協同作業環境を考慮したECAルール

5S－7

上林彌彦　野本政和

京都大学工学部

1　はじめに

現在著者らの研究室では，データベース技術を基礎としたオフィスにおける協同作業支援システムとして *VirtualOffice*の開発を行なっている[1]．ここで必要な機能として，システム内または利用者に関する何らかの事象の発生時に，その時の状態に基づいて，あらかじめ設定されている動作を自動的に起動するというものがあげられる．

本稿では，このような機能を実現する機構としてECA機構[2]をとりあげ，*VirtualOffice*上で必要となるECA機構の基本的な機能とSmalltalkを用いたその実現について述べる．また，応用例として作業依頼などに関するグループ内の複数メンバ間での対話の支援について示す．

2　ECAルール

2.1　能動的なルールの実行機構

ECA機構は，データベース内に登録されたルールに基づいて，利用者の関与なしに自動的にあらかじめ指定された処理を行なう機構である．このような機構は，能動データベースと呼ばれるある種のデータベース管理システム上で実現されている．個々のルールは事象（Event），条件（Condition），動作（Action），E-C結合，C-A結合の五つの属性を持っており，指定された事象が発生すると対応する条件が評価され，それが満たされていた時にはそれに応じた動作が起動される．またE-C結合，C-A結合とは，どのようなタイミングで条件の評価，動作の起動を行なうかをデータベース操作のトランザクションとの関係で指定するものである．

2.2　協同作業支援への応用

オフィスなどにおける利用者間の協同作業をサポートするためには，データベースは単に要求に応じてデータの操作を行なうだけでなく，データの変更にともなって種々の処理を行なうことが要求される．

例えばある作業スケジュールに変更が生じた時には，その作業に関係する利用者に連絡をし関連する作業の再スケジュールを行なうなどといった処理が必要になる．

さらに協同作業の支援のためには，ルール中の事象や動作の対象としてデータベース操作などのようなマシン側の動作だけでなく，人間であるシステム利用者の動きを記述できると有用である．

例えばルールの動作部に利用者への作業要求を記述できるようにすると，以下のような機能の実現が考えられる．

- グループ内のスケジュールをルールの集合として記述，管理する．前の作業が終了すると自動的に次の者への作業要求を送るというようなルールを使ったワークフロー管理を行なうことが可能となる．
- 特定の条件が満たされた際に必要となる作業要求をあらかじめ指定しておく．

3　協同作業のためのルール実行機構

3.1　利用者間の対話プロトコル

2.2節ではルール中の動作部として他の利用者への作業依頼を記述することを考えた．作業を行なう主体が機械である場合には，作業を要求する側は実行命令を発行するだけで，あとは作業が終了するまで処理は自動的に行なわれる．しかし人間が作業を行なう場合に，一方的に作業要求を送りつけるだけでは以下のような問題点が考えられる．

- 作業者側に他に優先度の高い作業がある時などにも，依頼を拒否することができない．
- 作業依頼時，または途中での条件や内容の変更要求に対応できない．
- 依頼者側で作業がどの程度進行したかを途中で把握することができない．

このため，人に対して作業要求を行なう場合には単に要求メッセージを送るだけではなく，依頼者側と作業者側の間での対話が欠かせない．そこでThe Coodinator[3]にあるような状態遷移を考慮した対話をサポートすることとした．その一例を図1に示す．このパターンはどの作業についても共通であると考えられる．以下，このようなパターンを利用者対話プロトコルと呼ぶ．

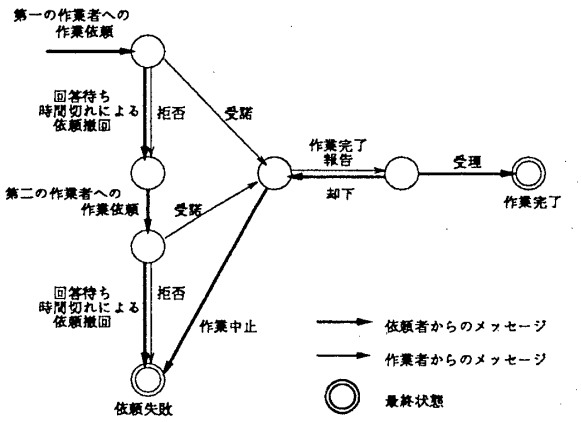

図 1: 利用者対話プロトコルの例（依頼先が二人の場合）

ECA Rules for Group Work Environment
Yahiko KAMBAYASHI and Masakazu NOMOTO
Faculty of Engineering, Kyoto University

