

4N-5

マルチエージェント分散対話スケジューリングシステムにおけるマシン対
マシン通信プロトコルの設計

Jonathan SHIH 荒木雅弘 堂下修司

京都大学 工学部 情報工学教室

1 はじめに

これまで我々用いてきた対話システムは全てのユーザが一つの対話管理プロセスと1対1の形式でマン-マシンインタフェースのレベルで対話を行い。ユーザの要求を一つ一つシーケンシャルに取り入れ、コンフリクトが生じるごとにシーケンシャルに解決法を求めるものである。

ここで提案するマルチエージェント分散対話スケジューリングシステムとは、分散管理を対話システムに適応するより、対話スケジューリングをマルチエージェントの研究の一例として扱うというほうが適切である。システムの設定は各ユーザのスケジュールに関する処理は同時に別々の対話管理プロセスに管理されている。それらのプロセスをエージェントとして扱い、複数のユーザが会議のようなイベントをスケジュールしたい時に発生する可能な障害状況を想像し、対応策の実行によって、イベントの登録を目指す。従来の対話システムとの一番の相違点は、各プロセス間のインタラクションを処理するために、もう一層の低レベルマシン-マシンインタフェースが必要であることである。今日このこのマシン-マシンインタフェースの通信プロトコルの設計を行った。

プロトコルはポリシー、言語、通信とアーキテクチャという四つの部分に分けて別々に設計する。今日の発表内容は主にポリシーと言語に関わっている。ポリシーでエージェントが対話を行う時の規則を定義する。即ち、対話を行う際、特定の状況に応じて起動すべきプロセスとそれらのプロセスの間の情報の流れを明確する規則である。言語の部分では、エージェントとの間に行う対話で使うエージェント通信言語

を定義し、メッセージの構成規則と意味解析のプロセスを定義する。そして、定義した規則と言語を使って、スケジューリングの過程に生じるコンフリクトを対応するための対話による問題解決手続きを指定する。

以下ではエージェントプロセスと情報の流れ、エージェント通信言語、そして、コンフリクトの解消という三つの部分に分かれ説明する。

2 エージェントの仕事と入力/出力の流れ

システムでは、二種類の対話がある。一つはエージェント内部で行う対話管理部とマシン-マシンインタフェースの対話、もう一つはエージェントの間の対話である。エージェン

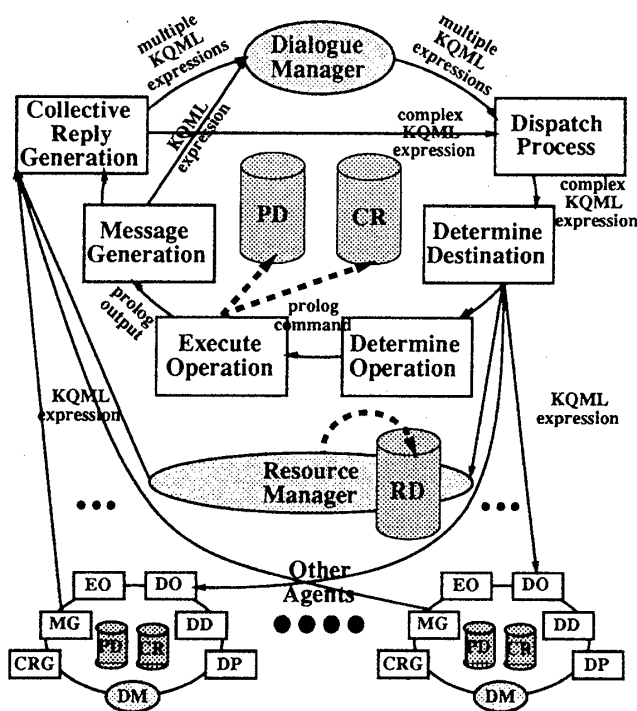


図1：エージェントの内部プロセスと情報の流れ
トの仕事と情報の流れは図1で示してあるように、まず、対話管理部はユーザが発する自然言語文を入力として取り入れ、その文に含まれている複数の要求を同数の KQML 表

現で表す、これはマシン-マシンインタフェースの入力になる。対話管理部で行う自然言語処理はこの研究では扱わない。KQML 文を取り入れた時点で、まず、要求文の数を確認し、一つずつ処理を行う。次に、要求文を送る対象になるエージェントを確認してメッセージを出す。メッセージを受け入れたエージェントは KQML の表現を prolog の命令に解釈し、データベース上の処理を実行する。実行の結果に基づいて、応答文になる KQML の表現を生成し、イベント管理者であるエージェントへ返事を出す。イベント管理者は各エージェントからの応答文を分類し、応答文のタイプの数と同数の KQML 表現を生成して、これは対話管理部へ出す応答になる。未処理の要求文が残ってあれば、同じ作業を繰り返して実行する。

