

OSI トランザクション処理システムの試作

楠 和浩、田中 功一、

中川路 哲男、勝山 光太郎、水野 忠則

三菱電機(株) 情報電子研究所

OSIでは、分散トランザクションの処理のために応用層でトランザクション処理を規定している。今回、我々は、OSIトランザクション処理システムを、オブジェクト指向の手法により、応用層での処理を、プロトコル処理における状態遷移と、データ構造を取り扱う処理に分けて独立性を高めたモデル化を行った。さらに、プロトコル処理をアソシエーション、トランザクション、ダイアログの処理レベルに分割して設計し、モジュール性の高いソフトウェア構成とした。本稿では、設計したモデルとオブジェクト指向言語によって試作したシステムの構成などについて述べた。

An Implementation of OSI Transaction Processing System

Kazuhiro KUSUNOKI, Kouichi TANAKA,

Tetsuo NAKAKAWAJI, Kotaro KATSUYAMA and Tadanori MIZUNO

Information Systems and Electronics Development Laboratory

MITSUBISHI ELECTRIC CORPORATION

5-1-1 OFUNA KAMAKURA-CITY KANAGAWA 247 JAPAN

In the standardization of OSI(Open Systems Interconnection), "TP(Transaction Processing)" is studied to support distributed transaction processing. In this paper, we present the design strategies and implementation methodology for OSI TP software. we have designed it by means of the object oriented approach, and implemented it by using the object oriented language "superC". The object oriented approach and implementation by the object oriented language make us efficient to develop software.

1. はじめに

異なるシステム内に存在する複数のプロセス間で相互に通信を行って1つのまとまった業務を遂行する分散トランザクション処理のためにOSI (Open Systems Interconnection) では、応用層でトランザクション処理応用サービス要素^{[1][2]} [3]を規定している。

今回、我々はOSIトランザクション処理の設計および実装を行ったのでこれを報告する。

2. OSIトランザクション処理の概要

2.1 OSIトランザクション処理のモデル

OSIトランザクション処理においては、OSIトランザクション処理サービス利用者(TPSU)が、OSIトランザクション処理サービス提供者(TPSP)が提供するサービスを利用して分散トランザクションを実行する。分散トランザクションを表わすために、OSIトランザクション処理では図1のような木構造モデルを定義している。

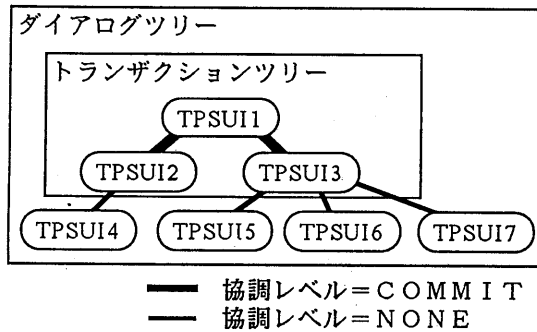


図1. ダイアログツリーとトランザクションツリー

図1においてTPSUIは、あるトランザクションを処理するために生成されたTPSUのインスタンスである。また、TPSU間の関係をダイアログ、ダイアログによって関係付けられたTPSUI全体のツリーをダイアログツリーと呼ぶ。TPSUIはダイアログを①データの転送、②エラー

の通知、③処理の同期、④トランザクションの開始・コミットメント及びロールバック、のために使用することができる。

TPSP提供するコミットメントサービスをTPSUIが使用する／しないによって、ダイアログのタイプが2つに分かれる。TPSUIがTPSPの提供するコミットメントサービスを使用する場合をダイアログの協調レベルがコミットである、使用しない場合をダイアログの協調レベルがNONEであると呼ぶ。ダイアログツリーの内ダイアログの協調レベルがコミットであるダイアログで結ばれた部分をトランザクションツリーと呼ぶ。

TPSPはTPSUに対してTPサービスを提供する複数のAE(応用エンティティ)からなり、応用プロセス実行時に生成されるAEI(応用エンティティインボケーション)は図2の様な構造をしている。

