

構造化マルチメディア文書を用いた協同編集システム Colleague

田中俊昭[†] 山田満[†] 羽鳥好律[†]

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。Colleague は、(1)構造化文書を用いて文書構造(オブジェクト)のアクセス管理を行うことにより構造上レベルの異なるオブジェクト間においても利用者の役割分担を明確にできる、(2)複数の操作権の利用およびレイアウトや可視化などの文書処理の制御により複数利用者からの共有文書に対する同時編集機能を提供する、(3)会議への途中参加や退席における円滑な文書管理機能を提供する、などの特徴を有している。本論文では、上記の特徴を実現する具体的なメカニズムについて述べるとともに、システムの利用評価を行い、その有効性を示す。

Colleague: Joint Editing System Based on Structured Multimedia Document

TOSHIAKI TANAKA,[†] MITSURU YAMADA[†] and YOSHINORI HATORI[†]

The goal of this paper is to realize synchronous joint editing of laid out document containing various media such like text, image, graphics. 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 session. Second, by managing multiple tokens and controlling document process such like laying out or imaging, users can edit the different part of the same shared document simultaneously. Finally, a user can manage the shared documents effectively even if a user may join or leave the existing conference dynamically. In this paper, we discuss the detailed 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うグループウェアの検討がさかんに行われている^{1),2)}。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³⁾⁻⁵⁾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⁶⁾やワークフロー^{7),8)}等のシステムが報告されている。ここで扱われる文書については、テキストに加え、画像、図形、オーディオなどを含むマルチメディア文書、さらには、メディ

アをリンクという関係で結合するハイパーメディア文書⁹⁾⁻¹¹⁾のモデルを用いた検討も行われている。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¹²⁾⁻¹⁴⁾。

文書をベースとしたグループウェアシステムとしては、Quilt⁶⁾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。また、シングルユーザAPをマルチユーザ対応に拡張する方式も提案されている¹⁵⁾。なかでも阿部¹⁶⁾らは、マウスやキーボードの入力情報などのイベントの授受をベースとしてAPを共有するための汎用的な方式を提案しており、この方式により既存のワープロやDTP

[†] 国際電信電話株式会社研究所
KDD R & D Laboratories

ソフトウェアがそのままマルチユーザ環境で使用可能となるが、一方でかな漢字変換の学習処理などイベントベースの AP 共有だけでは完全な処理の同一性が保証されない点も指摘されている⁴⁾。さらに、ハイパーメディアを用いた非実時間型システムとしては、gIBIS⁹⁾が報告されインフォーマル情報の収集などの AP で有効性が確認されている。MuHyme¹¹⁾ではハイパーメディアのデータ更新モデルやバージョン管理に関して有効な方式を提案した。しかしながら、文書作成のライフサイクルの最終段階である紙に印刷可能なレイアウトをもつオフィス文書については、いずれのシステムでも対象としていない。本論文では、協同文書作成が、(1)各執筆者による文書パーツの作成、(2)監修者による共有文書の基本構成(枠組み)の作成、(3)各パーツの共有文書への反映、(4)各執筆者間での相互作業、(5)文書体裁の推敲というライフサイクルで構成されると考えた。これまで協同編集システムは種々提案されているが、とくに(5)を含めた協同文書作成を支援しているシステムは提案されていない。しかも、昨今の個人の文書作成機能は、レイアウトを含む文書の体裁が編集可能であり、文書作成者(執筆者)によりレイアウトの体裁を整えた形でそのまま提出/出版されることが日常的になりつつある。このような個人環境を協同文書作成環境へスムーズに移行する技術は、グループ作業において不可欠な要素技術と考えられる。さらに、(5)のような最終段階の協同文書作成では、文書全体の論旨というよりは、読み合わせ、細かな編集上の修正、視覚的なレイアウト表現の変更などが多く、リアルタイムで行うほうが効率的であると考えられる。

本論文では、既に開発したポイント-ポイントでの文書会議システム¹⁷⁾を3者以上のマルチユーザへ拡張するにあたり、上記のコンセプトを実現するため以下の点について注目した。

(1)3者以上のマルチユーザを対象とした協同執筆、遠隔教育などでは、共著者の役割を明確に分離し、各利用者によって編集する能力が異なるなどの文書内のアクセス制御を厳密に行う検討が必要である。

