

オブジェクト指向分散プログラムのトレース

久保田 稔*

金 東輝**

*NTT交換システム研究所 **早稲田大学理工学部

オブジェクト指向に基づく分散プログラムのトレースを自動的に行う方式について提案する。これはオブジェクトとメッセージの両者にトレース状態を制御するデータを設定し、また同じアプリケーションに属するメッセージのトレース状態を伝播させることにより実現する。本方式は、複数の処理が同時に行われる分散システムにおいて、特定の処理のみのトレースを行うことを可能にするため、分散プログラムの機能確認、デバッグに有効である。本方式は通信網ワイド分散処理プラットフォームPLATINAのプロトタイプシステムの一部として試作し、その効果を確認した。

Object-Oriented Distributed Program Trace

Minoru KUBOTA*

Dongwhee KIM**

*NTT Communication Switching Laboratories

**Department of Electronics and Communication Engineering, Waseda University

In this paper, we propose a method for automatically tracing distributed object-oriented programs, by adding trace control data to both objects and messages, and propagating a trace status of a message to the next message belonging to the same application. This allows us to trace a specific application in a distributed system where multiple applications are executing simultaneously. It is thus useful for testing and debugging distributed programs. This method has been implemented in the prototype of the PLATINA, a network-wide distributed processing platform.

1 はじめに

ネットワークの大規模化，高速化に伴い分散処理がより一般的な技術となりつつある．通信分野においても分散処理を活用してネットワークワイドの高度なサービスを提供する方式が導入されている[1]．

このような背景から，我々は並行オブジェクト指向モデルに基づく通信網ワイド分散処理プラットフォームPLATINA[2]の検討を進めている．この上で動作するアプリケーションは並行動作するオブジェクトで実現され，それらが互いにメッセージを受送することにより処理を進める．

オブジェクトには，自律的に並行動作する能動オブジェクトと，能動オブジェクトの実行環境下で動作する受動オブジェクトの2種類があるが，本稿で単にオブジェクトという場合は能動オブジェクトを意味するものとする．オブジェクトは，従来のOSの並列実行単位であるプロセスあるいはタスクに相当する．

PLATINAはオブジェクトの実行を制御する実時間カーネル（以後，単にカーネルと呼ぶ）と，アプリケーションに共通な拡張サービスを提供するサーバからなる（図1）．カーネルはオブジェクトの実行とオブジェクト間のメッセージ通信を制御する．カーネルが提供する基本機能をカーネルプリミティブと呼び，代表的なものに，メッセージ送信，メッセージ受信等がある．

各オブジェクトはネットワーク内で一意に定まるオブジェクト識別子を持ち，これを指定することにより，オブジェクトの存在するノードにかかわらずメッセージを送ることができる．メッセージにはオブジェクト間でやりとりされるデータの他に送信元と受信側のオブジェクトの識別子等の制御データを含む．

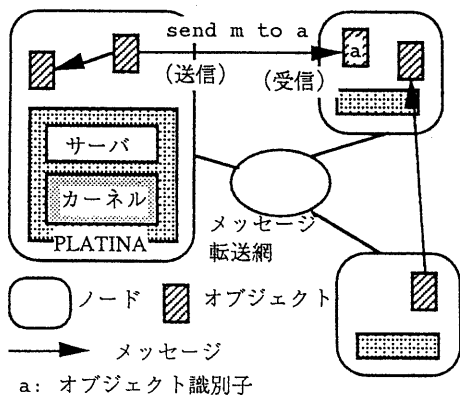


図1：PLATINA概要

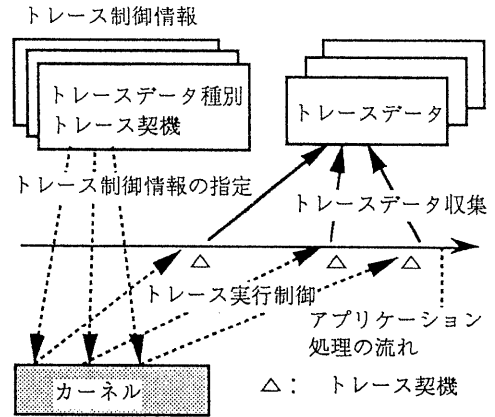


図2：オブジェクト実行のトレース

分散した多数のオブジェクトがメッセージを送受しながら協調的に処理を進めることで，分散プログラムが実現できる．このような分散プログラムの検証やデバッグを行うには，プログラムの構成要素であるオブジェクトの動作をトレースすることが有効である．各オブジェクトは単体として試験が完了していると仮定すると，どのようなメッセージがどのオブジェクトの間で送受されたかを調べるのが重要である．本稿では，分散オブジェクト間でのメッセージ通信のトレースを自動的に行う方式について提案する．

