

分散機能DF/UXにおける
X/Open XA及びOSI-TPの実装

4W-9

江口栄治郎*¹ 石川博道*²

*1 日立西部ソフトウェア(株) *2 (株)日立製作所

1. はじめに

近年、コンピュータシステムのダウンサイジング化、マルチベンダ化及びネットワークのオープン化に伴いマルチベンダ間の異機種分散システムが注目されている。このようなソフトウェアオープン化の波に従いX/Open、OSI等の業界標準、世界標準の仕様に準拠したソフトウェアの開発が一般的となりつつある。この様な状況の中、通信プロトコルとしてOSI規格であるRDA及びTPに準拠し、DBMSとのインタフェースとしてX/Open XAを実装したワークステーション上で動作する分散機能DF/UXの開発を行った。本稿では、DF/UXにおいてトランザクション制御インタフェースとしてX/Open XAを、またトランザクション制御プロトコルとしてOSI-TPを実装した際の仕様のマッピング及び実装マターとなっている仕様の実現方法についてその問題点と解決方法について報告する。

2. 標準分散処理システムモデル

2.1 分散トランザクション処理モデル

分散トランザクション処理モデルとしてX/Openにおいて図1に示すようなDTP(Distributed Transaction Processing)モデルが規定されている。DTPモデルでは、分散トランザクション処理を行うために必要な機能を3つの機能単位に分割して3boxモデルとして定義している。

(1) AP(アプリケーション・プログラム)

トランザクションのバウンダリを決めてトランザクションを構成する動作を定義する。

(2) RM(リソース・マネージャ)

共有資源(データベース等)へのアクセスを行う。

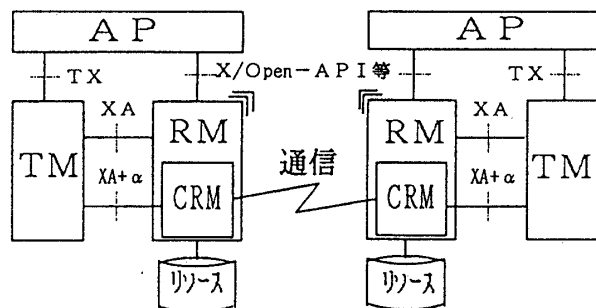


図1 X/Open DTPモデルの概要

Implementation of X/Open XA and OSI-TP on DF/UX

Eijirou Eguchi**, Hiromichi ISHIKAWA*
*Hitachi, Ltd. **Hitachi Seibu Software Co., Ltd.

(3) TM(トランザクション・マネージャ)

トランザクション識別子の割当て、処理のモニタ、トランザクション完了・障害回復を行う。

上記3 boxモデルにおいてRMは、自システム内のローカルなリソース管理を行うRMと他システム上のリモートリソースへのアクセスを行う通信RM(CRM)とに区別できる。

DF/UXは、このRM及びCRMに相当し、分散トランザクション処理におけるOSIプロトコルに準拠した通信管理機能を提供する。

2.2 システム間インタフェース

DTPモデルでは、TMとRMとのインタフェースをXAインタフェースとして規定している。また、DTPモデルにおいては特にAPとRMとのインタフェースは規定していないが、X/OpenのSQL(CLI)、XATMI、TxRPC、Peer to Peer(CPI-C)等の標準的なインタフェースを想定している。