(2)電子的な文書は従来の紙の印刷物に対するメタファであり、レイアウトされた文書を用いた協同作業の要求も指摘されている¹¹⁾。このようなレイアウトされた文書を用いて WYSIWIS の制約に妨げられず、個人の文書作成環境をそのまま会議形態に実現するためには、各利用者が任意のページを閲覧でき、かつ、文書を複数の利用者が同時に編集する機能が望まれる。

(3)柔軟な会議を実現するには既存の会議への途中参加や参加している会議からの退席などの会議機能が必要となる¹⁸⁾。このようなフレキシブルな会議形態における共有文書の管理機能が必要となる。

本論文では、これらの点を解決するため以下の提案を行う。

- ・文書内での各利用者の役割設定を可能とするため、ODA¹⁹⁾の文書構造に基づくアクセス権管理メカニズムを提案する。

- ・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、ODAの文書構造に基づく操作権の管理法を提案する。さらに、文書処理モデルで規定される編集処理、レイアウト処理、可視化処理を協同編集下で効率的に制御する方式を提案する。

- ・途中参加や会議終了時に最新の共有文書を取得できるようにする。また、退席時のバージョンをローカルに保管することにより最新のバージョンと各利用者が関与していた時点のバージョンとの比較を可能とする。

本論文では次の構成をとる。まず2章において本システムの概要を述べ、次に3章において本論文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²⁰⁾。

(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”⁴⁾が存在する。本システムでは、比較的低速度の N-ISDN や 10 Mbps の LAN など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

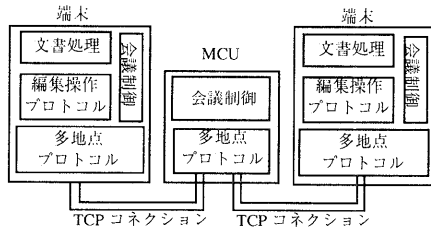


図1 システム構成図(2端末の場合)

Fig.1 System configuration in case of two terminals.

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(文書転送および操作)^{21),22)}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”および、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)²³⁾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/退席等の管理を行う会議制御部および、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および会議制御部から構成される。

2.3 文書作成のライフサイクル

本論文では、以下に示すような印刷などの最終段階に至るまでの過程(ライフサイクル)に基づき協同文書が作成されると考えた。

フェーズ1) 文書作成にあたってその目的を遂行す

るためのメンバーを選択し、各メンバーに役割を与える(たとえば、各人が文書の何章を担当するなど)とともに、各個人が割り当てられた役割に従い文書を作成する。すなわち、割り当てられた利用者は、各章を構成するパーツ(テキストや図形)を作成し、保管しておく。本フェーズはローカルあるいは非リアルタイムで行われる。

フェーズ2) 文書作成の監修者がフェーズ1)の割り当てに従い、文書全体の章構成などを規定し、共有文書の基本構造を作成する。

フェーズ3) 各利用者がリアルタイム会議において、フェーズ1)で作成したパーツをフェーズ2)で作成した共有文書の内容オブジェクトに貼り込む。ここで、文書の作成内容が割り当てられた利用者の知識や能力を超える場合にはメンバー内の他利用者による内容の作成を要請する。

フェーズ4) コメントや文書内容(テキストや図など)の作成を依頼された利用者は、それに従い、作成する。フェーズ3)~フェーズ4)を繰り返し、文書内容の完成度を高める。

フェーズ5) 最終的な文書の体裁(レイアウトも含む)を決定する。ここでは、各利用者がリアルタイム会議において、最終段階での校正や、各ページを1段組から2段組にする、ページ内の図の位置を変更する、表題のフォントを変えるなどレイアウトも含めた協同文書作成を行う。

3. 協同編集機能の実現手法

本論文でのポイントとなる協同編集を効率的に行うには、会議の運用形態、編集機能、文書管理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: WYSIWISの制約を緩和し個人文書作成環境をスムーズに複数利用者環境に移行できるよう、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、複数操作権の実現手法および、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。

