したいオブジェクトとの間に以下の関係がある場合、競合状態にあると解釈し、操作権の譲渡を拒否する。

$\{GrabbedObjectID_i \in Prefix(RequestedObjectID)\}$
 $\vee \{RequestedObjectID \in Prefix(GrabbedObjectID_i)\}$
 ここで、Prefix(A)は、Aの値(文字列)の左端から任意の数の文字列を取り出した文字列の集合を表わす。例えば、Prefix(3_0_2)={3_3_0, 3_0_2}となる。また、GrabbedObjectID_iは利用者iによって既に取得されたオブジェクトのオブジェクト識別子を、RequestedObjectIDは、取得したいオブジェクト識別子を表わす。

このようにMCUに少しの修正を加えることにより操作権の競合を管理することが可能となる。例えば、図2に示すように節1(ObjectID=3_1_0)が取得されている場合、その祖先である章2(ObjectID=3_1)、あるいは、その子孫である項1(ObjectID=3_1_0_0)などのオブジェクトの操作権は他者によって取得できない。

3.3 協同文書編集に適した文書処理部の制御

文書処理部の構成を図3に示す。文書処理部にはODAの文書処理モデルに従う文書処理カーネルが含まれる。本モデルは、文書の論理構造や文書内容の編集を行う編集処理部、論理構造からレイアウト構造を作成するレイアウト処理部、レイアウトされた文書を画面や紙に表示する可視化処理部から構成される。各処理部は、文書編集を行う度に編集処理部、レイアウト処理部、可視化処理部の順で逐次的に起動される。本システムでは、一貫性を保ちながら効率的に協同編集を行うために、各処理部に以下に示すような制御メカニズムを設けた。

(1) 編集処理の制御

編集処理部では、図3に示すようにローカルな利用者から編集イベントが生じる、あるいは、遠隔の端末から編集要求を受信すると、ローカルコ

マンドにより編集処理を起動し共有文書を更新する。この際、コマンド方式ではローカルに生じたコマンドを標準の通信サービスに変換する、あるいは受信の際にその逆変換を行う"通信サービス変換処理"が必要となる。本システムで用いるDTAM通信サービスは表1に示すように一般的な文書処理のコマンドと等価な通信サービスが提供されており、基本的にはローカルなコマンドからDTAM通信サービスへは1対1にマッピングされる。しかしながら、カット/ペーストにおいては例外的に以下の問題点が生じる。

ある利用者が他APなどで作成したローカルデータをペーストして共有文書に取り込む際には、カットバッファの内容を他の端末に送信する"CREATEサービス"を用いる。一方、共有文書内あるいは共有文書間のカット/ペースト処理の際には、各端末が同一の動作を行うのでペースト処理はカットバッファの内容を共有文書にペーストする"COPYサービス"を用いて実現できる可能性がある。しかしながら、通常ウィンドウシステム等では利用できるカットバッファは一つに限られるので、本システムのように複数の利用者が同時に編集するとカットバッファの同一性が保証されない。さらに、ウィンドウシステムの汎用的なカットバッファを用いるかぎり、カットバッファに蓄えられたデータがローカルデータであるのか共有文書から切り出したデータであるのかの区別が不可能である。

従って、本システムではローカルデータまたは共有文書の切り出しデータの如何にかかわらず、利用者からのペースト要求に対しては、"CREATEサービス"を用いてカットバッファのデータ内容を他の端末に同報転送することとした。すなわち、図3に示すようにローカルな編集処理部へのペースト要求を

PasteObj (obj_{id} , position)

とすると、通信サービス変換処理部においてこの要求を以下の通信サービス

Create(obj_{id} , position, objectdata)

に変換して発行する。ここで、 obj_{id} は作成すべき位置を表すオブジェクト識別子を、positionは作成すべきオブジェクトが上記の位置を表すオブジェクトの前後いずれかを指示する引数を、objectdataはカットバッファに保管された作成すべき文書データをそれぞれ表す。

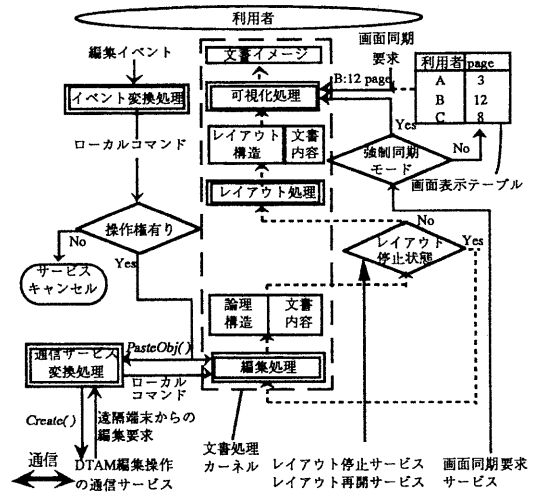


図3 協同編集に適用可能な文書処理部の構成

表1 編集操作のための主なDTAM通信サービス

		文書データの流れ
GET	オブジェクトの取得	remote -> local
SEARCH	オブジェクトの検索	none
CREATE	オブジェクトの作成	local -> remote
DELETE	オブジェクトの削除	none
MODIFY	オブジェクト属性の修正	none
COPY	オブジェクトの複写/貼込	none
MOVE	オブジェクトの移動/切取	none
POINT	オブジェクトの指示	none
RESERVE	オブジェクトの予約	none

