

分散機能DF/UXにおける
システムモデルの実装

9G-5

江口栄治郎*1 木村淳美*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。このようなマルチベンダ間の接続性を確保するためにプロトコル及びシステム間インタフェースの標準化が重要となってくる。現在、OSI、X/Open、INTAP等の標準化団体では、より一層のオープン化を押し進めるために多くの標準規格化作業がなされている。特に分散トランザクション及びデータベース関係ではさまざまな標準規格がすでに制定されており、それに準拠した製品の開発が急がれている。この様な状況の中、OSI規格であるRDAに準拠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におけるX/Open等の標準規格を基にした分散処理システムモデルの実装及びプロセスの制御構造について報告する。

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のXTPワーキング・グループにおいてDTP(Distributed Transaction Processing)モデルの検討が行われている。図1にX/Open DTPモデルの概要を示す。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3 box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)

共有リソース(データベース等)へのアクセスを行う。

(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とに区別できる。

DF/UXは、この通信RMに相当する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。XAインタフェースにおいてはRMと通信RMは同じ位置付けである。TMと通信RMのインタフェースではXAインタフェースにリモートなグローバルトランザクションを管理するために必要な機能を追加する必要がある。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CLI(Call Level Interface)、RPC、Peer to Peer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モデルを示す。

DF/UXは、OSI RDAプロトコルマシン部分を含む基本処理部分(通信RM)とインタフェース

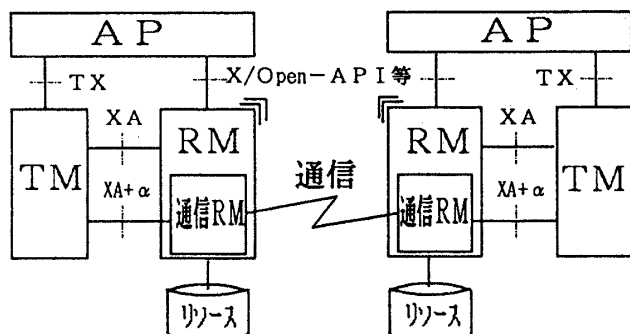


図1 X/Open DTPモデルの概要

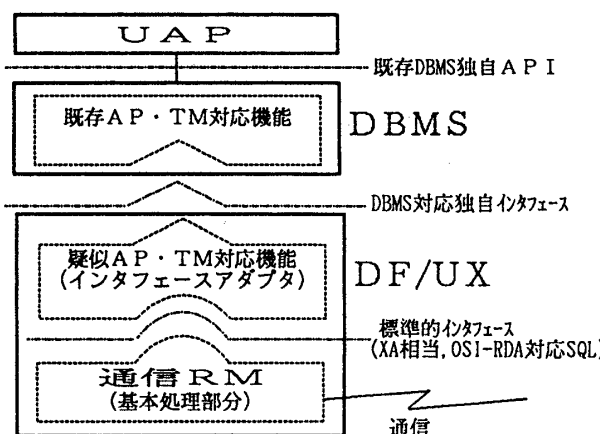


図2 DF/UXのシステムモデル

Implementation of System Model on DF/UX

Bijirou Eguchi*, Atsumi Kimura*, Hiromichi Ishikawa**
*HITACHI Seibu Software Co., Ltd. **HITACHI, Ltd.

部分(インタフェースアダプタ)とを分離独立した構造である。インタフェースアダプタは、各DBMS対応の独自インタフェースを吸収し基本部分の標準的インタフェースに変換する。このように、既存の独自インタフェースから標準インタフェースに変換するアダプタ部分を独立化し基本処理部分のインタフェースを標準的なインタフェースとして規定することにより、最少限の変更で様々なDBMSとの接続が可能である。基本処理部分のインタフェースは極力XAインタフェースとするが、当初はOSIRDA基本応用コンテキストのみ実現するため、トランザクション回復等を除いた基本的なインタフェースだけを実現することとした。OSIRDA TP 応用コンテキストによる2フェーズコミットを用いたトランザクション回復機能をサポートする時にはフルサポートの必要がある。DF/UX基本処理部分は通信RMとして動作するため上位にTM機能が必要である。しかし、既存DBMSには十分なトランザクション管理機能がないため不足機能はインタフェースアダプタで仮に実現する。

