

4 T-9

Virtual Office における仮想オフィス間の通信機能

国島丈生 石垣博康 上林弥彦

京都大学工学部

1 まえがき

Virtual Office は、オフィスにおける仕事をコンピュータ上でサポートし、特にオフィスを仮想化することによって共同作業をやりやすくするものである。オフィスの仮想化とは、例えばプロジェクトごとに仮想的なオフィスを作ることができるような機能のことである。このシステムは、ワークステーション、ネットワーク、FAX、電話、AV 機器など仕事に必要なものを統合し、利用者が簡単な操作で作業を進めることができるような使いやすい利用者インタフェースを提供する。

オフィスにおける共同作業においては、さまざまな通信が必要とされる。仕事の依頼や質問、システムからの通知などがその一例である。これらの通信を扱う道具としては、いままでもグループウェアなどにおいて電子メールを構造化したものなどが存在した。また、アクティブデータベースにおけるトリガーモダル[3]は、現実の事象と密接な関連のあるオフィスにおいては欠かせないものと考えられる。それに加えて、ある仕事を取消す際に単純な取消しではなく、埋め合わせる処理を行なうような補償トランザクション[2]の概念も必要となる。

本稿では、これらの考えを取り入れた共同作業における通信機能について検討する。

2 基本的事項

ここでは、Virtual Office の通信機能を実現するために必要となる概念について述べる。

2.1 ECA モデル

オフィスにおける共同作業では、事象の発生やその状況によって動作が起きる、いわゆるトリガーモデルが必要な場合が多い。たとえばメット前になった仕事を知らせるとか、ミーティングの日時を設定したらメンバーへ通知する、などである。また、報告書を提出せよ、といった命令もトリガーとなる。

これらを扱うための基礎として、アクティブデータベースにおける ECA(Event-Condition-Action) モデル[3]について述べる。アクティブデータベースとは、実世界の変化に応じてオブジェクトを変化させていくようなもので一般にトリガーが用いられる。ECA モデルは、アクティブデータベースの一つである HiPAC で用いられており、ある事象(Event)が起きると条件(Condition)が評価され、もしそれが満足されると動作(Action)がトリガーされる、というものである。これらの事象、条件、動作は一つのルールとして記述されシステム内に組み込まれる。

2.2 補償操作

補償操作[2]とは、ある仕事が取消されることになった時に単純に取消するのではなく、埋め合わせるような処理を行なうことによって取消することである。これはまず仕事を単純に取消することが難しい場合に必要となる。例えば人に依頼した仕事を取消す為には、取消す説明を送らねばならない。また、単純な取消が無駄を生ずる場合にも補償操作は有用である。前の例において、取消される側がある程度仕事を進めていた場合、新たに仕事を依頼する相手に対して、誰がそれまで仕事をしてきたかを伝えるという操作が含まれると、それまでの仕事の結果が無駄にならずに済む。

従来のデータベースにおける補償操作の研究は、それが並行処理の正当性を破壊しないように行なえることを目的としていた。ここでの補償操作は、オフィスにおける仕事の内容の正しさを求めるものである。これはオフィスのように人が働き、単純に物事を取消したりすることが難しく、無駄になったりする場合には必要なものであるといえ、現実的に行なわれている行為である。

2.3 グループウェアにおける通信

今までのグループウェアにおいて通信をサポートするものとしては、Information Lens, Object Lens[1]に見られるように電子メールを構造化し、それによりメッセージを優先度順やプロジェクトごとに分けるようにフィルターをかけた、データベースを用いた動的な宛先を実現したりするものがある。このように通信の際に受信側や送信側が優先度をつけたり選択する機能はオフィスにおいては必要となるものだろう。また、他にも The Coordinator[4]のように会話型モデルに基づいて共同作業を支援するものなどもある。

Virtual Office における通信機能は、以上に述べたものをまとめて扱える必要があるといえるだろう。

3 現在のモデルの問題点

現在のモデルでは実現が困難であるが共同作業では一般的なことであり、Virtual Office に必要とされるものについて述べる。

- 条件に優先順位がある。

Virtual Office における通信では、例えば報告書を依頼するにしても何人かの選択肢があり、その中から選ぶというような場合がある。この場合の条件は、相手が責任者や前回報告者、その他の場合などで優先順位があると考えられる。

- 相手の応答や状態により別の動作への要求が出る。

例えば仕事を依頼する場合に、相手が断ってきた時や留守だった時には他の人へと再依頼を行ったり、依頼を置きつつ他の人を探したり相手の応答や状態によってフィードバックが起きる場合がある。

4 Virtual Office における通信例

ここでは、実際の Virtual Office において想定される通信の例として、報告書作成依頼における通信を取り上げる。(図1)