2. 分散プログラムのトレース

2.1 オブジェクト実行のトレース

試験担当者等によりあらかじめ指定されたプログラム（命令のアドレス）をオブジェクトが実行した際に，その時点であらかじめ指定されたデータを収集することをオブジェクト実行のトレースと呼ぶ（図2）．トレースを行う時点をトレース契機，トレースにより収集されるデータをトレースデータ，トレース契機でどのようなデータを収集するかを指定する情報をトレースデータ種別，と呼ぶことにする．またトレース契機とそれに対応するトレースデータ種別の集合をトレース制御情報と呼ぶ．トレースを行うオブジェクト（処理）をトレース対象オブジェクト（処理）と呼ぶ．

トレース契機としてはプログラム中の任意の時点を指定することが可能であるが，ここでは，オブジェクトがメッセージを受信した時点をトレース契機とする．オブジェクトの単体の試験が完了していれば，メッセージ受信時のみトレースデータを収集しても，分散プログラムの処理の流れが把握でき，有効な情報が得られる．

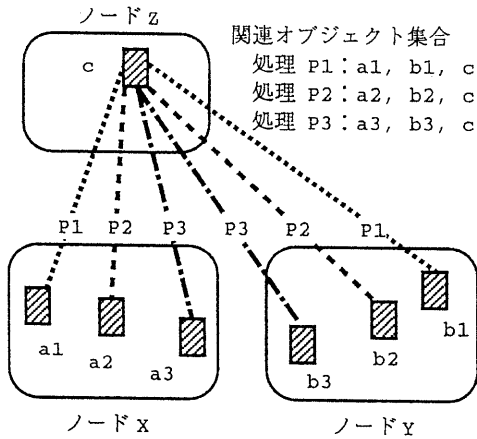


図3：分散オブジェクトのトレース

トレース制御情報は試験担当者等によりカーネルに通知される。カーネルは、オブジェクトがメッセージを受信する毎に、トレースデータを収集する。収集するトレースデータには、通信種別、送信/受信オブジェクトの識別子、メッセージの種別/内容等がある。

2. 2 分散プログラムのトレース

逐次的なプログラムにおいてはプログラムのトレースは一般的な技術である。しかし多数の分散したオブジェクト間でメッセージをやりとりする分散プログラムのトレースを行う場合には後述するような問題があり、分散環境に対応したトレース技術が必要となる。

まず分散プログラムにおけるトレースを説明す

オブジェクトopで収集されるトレースデータをTopと表わす。

c1, c2, c3はそれぞれ処理p1, p2, p3を実行中のオブジェクトcを示す。

(a) 全トレース

Ta1 Ta2 Tb1 Tc1 Tb2 Ta3 Ta1 Tb3 Tb2 Tc2 Tc3 Ta2 Tb2 Tc1 Ta1

(b) ノードxがトレース対象

Ta1 Ta2 Tb1 Tc1 Tb2 Ta3 Ta1 Tb3 Tb2 Tc2 Tc3 Ta2 Tb2 Tc1 Ta1
Tb1, Tc1を収集することができない。Ta2とTa3が不要な情報である。

(c) タスクa1がトレース対象

Ta1 Ta2 Tb1 Tc1 Tb2 Ta3 Ta1 Tb3 Tb2 Tc2 Tc3 Ta2 Tb2 Tc1 Ta1
Tb1, Tc1を収集することができない。