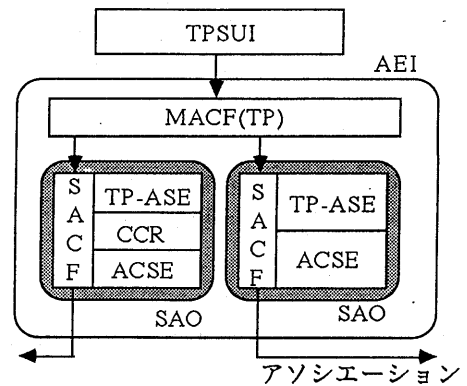


図2. AEIの構造

AEIは、単一のアソシエーションに関連する機能とその情報について制御するSAO(単一アソシエーションオブジェクト)と、複数のアソシエーションに関連した機能を制御するMACF(複数アソシエーション調整機能)からなる。さらにSAOは、複数のASE(応用サービス要素)と、ASE間およびプレゼンテーションサービスの利用を調整するSACF(単一アソシエーション制御機能)からなる。ダイアログの調整レベルがコミットの場合には、分散処理全体の処理の同

期をとるCCRをTP-ASEと同時に用いる。

2.2 OSIトランザクション処理のサービス

表1にOSIトランザクション処理サービスの一覧を示す。

表1. OSIトランザクション処理サービスの一覧

機能単位	TPサービス
カーネル	TP-BEGIN-DIALOGUE TP-P-REJECT TP-U-REJECT TP-END-DIALOGUE TP-DATA TP-U-ERROR TP-P-ERROR TP-U-ABORT TP-P-ABORT
全二重型	特定のサービスは無し
半二重型	TP-GRANT-CONTROL TP-REQUEST-CONTROL
ハンドシェイク	TP-HANDSHAKE TP-HANDSHAKE-AND-END TP-HANDSHAKE-AND-GRANT-CONTROL
コミット	TP-DEFERED-END-DIALOGUE TP-DEFERED-GRANT-CONTROL TP-COMMIT TP-CONTINUE-COMMIT TP-COMMIT-RESULT TP-DONE TP-COMMIT-COMplete TP-PREPARE TP-READY TP-ROLLBACK TP-ROLLBACK-COMplete
非連鎖トランザクション	TP-DEFERRED-NEXT-TRANSACTION TP-UNCHAIN-TRANSACTION TP-BEGIN-TRANSACTION

各機能単位の概略は次のとおりである。

(1) カーネル機能単位

ダイアログの開始・拒否・終了、データ伝送、エラー報告、異常終了のための機能

(2) 全二重機能単位

カーネル機能単位と同時に選択することにより全二重型の通信を実現する機能

(3) 半二重型機能単位

カーネル機能単位と同時に選択することにより半二重型の通信を実現する機能

(4) ハンドシェイク機能単位

通信するTP-SUI間の処理の同期をとるための機能

(5) コミット機能単位

分散トランザクション全体の同期をとるための機能

(6) 非連鎖トランザクション機能単位

ダイアログの協調レベルを変更してアトミックアクションの範囲を更新する機能

3. 基本設計

3.1 設計方針

設計に当たり次のような方針を設けた。

(1) ソフトウェアのモジュール性を高め、さらに既存のソフトウェアとの共存を図るためにオブジェクト指向によるモデル化を行った。

応用層での処理にはPDUのデータ構造を扱うデータ処理と状態遷移処理の2つの側面があり、状態遷移処理ではPDUのデータ構造を意識せずに、必要な値のタイプと意味のみを認識していればよい。したがって、PDUのデータ構造を持ち、データの設定/獲得を行うメソッドを持つオブジェクト(データオブジェクト)と状態遷移処理を行うオブジェクト(プロトコル処理オブジェクト)をそれぞれ独立に設計することにした。

(2) オブジェクト指向により設計したモデルを現実のソフトウェアに実現するための言語として、我々が通信ソフトウェア用に開発したオブジェクト指向言語superC^[4]を用いる。トランザクション処理は特に性能を重視する必要がある

