

Awareness アナロジーによる分散オブジェクトの緩やかな連携

岡田 誠 岩尾 忠重 牛嶋 悟 高田 裕志

富士通研究所 ネットメディア研究センター

okadamkt@flab.fujitsu.co.jp

本論文は、分散オブジェクトの緩やかな連携における Awareness アナロジーについて述べている。CSCW(Computer Supported Cooperative Work)における Awareness モデルについて述べた後、分散オブジェクトの緩やかな連携としての Awareness アナロジーについて述べる。さらに、Awareness アナロジーにおけるもっとも重要な特徴がメッセージとアクションとの分離にあることを述べ、このような分離が分散オブジェクトの緩やかな連携として参加・介入・連結といった相互作用へとつながることを示す。さらに、負荷分散のない系から契約ネットモデルに基づく負荷分散のある系への動的な機能拡張を、その例として示すことにする。

1. はじめに

分散オブジェクトシステム技術においては、CORBA[1]や DCOM[2]に代表されるように、ネットワークに対して透過的なオブジェクトアクセスを提供するための枠組みや、分散オブジェクトのインターフェイスに関する枠組みが徐々に整いつつある。

我々は、分散オブジェクトシステムにおける次の課題の一つに、オブジェクト間の連携をいかに動的に構築していくかという問題が存在すると考えている。特にインターネットや移動型ユーザーのことを考えると、オブジェクトの実行環境自体が多様化した動的に変化するような状況が今後ますます増えてくるだろう。

このような問題に対応するためには、オブジェクト間の連携をより柔軟にすることが必要となる。これは単純にアプリケーションをコンポーネント化していくだけでは解決できず、オブジェクト同士が相互に作用しながら動的に機能を構成するような枠組みが必要となる。

しかし、連携するオブジェクト同士が、互いの内部状態や内部関数に依存しあうような密な関係があれば、相互作用によって機能を動的に構成していくことは容易でない。したがって、連携によって機能を動的に構成しようと思えば、連携するオブジェクトの疎な関係を規定しなければならない。

CSCW (Computer Supported Cooperative Work) の分野では、以前から人間同士の連携モデルとして Awareness という概念が導入され議論されてきた[3-9]。このモデルは、人間同士のコミュニケーションにおいて

『気づき(Awareness)』という状態が存在し、Awareness に基づく情報のやり取りが人間のコミュニケーション行動にバリエーションと柔軟性をもたらしているとするものである。

分散オブジェクト技術の分野においても、Awareness モデルは連携する相手のオブジェクトが動的に変化してしまうような状況を想定した場合に有効となり得る。それは Awareness モデルが連携における疎な関係を規定し、連携に柔軟性をもたらすからである。

本論文では、分散オブジェクトの緩やかな連携モデルとしての Awareness アナロジーを示し、さらに Awareness アナロジーにおけるもっとも重要な特徴がメッセージとアクションの分離にあることを述べた上で、このような分離が分散オブジェクトの緩やかな連携として、参加・介入・連結といった相互作用へとつながることを示す。さらに、負荷分散のない系から契約ネットモデル[3]に基づく負荷分散のある系への動的な機能拡張を、その例として示す。

2. CSCW における Awareness モデル

CSCW における Awareness モデルは、人間同士のコミュニケーションにおいて、具体的で特定可能なコミュニケーション行動とは別に、『気づき(Awareness)』、というレベルの情報交換が存在するというものである。人の場合の『気づき』とは、人が部屋に入ってきたのがわかるとか、誰かと誰かが話しているのが聞こえるなどのことを指している。Awareness 情報として伝達される

これら『気づき』を導くメッセージが、人間がコミュニケーション行動を取るにあたって重要な役割を果たしている、というのが CSCW における研究の結論である。

たとえば Gaver[4]は、協調作業におけるコミュニケーションの深化は Awareness 情報に基づくものであることを指摘し、音をきっかけとする協調作業システムの有効性を述べている。また小幡ら[5,6]は、人間におけるコミュニケーション開始の過程を分析し、人間にはコミュニケーションを開始する前に、Awareness 段階といえるコミュニケーション行動開始判断の過程が存在すると述べ、この分析をもとに、遠隔地にいる共同作業者の協調を円滑にサポートするために、ビデオによって Awareness 情報を伝達することを提案した。また奥山ら[7]は、コミュニケーションを行う相手の状態をアイコンで示すアプリケーションを作成し、遠隔地にいる共同作業者の様子や現在の状態を知らせる Awareness 表示が、次のコミュニケーション行動を決定するきっかけとなりえることを示している。