(d) 処理P1の関連オブジェクト集合タスクa1, b1, cがトレース対象

Ta1 Ta2 Tb1 Tc1 Tb2 Ta3 Ta1 Tb3 Tb2 Tc2 Tc3 Ta2 Tb2 Tc1 Ta1
Tc2とTc3が不要な情報である。

(e) メッセージトレース

Ta1 Ta2 Tb1 Tc1 Tb2 Ta3 Ta1 Tb3 Tb2 Tc2 Tc3 Ta2 Tb2 Tc1 Ta1

下線部が実際にカーネルが出力するトレースデータを示す。

図4：トレース対象とトレースデータの収集

る。図3に示すように3つのノード, x, y, zからなる分散システムにおいて3つのアプリケーションの処理P1, P2, P3が並行して処理されているものとする。アプリケーションは複数のノードに分散するオブジェクトによって実現される。

すべてのノードにおいてすべての処理のトレースを行うことは、P1, P2, P3に関するトレースデータが混在して収集され、トレースデータが膨大になり現実的でない(図4(a))。特定のノード(ここではXとする)に着目して、そのノードのすべてのプログラムの実行をトレースすると、他のノードのプログラムのトレースデータが収集できず、またP1, P2, P3に関するトレースデータが混在して出力される(図4(b))。

ある特定の処理の機能確認やデバッグを行う場合、不要なトレースデータもあわせて収集されると、多重度が高い場合に、必要なトレースデータのみを取り出すことが困難になる。したがってトレースを行う特定の処理(トレース対象処理)が実行された場合にのみトレースデータを収集する方式が有効である。

2. 2. 1 オブジェクトに着目したトレース

トレース対象処理のみに着目したプログラムのトレース方法として、オブジェクト(タスク)毎にトレースを行う方法がある[3]。これをオブジェクト(タスク)トレースと呼ぶ。たとえば図3においてオブジェクトa1をトレース対象とすると、a1が実行した処理に関するトレースデータのみが出力される。しかし処理p1に関して、オブジェクトb1, cはトレース対象となっていないので、これらに関する

トレースデータは収集されない(図4(c)).

2. 2. 2 関連オブジェクト集合

トレース対象処理が複数のオブジェクトにまたがって行われる場合、これらのオブジェクトの集合を関連オブジェクト集合と呼ぶ。たとえば処理P1の関連オブジェクト集合は、a1, b1, cである。

オブジェクトトレースを用いてトレース対象処理をすべてトレースする場合、関連オブジェクト集合に属するすべてのオブジェクトをトレース対象としなければならない。複数のノードからなる分散システムにおいては、オブジェクト数が多くまたそれらが複数のノードに分散していることから、関連オブジェクト集合を決定すること、及びこれらをトレース対象として手動で指定することは、困難である。

2. 2. 3 サーバオブジェクトのトレース

オブジェクトトレースでは、関連オブジェクト集合をすべてトレース対象とすることができても以下のような問題がある。すなわち複数のオブジェクトからの処理要求を受けるオブジェクト(サーバオブジェクトと呼ぶ)では、トレース対象処理以外の処理のトレースデータも収集されてしまう。たとえば図3においてトレース対象処理がP1でも、オブジェクトcは、P1, P2, P3すべての処理を行うため、オブジェクトcの実行に伴って、P2, P3に関するトレースデータも収集される(図4(d))。

サーバオブジェクトにおいてトレース対象処理のみトレースすることは、実行中の処理がトレース対象かどうかの判定処理が必要となるが、オブジェクトトレースだけでこれを行うことはできない。アプリケーションにこれらの判定処理を組み込めば可能となるが、運用時には不要となる試験やデバッグのための処理をアプリケーションに記述することは望ましくない。

2. 2. 4 メッセージトレース

前述のように、オブジェクトトレースは、複数ノードからなる分散システムのように多数のオブジェクトにより複数のアプリケーションを並行に実行する環境でのプログラムのトレースには十分な機能をもたない。したがってこのような環境におけるプログラムトレースでは、図4(e)に示すように、トレース対象処理に関するトレースデータのみが収集できる方法でなければならない。

