

Virtual Office の基本的設計と基本部分の実現

上林弥彦 高田秀志 渡辺正子 稲森豊

京都大学工学部

Virtual Office は日常のオフィスでの共同作業を、データベースを基礎として支援するためのシステムである。このシステムでは、実際のオフィスに対して、仕事に適した仮想的なオフィスを複数個設定できることが特色である。また、ネットワーク、電話、FAX、AV 機器などをワークステーションに接続・統合し、利用者は端末に向かって簡単な利用者インタフェースによって操作するだけでこれらの機器をデータベースと統合して操作できることを狙いとしている。本稿では、Virtual Office の基本概念、オフィス空間記述言語、および、HyperCard を用いた基本的機能の実現例について述べる。

Basic Design and Implementation of Basic Functions of Virtual Office

Yahiko Kambayashi, Hideyuki Takada, Masako Watanabe, Yutaka Inamori

Faculty of Engineering, Kyoto University

Virtual Office is a system that supports usual cooperative work in a real office based on databases. One of the specific features of this system is the ability to set up more than one virtual office appropriate to the work, besides the real existing office. In Virtual Office, our aim is that computer networks, telephones, FAXes, audio-visual equipment are connected and integrated, users are able to control these equipment combined with databases in front of a terminal using an easy user interface. In this paper, we describe the basic concepts and space description language for Virtual Office, and introduce the implementation of basic functions of Virtual Office developed on HyperCard.

1 まえがき

計算機のダウンサイジングとネットワーク技術の発展によって、高速のネットワークに接続された100MIPS以上の性能を持つデスクトップ型のワークステーションが出現しており、分散環境での計算機システムの利用形態が主流になりつつある。本質的に人間によるオフィスなどでの作業形態は分散環境であり、人間の行なっている作業形態を分散計算機システムによって支援するためのシステム開発が急務となっている。

現在、人間の共同作業を支援するための通信システムとしては、FAX、電話、テレビ電話、テレビ会議などが存在している。また、近年、グループウェアと呼ばれる、ネットワークで結合された計算機システムによって共同作業を支援するシステムが多数開発され、注目を浴びている [1][2][3][4]。しかし、これらのシステムでは人間が日常に行なっているオフィスや研究室での作業を全面的に支援するまでには至っていない。また、日常のオフィスでは存在している個人の自由というものも、これらの共同作業支援システムによっては保証されない場合が多い。

Virtual Officeではこれらの点を考慮し、データベース技術を基礎として、使いやすい利用者インタフェース、通信機器との統合、複数の利用者間のコミュニケーション、セキュリティの保証などの観点から、日常のオフィスでの作業を、個人の独立を保ちつつ、共同作業を支援するためのシステムとしての開発を目指している。本システムは、実際のオフィスに対して、仮想的に仕事に適したオフィスを複数個定義できることが特色であるため、Virtual Officeと名付けた。

本稿では、Virtual Officeのシステム構成について、2章では環境・利用条件・利用例について述べる。また、3章では、Virtual Officeの利用者が自由に自分の作業環境を設計するためのオフィス空間記述言語について述べる。最後に4章では、Virtual Officeの利用者インタフェースがどうあるべきかを検討するために、MacintoshのHyperCard上でVirtual Officeの基本的な機能を実現した例について紹介し、これからの開発の展望を述べる。

2 Virtual Officeの基本概念

2.1 計算機による共同作業支援

計算機によって人間の共同作業を支援するシステムは、グループウェアとして近年盛んに研究開発が行なわれている。現在までに開発されているグループウェアとしては以下のようなものが代表的である。

- 対面型電子会議システム [5]
- 分散型電子会議システム [6][7]
- 分散型エディタ [1]
- 構造化電子メール [8][9][10]

これらのシステムは、マルチメディア、ネットワークなどの技術を基盤として研究開発されたものであるが、現在のオフィスでの日常的な作業を総合的に支援するものまでには至っていない。これに対して、我々が開発を行なっているVirtual Officeは、データベース技術を基礎として、マルチメディア、ネットワーク、セキュリティなどの技術の総合形態として、計算機の中に実際のオフィスを仮想化して存在させることによって、個人の独立と共同作業という相反する要素を共存させ、日常的に行なっている個人作業と共同作業を支援するものである。