CSCW における Awareness モデルはまた、人間のコミュニケーション行動を仮想空間上で表現するにあたって、共有仮想空間における相互作用のための概念としても利用されている[8,9]。たとえばオーラの接触というモデルでは、アバタ(仮想空間におけるヒトを表現するオブジェクト)がその周りの空間的に広がった一定領域にオーラ(Aura)を持ち、オーラが接触しているもの同士だけが相互作用を行い、コミュニケーション行動もその影響を受けるというものである。このモデルではさらに Awareness のパラメータとして、視覚域としての Focus や周りに対する影響力 Nimbus といった概念が導入されている。

このように CSCW における Awareness モデルとは、コミュニケーション行動とは別に、コミュニケーションの主体が直接的、合目的に関わらない状態もしくは情報のやりとりを認めた上で、そのような Awareness 情報が、実はコミュニケーション行動を含む様々な行動のバリエーションに影響を与える、とするものと考えることができる。

3. Awareness アナロジー

分散オブジェクトの柔軟で緩やかな連携のために CSCW で議論された Awareness モデルを適用することについて考えてみよう。ただしここでは、分散オブジェクトに人間のような高度な知的処理は期待していない。

まず、簡単のために Awareness のもっともシンプ

ルな以下の事例について考えてみる。

事務所で研究者 A と B が作業をしている。そこへ研究者 C が疲れた様子で入ってくる。A は C に提出を約束したレポートの準備のことを思い出し準備に取り掛かる。B は C の疲れた様子を見て C に頼もうと思っていた作業を自分で行うことにした。

この例では、研究者 A, B の行動は C の出現とその様子によって影響を受けている。しかし C 自体は A, B に対しアクションを要請する行動をとったわけではない。また C の入室という行為と A, B の取った行動の間には、あらかじめの双方の合意に基づいた取り決めは存在しない。

この例に見られるように、Awareness に基づく行動はあらかじめ十分に規定されたものではなく、人間同士の疎な連携として機能している。

この例において Awareness に基づく疎な連携は以下のような特徴を持つ。

- 1) Awareness 情報に相当するメッセージ(Cの入室、疲れた様子)が C(発信者)から A, B(受信者)に送られている。
- 2) A, B, C は時間および空間を共有しているが、メッセージ自体は受信者 A, B を特定して発信されたものではない。
- 3) 発信者 C の意図とメッセージ受信者のアクションとは独立している。
- 4) 発信者 C から送られた同一のメッセージに対して受信者 A, B はそれぞれが独自の行動を起こしている。
- 5) 同一のメッセージに対して A, B がそれぞれいつも同じ行動を取るとは限らない。

本論文では、上記の特徴を持ったオブジェクト連携を Awareness アナロジーとして考えることにする。この定義は CSCW における Awareness モデルと完全に一致するものではないが、本来目的とするオブジェクト間の緩やかな連携を議論するには十分であり、またここを出発点とすることで柔軟なオブジェクト連携について議論することが出来ると考える。

4. Awareness アナロジーの基本要素

上述した Awareness アナロジーに基づく分散オブジェクトの連携は、以下のような構成によって実現することが出来る。

- 1) Awareness メッセージは、共有もしくは Broadcast されている。

- 2) Awareness メッセージを受信するオブジェクトはそれぞれ独立に、反応し得るメッセージのパターンと反応としてのアクションとを持っている。
- 3) Awareness メッセージを送信もしくは受信するオブジェクトにおいて、メッセージとアクションとの関係は分離されている。このことにより送信側と受信側との関係もまた分離されている。

先の例に対応させれば、1)は事務所という場所とCの入室という時間が共有され、かつCの入室とその様子という Awareness メッセージが、音と光を媒体としてAとBに Broadcast されたことに相当し、2)はCの入室とその様子に対してA,Bそれぞれが独自の行動を起こしたことに相当し、3)は同じCの入室とその様子が、別の場合にはA,Bに別の行動を引き起こす可能性があるということに相当する。

我々がこれまでに発表してきた緩やかなオブジェクト連携モデル(Field-Reactor Model)[10,11]は上記の要件を満たしている。

Fig. 1にField-Reactor Modelの概念図を示そう。1)については、FieldはLANにおいてはUDPブロードキャストとして実装され、InternetにおいてはIRCチャンネルを用いて実装されていた。Fieldに流されたメッセージはすべてのReactor Objectに届く(Awarenessされている)。2) 3)についても、Reactor Objectは、Reactor Tableと呼ぶFieldに流れるメッセージとオブジェクトの持つアクションとを結び付けるマッチングテーブルを持っており、このテーブルはメッセージとアクションをそれぞれ独立に書き換えることが可能となっている。もちろん各オブジェクトの持つReactor Tableはオブジェクト毎に異なっている。