本稿では以下の要求条件を満たすトレース方法(これをメッセージトレースと呼ぶ)について提案する。

(1) 一つ以上のノードからなるシステムにおいて、

複数の処理に関し、それぞれの処理を行うために実行される複数のオブジェクトの実行過程の内、特定の処理(トレース対象処理)に関してのみトレースを行う。

- (2) トレース対象処理の関連オブジェクト集合に属するオブジェクトが自動的にトレース対象となり、複数の処理に係わる(複数の関連オブジェクト集合に属する)オブジェクトにおいては、トレース対象処理に係わる場合にのみトレースを行う。
- (3) (1), (2)の実現に際して、アプリケーションを変更しなくても、カーネルだけで対処可能とする。

3. 分散オブジェクト間メッセージ通信のトレース

3. 1 メッセージフロー

ある処理を実現するために実行されるオブジェクトとそれらが送受するメッセージの集合を、その処理に関するメッセージフローと呼ぶ(図5)。メッセージフローの中のオブジェクトの集合が前述の関連オブジェクト集合となる。

メッセージフローの中のメッセージの並びをメッセージシーケンス(MsgSeq)、オブジェクトの並びをオブジェクトシーケンス(ObjSeq)と呼ぶ。ObjSeqの最初のオブジェクトをメッセージフロー起動オブジェクトと呼ぶ。またトレース対象処理に対応するメッセージフローの起動オブジェクトを特にトレース開始オブジェクトと呼ぶ。

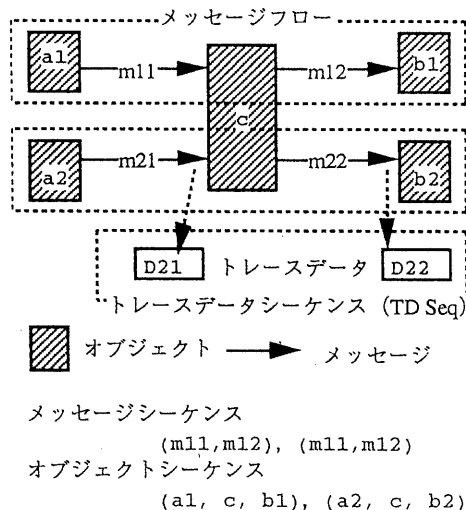


図5: メッセージフローと
トレースデータシーケンス

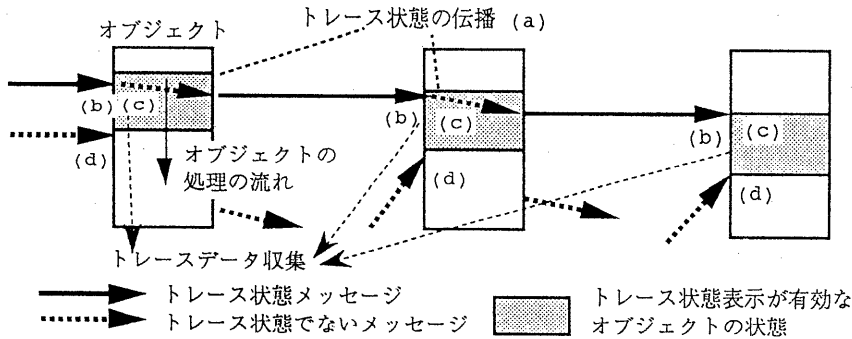


図6：トレース状態指示の伝播

3. 2 オブジェクトのトレース制御

2. 2. 4の要求条件を満たすトレースを行うため、カーネルのオブジェクトの実行制御に以下の機能を追加する。まずトレースを行うかどうかを示すデータ（トレース状態表示と呼ぶ）をオブジェクトに付加する。これは各オブジェクトの実行制御用データに格納される。

トレース状態表示が有効なオブジェクトがトレース対象オブジェクトとなり、このオブジェクトのトレースデータのみを収集することとする。トレース対象処理の関連オブジェクト集合に属するオブジェクトのトレース状態表示を有効にすることで、トレース対象処理のトレースのみを行うことが可能になる。

3. 3 トレース状態の伝播

前述のように、トレース対象処理の関連オブジェクト集合を事前に求め、これに属するオブジェクトのトレース状態表示を有効にしておくことは分散システムにおいては困難である。トレース対象の関連オブジェクト集合のオブジェクトを自動的にトレース対象とするため、次の方法を用いる。