本章では、Virtual Officeの目的や典型的な利用例を考え、このシステム実現のための基礎研究の課題について検討する。

2.2 Virtual Officeの環境と利用条件

Virtual Officeは次に示すような技術を基礎として開発する。

- データベース
- マルチメディア処理
- 高機能ワークステーションのネットワーク
- 通信機器 (電話、TV電話、FAX、電子メール) との統合
- ビデオ機器 (TV、ビデオ、光ディスク) との統合

過去において、オフィス用ソフトウェアは機能の統合という形ですすんできた。80年代の始めには、データベース、表計算、グラフの3つの機能が統合された第4世代言語が登場し、80年代後半には、さらにワープロや通信ソフトウェアとの統合が行われた。しかし、MacintoshやWindowsのようなシステムでは、ソフトウェアの統合利用がある程度可能なような利用者インタフェースが用いられており、機能の統合ができるようなシステムにと進みつつある。このため、Virtual Officeでは、ソフトウェアの統合利用が可能な環境を想定し、その上でさらに未来のオフィスに必要な機能について検討している。

Virtual Officeの1つの目的は、現実のオフィスの機能と計算機の機能との両立にあり、両者の特色を兼ね備えると共に、現実のオフィスからの移行も容易になるようにしている。この2つの機能の違いの例として、次の3つの項目をあげることができる。

- 共同作業と個人の独立

計算機の上ではデータの共有が容易であるため共同作業のための情報交換が簡単にできる。一方現実のオフィスでは、他人に知られないように仕事を進め、必要な情報だけを公開できる。システムとして、作業効率を上げるだけでなく、個人のある程度の自由を保証するものでなければ非人間的なものとなってしまふ。

- 論理的なデータ検索と物理的なデータ検索
 計算機ではデータの持つ性質によってデータを検索できるのが大きな特色となっている。データベースでは、データがどこにどのような形で格納されているかを利用者は知る必要がない。しかし、現実には「3日前に机の上にあった書類」を探すようなことも多く存在する。データの利用者から見た物理的な場所も手がかりにできるようにすれば、データ検索の機能が向上できる。セキュリティについても、公開されているデータでも一時的に金庫に入れて鍵をかける等の処置ができれば自由度が高くなると考えられる。
- 実際のオフィス空間と仮想オフィス空間
 実際のオフィス空間では机や本箱等の配置が固定化しており、作業に向けた配置を自由にとることはできない。Virtual Officeでは、プロジェクトごとにそれに適した仮想オフィス空間を定義してそこで仕事をすることができる。1つの仮想オフィス空間自体は現実のオフィスと類似の操作で作業することができる。また、仮想オフィスでは、「スーパー電話」というような現実に存在しない機器を定義することが可能である。スーパー電話としては、文書で書かれた連絡文を、電子メールで送り、電子メールの使えない人、ないしはある時間内に連絡のない人にはFAXで送り、さらにそれでも連絡ができない人には自動発声電話をかけ、さらに、相手からの対応する応答を出力するといったものが考えられる。

2.3 Virtual Officeの利用例

本節では、Virtual Officeを開発するために、どのような機能が必要であるかを整理する。ここでは簡単のためいくつかの代表的利用例を示す。

- 個人の統合オフィス環境の実現
 基本的には、ワープロ、マルチメディアデータベース、各種応用ソフトウェア、通信機器、ビデオ機器等が、現実のオフィス内で操作するのと類似の環境での利用を可能にする。出張で他のオフィスに居る時でもネットワークを通じて、自分の室や机を呼び出して全く同じ環境で作業をすることができる。現実のオフィスとの違いは、同じ書類を何ヶ所にも入れることができたり、内容で検索できたりする点である。過去に戻ることも可能にできるとさらに便利である。
- 共同作業の実現
 共同作業には、秘書に仕事を依頼するといった簡単なものから、共同で1つのプロジェクトを進めていくといったものまで考えられる。さらに、地域的に離れている者同士が、あたかも同じ室にいるように作業ができることも