(2) レイアウト処理の制御

レイアウト処理は、文字や図形などの文書内容を最初から順次にページやカラムなどに流し込む処理を行う。すなわち、文書の内容が変更されるそれ以降の内容はすべて再レイアウトされる。例えば、3ページ目の第1行を削除すると、レイアウト処理により3ページ以降全て1行繰り上げてレイアウトされる。従って、文書のボリュームが非常に大きい場合やDTPのような高度なレイアウト機能を持つ場合には、文書全体をレイアウトするのに一般的に時間を要する。このようなレイアウト処理を含んだ文書処理APを協同作業環境で適用する場合、遠隔端末から文書が頻繁に変更されるとその度に文書処理部において再レイアウトが行なわれ、受信側利用者からの編集要求が妨げられるという問題点が生じる。この問題を解決するため、以下の方式を提案する。

・ランダムに発生する各利用者の編集要求をMCUが蓄積し、送信側利用者が共有文書に対して更新を要求すると、各受信端末に一連の編集要求をまとめて送出できる構成とする。また、受信端末では、連続して受信する編集要求毎にレイアウト処理を起動せず、編集処理とレイアウト処理を非同期に実行する。すなわち、複数の編集処理を行った後、1回のレイアウト処理にまとめて実行し、その結果、複数の編集処理を反映できる構成とした。

具体的なレイアウト処理の制御及びプロトコルを図3、図4にそれぞれ示す。利用者1からの編集要求(図4 a,b)をMCUで蓄積し、その後更新要求を送出する(図4 c)。この際、MCUはレイアウト停止要求を各受信端末に送出し、レイアウト処理を停止させ(図4 d)、一連の編集要求をすべて送出する(図4 a',b')。この間は図3に示すようにレイアウト停止状態のために編集処理の結果がレイアウト処理を起動せず、編集処理のみを順次行い、論理構造や文書内容を更新する。次にレイアウト再開要求を各受信端末に送出する(図4 e)と、作成された論理構造/文書内容に基づき1回のレイアウト処理が受信側端末(端末2、3)で実行され、その結果図3に示すレイアウト構造/文書内容が作成される。

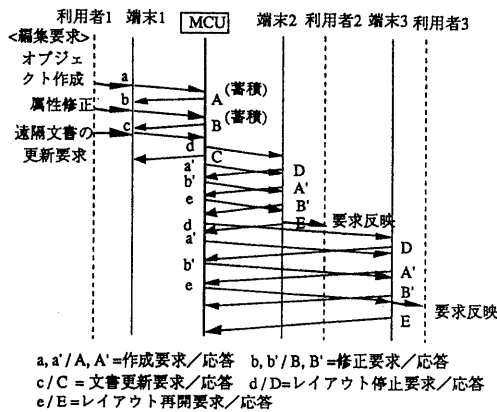


図4 遠隔文書編集プロトコル手順の例

(3) 可視化処理の制御

各利用者が同時に協同編集を行う場合、所望のページを閲覧/編集できる"自由閲覧モード"が必要となる。さらに、実時間型APでは遠隔の利用者がどのページを閲覧し、どのような編集を行っ

ているかを知る"強制同期モード"も必要である(この場合、ローカルな編集は禁止される)。さらに、これら両モード間をスムーズに移行できるメカニズムが必要となる。

・本システムでは、上記両モードを実現するため、図3に示すように他の端末から受信する画面同期要求サービスに対して、強制同期モードの場合はそのまま可視化処理を起動させ要求のあったページを表示し、自由閲覧モードの場合には各端末の表示ページを画面表示テーブルに登録しておく。強制同期モードへ切り替える画面同期要求がローカルの利用者から生じた場合は、画面表示テーブルを参照して同期させるべき利用者の表示位置を得た後、可視化処理を起動する。

4. Colleagueシステムの実装

本システムはSUN SPARC(OS: SUN OS 4.1.x)上に実装を行った。複数のワークステーションをEthernet LANで接続し、MCUを構成する多対地プロトコル部と会議制御部は常駐プロセスとした。また、MMIはOpenWindowsを用いて実現した。図1に示す各モジュールはC言語を用いて開発した。音声や動画については既存ソフトウェアをマルチウインドウで本システムと同時に起動させ実現した。図5に本システムの画面イメージを示す。

