

## リモートデータベースアクセス(RDA)

鈴木 健 司  
NTT 電気通信研究所

RDA (Remote Database Access) は、OSI環境にあるデータベースを遠隔から操作するためのサービスとプロトコルである。このRDAを国際標準にするための検討が、ISO/TC97/SC21のデータベース作業グループ(WG3)で行われている。現在、第2版作業草案をもとに検討が進められている。RDAは、汎用的なリモートデータベースアクセスを行うためのサービスを規定し、この中で、データ操作については、特定なモデルに基づいて規定している。このモデルとしては、ISOデータベース言語SQLが用いられている。本稿では、WG3でのRDAの標準化活動状況として、仕様案の内容と、その検討状況を報告する。

## Remote Database Access (RDA)

Kenji SUZUKI  
NTT Electrical Communication Laboratories  
1-2356, Take, Yokosuka-shi, Kanagawa, 238-03 Japan

The RDA standard is a specification for a Remote Database Access Service and Protocol for Open Systems Interconnection. This RDA standard is under development by Database Working Group (WG3) of ISO/TC97/SC21 and is currently in the second Working Draft stage. The RDA standard defines a general RDA Service for access to remote database. This RDA standard also defines a specific model for data manipulation to databases using SQL as defined in ISO Database Language. This paper is an overview of RDA specifications and WG3 activities.

## 1. はじめに

RDA (Remote Database Access)<sup>1)</sup>は、OSI (Open Systems Interconnection) 環境にあるデータベースを遠隔から操作するためのサービスとプロトコルである。

このRDAは、ISO (国際標準化機構)のTC97 (情報処理システム技術委員会)において、ECMA (欧州計算機製造連合)から標準化項目とすることが提案され、1985年11月にTC97/SC21に作業項目が割り当てられた。実際の活動は、SC21のデータベース作業グループであるWG3で行われている。

現在、RDAの第2版作業草案(WD)<sup>1)</sup>をもとに検討が行われている。計画では、1987年6月の東京会議を経て国際規格草案(DP)に、1988年6月に国際規格案(DIS)に、1989年6月に国際規格(IS)にする予定である。

本稿では、ISO/TC97/SC21/WG3でのRDAの標準化活動状況として、仕様案の内容とその検討状況について概要を報告する。

## 2. RDAの位置付けと構成

OSIの基本参照モデル(basic reference model)<sup>2) 3)</sup>は、システムの相互接続のレベルとして、7層の階層を設定している。RDAは、この7層の最上位である応用層(application layer)の応用サービス要素(Application service Element: ASE)の1つに位置付けられる。

### (1) リモートデータベースアクセスの構成

第2版WDでは、リモートデータベースアクセスのための構成要素を以下のように規定している。この構成要素を図1に示す。

リモートデータベースアクセスは、データベースの情報処理に関係する利用者要素(User Element: UE)と、通信処理に関係するRDA通信要素に分けられる。RDA通信要素は、RDAサービスとしてUEとのインタフェースを規定し、RDAプロトコルとしてRDA通信要素間のインタフェースを規定する。

UEは、その役割により、クライアントUEとサーバUE(以下、単にクライアントとサーバと呼ぶ)に分けられる。クライアントはリモートデータベースアクセスを依頼するUEであり、サーバは依頼されたデータベースアクセスを実行するUEである。

クライアントは、さらにアプリケーションとクライアントプロセスからなる。アプリケーションは、データベース処理を行う応用プログラムあるいは問合せ言語プロセッサの実行である。クライアントプロセスは、アプリケーションの発行するデータベース言語の手続きインタフェースを受けて、RDA通信要素のRDAサービスインタフェースに変換する。RDA通信要素は、クライアントとサーバの代りに通信を操作する要素である。

サーバは、サーバプロセス、DBMS、データベースからなる。サーバプロセスは、クライアントに代って、データベース手続きインタフェースへの変換と発行を行い、その処理結果をRDA通信要素を用いて送り返す。

この構成要素は、リモートデータベースアクセスを理解するうえで有用であるが、UE内の構成方法や、データベース言語との手続きインタフェース方法は、インプリメンテーションにかかわるものであり、規定対象とすべきではないことが指摘され、これらについては、チュートリアルとすることで改訂される予定である。また、UE、RDA通信要素などの用語も、最新の応用層構造(Application Layer Structure)と整合していないので、これらも改訂される予定である。従って、応用サービス要素であるRDAとは、RD