3 エージェント通信言語

エージェント通信言語として KQML の performative のサブ集合と自分で定義したスケジューリングを対象にしたオントロジの結合を用いる。ここでは、一つの例を使って言語の文法と意味論を簡単に説明する。エージェント 1 はエージェント 2 に 7 月 20 日の午後 1 時と 2 時の間に空いているかどうか確認したい時に、次のような KQML 表現を出す。

```
ask-if:content (? (open (((from 1300) (til 1400))
(startdate 7/20/94)))) :reply-with agent1
```

ここで、ask-if は KQML の performative であり、:content の下のメッセージはスケジューリングのオントロジで構成したメッセージである。メッセージの一貫性を保証するために、各 KQML の performative とオントロジ単語の使い方は BNF で制限し、使用可能な状況も明確に限定する。メッセージを生成する際、状況に応じて、適切なテンプレートを選んで、オントロジの BNF に従って、テンプレートに適切な単語を埋め込む。メッセージの意味解釈はこのプロセスを逆にするだけである。すなわち、両方ともマッピングの問題に還元される。

4 マルチエージェントによるコンフリクト解消

マルチエージェントによるコンフリクトの解消はこの研究の最大の課題になる。現在扱っているコンフリクトの種類について図 2 を参照して下さい。ここで示してある四タイプのコンフリクトの解決法を定式することによって、全てのスケジューリングコンフリクトをカバーできるわけではない。まず、最もよく生じると考えられる一般的なコンフリクトに対応して、その後は評価によって、扱うコンフリクトの種類を増やすアプローチをとる。難点は、どこまで細かく例外処理を行わなければならないか、各コンフリクトに対する解決法が完全であるかどうかの判断が難しいことである。

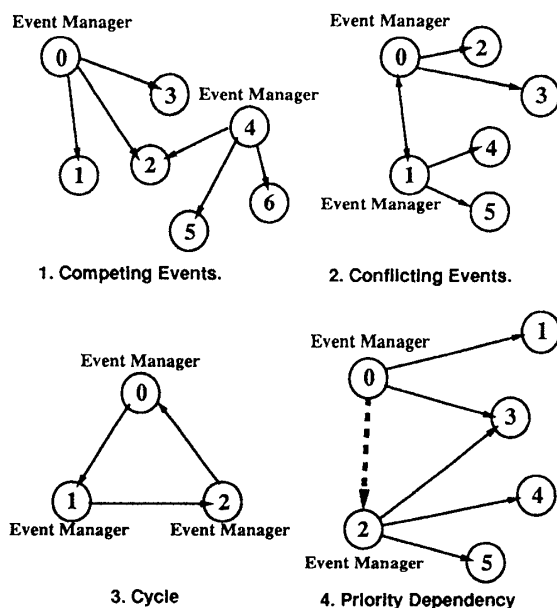


図 2：扱うスケジューリングコンフリクト

コンフリクトの解消の前提として、エージェント及びイベントはある半順序に従うことを仮定した。全てのエージェントとイベントの大切さを平等にすれば、決着できないコンフリクトがあるから。

5 おわりに

この研究の一番目的は、従来の対話システムと比べてより優れる点を挙げるのではなく、システムの分散化によってどのような面白い問題を生じるか、そして、エージェント間の対話による交渉によって、どこまで問題を解決できるかを評価することになる。即ち、我々にとって重要な問題はエージェントがどこまでの自律度を達成できるか、そして、サーバにある資源管理部の仕事はどうやって最小限にするかになる。2 節で紹介したエージェントの通信規則と 3 節で述べたエージェント通信言語は全て 4 節で示したようなコンフリクト問題の解決に向かう必要なツールと考えられる。

参考文献

- [1] T. Finin, J. Weber, G. Wiederhold, M. Genesereth, R. Fritzson, D. McKay, J. McGuire, P. Pelavin, S. Shapiro, and C. Beck. (1993) Specification of the KQML Agent-Communication Language. Enterprise Integration Technologies, Palo Alto, CA, Technical Report EIT TR 92-04, Updated July 1993.
- [2] Gruber, T.R. (1992) A Translation Approach to Portable Ontology Specifications, Proceedings of the Second Japanese Knowledge Acquisition for Knowledge-Based Systems Workshop 1992 pp.89-107.