ので、実行速度向上のために、プログラムのコンパイル時に実行すべきメソッドを決定する `superC` のスタティックバインディング機能を積極的に使用する。

(3) 応用層のプロトコルデータは ASN.1 (抽象構文記法1) で記述されている。OSI トランザクション処理のプロトコルデータを処理するために、我々が既に開発した ASN.1 ツール `APRICOT`^[5] を使用する。

3.2 プロトコル処理オブジェクトの設計

OSI トランザクション処理標準において、プロトコル全体で1つに規定されている状態遷移を、本システムでは次の3つに分割することにより、処理レベルを分割し、動作に独立性を持たせることにした。

(1) アソシエーションに関係した状態遷移

(2) 単一ダイアログに閉じた処理に関係した状態遷移

(3) 複数ダイアログにまたがるトランザクション処理に関する状態遷移

それに伴い、本システムでは状態遷移の実行主体によってオブジェクトの構成を3つに分けることにした。すなわち、

(1) アソシエーション管理オブジェクト

ダイアログのアソシエーションへのマッピングなどのアソシエーションに関わる状態遷移を行う。

(2) MACFオブジェクト

コミットメントの調停などの複数のダイアログにまたがるトランザクションの関わる状態遷移を行う。

(3) TPダイアログオブジェクト

データ送信権制御などの単一ダイアログに閉じた処理に関係した状態遷移を行う。

の3つのオブジェクトである。さらに、トランザクションのコミットメント制御を行うために `CCR` プロトコル処理を行う `CCR` オブジェクトを設計した (図3)。

以上のようなオブジェクトの構成によりモジュール性、保守性が高くなり、また同一の `TPSU` が同時に複数のトランザクションツリーに含まれたり、ダイアログの協調レベルの変化に伴うトランザクションツリーの動的変化に対応することが可能である。

`MACF` オブジェクトのインスタンスはトランザクションツリー生成時に生成される。`TP` ダイアログオブジェクトのインスタンスはダイアログ確立時に生成される。いま、図4(1)のような物理リンクで接続されたノード間の関係を仮定する。この関係に於て図4(2)の様なトランザクションツリーを持つトランザクションが同時に発生した場合を考える。このときノードBにおける `MACF` オブジェクトのインスタンスおよび `TP` ダイアログオブジェクトのインスタンスの関係は図4(3)の様な構成となる。

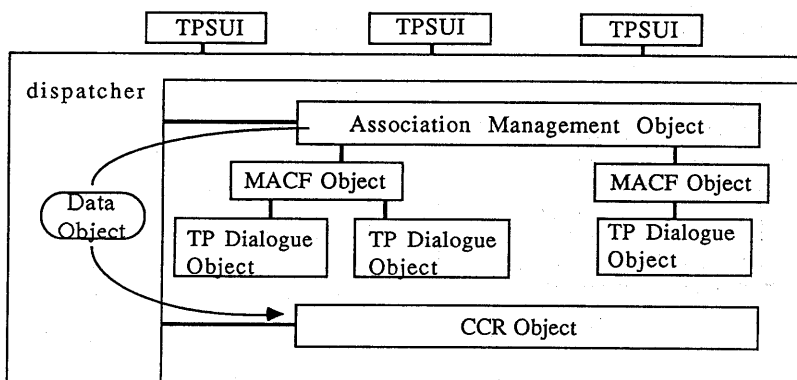


図3. オブジェクト指向によるモデル化

MACFオブジェクトのインスタンスは、それぞれのトランザクションに対応して生成され、個別のトランザクションに関する処理のみを実行する。また、各MACFオブジェクトは、実行中のトランザクションに関係したTPダイアログオブジェクトのインスタンスを管理する。TPダイアログオブジェクトは、あるトランザクションの、単一ダイアログに閉じた状態遷移のみを行い、他のダイアログの状態、複数のダイアログにまたがる状態に関して意識する必要はない。このように、処理レベルを分割することにより、独立性およびモジュール性を高めた。