要件3) 柔軟な会議運用における共有文書の管理機能: 会議の途中参加や途中退席における文書の効率的な管理を行う必要がある。このため、会議の途中参加や途中退席時の共有文書のバージョン管理メカニズ

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 操作権による文書内のアクセス管理

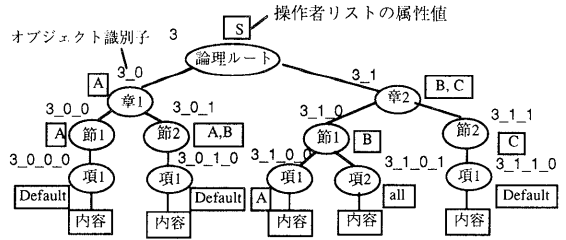
協同編集の一貫性を保証しながら実時間型システムを実現するには、複数の利用者が文書内の同一オブジェクトに対して時分割で操作する必要がある。本システムでは構造上の上下関係を含む柔軟な役割設定を可能とするため、文書構造である各オブジェクトに対して、操作権と操作者リストの概念を導入する。ここで、文書構造の最小単位は、ODA で規定される基本論理オブジェクトに相当し、基本論理オブジェクトは利用者によって実文書の論理的な単位に対応づけられる。従って、1センテンスなどの細かなグレインサイズに対する操作権の付与も可能である。

まず、操作権は“任意の時点で文書中のあるオブジェクトに対して唯一の利用者が取得でき、そのオブジェクトの属性を変更する権利、および構造上の配下にあるオブジェクトに対して、生成、削除、順序の入れ替えなどの変更可能な権利（ただし、配下のオブジェクトが保有する後述の操作者リスト属性に当該利用者が含まれない場合は、削除は不可能とする）”と定義する。ここで、文書自体に変更を伴わない Read する権利は暗黙的に全利用者に付与する。

さらに、操作者リストは“文書内の各オブジェクトに対して変更操作可能な利用者を列挙した静的な（会議中是不変の）リスト”を表す。すなわち、操作権と操作者リストの関係は、操作者リストに対するある時点での実体（インスタンス）が操作権であり、操作権は操作者リストで指定された利用者間で時々刻々変わっていく権利を表す。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操作者リストに関して、運用上、以下の場合を満足する管理方法が必要となる。

- 2章全体は利用者 B および C の担当で、2章に属する 2.1 節は利用者 B が、2.2 節は利用者 C がそれぞれ担当する。
- 2章の操作者リストを設定することにより、2章の下位に属するすべてのオブジェクトの操作者リストを暗黙的に設定する。

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操作者リスト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操作者リストが省略された場合には、その上位の操作者リストを 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操作者リストの値として任意の数



ObjectID	操作者リスト	操作可能な利用者のリスト	ObjectID	操作者リスト	操作可能な利用者のリスト
3	S	S	3_1	B,C	B,C
3_0	A	A	3_1_0	B	B
3_0_0	A	A	3_1_0_0	A	A
3_0_0_0	Default	A	3_1_0_1	all	all
3_0_1	A, B	A, B	3_1_1	C	C
3_0_1_0	Default	A, B	3_1_1_0	Default	C

図2 操作者リストの例

Fig. 2 Example of users' list who can grab token.

の“参加者識別子 (ParticipantID)”，事前に指定された“グループ識別子 (GroupID)”，すべての利用者が対象となる“all”，および省略時値“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、操作者リストが省略 (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) の操作者リスト (利用者 A および B が操作権が取得できる) を継承する。また、節1 (ObjectID=3_1_0) は、B 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) の操作者リストが 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 のみに変更操作が可能となる。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、個別に操作者リスト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。ただし、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。

3.2 複数操作権の実現手法

本システムでは、簡単化のため、各利用者は一度に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

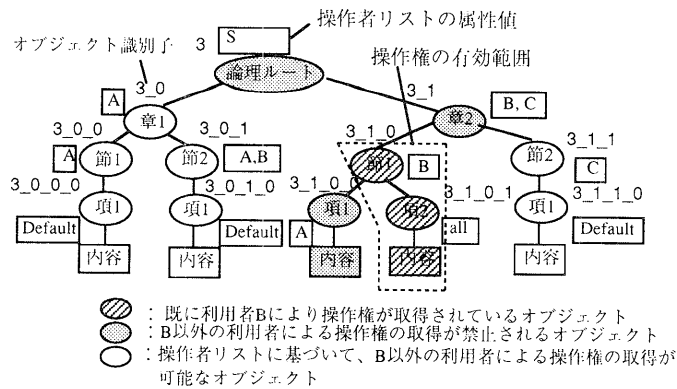


図3 操作権の取得例 (利用者Bが節1 (ObjectID=3_1_0) の操作権を取得した場合)

Fig.3 Example grabbed token (In the case that user B has grabbed a token of section 1 (ObjectID=3_1_0)).