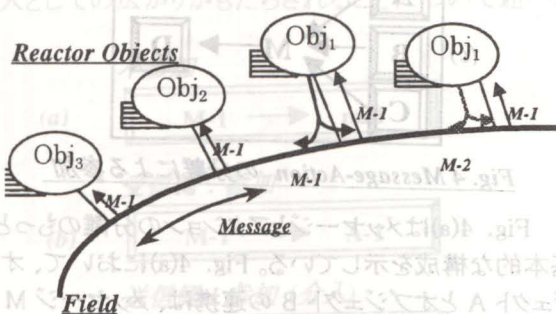


Fig. 1 Field-Reactor Model

Awareness メッセージの共有という意味では、いわゆる掲示板型サービス(Notice Board Type Storage Service)も Awareness アナロジーの1)の要件を満たすことができる。たとえばエール大学で開発された並列

分散プログラミングのための通信モデルであるLinda[12]は、すべてのプロセスによって共有される Tuple-Space と呼ばれる空間を考え、そこに Tuple と呼ばれるデータを送り出すことで通信を行うモデルであるが、Tuple-Space を Awareness メッセージの共有ボードとして利用することは十分可能である。また Tuple-Space を用いた通信は、Tuple の利用方法については一切関与しないモデルであるため、2)3)の要件を満たす実装も可能である。現段階では、Linda アプリケーションの多くは大規模並列計算への適用にとどまり、ここで述べたような Awareness アナロジーに類する分野への適用は一部[13]を除いて多くはない。が、今後は、Sun からは JavaSpace[14]として、IBM からは T-Space[15]として、従来の Linda に Notification 機能を追加した派生モデルを DHCP に相当する機器の Awareness とも言える分野に適用することが提案されているように、分散オブジェクトの緩やかな連携モデルとして用いられる例は増えていくことだろう。

この他、1)に類するメッセージ配信の実現においては、Server 自身はクライアントオブジェクト間でやりとりされるメッセージの内容や構造には関知せずその配信のみに専念する通信プロトコル(Notification Service Transfer Protocol)[16]が Lotus から提案されている。OMG においても現在、独立に Notification Services が検討されている。また TIBCO の Information Bus[17], Lotus の InfoBus など Publish/Subscribe の枠組みも存在し、特に InfoBus についてはその枠組みをネットワークワイドに拡張するために Reliable IP-Multicast を利用した InfoBus Repeater[18]が提案されている。また IONA の Orbix-Talk[19]は CORBA の Event Service を独自拡張したもので、やはり Reliable IP-Multicast を用いた Publish/Subscribe を作っている。

いずれにせよ、Awareness メッセージの配信自体はさまざまな方法が可能であるが、Awareness メッセージが共有される、あるいは Broadcast されるというだけでは十分ではない。Awareness アナロジーとして、より柔軟なオブジェクト連携モデルを目指すには 1)のレベルに留めるのではなく、2)3)まで踏み込むことが必要である。特に 3)のメッセージとアクションとの関係が十分に分離されていることは、系としての柔軟性の実現において特に重要となる。