更に、あるTPSUIのダイアログの協調レベルの変化に対する対応について考える。同一のダイアログの協調レベルの変化に対応するために、本システムではMACFオブジェクトに2つのタイプを用意する。1つは協調レベルがNONEのダイアログに対応したTPダイアログオブジェクトのみを管理するnon-COMMIT-MACFオブジェクト、もう1つは協調レベルがコミットであるトランザクションツリーに属するTPダイアログオブジェクトを管理し、TPダイアログオブジェクト全体の動作を制御するCOMMIT-MACFオブジェクトである。

動的に変化するダイアログの協調レベルに対しては次のように対処する。例えば図5に示す様なダイアログツリーによって構成されているトラン

ザクションを考える。

このときTPSUI1に対するトランザクション処理部のMACFオブジェクトおよびTPダイアログオブジェクトは図6の様な構成をしている。

いま、TPSUI1とTPSUI2との間のダイアログの協調レベルがTP-BEGIN-TRANSACTIONによりNONEからCOMMITに変化する場合を考える。このときTPSUI1に対するトランザクション処理部のMACFオブジェクトおよびTPダイアログオブジェクトは次の様に動作する。

(1) COMMIT-MACFオブジェクトはTP-BEGIN-TRANSACTIONサービスプリミティブの処理の前にnon-COMMIT-MACFオブジェクトに対して処理対象となるTPSUI2に対応するTPダイアログオブジェクトのインスタンスの管理権を移すように要請する。

(2) non-COMMIT-MACFオブジェクトは要請されたTPダイアログオブジェクトの管理権をCOMMIT-MACFオブジェクトに移す。

(3) トランザクションが終了したときにはCOMMIT-MACFオブジェクトはnon-COMMIT-MACFオブジェクトから管理権を移されたTPダイアログオブジェクトの管理権を再びnon-COMMIT-MACFオブジェクトに移す。

以上の様な制御によりダイアログはそのまま、容易にダイアログの協調レベルを変化させることができ、トランザクションツリーの複雑な変化に

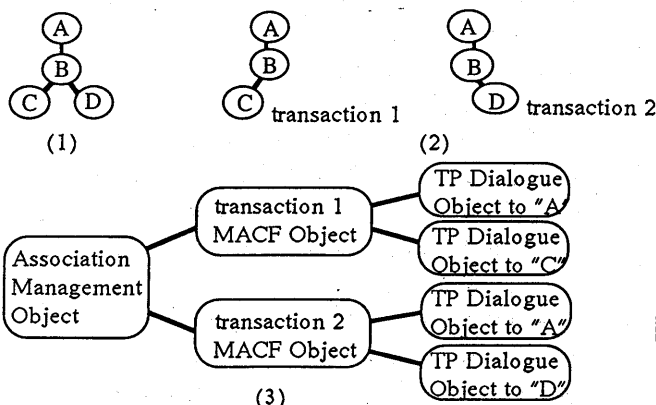


図4. MACF Object と TP Dialogue Object との関係

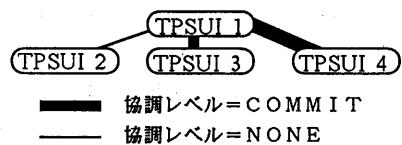


図5. ダイアログの協調レベルの変化

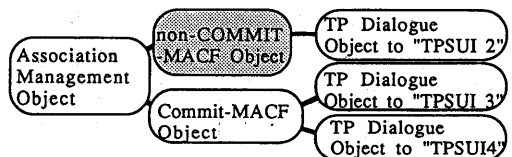


図6. プロトコル処理部の構成

対しても対応することができる。

また、障害（アソシエーション障害／システム障害）に対処するためにOS Iトランザクション処理ではTPチャンネルという特別なアソシエーションによるトランザクション障害回復を規定している。障害に対処するために本システムでは、障害回復対象TP S U Iに対応するTPチャンネルオブジェクトと、TPチャンネルオブジェクトを管理するTPチャンネル管理オブジェクトを設計した。さらに、トランザクション実行中にM A C Fトランザクションオブジェクトからファイルに書き出すことにより障害回復のためのトランザクション状態ログを記録することとした。システム起動時に障害回復対象ログが存在する場合にはシステム障害回復処理が必要であるとみなし、TPチャンネルを使用してシステム障害回復処理を実行する。TPチャンネルは後で述べるアソシエーション構成ファイル内に定義して予めプールする方法と、障害回復用に新規に確立する方法がある。