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 上に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り当てられる 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。

$$\{GrabbedObjectID_i \subseteq Prefix(RequestedObjectID)\} \vee \{RequestedObjectID \subseteq Prefix(GrabbedObjectID_i)\}$$

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、取得すべきオブジェクト識別子を表す。

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3は節1 (ObjectID=3_1_0) の操作権が利用者Bによって取得されている状態を示す。この場合、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。ここで、上位オブジェクトを取得禁止としているのは、もし、図3の場合でBが節1を取得している状態で利用者Cが章2のオブジェクトを取得できるとすると、操作権の定

義に従い、Cの権限で章2に属する節1と節2 (ObjectID=3_1_1) の順序を入れ替えるなどの処理が可能となり、Cが節1と節2の順序を入れ替えると同時に、Bが節1の内容を変更する場合は生じ、その結果、利用者端末間でオブジェクト識別子に相異が生じて処理に矛盾をきたすので、これを回避するためである。また、Bが節1の操作権を取得している場合には、その配下にあるオブジェクトで操作者リストにBが含まれる項2 (ObjectID=3_1_0_1) もBにより操作可能となり、節1および項2が操作権の有効範囲となる。

ここで、図2の例を用いて本論文の2.3節の各フェーズに関して詳述すると、フェーズ2)では監修者Sが論理ルートの下に2つの章オブジェクトを作成し、章1 (ObjectID=3_0) を利用者Aに章2 (ObjectID=3_1) を利用者BおよびCに割り当てる。フェーズ3)では利用者Aは、章1の操作権を取得することにより節1 (ObjectID=3_0_0) や節2 (ObjectID=3_0_1) オブジェクトを作成し、ローカルに作成した内容情報を貼り込む。ここで、他の利用者であるBのコメントを要請する場合、節2 (ObjectID=3_0_1) の操作者リストをAおよびBとしておく。また、フェーズ4)ではAからの要請に応じて項1 (ObjectID=3_0_1_0) に対してBが操作権を取得し、コメントしたり修正を加える。さらに、フェーズ5)では文書全体の操作権 (論理ルート) を監修者Sが取得し、文書全体のレイアウトを変更する。また、利用者Aが章1の操作権を取得し、節1と節2の順序を入れ替えるなどを行い最終的な文書を完成させる。

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4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4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4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。

(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ページ以降すべて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、以下の方法を提案する。

- 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ごと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

表1 DTAMで規定される主な通信サービス
Table 1 Example of DTAM service primitives.

		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/ペースト	none
MOVE	オブジェクトの移動/カット	none
POINT	オブジェクトの指示	none
RESERVE	オブジェクトの予約	none

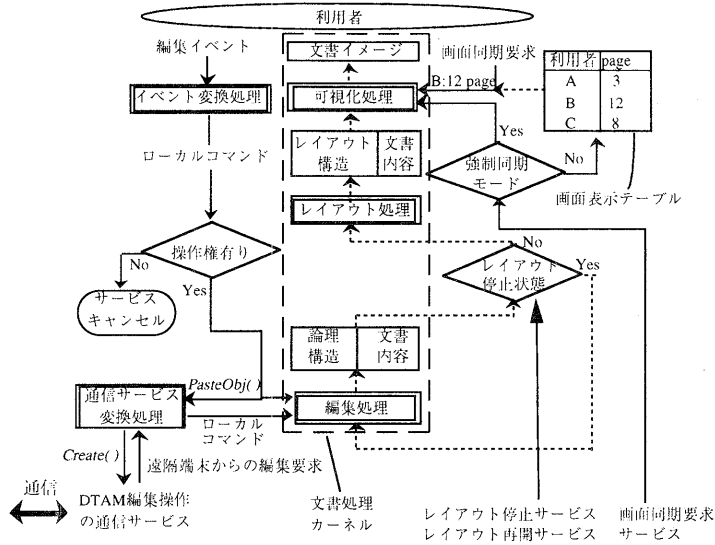


図4 協同編集に適用可能な文書処理部の構成

Fig. 4 Configuration of document process applicable to joint editing.