5. メッセージとアクションの分離

分散オブジェクトの緩やかな連携において、メッセージとアクションとが分離されている、あるいは分離可

能であるということは、オブジェクト間の相互作用として参加と介入という二つの重要な要素を付け加える。

Fig. 2 に、Field-Reactor Model で示した2台のマシンにおける入力共有の例を示す。これを用いて参加の例を示そう。

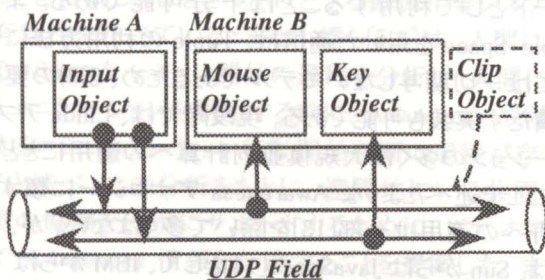


Fig. 2 入力共有と Application の追加

この例では、Machine-A の Input オブジェクトは、Machine-A におけるマウスイベント、キーボードイベントを Field に流している。一方 Machine-B では、Field に流れるマウスイベントやキーボードイベントを自分のマシンのイベントに変換する Mouse オブジェクトと Key オブジェクトが動作している。ここで重要なことは、Machine-A の Input オブジェクトが流す情報は、べつに Machine-B の Mouse オブジェクトや Key オブジェクトのためだけのものではないことである。先に示した研究者 C の入室における Awareness の例と同じく、Input オブジェクトの流す情報はそのまま Awareness 情報としても取り扱うことができる。したがってたとえば、Field に流れる Input オブジェクトの情報を聞くことができる Clip オブジェクトを新規にこの Field に導入すれば、それまでは {Input Object, Mouse Object, Key Object} のセットでマウスキーボード共有としてユーザーに見えていたアプリケーションとは別に、{Input Object, Clip Object} のセットでクリップボード共有アプリケーションが新たに生成されたことになる。Fig. 3 はこのことを模式的に表している。Fig. 3 では、Object 1

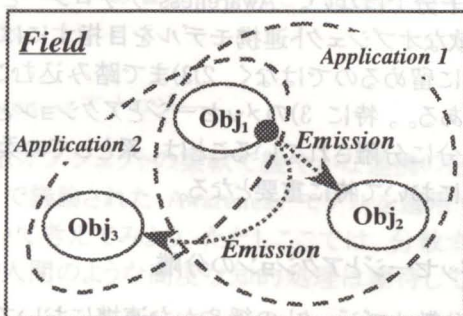


Fig. 3 Awareness を用いた Application の追加

と Object2 の組み合わせで作られていた Application 1 が存在する Field に、Application 1 とは独立な Object3 が参加することによって、ユーザーにあたかも Application 2 が生成されたように見えることを示している。Fig.2 の例でいえば Input オブジェクトが Object1 に相当し、Mouse オブジェクトと Key オブジェクトが Object2 に、Clip オブジェクトが Object3 に相当する。

Fig. 4 は、Awareness アナロジーにおけるメッセージとアクションの分離を模式的に示したものである。Awareness アナロジーにおける参加がこの模式図で表現される。

Fig. 4 において二重の□で囲まれた文字 A, B, C, D はそれぞれ Awareness メッセージを送受信する主体であるオブジェクトを示し、□で囲まれない文字 M は Awareness メッセージを示す。また図中の矢印は、起点がオブジェクトのときメッセージの送出を、終点がオブジェクトのときメッセージの受信を示している。またそれぞれのオブジェクトはその内部にメッセージに対応するアクションをそれぞれ独自に持っているとする。

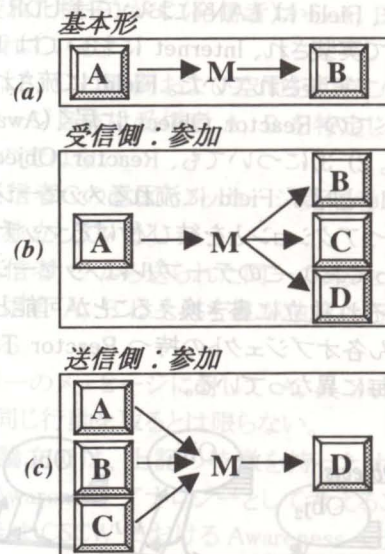


Fig. 4 Message-Action の分離による参加

Fig. 4(a)はメッセージとアクションの分離のもっとも基本的な構成を示している。Fig. 4(a)において、オブジェクト A とオブジェクト B の連携は、メッセージ M を介して行われる。Awareness アナロジーにおいて A は、B についても B の持つ内部アクションについても基本的に関知せずにメッセージを送出する。A と B とは有意な連携を取り得るが、それは A によるものではなく、M と B とが結び付けられている帰結といえる。

Fig. 4(b)では、オブジェクト A の送信したメッセー

ジ M に対し、オブジェクト B, C, D がそれぞれ独自に反応している様子を示している。もちろん B, C, D におけるアクションはそれぞれ別であってかまわない。このときオブジェクト B にとって、他のオブジェクト C, D が存在しても存在しなくても影響はない。したがって Fig. 4(b)におけるオブジェクトの追加・削除は Fig. 3 においては、新規オブジェクトの参加や離脱による Application の生成・消滅に他ならない。あるいは受信側参加といえる。

Fig. 4(c)は Fig. 4(b)の反転形である。オブジェクト A, B, C が同一のメッセージ M を送出し、オブジェクト D と M とが結び付けられている。この場合、D にとって M が A, B, C いずれかからのメッセージであろうと受け取って処理することができる。これはオブジェクトの連携において送側での参加を許容しているということになる。