オブジェクトだけでなくメッセージにもトレース状態表示を付加し、同一メッセージフローに属するオブジェクトとメッセージの間でトレース状態表示を引き継いでいく機構を提供する（図6(a)）。これをトレース状態の伝播と呼ぶ。メッセージのトレース状態表示はメッセージの制御データエリアに格納される。

トレース状態表示が有効なメッセージをトレース対象メッセージと呼ぶ。オブジェクトが受信したメッセージがトレース対象メッセージであれば、カーネルはトレースデータを収集する（図6(b)）。

トレース状態の伝播を実現するため、カーネルに以下の機能を追加する。

- (1) トレース対象オブジェクトから送られるメッセージを自動的にトレース対象とする（メッセージのトレース状態を有効にする）。
- (2) オブジェクトが受信したメッセージがトレース対象かどうかを調べ、トレース状態ならばオブジェクトのトレース状態表示を一時的に有効にする（図6(c)）。トレース対象でなければオブジェクトのトレース状態表示は変化しない。別のメッセージフローに属する次のメッセージがくるまで（図6(d)）、このオブジェクトから送られるメッセージのトレース状態表示は有効となり、トレース状態表示が伝播される。オブジェクトが次のメッセージを受信すると、トレース対象メッセージ受信前の状態にオブジェクトのトレース状態表示を戻し、(2)の処理を繰り返す。

以上によりオブジェクトが受信したメッセージのトレース状態表示が同一メッセージフローに含まれる次のメッセージに引き継がれる。試験担当者等がメッセージフロー起動オブジェクトのトレース状態表示を有効にすることで、同一メッセージフロー中のすべてのオブジェクト、メッセージのトレース状態表示がカーネルにより自動的に有効となる。これによりトレース対象の関連オブジェクト集合に属するオブジェクトを個別にトレース対象としなくても、これらの中でやりとりされるメッセージが自動的にトレースされる。

3. 4 トレースデータの収集

あるトレース対象処理の実行をトレースすることにより生成されるトレースデータの並びをトレースデータシーケンス（TD Seq）と呼ぶ（図5）。一つのトレース対象処理に一つのTD Seqが対応する。図4におけるTD Seqは次のようになる。

処理P1: Ta1 Tb1 Tc1 Ta1 Tc1 Ta1

処理P2: Ta2 Tb2 Tb2 Tc2 Ta2 Tb2

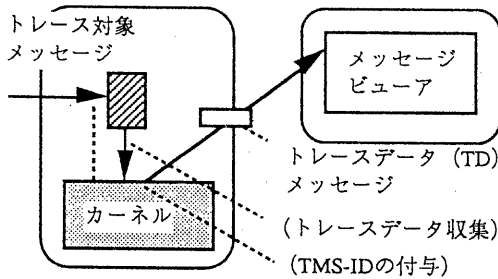


図7：トレースデータの収集

処理P3: Ta3 Tb3 Tc3

メッセージトレースを用いて、プログラムの機能確認、試験を行う場合、トレース対象のメッセージフローの内容を表示する必要がある。これを表示するプログラムをメッセージビューアと呼ぶことにする。メッセージビューアは、複数のノードから送られるトレースデータを分析して、トレース対象のメッセージフローの内容を再現する。

図7に示すように、トレース対象メッセージが受信されると、カーネルがトレースデータを収集し、メッセージビューアにメッセージとして送る。このメッセージをトレースデータメッセージ (TDメッセージ) と呼ぶ。TDメッセージには、トレース対象メッセージの内容と、TD Seqを識別するための情報が含まれる。

3. 5 メッセージフローの識別

3. 5. 1 トレースシーケンス番号

メッセージビューアは保守・運用を目的とする特定のノードに配置され、トレース対象処理が行われる複数のノードのカーネルからTDメッセージがメッセージビューアに送られる。メッセージトレースを行う分散システムでは、図8に示すように、トレースデータの生成順序と、これらがメッセージビューアに到着する順序が一致する保証はない。したがって、メッセージビューアでは、一つのトレース対象処理に関するトレースデータをその発生順序に対応して並び替える必要がある。