まず依頼者がある仕事における2つの互いに関連した部分、aパートとbパートにおける報告書を依頼する。これがEventとなり、条件をトリガーする。優先度の高い第一条件は、a,b両方を同時に書ける人であり、データベースを参照してA氏、B氏に依頼する。しかしA氏は留守、B氏は拒否であった。そこで次の第二条件を参照し、a,bを別々の人に頼むことにする。結局aはC氏、bはG氏に受理された。また、A氏は留守だったので受理する可能性があり、そのための情報(仕事の内容や現在その仕事を行なっている人(C氏,G氏)など)をA氏に通知しておく。もしA氏が戻ってきてからその仕事を行なうならば、C氏とG氏には補償操作が働き、C氏とG氏にはその仕事をA氏に送るように通知される。

このような例はオフィスにおいて日常起りうることであり、Virtual officeではこれをサポートできるような通信機能を記述する必要がある。次の節ではこれを実現できる一般的通信機能の記述方法について述べる。

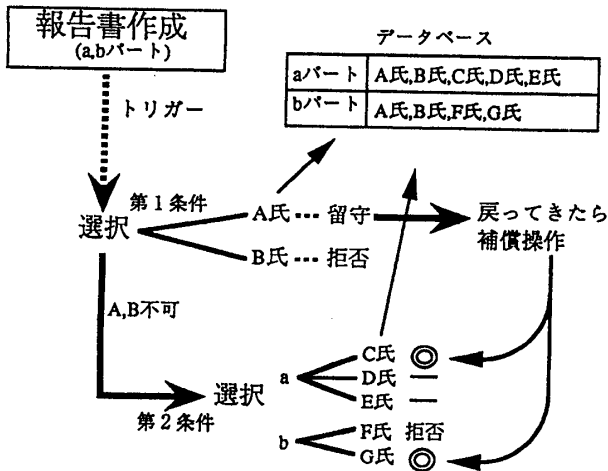


図1: Virtual Office における通信例

5 Virtual Office における一般的通信機能の記述

3章で述べたような問題点があるため、我々の考えている Virtual Office にECA モデルをそのまま適用することはできない。また、上の例で見られるように、通信にはその前の状態によって次の動作が限定されるというオートマトンの要素があり、それをここでとりいれることは必要である。しかし、ここではまず通信部分のみに絞って、以下のようにECA モデルを拡張することにする。

エージェントの動作の満たすべきルールは

$$rule \equiv (event, \{(condition, envelope)\})$$

という形で記述される。ここに、eventは外部からエージェントに送られるトリガイベントを表す。conditionはイベントeventが送られてきた時に評価する条件である。またenvelopeはconditionが成立した時に起こす動作の記述であり、次のように記述される。

$$envelope \equiv (to, action, \{(response, responseAction)\})$$

ここにtoは動作を起こす相手のリスト、actionは動作、responseはactionを起こした時に相手から返ってくる返答、responseActionはresponseが返ってきたときにこのエージェントが起こすべき動作をそれぞれ表す。ただし、responseには「返答が返ってこない」という特殊なケースも書けることとする。responseActionには次のようなものしか許さないこととする。

1. addConditions
rule中のcondition集合に対する追加。
2. deleteConditions
rule中のcondition集合に対する削除。
3. changeConditions
rule中のcondition集合に対する変更。
4. changeState
エージェントの状態の変更。
5. triggerEvent
イベントを新たにトリガーする。

eventが発生した時に起こし得る動作は一般には複数あり、通常はそれらには優先順位がある。そこで、conditionにその評価順序を表す優先順位を付けられるものとし、eventが発生した場合その優先順位にしたがってconditionが評価され、最初に成立したものに对应的envelopeが実行されることとする。conditionの優先順位の変更はchangeConditionsで行えるものとする。

responseに一致する返答が返ってくると、responseに对应的responseActionが実行された後、1~4の場合はconditionが再評価されて、再度対応するenvelopeが実行される。

この動作モデルでは補償操作は通常のトリガおよびそれに対する操作と同様に書くことができる。

6 あとがき

Virtual Officeにおける通信機能とその記述方法について考察を行なった。今後は実際の言語の定義や、それを容易に記述できるような利用者インタフェースの開発を行なう予定である。

参考文献

- [1] 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.
- [2] Henry F.Korth, Eliezer Levy, Abraham Silberschatz. :A Formal Approach to Recovery by Compensating Transactions, *Proceedings of the 16th VLDB Conference*, Brisbane, Australia 1990, pp.95-106.
- [3] Dennis R.McCarthy, Umeshwar Dayal. :The Architecture Of An Active Data Base Management System, *ACM SIGMOD*, 1989, pp.215-224.
- [4] Winograd, T. :A Language Perspective on the Design of Cooperative Work, *CSCW'86*, December 1986, pp.203-220.