Fig. 4 に示した Awareness アナロジーに基づく参加では、1)のブロードキャストという枠組みに加え、メッセージが、送信者・受信者のそれぞれおよびその内部処理と分離されているが重要である。このことによって Fig. 4 に示した参加や介入はより広がりが増すことになる。このことを以下に示そう。

Fig. 5 は、ひとつのオブジェクトに注目したときのオブジェクトの内部におけるメッセージとアクションの分離を模式的に示したものである。外部からの介入を許すインターフェイスを設けて内部動作の定義をランタイムに変更できる能力は Reflection として研究が行われている[20,21]が、ここではオブジェクトの内部でメッセージとアクションとが分離され、その結合や付け替えが可能となることによって Fig. 4 における参加に介入としての広がりをもたらされることについて述べよ

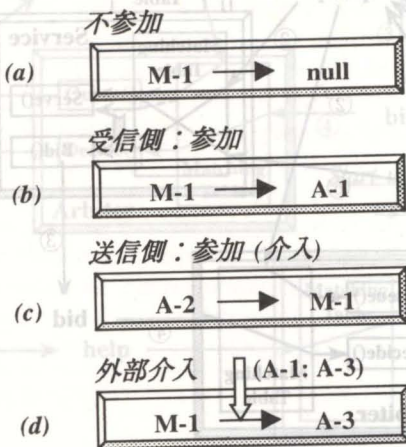


Fig. 5 内部連鎖の変更

う。

Fig. 5(a)は、オブジェクトの内部でメッセージ M-1 が特定のアクションと結合されていない状態を示している。この場合、オブジェクトが M-1 を受信してもなんの反応も示さない。したがってこのオブジェクトは M-1 を仲介とした連携を他のオブジェクトとの間で持っていないことになる。

Fig. 5(b)では、M-1 がアクション A-1 に結合されている。その結果、(a)の状態では M-1 を受信してもなんのアクションも起こさなかったオブジェクトが A-1 を起こすようになる。もし M-1 を受信するオブジェクトが他にもあれば、これは Fig. 4(b)に示した参加に他ならない。

Fig. 5(c)は、(b)の反転形であり、メッセージ送信側に相当する。この場合、もし他に M-1 を送出手のオブジェクトが存在すれば Fig. 4(c)で示した介入を行ったことになる。

さらに Fig. 5(d)は、外部からオブジェクト内部への介入によるアクションの付け替えを示している。このようなメッセージとアクションの結合に対する外部からの介入を認めると Fig. 4 で示した参加、介入にオブジェクト間の相互作用の要素を加わり、Message-Action の連鎖へと発展させることが可能となる。

Fig. 6 は、外部からの介入による Message-Action の挿入と連結を図式化したものである。

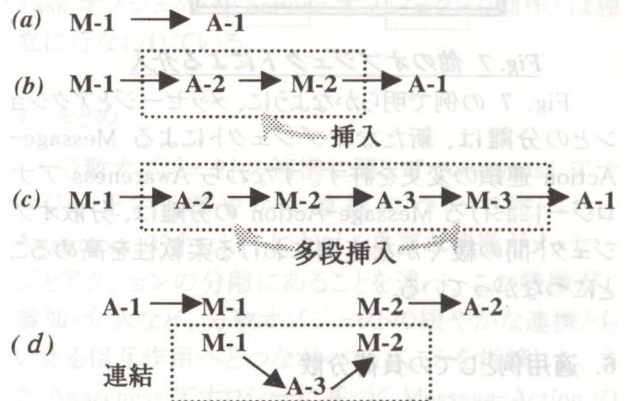


Fig. 6 Message-Action の挿入と連結

Fig. 6(a)は、メッセージとそれを受信するオブジェクトのアクション部分を切り出して表示したものである。メッセージとオブジェクトの分離が可能であることは、Fig. 6(b)に示すようなメッセージ・アクションの挿入を可能とする。Fig. 5(d)に示したように、このような挿入は外部からの介入としてオブジェクトの内部の連鎖に対する付け替えとして行うことも可能である。また、Fig. 6(c)に示すように挿入を多段にすることも可能である。さら

に、Fig. 6(d)に示すように、完全に独立した二つのメッセージ・アクションのセットに対して、別のメッセージアクションセットを用意するだけで、これらを連結させることも可能である。

Fig. 7 は、Fig. 6(c)に示した多段挿入による介入の過程を他のオブジェクトによる外部からの介入として模式的に表したものである。Fig. 7 の例の介入は、オブジェクトAの内部で1) メッセージ・アクションの切断、2) 付け替え、3) 新たなメッセージ・アクションの結合、という三段階を経て行なわれている。この介入によって Object A のみでは存在しなかった機能が、Object B との相互作用を通して追加されたことになる。

