

## アトミックマルチエージェントによるグループウェアシステム

5 Y-3

白川 洋充

近畿大学工学部経営工学科

## 1 まえがき

信頼性を有する分散システムの実現を目的として70年代終わりにグループ通信の研究が始まった。それ以後、グループ通信は理論面で著しく進歩した。実際面では、コーネル大学で開発されたISISが信頼性を有する大規模システムの構築の可能性を示したことを契機として、グループウェアシステムにグループ通信を適用する研究が進められている。

信頼性を有するグループウェアシステムを実現する場合、オフィス環境で使われるグループウェア、例えば、協同文書作成システムなどでもグループ通信を適用して信頼性を向上させる試みがなされている。しかしながら、これらの研究は単にグループウェアシステムの信頼性を高めるためにISISシステムを採用したにすぎない。我々の研究の目的は単にグループ通信を採用するだけでなく、グループウェアに必要なグループ通信を根本的な観点から見直し、エージェントの協調によって信頼性を高めることができることを示すものである。

## 2 グループウェアシステムのプロトコルの基本的要件

本節では、グループウェアシステムのプロトコルの基本的要件としてメンバーシップと協調のパターン、合意について述べる。

## 2.1 メンバーシップと協調のパターン

グループとはデータやファイルなどのようなパッシブな実体のまとまり、あるいは、エージェントやプロセッサのようなアクティブな実体の集合と考えることができる。人間は、アクティブなエージェントと関連し、パッシブな実体を操作すると考える。ここで、グループの正当なメンバーを何に選ぶかが問題になる。後述する耐障害性を考えてグループの正当なメンバーとしてはプロセッサを選ぶことが合理的である。この考え方によると、プロセッサとエージェントは同時にアクティブにし、エージェントがインアクティブになると同時にプロセッサもインアクティブにする機構が必要になる。この機構を連座制ということにする。したがって、グループから見ると、プロセッサがアクティブであるとエージェントは常

にアクティブであると仮定する。

さて、エージェントはプロトコルを司るエージェントとその他のエージェントに分けられる。その他のエージェントには、ユーザとのインタフェースをとるもの、システム資源に唯一アクセスできるもの、モバイルエージェントを受け入れるアクセプタエージェントと移送されたエージェントの実体であるアプリケーションエージェント、アプリケーションの結果をマルチキャストするレスポンドエージェント、他のアプリケーションの結果を収集するコレクタエージェントがある。

## 2.2 合意

グループ通信における合意とは、1. 障害のないすべてのリシピエントは同じ値で合意する。2. センダであるプロセッサが障害のない場合、障害のないリシピエントはセンダが送った値で合意することをいう。ここで扱う合意は次の通りである。

At-least-N: あるリシピエントに配送されたメッセージは、正当なアドレスを持った少なくともN人に配送される。

At-least-To: リシピエントの部分集合  $P_i$  を考えたとき、あるリシピエントに配送されたメッセージは、 $P_i$  の正当なすべてのリシピエントに配送される。

Best-effort-N: センダに故障が無いとき、あるリシピエントに配送されたメッセージは、正当なアドレスを持った少なくともN人に配送される。

Best-effort-To: センダに故障が無いとき、あるリシピエントに配送されたメッセージは、 $P_i$  の正当なすべてのリシピエントに配送される。

グループ通信において全員一致の合意を目指してセンダがメッセージを送信しても、センダがプロトコルを終了する直前に障害を起こした場合に、あるリシピエントのボランティアが代わりにプロトコルの終了を責任もって行う。また、リシピエントがオミッション障害を起こしているときプロトコルがその障害を回復できない場合、一時的に不本意ながらも離脱する場合である。これは、通常はオミッション障害の最大発生数を仮定し、その範囲内では障害の回復を行うが、これを越えると回復できない場合である。センダはリシピエントがクラッシュしているか、オミッション障害をおこしているのか判断できない。それで、一時的に、正当なリシピエントとしての取り扱いをしないが、メッセージを送り続け追跡調査する。もし、ある時間たって、障害が回復すれば、