重要である。時間的にも同時である場合と、そうでない場合(時差のある場合も含めて)が扱える必要がある。

- 会議
 共同作業を実現するための機能としても重要であるが、臨時に数人が集まる必要のあるときに、机を離れずに会議ができる機能が必要である。現在のTV会議システムとワークステーションによる書類の提示機能(データベースも利用できる)のほか、会議内容をテキストデータベース、あるいは画像まで含めたビデオデータベースとしておいて後で利用することも可能である。
- 仮想オフィス
 1人の人間がいくつものプロジェクトに関係している場合、各プロジェクトごとに使う書類や共同作業に関係する人が異なるのが普通である。このため、各プロジェクトに対して1つづつそのプロジェクトの遂行に適した仮想オフィスを定義することができる。1つの仮想オフィスで作業しているときには、他の仮想オフィスからの割り込みを禁止することもできるし、同時に複数の仮想オフィスで仕事することも可能である。仮想オフィスでは、先に述べた機器の仮想化のほかに人間の仮想化も可能で、1人をプロジェクトの種類で複数人のように扱ったり、複数人を1人とみなしたりすることも可能である。

2.4 Virtual Office 開発のための基本的研究課題

Virtual Officeはある意味で共同作業を実現できるような分散データベースの一般化であるが解決すべき問題も多い。いくつかの基本的課題をまとめてみる。

1. システム開発言語
 オフィスの配置や機能が高水準言語で定義できる必要がある。これは、空間記述言語として開発中で、3章で述べる。また、以下に述べるものも含めた言語の開発も必要である。
2. 共同作業管理
 仕事を依頼したときにさらに別の人に依頼したりするため、依頼関係を含めると有限オートマトンよりプッシュダウンオートマトンの方がモデルとして使い易い。さらに期限の管理や、やりかけた仕事ができなくて他人が手伝う等あらかじめ決めていた作業以外のものがある範囲で許す自由度も必要である。このようなオートマトンモデルについて検討中である。
3. 通信機能

通信もデータベースのトリガーのように一方のものではなくフィードバックも考えたものが重要である。また、相手に届けばよい程度のもので、速達や親展といったものもあり、非常に多様な通信手段を表現できる必要がある。

4. セキュリティ

個人の作業の秘密を守る上でも、安全に共同作業管理をする上でもセキュリティは重要である。利用者、データ、物に対してセキュリティレベルを設定するだけでなく、鍵による管理のような機構も重要といえる。さらに利用権の動的制御やメタデータ（共同作業管理データ）の保護も重要といえる。

5. データベース

過去のオフィス状況の再現、共同作業管理、セキュリティ管理といった面からは履歴データベースが重要である。また、マルチメディアが扱えることも不可欠である。重要なデータについては、コピーが複数のワークステーション上に作られるため、更新時のコピー管理も重要である。さらに、仮想オフィス間でデータを共有しているため、データベースにおける高度のビュー機能も重要といえる。

6. 並行処理

トランザクション処理と異なり、長時間処理が一般的であるため、従来とは異なる正当性のもとで並行処理を実現する必要がある。このように、ある程度の自律性を認めた分散システムの並行処理は大きな研究課題である。

7. 一貫性制御

セキュリティ管理、個人データ管理、矛盾を許したデータの管理等種々の側面で、複数の一貫性管理を行わなければならない。独立性と統合との関連で、今までのデータベースより一貫性制御は複雑になる。

3 Virtual Office のための空間記述言語

Virtual Office の機能として、システムの利用者が、誰でも自分の作業環境に応じて自由にオフィス空間を設定できることが要求される。本章では利用者がオフィス空間を設定するためのオフィス空間言語を定義する。

3.1 オフィス空間記述言語

実際のオフィス空間を考えてみると、オフィスの建物の中に共有の各部屋があり、各部屋には共有の空間や機器、各個人用の空間や機器が存在している。共有の空間においては、複数の人間が存在することによって空間を形成し、個人の空間やオブジェクトは各人の作業環境に合わせて設定している。ま

た、各人は複数の共有空間に所属することも可能である。

これによって、オフィス空間記述言語によって記述できるべきものについては以下のようなものが考えられる。

- 空間・機器の配置
- 複数の空間による空間・機器の共有
- 仮想的な空間・機器の設定

以上のようなことを考慮に入れて、オフィス空間記述言語を以下のように定義する。

3.1.1 オフィスの空間的定義

オフィス空間は、その空間に存在する機器や家具、書類などのオブジェクトと、その空間に属する各研究室などのサブオフィス空間の集合によって表現できる。

Office → *Space*

Space → {*Space|Object*}*

Space : オフィス空間

Object : オフィス内のオブジェクト

例えば、情報工学科の建物内に機器として学科共有の電話やソファがあり、そのサブ空間として事務室や各研究室があり、事務室や各研究室にはまた電話やワークステーションなどの機器があるという具合である。