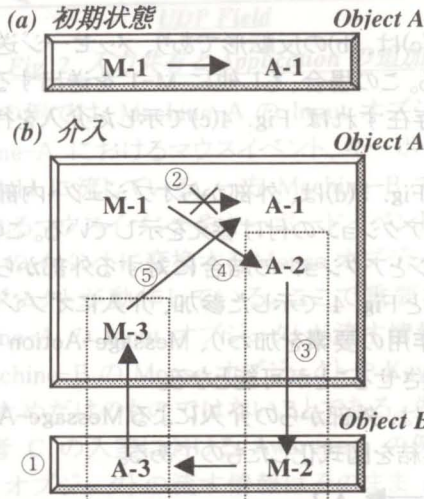


Fig. 7 他のオブジェクトによる介入

Fig. 7 の例で明らかなように、メッセージとアクションとの分離は、新たなオブジェクトによる Message-Action 連鎖の変更を許す。すなわち Awareness アナロジーにおける Message-Action の分離は、分散オブジェクト間の緩やかな連携における柔軟性を高めることにつながっている。

6. 適用例としての負荷分散

ここまで示したように、Awareness アナロジーに基づく分散オブジェクトの緩やかな連携において、メッセージとアクションとの分離はオブジェクト間の相互作用として参加と介入という二つの重要な要素を付け加える。このことを用いて、複数のオブジェクトが参加する系自体の振る舞いを動的に変化させ、負荷分散機能を持たない系(Stage I)が負荷分散機能を持つ系(Stage II-1, II-2)へと変化する例を示そう。

Fig. 8 は負荷分散機能を持たない系(Stage I)を示

している。系は Field-Reactor Model に基づいて動作している。Fig. 8 は、Task オブジェクト群が Field に query メッセージを流し、これを受けていた Service オブジェクトがアクション Serve()を実行する様子を示している。この系が Fig. 4 に示した送信側参加および受信側参加を許す系であることにも注目してほしい。query メッセージ自体は Awareness を媒介するためのメッセージであり、これを受けて実行されるアクションは必ずしも Serve()である必要はなく、また Service オブジェクト毎に実行するアクションが異なってもよい。

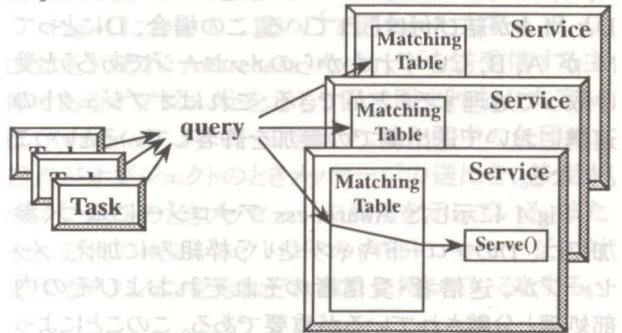


Fig. 8 参加による Action の実行

Fig.8 の系では、各 Service はそれぞれ独立に動作しており、お互いが協調して系全体で負荷分散を行なうような機能は有してはいない。しかしここに新たに Arbiter オブジェクトを加わることによって Fig.9 に示す契約ネットモデル[3]を用いた負荷分散機能を持つ系(Stage II-1)へと変化させることができる。