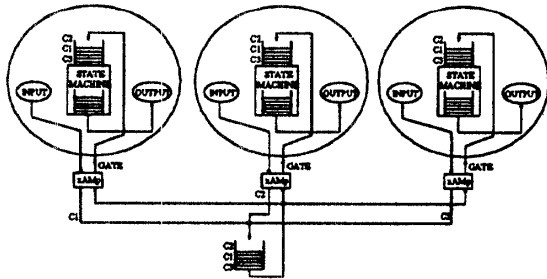


図 1: 状態マシンによる複製

そのレスピエントは速やかに回復作業を行って、もとの状態に戻る。これは、スタートアップの状態から回復するより効率的である。このような合意はフェイルサイレントのシステムを対象としているときには適当である。

### 3 複製の決定性

グループウェアの信頼性を向上させるために、各プロセッサで生成した結果はすべてのプロセッサに複製を作る。前節で述べたようにグループウェアではフェイルサイレントを仮定していることと、合意は少なくとも  $N$  台が保証しているの、正当なメンバーである限り  $N$  台に複製を作ることになる。複製をどのようにして作るかという問題は「複製の決定性 (replica determinism)」というが、本稿ではこの手法は状態マシンの手法として知られているものを使う [2]。状態マシン (state machines) とは、状態をエンコードする状態変数 (state variables) とその状態を変換するコマンド (commands) から成る。また、状態マシンは各プロセッサごとに存在しコマンドをコンカレントに、かつアトミックに処理する。ここで、クライアントの発行するコマンドが各状態マシンで処理される順序について次の仮定をする。

- 複数のクライアントによって発行されるコマンドはすべてのプロセッサの状態マシンで同じ順序で処理される。

クライアントの要求が各状態マシンでコンカレントに処理される順序は全順序であったことから分かるように、グループウェアの順序関係は全順序でなければならない。

### 4 グループ通信の応用

グループ通信の応用として、アンケートの集計を考える。前節で述べたように、アンケートの配布はあるエージェントがマルチキャストするが、一旦、アンケートがマルチキャストされて、レスピエントがアンケートをデリバすると、全エージェントが回収に携わる問題である。各メンバーに対するアンケートは Java Applet を含む HTML ファイルをマルチキャストする。この HTML ファイルをレスピエントがロードして、ブラウザで Java Applet を

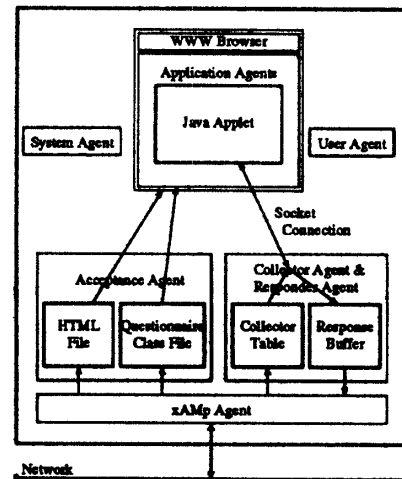


図 2: アンケートの集計

表示するためには、アンケートに対するさまざまなデータと操作が同時にマルチキャストされる必要がある。これが Questionnaire Class ファイルである。各レスピエントはこのクラスをインスタント化して、Questionnaire オブジェクトを生成する。図 2 がこのシステムのブロック図を示している。ブラウザでまず表示される Java Applet はアンケートの回答の集計表 (Collector Table) である。この結果は全員には非公開の場合があるが、説明を簡単にするため公開されていると考える。最初は集計の結果は表示されないが、回答が集められると途中の結果が表示される。個人の回答は、Java Applet のボタンを押して回答のブラウザを開く。必要な欄に記入するとこれがレスポンスバッファ (Response Buffer) に送られ、全員に送付される。

### 5 あとがき

グループウェアを実現する場合に、システムの信頼性、一貫性を保つためにグループ通信が必須になる。しかし、グループ通信はグループウェアにとって一番適切な QoS を決定しないと実現できないという問題がある。ここではグループウェアシステムに適切な QoS を使用する。また、マルチエージェントの協調によって障害の回復と複製を作ることが可能であることを示した。

### 参考文献

- [1] L. Rodrigues and P. Verissimo, xAMP: a Multi-primitive Group Communications Service, Proceedings 11th Symposium on Reliable Distributed Systems, pp. 112-121, 1992.
- [2] F. B. Schneider, The State Machine Approach: A Tutorial, ACM Computing Surveys, 22(4), pp. 299-319, 1990.