3.1.2 HyperCard による実現

Virtual Office は4章で述べるように、現在開発の容易な Macintosh の HyperCard 上で開発されている。オフィス空間は HyperCard 上ではカードに対応させ、オフィス内のオブジェクトはアイコンやボタン、フィールドに対応させる。ある空間の中でのサブ空間の記述は、その空間に対応するカードに移動するためのリンクが張られたボタンとして表現し、そのボタンが押されたときはサブ空間のカードを開き、空間の移動を達成する。

また、空間や機器の機能は、他のオブジェクトからのメッセージ(マウスイベントなどのようなシステムによって報告されるものも含む)に対するオブジェクトの動作として、HyperTalk とよばれる Script を記述する。

3.1.3 オフィス空間記述言語

前節で述べたオフィスの空間的定義に基づくと、オフィス空間は入れ子構造をもった言語によって表現が可能である。

以下にオフィス空間記述言語の構文について述べる。

- 空間の記述

```
Space → BEGIN space
        [SpaceBody]
        END space
SpaceBody → [SpaceAttribute]
             {Space|Object}*
```

[SpaceAttribute]には、システムが処理するための空間の属性(座標値・名前など)、および、他の空間やオブジェクトからのメッセージに対する動作を表す Script を記述する。

- オブジェクトの記述

```
Object → BEGIN object
         [ObjectAttribute]
         END object
```

[ObjectAttribute]には、オブジェクトの属性や他の空間やオブジェクトからのメッセージに対する動作を表す Script を記述する。

また、オフィス空間記述言語の持つ意味としては、以下のようなものとする。

- 空間・オブジェクトの共有

複数の空間で1つの空間やオブジェクトを共有したい場合には、属性をそれぞれ他の空間やオブジェクトへの Alias として記述する。これにより、1つの書類をいくつもの空間に入れることなどが可能になる。

- 空間の仮想化

プロジェクトごとに作業部屋を設けたい場合などは、それぞれのプロジェクトに対して空間の記述を行なう。他の空間との共有の場合には、前述の Alias によって対応する。

- オブジェクトの仮想化

仮想的なオフィスを実現する際には、実際に存在している機器の機能の一部を実現するような機器を設定したい場合や、前述の「スーパー電話」などのような種々の機器の機能を合成したようなものも必要である。オブジェクト、すなわち機器の動作記述は Script によって行なうため、オブジェクトの仮想化には Script の記述で対応する。

3.1.4 言語記述例

図1に示すようなオフィス空間例をもとに、前節で述べたオフィス空間記述言語でこのオフィス空間を記述した例を図2に示す。なお、空間やオフィスの属性の種類は単純化されており、ここでの例は暫定的なものである。

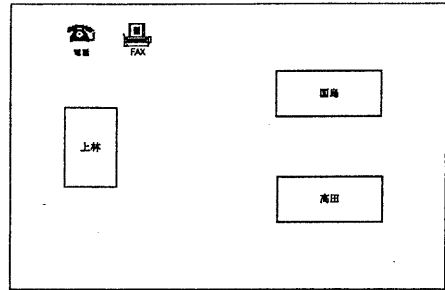


図 1: オフィス空間の例

3.2 言語記述のための視覚的インタフェース

前章で述べたオフィス空間記述言語を用いることによって、利用者が独自に自分の作業環境を構築することが可能となる。しかし、仮想オフィスシステムの利用に際して、言語記述という利用者インタフェースは誰でも使えるシステムという点から見て必ずしも適切なインタフェースではない。また、オフィス空間内のオブジェクトの動作はオブジェクトに記述された Script によって決定されるが、これは、電話や FAX など、それぞれの機器に固有のものであり、あらかじめ記述されたものに対して座標値などの属性値を若干変更するだけで、コピーして用いることができる。