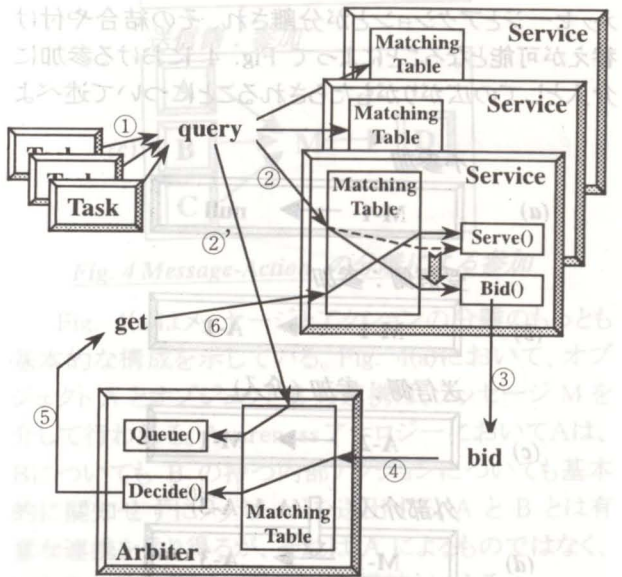


Fig.9 Arbiter の介入によって生じた負荷分散系

Arbiter オブジェクトは Fig.8 の系への参加時に Fig.7(2)における Object B と同様の介入を Service オブジェクトに対して実施する。すなわち、まず Service オブジェクトの Matching Table を書き換えることにより、Fig.8 で結び付けられていた query メッセージと Service()アクションの関係を切断する。次に Service オブジェクトに Bid()アクションを追加し、query メッセージに対して Bid()が反応するように Matching Table を書き換える。また従来の Service()アクションに対しては Field に自らを示す id が付加された get メッセージが流れたときパターン発火が起こるように Matching Table を書き換えておく。一連のこの処理によって、メッセージ bid, get を介した Arbiter オブジェクトにおける処理の挿入が可能となっている。

実際の負荷分散は Bid()アクションが Field に、それぞれの Service オブジェクトの id とその Service オブジェクトのその時点での処理の重みを付加した bid メッセージを流し、これを受けた Arbiter オブジェクトが Field に、負荷のもっとも軽い Service オブジェクトの id を付加して get メッセージを流し処理を依頼することによって行なわれる。

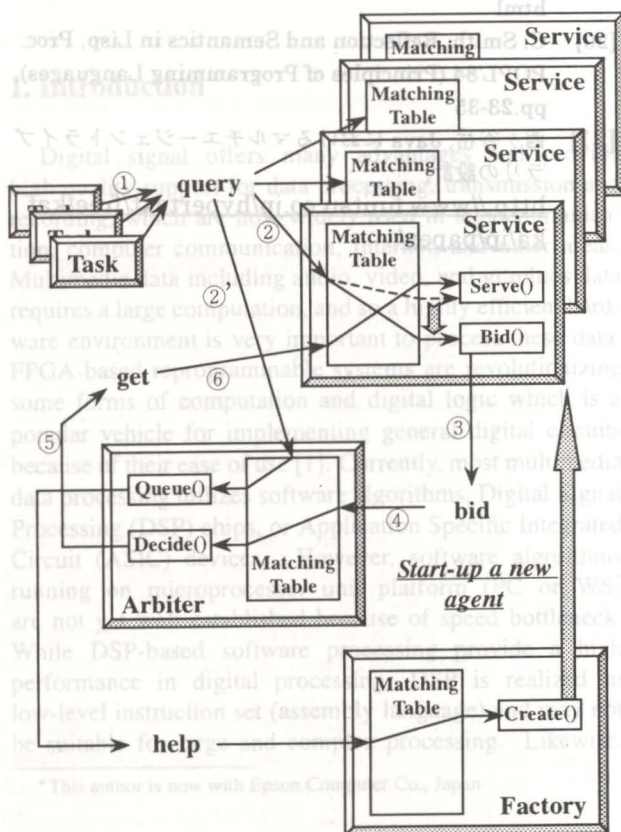


Fig.10 Factory による Service の増加

Fig.9 の例では、query メッセージを出す Task オブジェクト群に変化は加えられていない。また元々は Service オブジェクトをターゲットに投げられた query メッセージが、Fig.9 中の②として Arbiter オブジェクトにも利用されている。これらは、本論文における Awareness アナロジーの性質によるものといえる。さらに Fig.9 は Message-Action の分離がもたらす介入が系全体の有機的な動きに変化をもたらすこと(この例では Fig.8 の負荷分散を持たない系から Fig.9 の負荷分散を持つ系への変化)を示している。

Fig.10(Stage II-2)ではさらに、Factory オブジェクトが Arbiter オブジェクトの Query()の機能を交換している。新たな Query()は query メッセージと bid メッセージとの個数のバランスをモニターし、バランスが大きく崩れたときに help メッセージを出すようになっている。このことによって Service オブジェクト群が処理しきれない数の query が投げられたとき、help メッセージを受けて Factory オブジェクトは新たに Service オブジェクトを Field に投入する。

Fig.9 で示した Stage II-1 の系が、有限の Service オブジェクト群の中での負荷分散であったのに対して、Fig.10 で示した Stage II-2 の負荷分散は Service オブジェクトの個数を増減させることまで含めた負荷分散の系になっている。また Fig.9 における Task オブジェクトへの影響と同様、Fig.10 における機能拡張は、Task オブジェクトや Service オブジェクトの動作とは独立に行なわれている。