3. 3 データオブジェクトの設計

TPプロトコル処理オブジェクトを処理の順に従って通過するデータオブジェクトのインスタンスは、利用者からのプリミティブ要求を受けたとき、あるいは下位層からP D Uを受け取ったときに生成される。データオブジェクトは各プロトコルのP D U定義を内部構造として持ち、プロトコル処理オブジェクトからはP D Uへの値の設定または獲得というメソッドを受け付けることによりP D Uへの値の設定または獲得を行う。このためプロトコル処理オブジェクトはP D Uの構造を意識する必要がなく設定又は獲得する値のタイプと値の意味のみを知っていればよい。また、データオブジェクトは各プロトコル固有のデータ構造と全てのプロトコルに共通なデータ構造（例えばプリミティブ識別子）を持つ。そのために共通なデータ構造を持つ共通データオブジェクトを設計し、各P D Uに対応したデータオブジェクトは全て共通データオブジェクトを上位クラスとして定義す

ることとした。

応用層ではP D U間の連結又は埋め込みを考慮する必要がある。図7に連結及び埋め込みの状態を示す。

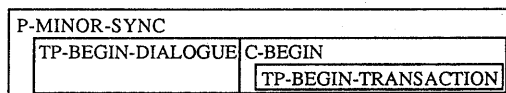


図7. P D Uの連結と埋め込み

このP D U間の連結又は埋め込みを表現するためにデータオブジェクトには連結ポイントと埋め込みポイントという2つのチェインポイントを設定した。2つのチェインポイントを使用することにより図7を図8の様なデータオブジェクトチェインとして表現することができる。

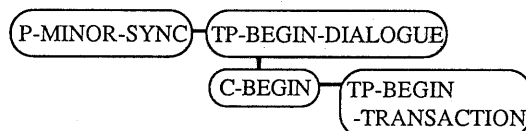


図8. データオブジェクトチェイン

図8において縦のつながりは連結を示し、横のつながりは埋め込みを示す。

P D U送信時のデータオブジェクトの処理は以下のようにして行うものとする。

各プロトコル処理オブジェクトでは次に処理されるプロトコルのデータオブジェクトのインスタンスを生成し、次に処理すべきプロトコルのデータオブジェクトを連結または埋め込んで次のプロトコル処理オブジェクトに送信する。上位層の全てのプロトコルを通過したデータオブジェクトは連結管理オブジェクトにキューイングする。連結管理オブジェクトではキューイングされてくるデータオブジェクト同士の連結又は埋め込みを行う際に必要なプレゼンテーション層データオブジェクトの強弱の判定やデータオブジェクトチェインのつなぎ替えを行う。

P D Uを受信した場合には受信したP D Uから図8の様なデータオブジェクトチェインを生成し、各プロトコル処理オブジェクトを通過することにより処理を実行する。通過の過程で処理が済んだ

データオブジェクトには処理済みのフラグを立てることにより次にどのプロトコル処理オブジェクトにデータオブジェクトチェーンを送信すればよいか分かる。

3.4 アソシエーションの取り扱い

OSIトランザクション処理ではダイアログをマッピングするアソシエーションはアソシエーションプールにプールされているものとしている。本システムではこのアソシエーションプールをアソシエーション管理オブジェクト内に設定している。しかしながらこのアソシエーションプールにアソシエーションをプールするタイミングに関しては標準では何も定義していない。

アソシエーション確立タイミングには次の3つが考えられる。

- (1) システム起動時
- (2) ダイアログ確立時
- (3) 外部からの非同期な要求時

本システムでは以上の3つの場合に対応するために次のような設計を行った。

まず、アソシエーション情報を定義するアソシエーション構成ファイルを定義する。このファイルには確立することができるアソシエーションに関して次のような情報が格納されている。