以上のようなことを考慮して、オフィス空間記述言語を容易に記述できるような視覚的利用者インタフェースについて検討する。

3.2.1 空間・オブジェクトの配置

前述の図1の空間に、プリンタを追加したいとする。図3はその際の画面表示である。

画面右上のウィンドウに表示されているのは、現在のオフィス空間の状態であり、その下のウィンドウにはその現在の空間を記述した言語が表示されている。利用者が現在のオフィス空間にプリンタというオブジェクトを追加しようとするとき、左側の機器メニューからマウスカーソルによってプリンタを選択し、オフィス空間の目的の位置に置く。すると、システムが自動的にプリンタとして記述されたオブジェクト記述言語を、適当な属性値を設定して現在のオフィス空間記述の Body 部に挿入し、新たにプリンタが追加された言語が生成される。

3.2.2 空間・オブジェクトの共有

メニューにより新しい空間やオブジェクトを作成しようとした時には、その空間やオブジェクトが新規に作成されるべきものであるか、あるいは他の空間やオブジェクトへの Alias として作成されるべきものであるかの選択を利用者に要求する。新規に作成するべきものであれば、上記のような手順で空間あるいはオブジェクトを作成し、Alias として作成されるべきものであれば、どの空間あるいはオブ

```

BEGIN space
  NAME = "上林研究室"
  KIND = room
  BEGIN object
    NAME = "電話"
    KIND = phone - オブジェクトの種別
    LOC = {20,10} - 座標の指定
    SCRIPT =
      { on mouseDown
        ... - 電話をかける
        end mouseDown }
  END object
  BEGIN object
    NAME = "FAX"
    KIND = fax
    LOC = {20,10}
    SCRIPT =
      { on mouseDown
        ... - FAXを送る
        end mouseDown }
  END object
  BEGIN space
    NAME = "上林"
    KIND = desk
    .....
  END space
  BEGIN space
    NAME = "高田"
    KIND = desk
    .....
  END space
  BEGIN space
    NAME = "国島"
    KIND = desk
    .....
  END space
END space

```

図 2: オフィス空間記述言語での記述例

ジェクトへの Alias であるかを指定するためのウィンドウを開き、利用者に指定を要求する。

3.2.3 空間・オブジェクトのビュー設定

仮想オフィスは、実際に存在している物理的な空間やオブジェクト、またオブジェクトの機能の一つのビューであるということができる。したがって、このシステムにより、新しく空間やオブジェクトが生成された時は、あらかじめシステムによって用意された数種類の機能から、仮想的な空間やオブジェクトが持つべき機能を利用者に選択することを要求する。例えば、電話を生成する際にそれを NTT の電話回線につなげるか、あるいは、建物内だけの回線につなげるか、などが考えられる。

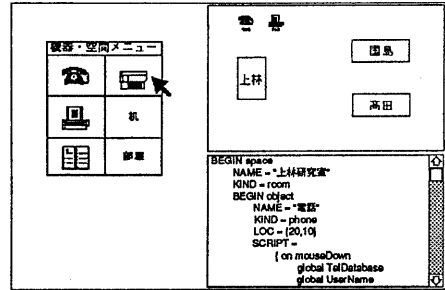


図 3: 視覚的インタフェースを用いた言語記述

4 Virtual Office の基本機能の実現

前章までに Virtual Office の環境や利用例、オフィス空間の設定などについて述べてきたが、Virtual Office の利用者インタフェースがどうあるべきか、また、Virtual Office を構築するための環境がどのような機能を持つべきかを検討するために、基本的機能のみを Macintosh の HyperCard を用いて実現した。本章では、これを紹介し、これからの展望を述べる。

4.1 Virtual Office のための利用者インタフェース

Virtual Office のための利用者インタフェースにおいては、計算機を専門としない人々にとっても目的的操作を容易に行えることが重要であるが、従来、計算機の操作方法として使われてきた、コマンド入力という利用者インタフェースは、複雑なコマンドを利用者に覚えさせることを要求し、全く適していない。そこで、J-Star や Macintosh に代表されるような、ディスプレイ上に開かれたウィンドウ(作業盤)に描かれているアイコン(書類など)をマウス操作などで移動することができるような、いわゆるオブジェクト指向利用者インタフェースによって Virtual Office の利用者インタフェースを構築する。

実際に Virtual Office を構築する際には、利用者インタフェース部分やオフィス内の機器とのインタフェースをとる部分、書類や住所録などを格納しておくデータベースとのインタフェース部分、およびそれらのセキュリティ管理を行なう部分などが必要になると考えられる。また、実際のプロジェクトと Virtual Office での仮想空間の対応など、システム実現のためには検討するべき課題が多く残っているが、とりあえず利用者インタフェース部分と基本的機能のみを構築し、これから開発を行っていく Virtual Office の全体像をイメージし、検討の参考にするこの価値は十分存在すると考えられる。