3. 標準システムモデルの実装

図2にDF/UXにおいて実現したDTPモデル対応のシステム構造と制御構造を示す。

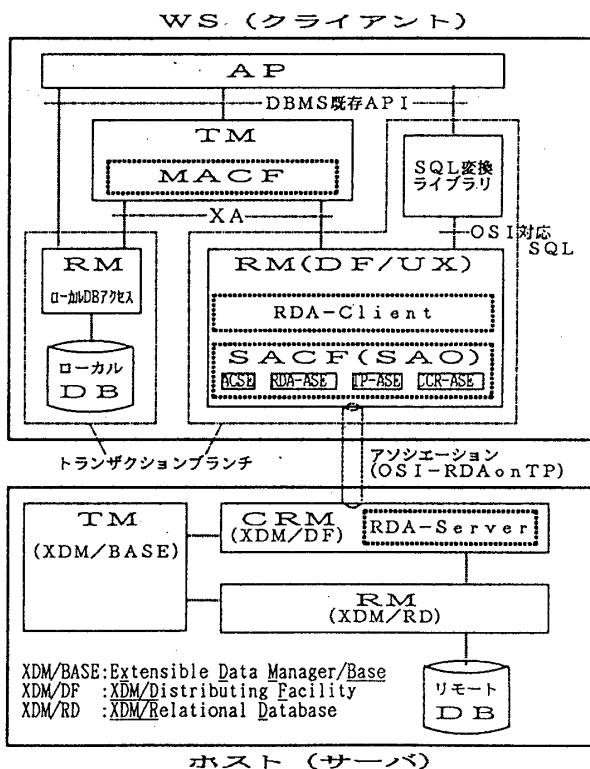


図2 DF/UXシステム構造と制御構造

DF/UXは、OSI RDA、TPプロトコルマシン部分を含む基本処理部分とSQL変換を行うライブラリとに分かれている。DF/UX基本部分はOSI-RDAプロトコルを制御するRDA-ClientとOSI-TPにおけるシングルアソシエーションを管理するSACF(SAO)からなる。DBMS(TM)は、DF/UXをローカルなリソースを管理するRMと同様に管理する。この時、OSI-TPにおけるマルチアソシエーションを管理するMACF機能はTMにおける複数トランザクションランチの管理と同等であるため「トランザクションランチ=アソシエーション」と対応付けることによりMACF機能はTMが受け持つ。また、SQLインタフェースとして各DBMS対応の既存APIを用いるが、OSI-RDA対応の標準的SQLとの差異は、SQL変換を行うライブラリにて吸収し基本部分への影響を最小限に止めている。

このように、各コンポーネントの機能範囲を明確にすることにより、独立性を高めオープン環境に柔軟に対応可能である。

4. XAインタフェースサービスのOSIプロトコルへのマッピング

表1にXAサービスとDF/UX内のサービスとのマッピングについて記述する。

- (1)トランザクション制御インタフェースとしてXAを用いるためDF/UX内部でXAサービスをOSI-TPのAFサービスプリミティブ及びCサービスプリミティブにマッピングしている。
- (2)DF/UXは、OSI RDA基本応用コンテキストによる一相コミットとOSI RDA TP応用

表1 XAコールの処理マッピング

XAサービス	機能概要	DF/UXのマッピング
xa_open	RM初期化	DF/UXリソース確保
xa_close	RM使用終了	DF/UXリソース解放
xa_start	使用せず	-
xa_end	トランザクションランチ終了	SQL受付終了
xa_prepare	コミット準備問合せ	C-PREPARE-RI
xa_commit	コミット指示	C-COMMIT-RI C-RECOVER-RI
xa_rollback	ロールバック指示	C-ROLLBACK-RI
xa_recover	ランチ情報取得	未完了XID取得
xa_forget	ランチ情報消去	DF/UXランチ解放
xa_complete	使用せず	-
xa_reg	トランザクションランチ登録	C-BEGIN-RI
SQLコール	SQL処理要求	ファースト A-ASSOCIATE TP-BEGIN-DIALOGUE-RI R-Initialize-RI ^{*1} R-Open-RI ^{*2} R-ExecDBL-RI ^{*2} 以外 R-ExecDBL-RI ^{*2}
Connect	CLI-Connect相当	DF/UXへのユーザ登録
Dis-connect	CLI-Disconnect相当	R-Close ^{*2} TP-END-DIALOGUE-RI A-RELEASE

* 1 : TP-BEGIN-DIALOGUE-RIにマッピング
* 2 : TP-DATAにマッピング

コンテキストによる2相コミットをサポートしている。OSIプロトコルにおいては使用する応用コンテキストをアソシエーション確立時に決定する必要がある。また、接続するDBMSにより使用するSQL制御インタフェースが異なる事が考えられる。そこでDF/UXではこれら機能を動的に選択可能とするため、実装マターとなっているxa_openにおけるオープン文字列の指定で選択可能とした。

- (3)DF/UXにおけるトランザクション開始のタイミングは、ファーストSQL実行時に動的に発生するためTM側では制御が難しい。このため、TMからのトランザクションランチ開始宣言(xa_start)を使用せず、ファーストSQL実行時にDF/UXから動的登録(ax_reg)を用いることとした。
- (4)xa_prepare、xa_commit、xa_rollbackサービスに関しては、直接OSI-CCRのサービスにマッピングした。この時、XAにおけるXIDのグローバルトランザクションIDをOSI-CCRのアトミックアクションIDに、XIDのランチ修飾子をOSI-CCRのランチ修飾子にマッピングすることにより、容易にトランザクションを識別、管理できるようにした。
- (5)リカバリ処理に関しては、DF/UX内部でリカバリ用情報、状態を保持、管理することにより自動的に通常プロセスかりリカバリプロセスかを判断し通信プロトコルを制御するため、DBMS(TM)側では通信プロトコルを意識することなくコミットメント制御可能である。

このように、XAサービスをDF/UX内のサービス及びOSIサービスにマッピングをすることにより、XAインタフェースを用いたDBMSとの接続性を確保することができた。

5. おわりに

本稿では、X/Open及びOSI標準規格の実装について述べた。本実装方式により、標準的な既存DBMSとの接続性を確保すると共に、標準規格に基づいた異機種分散を実現しマルチベンダ指向のオープンシステムに対応することができた。

参考文献

- [1] X/Open Guide, Distributed Transaction Processing Reference Model
- [2] X/Open CAE Specification, Distributed Transaction Processing: The XA Specification
- [3] OSI実装規約 上位層共通実装規約書 V2.0 (財)情報処理相互運用技術協会
- [4] 情報処理学会第46回全国大会(90-6) 「分散機能DF/UXにおけるシステムの実装」