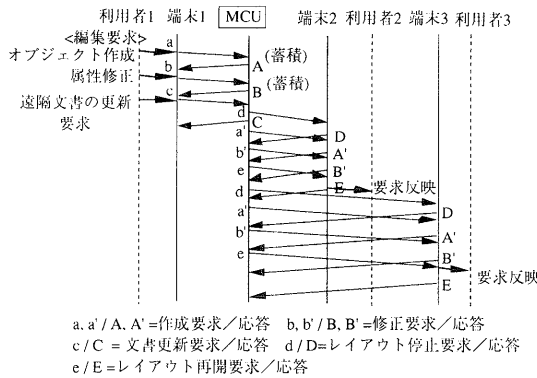


図5 遠隔文書編集のプロトコル手順の例

Fig. 5 An example of communication procedures for remote document editing.

映できる構成とした。

具体的なレイアウト処理の制御およびプロトコルを図4, 図5にそれぞれ示す。利用者1からの編集要求(図5 a, b)をMCUで蓄積し、その後更新要求を送出する(図5 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5 d)、一連の編集要求をすべて送出する(図5 a', b')。この間は図4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5 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4に示すレイアウト構造/文書内容が作成される。

(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4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。

3.4 協同編集における共有文書の管理法

実時間型の協同編集は一種の会議形態と考えられるが、会議APにおいては利用者が会議の途中から既存の会議に参加する機能および会議の途中で退席する機能が必須となる。従って、このような会議に参加する参加者の数が時変の場合での共有文書の管理法を確立する必要がある。本論文では共有文書の管理法について以下の点に注目した。

表 2 参加の形態における会議利用者の分類

Table 2 Classification of users from the point of the conference joining status.

TYPE 1	会議を開始する時点から終了する時点まで継続的に会議に参加している利用者
TYPE 2	途中参加し、会議が終了する時点においても会議に参加している利用者
TYPE 3	会議が終了するまでに退席し、会議が終了する時点では会議に参加していない利用者

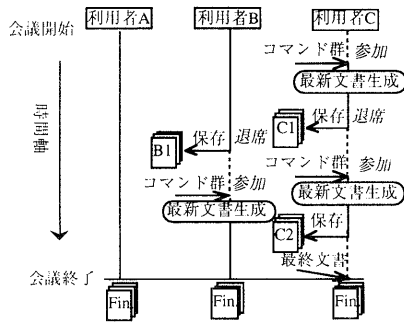


図 6 共有文書の管理

Fig. 6 Management of the shared documents.

- ・途中退席の後、途中参加した利用者は退席中に行われた共有文書の変更内容を把握したい。
- ・会議終了以前に退席した利用者も最終的な共有文書を手にいれたい。

上記の点を満足させるために、本論文では会議の参加者を表 2 に示す 3 つの TYPE に分類し、各 TYPE の利用者の文書管理を以下のように実現した。なお、ここでは共有文書は事前に各端末に転送されているものとする。

終始会議に参加している TYPE 1 は共有文書の途中のバージョンに対してはシステムが自動的に保存せず、利用者が必要な際に手で共有文書をローカルに保存できるようにした。また、TYPE 2 では途中参加時にその時点の最新の共有文書を反映するために、MCU において会議開始時点からの編集要求のコマンドを蓄積しておき、途中参加時にこれらを一括してその参加者に対して送付し、途中参加者の文書処理部において最新の共有文書に更新できるようにした。さらに、TYPE 3 では、すでに退席した参加者が会議を終了している可能性があるため、最終的な共有文書をファイルとして保存し、これを退席した参加者の端末にバルク転送することとした。また、退席/参加を繰り返す場合には、その間に生じた共有文書の変更を把握する必要があるため、退席時点での最新文書をシステム