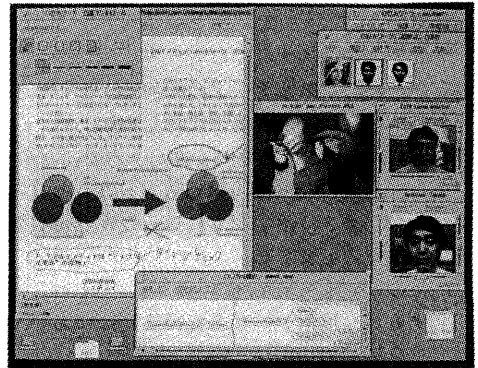


図5 Colleagueの画面例

・操作権のMMI

本システムでは、画面上で編集したい文書の位置にキャレット(カーソル)を置く、あるいは文

書構造ブラウザに表示されるオブジェクトをマウスで選択すると、そのオブジェクトに対する操作権を取得するための操作権要求プロトコルを発行する。また、キャレットを解放する、または、別のオブジェクトにキャレットを移動すると、先に取得された操作権が解放される。ここで、他の利用者がすでに取得しているオブジェクトについては、図6に示すように画面上でその領域にカラーで網掛けを行い、その部分が選択不可能であることを利用者に通知する¹⁾。これにより、利用者が意識せず操作権の授受がスムーズに行なわれ、通信時に係わる操作権の手続きを簡略化できる。

・アノテーション機能

文書編集機能に加えて、手書き注釈が可能なアノテーション機能を設ける。具体的には、ワープロ用の文書プレーンとアノテーション用のアノテーションプレーンを設け、2つのプレーンを透過的に重ねることにより、共有文書上に便宜上、手書きのコメントを追加可能とする。図6にアノテーションの様子を示す。

・画面表示モードの切り替え

自由閲覧モードから強制同期モードに移行するには、画面に表示された利用者の顔(静止画)をマウスで選択することにより当該端末の画面に同期させる。自由閲覧モードに戻るには自己の顔をマウスで選択する。これにより、両モード間のスムーズな切り替えが可能となる。

5. 利用評価

・協同編集機能の実現に関して、要件1)については3.1節のアクセス管理機能により実現した。要件2)については3.2節の操作権管理機能及び、3.3節の文書処理部の制御により実現した。

・本システムの有効性を検証するため作業効率の評価を行った。具体的には、文字、画像、図形を含む2種類(各3ページと10ページの文書)の文書を用いて、予め定めた編集シナリオに基づき、複数利用者からの同時編集による作業時間を測定した。作業形態としては、会議中に操作者が唯一1人に限られる強制同期モード及び自由閲覧モードの両方を評価した。さらに、後者のモードでは、編集シナリオとして各利用者が互いに異なるオブジェクトを編集する排他シナリオ(例えば、利用者Aが2章と4章を、利用者Bが1章と3章

を編集するシナリオ)と、各利用者の編集すべきオブジェクトが競合する可能性のある競合シナリオを用いた。また、レイアウト処理を非同期に行うことの有効性を確認するために、各シナリオにおいてMCUで編集処理の蓄積やレイアウト処理の非同期制御を行わない処理(以下、非制御方式とよぶ)も実現し、本稿での3.3節(2)の提案方式と比較した。ここで利用者は2人及び3人により同時編集を行った。その結果以下の知見が得られた。

- 1) 自由閲覧モードの場合は強制同期モードと比較して作業時間が明かに減少した。
- 2) 排他シナリオは、編集したいオブジェクトの操作権が他利用者により取得されることがないのでトータルな作業時間は、いずれの場合も競合シナリオより減少した。しかしながら、競合がある場合でも、シナリオによっては事実上競合が起きないようにすることも可能であり、作業時間を排他シナリオに近づけられることも判明した(例えば、編集したいオブジェクトが他の利用者取得されている場合には、その間に他のオブジェクトを選択できるようなシナリオにして、競合による編集の待ち時間を少なくするなど)。
- 3) 非制御方式では、他利用者からの文字入力等に対して頻繁にレイアウト処理が行われ、利用者はかなり不満を感じた。一方、本稿の提案方式では、各利用者は他者からのレイアウト処理による割り込みが減少し、利用者の満足感が得られた。さらに、提案方式では非制御方式と比較していずれのシナリオにおいても作業時間が平均して10%~20%程度改善された。

