

M I A分散トランザクション処理におけるインタオペラビリティ評価法

3 E - 5

菊地 寿 武藤 裕 藤田 敏幸  
 日本電信電話 日本電信電話 コムニック

1. はじめに

M I A (Multivendor Integration Architecture) は、複数ベンダから構成されるコンピュータシステムにおいて、以下の目的を達成するために制定された：(1) アプリケーションプログラムのポータビリティの確保、(2) システムを構成するノード間の相互接続性ならびにインタオペラビリティの確保、(3) 操作性の統一。現在、M I AそのものはNTTならびにNTTデータの調達仕様として規定されているが、その主要部分は、各国主要キャリアの調達仕様として検討されているS P I R I T (Service Providers Integrated Requirements for Information Technology) に引き継がれており、M I Aが当初目標としたポータビリティ、インタオペラビリティ等がどの程度達成されたかを評価することが重要となった。

上記の評価対象のうち、本検討ではM I A分散トランザクション処理におけるインタオペラビリティに関して、評価法を確立し評価を実施する。

2. M I Aが目指すインタオペラビリティ

M I Aは、S T D L言語(Structured Task Definition Language)とR T I (Remote Task Invocation)プロトコルの組合せによって、複数ベンダのマ

Evaluation Method on Interoperability in MIA  
 Distributed Transaction Processing

Hisashi KIKUCHI, Yutaka MUTO

NTT Information and Communication Systems Labs.  
 1-2356 Take, Yokosuka-Shi, Kanagawa 238-03,  
 Japan

Toshiyuki FUJITA

COMNIC Corporation

2-26-9 Hachoubori, Chu-o-Ku, Tokyo 104, Japan

シンのクライアント・サーバ型分散トランザクション処理について、インタオペラビリティを保ちつつポータブルに記述することを目標のひとつとしてきた。ルーチンコールを Call Task という形で、同一ノードに対しても別ノードに対しても同じ記述で記述可能とした点が代表的な特徴である(図1参照)。

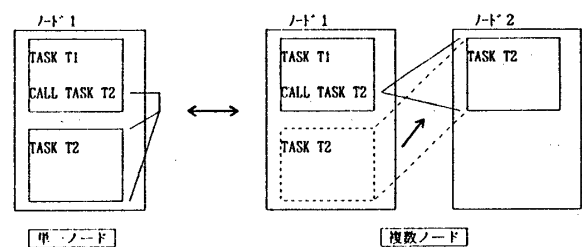


図1. Call Taskによる分散トランザクションの記述

さらに、このバリエーションとして、キューを利用したトランザクション転送を行い、タスクの繰返し起動や時間指定起動を可能にした形の Submit Task がある。

通信部分を司るR T Iプロトコルには2系統があり、各々、(1) OSI TP (Open Systems Interconnection / distributed Transaction Processing)とOSF RPC (Open System Foundation / Remote Procedure Call)データ転送部分の組合せ、(2) TCP/IPとOSF RPCの組合せとなっている。このうち(2)の組合せはOSFで発行しているソースコードを各社がコピーして実装しているためインタオペラビリティ上の問題点の数は少ないと推測し、本検討では、(1)のOSI TP系を評価対象とした。

3. インタオペラビリティ評価法確立の必要性

S T D LとR T Iの組合せは、仕様を先に作成して各社がこの仕様に準拠する実装を行ったため、評価が必須である。M I Aの実装製品が出現し始めたのに伴い、こうした分散トランザクション処理におけるインタオペラビリティ評価法の確立が急務とな

った。ここでいうインタオペラビリティとは、単に接続ができることを指すのではなく、複数ノード間で双方のアプリケーションが矛盾無く協調動作可能であることを意味する。

#### 4. 評価法の確立

マルチノードのインタオペラビリティを評価するに当たっては、大きく次の2つの評価基準がある：  
 (1) 単ノードのタスクコール等で実現していた処理がマルチノードに分散させた後も矛盾無く実行できるか、(2) マルチノード固有の問題を解決しているか。M I Aに関する、それぞれの評価基準に属する評価項目は次の通り。(1)に属するのは、CALL TASK、SUBMIT TASKの基本形、引数伝達、例外伝達、制限値、排他制御等、また、(2)には、通信路管理、宛先管理、等がある。これらの項目の中からMIA-STD Lインタフェースを用いて評価可能な項目を抽出し、テストケースを作成する。このテストケースの中から、重要性や製品動向を踏まえてテストプログラム化を行い、複数ノードで走行させるプログラム自身に搭載された判定ロジックと、各ノードに残されたログデータ等を基に問題点を拾う。以上のような手順を評価法としてまとめた。

#### 5. 評価項目の選択とテストプログラム作成

前節で述べた評価項目のうち、排他制御、通信路管理については、直接的に確認に使えるMIA-STD Lインタフェースが無いため、本検討では対象から外す。なお、宛先管理については、M I Aでは機能レベルの規定しかないので、テストプログラムを走行させていく中で問題点を拾うことにする。

以上の方針から、STD L言語要素を基にしてテストケース92317件を編み出した。テストケースを分類すると図2のようになる。

全テストケース92317件の中から、現在の製品の主流であるアプリケーション・サポート・トランザクションで、なおかつ、2ノード間のものを選択し、合計808件のテストケースに対応するテストプログラムを作成した。

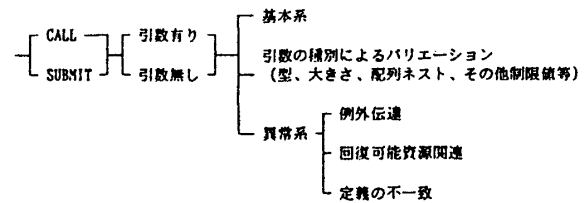


図2. テストケースの分類

#### 6. インタオペラビリティ評価結果

上記のテストプログラムを、3ベンダのマシン上で計6方向のクライアント・サーバの組合せに対して走行させた。

評価の初期の段階で、M I A規定内の引数伝達に関する規定の一部にバージョン間で互換性のない箇所を抽出した。評価対象の3マシン上のM I A製品に対し、この箇所の実装を最新バージョンに揃えるという反映を実施することにより、この問題を解決した。

また、M I A規定中の例外伝達に関する曖昧箇所2件を抽出し、速やかに規定に反映した。その他、宛先に関して、規定を追加した方が良いと思われる箇所1件を抽出した。これについては反映方法を考えて提案を行う予定である。

#### 7. まとめ

MIA-STD Lの特性に基づいてインタオペラビリティ評価法を確立し、作成したテストプログラムを実際の製品に適用した。その結果、本評価法がM I A規定および製品の完成度を一層高めるために役立つことが判明した。

#### 8. 今後の課題

本検討において未解決であった次の項目を今後の課題としたい：引数伝達（特にASN.1変換）に特化したテスト法、分散排他制御・通信路管理の確認法、3ノード以上の他ノード接続、2フェーズコミットなどノードをまたぐトランザクションの評価法。