が自動的に保存する機能を具備した。保存方法としては、他人の退席時のバージョンに対する必要性が少なくと考え、退席者のみがローカルに保存できるようにした。図 6 に会議の進行と文書の関係を示す。ここで、利用者 A が TYPE 1 に、利用者 B が TYPE 2 に、利用者 C が TYPE 3 にそれぞれ属する。また、B1、C1 および C2 は退席時に退席者の端末にローカルに保管されるバージョンを、Fin は会議終了時に保管される最終的な共有文書を表す。

4. Colleague システムの実現

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ィンドウで本システムと同時に起動させ実現した。図 7 に本システムの画面イメージを示す。

・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 7 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 7 にアノテーションの様子を示す。この結果、上記の両機能用途に応じて使い分けでき、キー入力に不慣れな利用者からシステムを使い慣れた利用者まで幅広い利用者に適用可能なシステムが実現できる。

・画面表示モードの切り替え

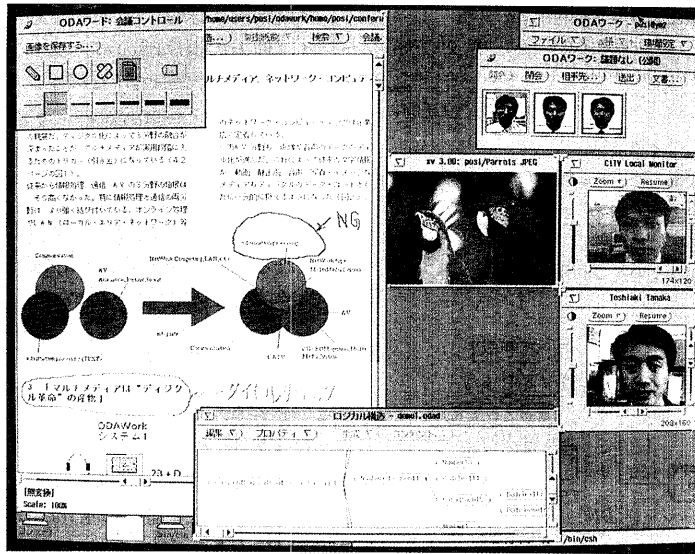


図7 Colleagueの画面例
Fig. 7 An example of Colleague screen.

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

以下に本システムの利用評価を示す。

- 協同編集機能の実現に関して、要件1)については3.1節の操作権および操作者リストにより実現した、要件2)については3.2節の複数操作権の実現手法および、3.3節の文書処理部の制御により実現した。さらに、要件3)については3.4節のバージョン管理機能により実現した。

- 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。ここで、用いた文書は実作業において本システムを用いて作成した文書であり、編集シナリオは複数利用者による実際に行われた編集過程をモデル化したものである。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およ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

表3 各文書例を用いた平均編集作業時間
Table 3 Mean time of editing work using sample documents.
(単位：秒)

		自由閲覧モード		強制同期モード
		提案方式	非制御方式	
文書例1	排他シナリオ	201	230	450
	競合シナリオ	303	388	
文書例2	排他シナリオ	157	227	355
	競合シナリオ	224	280	

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、本論文での3.3節(2)の提案方式と比較した。ここで利用者は2人および3人により同時編集を行った。その実験結果を表3に示す。

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減少した。具体的には、最も作業時間が少なかった排他シナリオによる自由閲覧モード/提案方式を用いた場合では、強制同期モードと比較して文書例1では450秒から201秒に、文書例2では355秒から157秒にといずれの場合も半分以下の時間で編集を行うことができた。

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、20%~40%の範囲で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により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、本論文の提案方式では、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、操作権の解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。また、操作権の適用範囲に関して、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²⁴⁾、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。従って、操作権の取得を円滑に行うためには、ある程度適用範囲が固定されているほうが望ましいといえる。さらに、操作権解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解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解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としては、よく知られたUnixのファイルシステムなどがあるが、Unixファイルシステムのアクセス管理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。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。