7. まとめ

分散オブジェクトの連携に関して Awareness アナロジーという緩やかな連携について述べた。Awareness アナロジーにおける重要な特徴がメッセージとアクションの分離にあることを述べ、この特徴が、参加・介入など、分散オブジェクトの緩やかな連携ともいえる相互作用へとつながっていくことを指摘した。また Awareness アナロジーに基づく Message-Action の分離を使い、負荷分散の機能を持たない系が契約ネットによる負荷分散を持つ系へと機能拡張する例を示した。

Awareness アナロジーに基づく緩やかな分散オブジェクト連携モデルは、メッセージ送信者とメッセージ受信者との間の関係を、メッセージを仲介する媒体によって切断したモデル、と言い換えることができ、その切断が連携における疎な関係を生み出している。また従来送信者主体で決定されていた系の動作を、受信

者主体で決定する枠組みへと変えることによって、これまでの分散システムモデルとは異なる自由度が得られている。全体系の記述に関する方法論をより明確に確立することは今後の課題のひとつといえよう。

いずれにせよ、このようなモデルは、従来の分散システムを補完し、より柔軟で動的なシステムを構築するための手法のひとつとなるものと、我々は考えている。

謝辞

本研究を進めるにあたり、有益な様々な議論をしていただいた富士通研究所(パーソナルシステム研究所のみなさんに感謝いたします。特に小幡・奥山両氏には Awareness に関する有益な議論をしていただいたことをここに感謝いたします。

Reference

[1] OMG, OMG Formal Documentation, <http://www.omg.org/corba>

[2] Microsoft, DCOM Architecture: White Paper, <http://www.microsoft.com/com/comPapers.asp>

[3] R. G. Smith, The Contract Net Protocol: High-Level Communication and Control in a Distributed Problem Solver, IEEE Transaction on Computers, Vol. C-29, No.12(1980), pp1104-1113

[4] W. W. Gaver, Sound Support for Collaboration, Proc. ECSCW91,

[5] 小幡明彦, 佐々木和雄, 佐藤義治, 上野英雄, コミュニケーション行動モデルに基づく偶発的会話支援, 情処研報 1996 GW19-1, pp.1-6

[6] Obata, K. Sasaki, OfficeWalker: A Virtual Visiting System Based on Proxemics, to be published in Proc. CSCW98,

[7] 奥山敏, 竹林知善, 福山訓行, 岡田純代, 松本 安英, 真鍋愛, オフィスコミュニケーションのための Awareness 支援サービスの一検討, 信学技報 CQ97-13, pp.29-36

[8] C. Greenhalgh, S. Benford, MASSIVE: A Collaborative Virtual Environment for Teleconferencing, ACM Transaction on CHI, Vol.2, No.3(1995), pp239-261

[9] T. Rodden, Populating the Application: A Model of Awareness for Cooperative Applications, Proc. CSCW'96, pp.87-96

[10] 岩尾忠重, 岡田誠, 竹林知善, 分散環境下における自律型オブジェクトによるアプリケーション構築モデル, 第 55 回情処全国大会, 1996, 3T-1, pp659-660

[11] 岩尾忠重, 岡田誠, 竹林知善, 緩やかなオブジェク

ト連携モデル: Field-Reactor Model, 情処研報1998 DPS86-3, pp13-18

[12] N. Carriero, D. Gelernter, Linda in Context, Communication of the ACM, Vol32, No.4(1989), pp.444-458

[13] 立山義祐, (修士論文)タプル空間通信を使ったユーザインタフェースの構築法, <http://www.sanpo.t.u-tokyo.ac.jp/people/tateyama.html>

[14] SUN, Java Spaces Specification, <http://java.sun.com/products/javaspaces/specs/index.html>

[15] T. J. Lehman, S. McLaughry: T Spaces: The Next Wave, <http://www.almaden.ibm.com/cs/TSpaces/>

[16] J. F. Patterson, M. Day, J. Kucan, Notification Servers for Synchronous Groupware, Proc. CSCW'96, pp.122-129

[17] TIBCO, TIB/Rendezvous: White Paper, <http://www.rv.tibco.com/rvwhitepaper.html>

[18] H. Maruyama, N. Uramoto, Infobus Repeater: A Java-based Publish/Subscribe Middleware, IBM research Report as RT0245, <http://www.trl.ibm.co.jp/projects/ibr/paper.htm>

[19] IONA, Orbix Talk White Paper, <http://www-usa.iona.com/support/whitepapers/orbixtalk/index.html>

[20] C. Smith, Reflection and Semantics in Lisp, Proc. POPL'84 (Principles of Programming Languages), pp.23-35

[21] 西ヶ谷岳, Java におけるマルチエージェントライブラリ設計, <http://www.fujitsu.co.jp/hypertext/free/kafka/jp/paper/>