以下では、HyperCard を用いて Virtual Office のプロトタイプを実現した例を示す。

4.2 HyperCardによるVirtual Officeの利用者インタフェースの構築

HyperCardでは、カード上に絵、ボタン、テキストなどを配置し、それぞれのオブジェクトについてHyperTalkと呼ばれるScriptを記述することにより、マウス操作など、システムから報告されるメッセージに対するオブジェクトの動作(オブジェクトの移動や他のカードへの移動、他のオブジェクトへのメッセージの送信)の実現が可能である。

Virtual Officeの空間記述とHyperCardのオブジェクトとの対応は以下の通りとする。

オフィス内の空間：カードで表現する。

オフィス内のオブジェクト：ボタン、またはフィールドで表現する。

4.2.1 機器の機能の実現

ボタン、あるいはフィールドに、それぞれが表しているオフィス内のオブジェクトの機能を実現するためのScriptを記述する。例えば、文書をFAXで送信するには、「文書アイコンをマウスでドラッグし、FAXボタンに重ねてから、発送先をボタンにより選んで発送する」というような動作を行なうようなScriptを記述する。これにより利用者に、実際に書類をFAXによって送信しているかのような印象を与えるようにすることが可能である(図4)。このような利用者インタフェースによって、計算機の操作に慣れていない人でも、容易にVirtual Officeの機能を利用することが可能である。

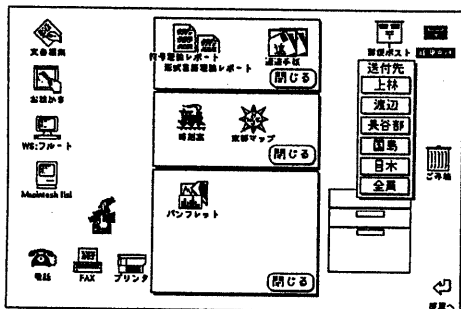


図 4: FAXの発送

4.2.2 データベース機能との結合

前述のFAXの送信の際に、発送先の電話番号はFAX番号データベースより検索し、自動的にダイヤルされる。このとき、FAX番号データベースは背後に住所データベースを想定すると、その一つのビューである。

4.2.3 セキュリティ管理

Virtual Officeの目標とする、共同作業と個人の独立という相反する機能を実現するためには、セ

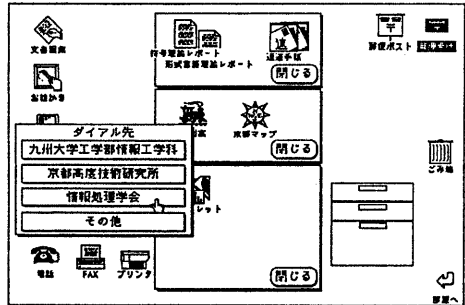


図 5: FAX番号データベース

キュリティ管理は非常に重要である。HyperCardを用いて実現したプロトタイプでは、オブジェクトに対するアクセス権制御と鍵による空間へのアクセス権制御とを併用した(図6)。また、オブジェクトの中身だけを秘密にしたい場合だけでなく、オブジェクトの存在自身を秘密にしたい場合などもあることから、「嘘つき空間」などの実現も行なった。

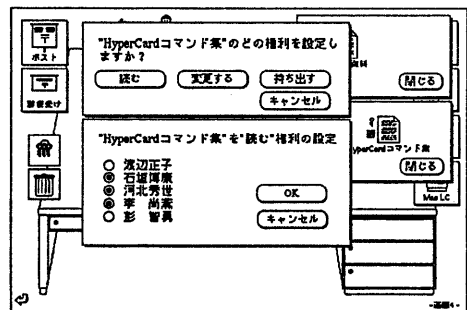


図 6: セキュリティ管理の例

4.3 HyperCardの限界

HyperCardは、優れた利用者インタフェースを容易に実現できるツールであるが、反面、実際にこれを用いてVirtual Officeを実現しようとすると、限界が見えてくる。今回、実際にこのシステムを構築した際の問題点をあげてみる。