・途中参加時の最新文書の更新に関しては、編集要求(数10バイト)を蓄積して送るので、文書全体を再送するよりもはるかに少ないデータで更新され、途中参加の処理が非常にスムーズに行われた。また、本システムでは退席時のバージョンが退席者のみがローカルに保存することとしたため、無駄な中間バージョンを共有文書として保存することがなくなり文書の管理が簡素化されるとともに、退席時のバージョンが必要な際には、共有文書として再利用も可能となった。一方で、共有文書に異常が生じた場合、ある時点に戻る必要性も指摘された。

・レイアウト処理の頻度に関しては、試作時にはある一定以上の編集データ量になった場合に強制的にレイアウトを行う方法や、タイマー監視により強制的にレイアウトを行う方法などで評価を行った。しかしながら、レイアウトを含む文書については、論理的な文書内容の変更、文書の体裁であるレイアウトの変更という2つの異なる観点があり、一概にデータ量や時間で判断するのは適切でないという結論に達し、本システムの実現時には文書を変更した利用者がマニュアルであるいは別のオブジェクトの選択した際、レイアウト変更を行うようにした。実際に、本システムを利用した結果、変更を行いたい利用者が変更を意図した際の意味のある単位でレイアウト変更をかける利用者が最も多かった。例えば、ある利用者がパラグラフを修正したい場合には、文節の単位やセンテンスの単位ではなく、パラグラフを修正し終わった時点などである。

・会議形態の評価を行った結果、本システムは以下の3つの用途に有効であることが確認された。

1) 実時間AP:強制同期モードではある利用者の画面に同期させることができ、しかもその編集がリアルタイムに各端末に反映される。本モードでは利用者に明確な主従関係がある遠隔教育などの場合に有効であった。

2) 疑似実時間型AP:自由閲覧モードでは、各自独立して文書の編集ができる。この場合は、相手の編集結果が真の意味で実時間に反映されず、送信側利用者の更新要求を契機として更新される。本形態は各モジュールを結合して協同デバッグを行うソフトウェア

開発などに有効であった。

3) 非実時間型 AP: 本システムでは会議という概念的なダイアログに途中参加/退席することができるので、非実時間型の AP にも応用できる可能性もある。ただし、非実時間 AP を実現するには、退席中に文書の変更が行われた際にそれを通知する、あるいは時々刻々変化する参加者を通知するなど、付加的な会議状態通知機能が新たに必要となる。

6. おわりに

本論文では、構造化文書を用いて協同編集を行うためのアクセス管理法、およ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を実装/評価した。その結果、構造化文書に基づく操作権および操作権リスト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、会議途中参加/退席時の文書管理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。