発生順序に対応してトレースデータを並び替えるために、各トレース対象メッセージに含まれる制御データにトレースデータシーケンス番号 (TD Seq番号) を付与する (図8)。TD Seq番号は、TDメッセージに含めてメッセージビューアに送られる。トレース開始オブジェクトが最初に出すメッセージのTD Seq番号を1とする。

トレース状態伝播機構により、トレース状態が次

に引き継がれるときに、メッセージのTD Seq番号は1増やされる。メッセージビューアでは、到着したトレースデータメッセージをTD Seq番号の順番で並び替えることにより、発生順に並んだTD Seqを復元できる。

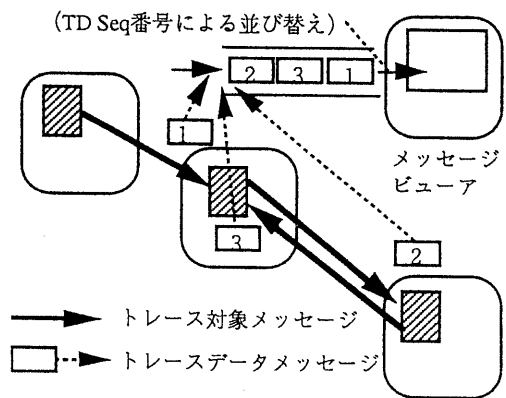
3. 5. 2 メッセージシーケンス識別情報

トレース対象処理が一つだけの時、すなわちTD Seqが一つだけの時、メッセージビューアは、TDメッセージを上記TD Seq番号順に並び変えて表示すればよい。トレース対象処理が同時に複数存在するとき、メッセージビューアには異なるトレース対象処理に関連するTDメッセージが混在して送られてくる。このためメッセージビューアは、これらのTDメッセージをトレース対象処理毎に分類する必要がある。たとえば図4において、処理P1とP2をトレース対象とした場合、TD Seq (Ta1 Ta2 Tb1 Tc1 Tb2 Ta1 Tb2 Tc2 Ta2 Tb2 Tc1 Ta1) から (Ta1 Tb1 Tc1 Ta1 Tc1 Ta1) と (Ta2 Tb2 Tb2 Tc2 Ta2 Tb2) を分離する必要がある。

この分類を行うためには、TDメッセージ毎にトレース対象処理を識別するための情報を付加しておく必要がある。この情報をトレースメッセージシーケンス識別情報 (TMS-ID) と呼ぶ。TMS-IDは、TDメッセージに付与されてメッセージビューアに送られる。

TMS-IDは分散システム内で一意に識別できなければならない (重複があってはいけない)。分散システム内で一意にTMS-IDを与える方法としては、以下のような方法が考えられる。

- (a) トレース対象処理対応にTMS-IDを一つだけ与える方法。TMS-IDをトレース対象メッセージに含め、トレース伝播機構を用いて、次のメッセージ



1, 2, 3 : トレースデータシーケンス番号

図8：トレースデータメッセージ

に引き継いでいかなければならない。メッセージ制御情報（ヘッダ）が大きくなる。

- (b)各ノードでTDメッセージ毎に付与し、全TDメッセージ内のTMS-IDを解析することにより、TDメッセージが対応するトレース対象処理を識別する方法。

以下(a),(b)の方式についてさらに述べる。(a)の方式として次の3つが考えられる。

(a-1) 集中管理方式

TMS-IDを与えるプログラム（TMS-IDサーバと呼ぶ）を分散システム内で一つ設ける。メッセージフローの起動オブジェクトのトレース対象になったときに、TMS-IDサーバがこれにTMS-IDを付与する。TMS-IDの一意性は、上記サーバがTMS-IDを一括管理することにより保証される。

TMS-IDサーバは、十分長い間、重複したTMS-IDを与えないことが必要であるが、トレースは常に行われるわけではない、すなわち同時に存在するトレース対象メッセージフローは多くないので、TMS-IDとしてたとえば16ビットの整数でも十分である。本方式の問題は、トレース開始時TMS-IDサーバにアクセスする必要があるため、処理の開始に遅延が生じることである。