ルール中の動作として利用者に対する作業の要求を指定した場合には，単に相手に対して要求メッセージを送るのではなく，以後この対話プロトコルを起動させる必要がある．

3.2 利用者対話プロトコルのルールによる表現

前節で示した対話プロトコルは依頼者，作業者の両者の間でのメッセージの送受という事象によって状態遷移を行なうモデルとしてとらえることができる．

ここでECAルールを用いると，以下により対話プロトコルを実現することができる．

i) 対話プロトコルの現在の状態をデータベースで管理する．

ii) 新たな対話の開始，その状態の変化に伴い，相手に対してメッセージを送るようなルールをシステムが準備する．

iii) 対話のための利用者インタフェースを与える．

4 ルール実行機構の実現

4.1 基本機能

まず，先に述べたECA機構の基本的な部分を作成した．ここで実現したルールを以下基本ルールと呼ぶ．事象は事象名をシンボルとした正則表現で表すこととしている．条件，動作はいずれもSmalltalkのメッセージ式で記述する．

この基本ルールをルールマネージャに登録すると，以後指定された事象が発生するたびに条件式が評価され，その結果が真であれば動作が起動される．また，ルールを削除することなく不活性にすることができ，この状態の時には指定された事象が発生してもルールは発火しない．

基本ルールは動作としてSmalltalkの任意のメッセージ式を認めるため，利用者に直接ルール定義を行なうことを認めると，誤りや故意によってシステムに破壊的な動作を行なわせる可能性がある．そのため，このルールはシステム内部によってのみ定義，削除が行なわれるものとし，一般利用者による直接操作は認めていない．

4.2 応用例

さらに，前節の基本ルールを用いて，利用者によって自由に定義できる利用者ルールを用意した．ここではシステム保護と利用者の設定の簡易化のため，指定できる事象や条件，動作はシステムが準備したものの中から選択するようになっており，任意の処理を行なうようなルールの定義はできない．利用者ルールは図2のようにウィンドウに表示された各項目を利用者が埋める形で対話的に定義される．

利用者ルールの動作には作業依頼が指定でき，この場合対話プロトコルに沿った依頼先の相手との対話が開始される．この利用者間の対話は4.1節の基本ルールの組合せによって実現されている．そのためにウィンドウやメッセージの表示などの基本的な機能を提供するモジュールを用意している．また，これらのクラスを用いることによって，対話プロトコルを変更するような場合にもルール群の定義

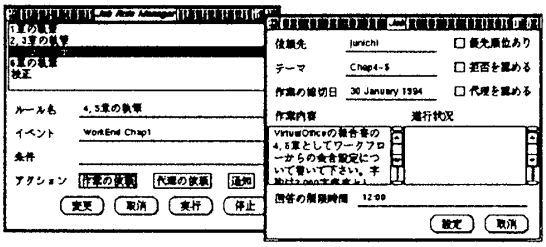

図 2: 利用者ルール定義の例

を変更することによって対応することができる．

対話の中では，受けた依頼の代理実行を他の利用者に依頼したり，作業中にその作業を分割し，他の利用者への下請け依頼を行なうこともできる．これらの場合には，代理や下請けへの依頼が行なわれるという情報が元の依頼者のもとに送られる．また，新しい依頼の条件には元の作業との関係でいくつかの制約がある．例として

- 代理，協力依頼先の利用者の中に，元の依頼者が含まれていてはならない．
- 新しい作業の締切日は元の作業のそれよりも早くなければならない．

などがあげられる．このような制約は依頼の定義の際にチェックされ，誤りがあれば訂正を要求する．

5 おわりに

本稿では能動データベースで実現されているルール実行機構について，協同作業支援システムへの応用と，それによる利用者間の作業依頼に関する対話の実現について述べた．今後は実際のオブジェクト指向データベースとの結合やトランザクション管理の機能を実現するとともに，作業スケジュール管理機能との統合を行ない，また利用者に対してもより自由度の高い機能を提供できるよう拡張していく予定である．

謝辞

本研究について御討論いただいた國島丈生氏をはじめとする上林研究室の皆様に感謝致します．

参考文献

[1] Hideyuki Takada and Yahiko Kambayashi: An Object-Oriented Office Space Description Model and an Office View Management Mechanism for Distributed Office Environment, 4th Int. Conf. on FODO, pp. 362–377, Springer Verlag, (Oct. 1993).

[2] Dennis R. McCarthy and Umeshwar Dayal: The Architecture Of An Active Data Base Management System, Proc. ACM SIGMOD, pp. 215–224, (Jun. 1989).

[3] Terry Winograd: Where the Action Is, BYTE, Vol. 13, No. 13, pp. 256A–258, (Dec. 1988).