- ・アソシエーション確立タイミング
(システム起動時/システム動作中)
- ・アソシエーション確立要求発行サイド
(自システム/相手システム)
- ・アソシエーション確立時の競合状態
(競合勝者/競合敗者)
- ・アソシエーションのアプリケーションコンテキスト
(コミット機能単位有り/無し)

システム起動時に、このアソシエーション構成ファイルをシステム内に読み込み、アソシエーションの確立タイミングがTPSP起動時に設定されているアソシエーションについては、この時点でアソシエーションの確立を行いアソシエーシ

ンプルに登録する。ダイアログの確立要求がアソシエーション管理オブジェクトに到着した場合にはアソシエーションプール内に適当なアソシエーションがあるかどうかを検索し、もしも適当なアソシエーションが存在する場合には、そのアソシエーションにダイアログをマッピングする。もしも適当なアソシエーションがない場合にはアソシエーション構成ファイルからアソシエーション未確立でダイアログの属性に合致するアソシエーションを検索しダイアログ確立前にアソシエーションを確立する。

4. システム構成

4.1 開発環境

実装した計算機は三菱エンジニアリング・ワークステーション MELCOM ME200であり、OSはUNIXである。

4.2 ソフトウェア構成

オブジェクト指向言語による実装を行ったためにソフトウェア構成はオブジェクトのクラスの集合となる。クラス構成を図9に示す。

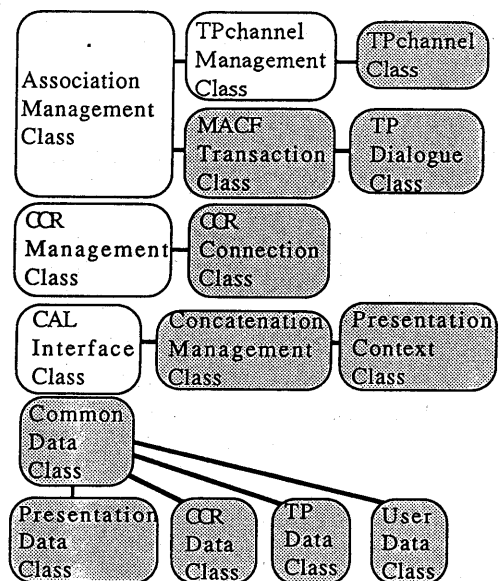


図9. ソフトウェア構成

ここで、網掛けのクラスは複数のインスタンスが生成されることを示しており、横のつながりはオブジェクト間の包含関係を、縦のつながりはクラスの継承関係を示す。

データクラスのインスタンスはデータ毎に存在する。それ以外のクラスのインスタンスのうち網掛けのなされていないクラスのインスタンスはプロセス起動時にただ1つのインスタンスが生成される。

4. 3 ユーザインタフェース

全てのプリミティブをユーザに提供しており、またサービスプリミティブを複数指定することを可能にすることでユーザレベルでPDUの連結を意識することが可能なインタフェースとなっている。

5. おわりに

OSIトランザクション処理の設計及び試作を行った。オブジェクト指向によるモデリングと、オブジェクト指向言語による実装によりモジュール性の高いソフトウェアを構築することが可能であることが分かった。今後はシステムの性能評価などを行う予定である。

参考文献

- [1] ISO/IEC JTC1:Information technology - Open Systems Interconnection - Distributed transaction processing - Part1:TP Model, ISO/IEC DIS 10026-1, (1990)
- [2] ISO/IEC JTC1:Information technology - Open Systems Interconnection - Distributed transaction processing - Part2:Service Definition, ISO/IEC DIS 10026-2, (1990)
- [3] ISO/IEC JTC1:Information technology - Open Systems Interconnection

- Distributed transaction processing - Part3:Protocol Specification, ISO/IEC DIS 10026-3, (1990)

- [4] 勝山他：通信ソフトウェア向けオブジェクト指向言語superC, 情報処理学会論文誌, Vol.30, No.2, pp.234-241 (1989)
- [5] 中川路他：OSI抽象構文記法支援ソフトウェアAPRICOTの開発と評価, 電子情報通信学会論文誌, Vol. J73-D-1, No.2, pp.225-234 (1990)