(a-2) オブジェクト識別子方式

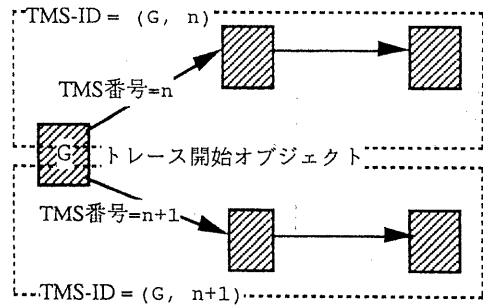
トレース開始オブジェクトのオブジェクト識別子をTMS-IDとする方式である。前述したように、オブジェクト識別子はシステム内での一意性が保証されているので、TMS-IDとしてもその一意性が保証される。トレース開始オブジェクトが、同時に複数のトレース対象処理に属する場合は、TMS-IDとしての一意性が保証されない。この場合は次のように対処する（図9）。

トレース対象オブジェクトにトレースオブジェクトメッセージシーケンス番号（TMS番号）と呼ぶデータを付与する。オブジェクトが生成された時に付与されるTMS番号は0とする。あるオブジェクトが、トレース開始オブジェクトになったとき、該オブジェクトの識別子とTMS番号の組をTMS-IDとする。オブジェクトがトレース開始オブジェクトとなるたびに、TMS番号を増分していく。これにより、TMS番号の一意性を保証する。

(a-3) 乱数方式

TMS-IDとして各ノードで乱数を用いる方法。トレース対象メッセージフローが少なければ、実用上問題ない。

次に各ノードでローカルにTDメッセージ毎にTMS-IDを付与する方法を提案する。これには以下



G: オブジェクト識別子

図9: トレース対象メッセージフローの識別 (1)

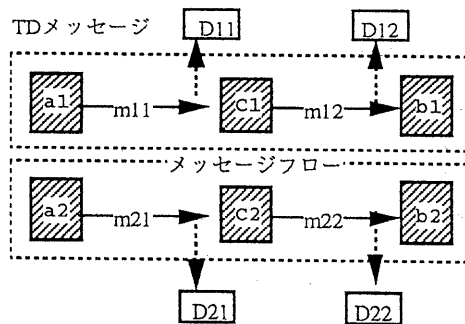
の2つが考えられる。

(b-1) 送受信オブジェクト識別子

前提条件で述べたように、メッセージのヘッダには、送信オブジェクト識別子と受信オブジェクト識別子が含まれる。これはTDメッセージにも含まれている。

トレース対象メッセージフローのとなりあうメッセージをMnとMn+1、トレース対象メッセージMnに対応するトレースデータDnとする。Dnに関し、トレース対象メッセージMnの送信オブジェクト識別子をDn.s、受信オブジェクト識別子Dn.r、TDS番号をDn.tと表す。またDn.srはMnの送信/受信オブジェクト識別子の組を表わすものとする。このとき、Dn.r=Dn+1.sとなる。

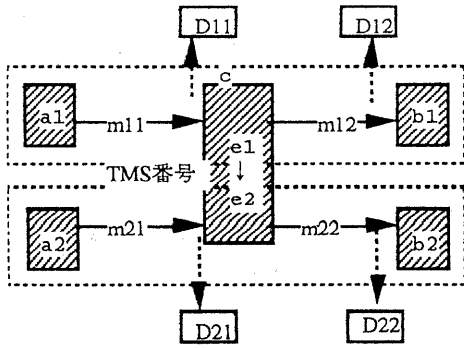
トレース対象メッセージフローに関し、それぞれObjSeqに共通するオブジェクトがなければ、



D11.sr = (a1, c1), D12.sr = (c1, b1)
 D21.sr = (a2, c2), D22.sr = (c2, b2)
 D11.t=t1, D12.t=t1+1
 D21.t=t2, D22.t=t2+1

TD Seq: (D11, D12) (D21, D22)

図10: トレース対象メッセージフローの識別 (2)