・操作権のMMIを評価する指標として操作権を取得するまでの時間、操作権の適用範囲、操作権の開放タイミングが重要であると考えられる。本システムでは、操作権をMCUが管理することにより約0.5秒程度の応答性が確保でき利用者の満足感が得られた。次に、操作権の適用範囲に関しては、オブジェクト単位とすることは逆に選択可能な操作権の最小単位が事前に限定されることを意味する。文字単位の操作権が指定可能なシステムも報告されているが¹³⁾、文字単位の操作権はその範囲を利用者が指定するのに煩わしさがあると

考えられる。さらに、操作権開放のタイミングに関しては、操作権取得者が故意に操作権を開放しない限り保持されたままになる。これは、会議に参加したまま離席するとそのオブジェクトは他の利用者が取得できないなどの問題が残るが、タイムアウト等で自動的に操作権を開放する処理などで解決できる。

・ ツリー構造を構成するオブジェクトのアクセス管理を行なうUnixファイルシステムと本システムとの異なる点は、Unixファイルシステムでは、上位オブジェクトのアクセス権が優先されるのに対し、本システムでは下位オブジェクトのアクセス権が優先される点である。これは、Unixファイルシステムのディレクトリが、あるアカウント用あるいはグループによる共有といった比較的単純なアクセス指定で十分であるのに対し、文書構造のアクセス権は文書の内容に依存してアクセス権が上下のオブジェクト間で異なる場合が想定されるためである。例えば、1章のアクセス権を利用者Aに2章のアクセス権を利用者Bに全体のアクセス権を利用者Cに設定し、1章、2章のレイアウト属性を省略することにより、Cが、1章、2章の文書内容は変更できないが、レイアウトは変更できるといった明確な役割分担が可能となり、本アクセス管理は有効であることが確認された。

6. おわりに

本稿では、構造化文書を用いて協同編集を行うためのアクセス権、操作権の管理法、及びレイアウト処理制御法などの実現メカニズムを示し、協同文書編集システムColleagueの実装について報告した。その結果、構造化文書に基づくアクセス権を用いた役割設定機能、複数操作権の管理や文書処理の制御による同時編集機能の有効性を示した。今後は、ISDNなどのTCP/IP以外のネットワークへの適用や、文書のハイパー化にともない各メディアが異なるエンティティに分散して存在するような分散文書環境での協同編集機能への拡張検討を進める予定である。最後に、日頃御指導頂く国際電信電話(株)研究所 浦野所長、村上次長、羽鳥画像情報処理グループリーダーに感謝します。

参考文献

1) Ellis, C. A., Gibbs, S. J. and Rein, G. L. :

Groupware Some Issues and Experiences, Communication of the ACM, Vol. 14, No. 1, pp.33-58 (1991).

2) Watabe, K., Sakata, S. et al. : Distributed Desktop Conferencing System with Multi-user Multimedia Interface, IEEE Journal on Selected Areas Communications, Vol. 9, No.4, pp. 531-539 (1991).

3) 中山良幸, 森賢二郎, 中村史郎, 山光忠 : 多者間電子対話システム ASSOCIA, 情報処理学会論文誌, Vol.32, No.9, pp.1190-1199 (1991).

4) Leland, M. D. P., Fish, R. S. and Kraut, R. E. : Collaborative Document Production Using Quilt, Proceedings of the 1988 Conference on Computer Supported Cooperative Work (CSCW'88), pp.206-215 (1988).

5) Froles, F., Graves, M. et al. : Computer System and the Design of Organization Interaction, ACM Trans. on Office Information Systems, Vol. 6, No.2, pp.153-172 (1988).

6) Tanaka, T. and Yamada, M. : Introduction of CDH(Cooperative Document Handling) and its feasibility study on Multimedia Document Conferencing System, Proceedings of the Conference on Multimedia'94, pp.4.3.1-4.3.6 (1994).

7) Palme, J. : Standards for asynchronous group communication, Computer Communications, Vol. 16, No.9, pp.532-538 (1993).

8) 山田満, 田中俊昭 : 文書会議アプリケーションに適用可能な開放型文書プロセッサ, 画像電子学会, Vol. 22, No.5 pp.540-549 (1993).

9) ITU-T Recommendation T.410 series : Open Document Architecture (1988).

10) 田中俊昭, 山田満 : マルチポイント協同文書編集システムColleagueの実装, 情報処理学会第49回全国大会, 3E-4 (1994).

11) ITU-T Draft Recommendation T.435 : Abstract Service Definition for Document Manipulation (1994).

12) ITU-T Recommendation T.122 : Multipoint Communication Service for Audio Graphic and Audio Visual Conference Service Definition (1993).

13) Baecker, R., Nastos, D. et al. : The user-centered interactive design of collaborative writing software, Proceeding of the Conference on Human Factors in Computing Systems (INTERCHI'93), pp. 399-405 (1993).