謝辞 日頃御指導頂く国際電信電話(株)研究所 浦野義頼所長、村上仁己次長に感謝します。

参 考 文 献

- 1) Ellis, C. A., Gibbs, S. J. and Rein, G. L.: Groupware Some Issues and Experiences, *Comm. ACM*, Vol. 14, No. 1, pp. 33-58 (1991).
- 2) 石井 裕: グループウェア技術の研究動向, 情報処理, Vol. 30, No. 12, pp. 1502-1508 (1989).
- 3) Watabe, K., Sakata, S. et al.: Distributed Desktop Conferencing System with Multi-user Multimedia Interface, *IEEE J. Selected Areas Communications*, Vol. 9, No. 4, pp. 531-539 (1991).
- 4) 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).
- 5) 島村和典, 正木茂樹, 谷川博哉: B-ISDN 用多地点間マルチメディア通信会議システム PMTC, 電子情報通信学会技術報告, IE 90-49, pp. 31-36 (1990).
- 6) Leland, M. D. P., Fish, R. S. and Kraut, R. E.: Collaborative Document Production Using Quilt, *Proc. the 1988 Conference on Computer Supported Cooperative Work (CSCW'88)*, pp. 206-215 (1988).
- 7) Froles, F., Graves, M. et al.: Computer System and the Design of Organization Interaction, *ACM Trans. Office Information Systems*, Vol. 6, No. 2, pp. 153-172 (1988).
- 8) 市村 哲, 松下 温: 発言と行動の管理に基づいた協同作業支援電子メール PilotMail, 情報処理学会論文誌, Vol. 33, No. 7, pp. 955-963 (1992).
- 9) Conklin, J. and Begeman, M.: gIBIS: A Hypertext Tool for Exploratory Policy Discussion, *Proc. the 1988 Conference on Computer Supported Cooperative Work*, pp. 140-152 (1988).
- 10) Press, L.: Personal Computers as Research Tools. *SIGCHI and CSCW'90, Comm. ACM*, Vol. 34, No. 4, pp. 19-25 (1991).
- 11) 村永哲郎, 守安 隆, 友田一郎, 水谷博之: ハイパーメディアに基づく共同文書作成環境 MuHyme, 情報処理学会論文誌, Vol. 34, No. 6, pp. 1395-1405 (1993).
- 12) Tanaka, T. and Yamada, M.: Introduction of CDH (Cooperative Document Handling) and Its Feasibility Study on Multimedia Document Conferencing System, *Proc. the Conference on Multimedia'94*, pp. 4.3.1-4.3.6 (1994).
- 13) Palme, J.: Standards for Asynchronous Group Communication, *Computer Communications*, Vol. 16, No. 9, pp. 532-538 (1993).
- 14) Delgado, J., Jordan, F. and Perramon, X.: FODATEC: An Open Document Manipulation System, *Proc. Conference on ULPAA*, pp. 221-234 (1994).
- 15) Prakash, A. and Knister, M. J.: Undoing Actions in Collaborative Work, *Proc. the ACM Conference on Computer Supported Cooperative Work (CSCW'92)*, pp. 273-280 (1992).
- 16) 阿部豊子, 前野和俊, 阪田史郎, 福岡秀幸: マルチメディア分散在席会議システム (MERMAID) を利用したグループアプリケーションの分散協調制御方式の提案, 情報処理学会論文誌, Vol. 34, No. 6, pp. 1406-1416 (1993).
- 17) 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, 画像電子学会誌, Vol. 22, No. 5, pp. 540-549 (1993).
- 18) Chung, G., Jeffay, K. and Abdel-Wahab, H.: Dynamic Participation in a Computer-Based Conferencing System, *Computer Communications*, Vol. 17, No. 1, pp. 7-16 (1994).
- 19) ITU-T Recommendation T. 410 series: Open Document Architecture (1988).
- 20) 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, 第 49 回情報処理学会全国大会論文集, 3E-4 (1994).
- 21) ITU-T Draft Recommendation T. 435: Abstract Service Definition for Document Manipulation (1994).
- 22) 田中俊昭, 中尾康二: グループウェアを実現する

DTAM 編集操作機能の一検討, 第 46 回情報処理学会全国大会論文集, 5 Q-3 (1993).

- 23) ITU-T Recommendation T.122: Multipoint Communication Service for Audio Graphic and Audio Visual Conference Service Definition (1993).
- 24) Baecker, R., Nastos, D. et al.: The User-Centered Interactive Design of Collaborative Writing Software, *Proc. the Conference on Human Factors in Computing Systems (INTERCHI'93)*, pp. 399-405 (1993).

(平成 6 年 9 月 5 日受付)

(平成 7 年 1 月 12 日採録)



田中 俊昭 (正会員)

1960 年生. 1984 年大阪大学工学部通信工学科卒業. 1986 年同大学院修士課程修了. 同年国際電信電話(株)入社. 現在, 研究所画像情報処理グループにおいて, マルチメディア通信プロトコル, グループウェア, ネットワークセキュリティの研究に従事. 電子情報通信学会会員.



山田 満

1957 年生. 1980 年東京大学工学部電子工学科卒業. 1982 年同大学院修士課程修了. 同年国際電信電話(株)入社. 現在, 研究所画像情報処理グループにおいてマルチメディア・グループ通信の研究に従事. 平成元年度電子情報通信学会篠原記念学術奨励賞, 平成 4 年度画像電子学会論文賞受賞. 工学博士. 電子情報通信学会, 画像電子学会各会員.



羽鳥 好律 (正会員)

1948 年生. 1971 年東京大学工学部電気工学科卒業. 同年国際電信電話(株)入社. 現在, 研究所画像情報処理グループリーダー. 画像通信, 文書・図形処理, マルチメディア情報通信等の研究に従事. 工学博士. 電子情報通信学会, TV 学会, 画像電子学会, IEEE, AVIRG, SICE・HI 部会各会員.