オブジェクト識別子.TMS番号=
 オブジェクト拡張識別子
 Dn.SR = (送信オブジェクト拡張識別子,
 受信オブジェクト拡張識別子)
 TMS番号: j1, j2, e1, e2, k1, k2
 (e2=e1+1)
 D11.SR = (a1.j1, c.e1)
 D12.SR = (c.e1, b1.k1)
 D21.SR = (a2.j2, c.e2)
 D22.SR = (c.e2, b2.k2)
 D11.t=t1, D12.t=t1+1
 D21.t=t2, D22.t=t2+1

図11: トレース対象メッセージフローの識別 (3)

図10に示すように、 $Dn.r=Dn+1.s$,
 $Dn.t=(Dn+1.t)+1$ となるように、メッセージを
 並び変えることにより、トレース対象メッセージ
 フロー毎にTDメッセージを分離することができ
 る。

(b-2) 送受信オブジェクト識別子とTMS番号

図11に示すように、複数のトレース対象メッ
 セージフローのObjSeqに共通するオブジェクトがあ
 る場合、(b-1)の方式ではメッセージフロー毎に
 TDメッセージを分離することはできない。たとえ
 ば図10において、c1とc2が同じ場合に(b-1)の方
 式を適用すると、図中のTD Seqの他に(D11, D22),
 (D21, D12)もTD Seqとなり、正しいTD Seqを識別で
 きない。

このため、(a-2)と同様に、各オブジェクト毎に、
 TMS番号を管理させる。(a-2)の場合と異なり、
 TMS番号は、メッセージを受信するたびに、TD
 メッセージにつけ加えられると同時に、増分され
 る。またこの値は、トレース状態メッセージにも付
 与され次のオブジェクトに伝えられる。オブジェ
 クト識別子とTMS番号の組を一つのオブジェクト識
 別子(オブジェクト拡張識別子と呼ぶ)とみなせば、
 (b-1)と同様の方式となる。

4 メッセージトレースプロトタイプ

PLATINAの研究・開発の一貫として、分散処理
 環境を擬似する環境をUnixワークステーション上
 に構築した。本システムは、ノードを1つ以上の
 Unixプロセスで擬似し、オブジェクトをSunOSの
 Light Weight Process機構を用いて擬似している。

Unixプロセスとして実行するアプリケーション
 プログラムには、PLATINAカーネルを擬似する
 プログラムをリンクする。これをPLATEと呼ぶ。
 PLATE内にメッセージトレースの機能を組み込ん
 でいる。またメッセージビューアを擬似環境の一
 部として実現し、XWindowを用いて、グラフィカ
 ルな表示を行っている。

アプリケーションプログラムは我々のグルー
 プで開発した並列オブジェクト指向言語COOL[4]
 を用いて記述されており、網ワイドの通信サー
 ビスのプロトタイプとなっている。このアプリ
 ケーションの試験にメッセージトレースを用い、
 デバッグの効率化に寄与することを確認した。ま
 たTDシーケンスを解析することにより、プログラ
 ムの動的特性(メッセージと統計情報)の解析に
 も役立つ。

5 まとめ

オブジェクトとメッセージの両者にトレース状
 態を制御するデータを設け、またメッセージのト
 レース状態を伝播させることにより、分散プロ
 グラムのトレースを自動的に行う方式について提案
 した。本方式は、複数の処理が同時に行われる分
 散システムにおいて、特定の処理のみのトレース
 を行うことを可能にするため、分散プログラムの
 機能確認、デバッグに有効である。本方式は
 PLATINAプロトタイプシステムとして試作し、そ
 の効果を確認した。

参考文献

[1] 青木利晴, 岡田和比古, 重松直樹: インテリ
 ジェントネットワーク, 信学会誌, Vol.73, No.7,
 pp.737-743 (1990)
 [2] 久保田稔, 丸山勝己, 田中聡: オブジェクト
 指向に基づく通信網ワイド分散処理プラッ
 フォーム: PLATINA, 1992信学秋季全大SB-6-1
 [3] 渡辺次郎, 城正彦, 五十嵐譲, 石橋奈津子,
 松下政好: マルチタスク交換向きCTRON用
 並行タスクデバッガ, 信学技報, Vol. 88, No.352,
 SSE88-161, pp.1-6 (1988)
 [4] 丸山 勝己: 並列オブジェクト指向言語COOL,
 情処学論, Vol. 34, No. 5, pp.963-972 (1993)