Script実行の逐次性 あるオブジェクトに書かれたScriptを実行している間は、他のイベントが発生しても、そのイベントに対して実行されるべき処理がなされないため、会議などのリアルタイム性を要する作業に対して大きな障害となる。また、次に述べるイベントの拡張性に対しても障害となる。

メッセージの非拡張性 HyperCardの標準として用意されていないFAXの到着などのようなイベントに対しては実現が不可能である。新たなイベントに対応するには、目的のイベントを常に監視するようなオブジェクトを作成し、

並列に実行することで実現できるが、並列性は現在の HyperCard では実現されていない。

非共有性 Macintosh ではマルチユーザをサポートしていないために、同一のドキュメントは同時に一人しか開くことができない。このため、共同作業の際の書類のアップデートなどの通信と制御が非常に複雑になる。

メタデータの非保護性 HyperCard においては、セキュリティ管理は Script の記述によって行なうが、Script によっていくら安全な保護をかけても、利用者による Script を書き換えには対処のしようがない。これには、システム自体を記述するメタデータまでも含めたセキュリティが必要であるが、HyperCard では困難である。

4.4 Virtual Office 構築ツールに必要とされる機能

前節および他稿の内容をふまえて、実際に Virtual Office を構築する上で必要と思われる機能についてまとめる。

1. オブジェクト間の通信機能
2. データベースと接続できる機能
3. セキュリティ管理機構が構築可能な機能
4. 複数のオブジェクトが並列に動作できる機能
5. 複数の利用者で同一のオブジェクトを共有できる機能

前節で述べたように、これらの機能のうち、3、4、5 については、HyperCard ではサポートされていない。マルチユーザをサポートしている UNIX ワークステーション上に同様のツールが開発されているので、その上での Virtual Office の開発を予定している。

5 あとがき

現在研究室で開発中の Virtual Office システムについて、その基本概念と、開発のための基礎研究課題の一部について述べた。また、実際のシステム構成時の利用者インタフェースの実現例について紹介した。ビデオで相手の顔を見ながら討論できる機能はぜひとも必要であるが、このような AV 機器との統合は、研究としては新しい面が少ないので、今後の課題とした。Virtual Office の研究開発は始めたばかりであり、システム開発等を通じて発展していくべき性質のものであるため、今後さらに検討が必要となる所が見つかると考えている。

謝辞

本研究に対し御討論いただいた松下電器産業(株)情報システム研究所の今井良彦氏をはじめとする諸氏に感謝いたします。

参考文献

- [1] C.A.Ellis, S.J.Gibbs, G.L.Rein : GROUP-WARE: Some Issues and Experiences, *Comm. ACM*, Vol.34, No.1, pp. 39-58 (1991).
- [2] 石井 裕: グループウェアのデザイン, *bit*, Vol.23, No.3, pp. 273-283 (1991).
- [3] 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).
- [4] 垂水浩幸: グループウェアのソフトウェア開発への応用, 情報処理, Vol.33, No.1, pp.22-31 (1992).
- [5] Stefik, M. et al.: Beyond the Chalkboard: Computer Supported For Collaboration and Problem Solving in Meetings, *Comm. ACM*, Vol.30, No.1, pp. 32-47 (1987).
- [6] H.Ishii: TeamWorkStation: Towards a Seamless Shared Workspace, *CSCW '90, Proceedings*, ACM, pp.13-26 (1990).
- [7] 渡部和雄 他: マルチメディア分散会議システム MERMAID, 情報処理学会論文誌, Vol.32, No.9, pp. 1200-1209 (1991).
- [8] Kum-Yew Lai, Thomas W. Malone, Keh-Chiang Yu. :Object Lens:A "Spreadsheet" for Cooperative Work, *ACM Transactions on Office Information Systems*, Vol.6, No.4, October 1988, pp.332-353.
- [9] 松尾 朗 他: 電子メールにおけるエージェントシステム, 情報処理学会第 43 回全国大会, 2J-13 (1991).
- [10] 奥村晃弘, 北 英彦: 電子メールに基づくグループウェア Brownie 他, 情報処理学会第 43 回全国大会, 3N-5, 3N-6 (1991).
- [11] Haiyan Xu: A Cooperative Transaction Model Handling Multiple Correctness Levels, *DASFAA '91, Proceedings*, pp.517-526 (1991).