4. プロセス制御構造

図3にDF/UXのシステム概要と処理プロセスの割当て等の制御構造について示す。
 (1)インタフェースライブラリは処理要求を受けると共有メモリ内の共有テーブルを参照し待機中の処理プロセスを割当て(①,②)。待機中の処理プロセスがない場合は、マネジャプロセスに対して処理プロセスの生成・割当てを要求する(④)。マネジャプロセスは新規に処理プロセスを生成し割当て(⑤,⑥)。DF/UX起動時に予め生成しておく処理プロセス数と生成可能な最大の処理プロセス数をユーザが定義できるようになっており、プロセス生成に

よるオーバーヘッドとシステムリソースの最適化が可能である。

(2)処理プロセスの割当ては、ユーザ(UAP)単位に行う。この時、DF/UXはユーザに対してコネクションを割当て(③,⑦)。以後ユーザは割当てられたコネクションを指定してDF/UXに対して処理要求する。

(3)DBMSプロセスと処理プロセスとの遣り取りはプロセス間通信処理用のメッセージルーチンを用いて行う。

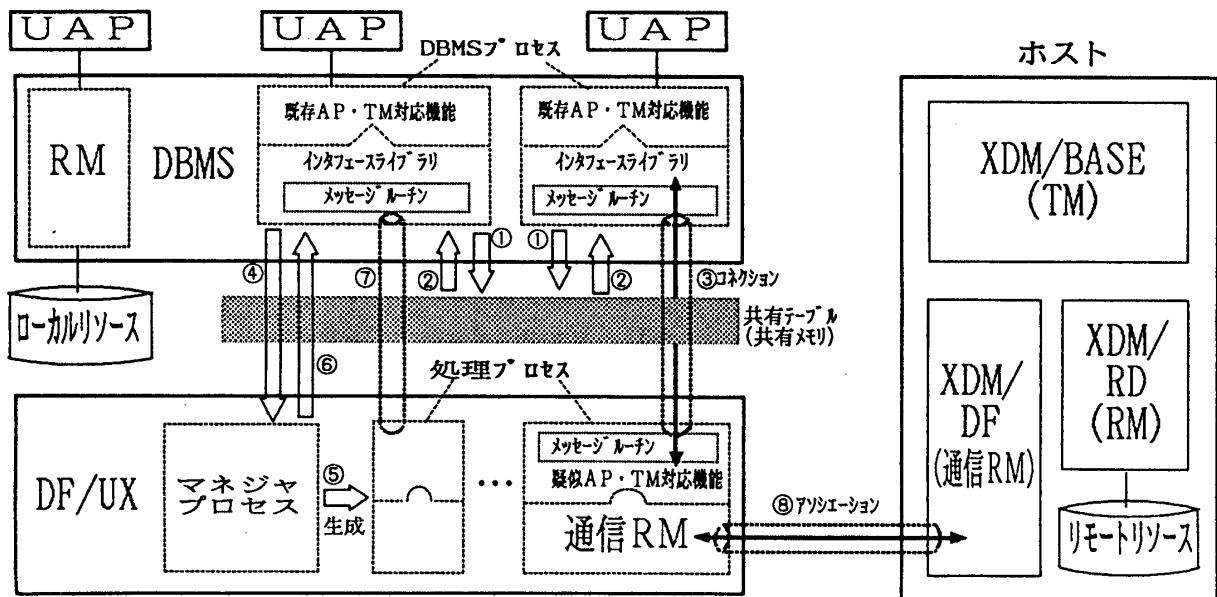
(4)DF/UXは、リモートリソース名称を基にアクセス対象のシステムを求め通信路(アソシエーション)を確立しリモートアクセスを行う(⑧)。

5. おわりに

本稿では、ワークステーション-ホスト間での分散データベースアクセスにおけるX/Openの標準規格を考慮したシステムモデルの実装について述べた。本システムモデルの実装方法により、標準的なDBMSとの接続性を確保するなど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] OS I 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



XDM/BASE : Extensible Data Manager/Base
 XDM/DF : XDM/Distributing Facility
 XDM/RD : XDM/Relational Database

図3 DF/UXシステム概要と制御構造