A通信要素であり、そのサービスとプロトコルが規定対象である。

## (2) アクセスモデル

RDAサービスインタフェースは、リモートデータベースアクセスのためのモデルとして、次のものを設定している。

- ・リモートデータベースアクセスのための汎用モデル (generic model)
- ・特定のデータベース言語に基づくデータベースアクセスのための特定モデル (specific model)

この2つのアクセスモデルは排他的な関係ではなく、リモートデータベースアクセスの枠組みが汎用モデルで規定され、その実際のデータ操作の内容が特定モデルで規定されることを意味している。特定モデルとして、現状ではISOデータベース言語SQL<sup>4)</sup>に基いて規定されている。

## 3. RDAサービス

RDAサービスは、リモートデータベースアクセスのための汎用モデルを規定する。それは、次のものから構成される。

- ・アソシエーション管理
- ・資源ハンドリング
- ・トランザクション管理
- ・データ操作

これらのRDAサービスの一覧を表1に示す。表中のオペレーションは、RDAサービスプリミティブを定義する。

### 3.1 サービス概要

現状のRDAサービスの内容の概要は以下のとおりである。

#### (1) アソシエーション管理

クライアントは、リモートデータベースアクセスを行うのにあたり、接続すべきサーバとの間に論理的な関係を確立する必要がある。この関係をアソシエーション(association)という。アソシエーションの確立と解放は、クライアントによりそれぞれ R-Associate、R-Release により行われる。

#### (2) 資源ハンドリング

クライアントはデータ操作の前に、アクセスするサーバのデータ資源を利用可能にする必要がある。データ資源は、データ操作において使用されるデータの単位を一意に識別するものである。これは特定モデルの資源を指し、データベース、スキーマ、サブスキーマなどがある。データ資源の利用の要求と終了は、それぞれ R-Open、R-Close により行われる。

#### (3) トランザクション管理

トランザクションは、クライアントにより決定されるデータベース処理のコミットメント単位である。これは、データベース処理がすべて完了したか、すべて完了していないかのどちらかの状態を識別するものであり、これによりデータベースの一貫した状態を保証

するものである。これをアトミックアクションともいう。

RDAでは、トランザクションのコミットメントレベル（注）として以下の3つを設定している。これは、アソシエーションを確立するR-Associateで指定できる。

（注）第2版WDではCCRレベルと呼ばれているが、後述する他の応用サービス要素であるCCRと混乱するために次版でこのように変更される予定。

- ・レベル1：ノートランザクション（no transaction）処理。非保護（unprotected）環境下での検索のみのアクセスであり、繰り返される検索処理は、異なった結果となる可能性がある。
- ・レベル2：1相コミットメント（one-phase commitment）処理。
- ・レベル3：2相コミットメント（two-phase commitment）処理。

このようなレベルは、次のような処理環境に適用できると考えられる。レベル1は検索のみのアプリケーションに適用できる。ただし、その検索結果が常に同じであるためには、レベル2か3を使用する必要がある。レベル2はクライアントがただ1つのサーバとの間でデータベース処理を行う環境に適用でき、性能上有益である。レベル3は複数のサーバとの間で分散処理を行う環境に適用できる。

表1のトランザクション管理のオペレーションは、レベル2とレベル3に適用される。トランザクションの開始は R-BeginTransaction により行われ、終了は R-Commit あるいは、R-Rollback により行われる。レベル3の2相コミットメント処理の場合には、R-PrepareToCommit により第1相処理を行い、R-Commit か R-Rollback で第2相処理を終了する。処理の途中で障害があり、処理が中断された場合には、その後の再開時にクライアントとサーバの処理を同期させる必要がある場合がある。このために、R-Restart により、クライアントはサーバがトランザクションをコミットしたか、ロールバックしたかを確認することができる。このようなトランザクション管理を行うためには、クライアントのデータベース言語のトランザクション関連の文（statement）、たとえばSQLのCOMMIT文やROLLBACK文は、これらのオペレーションにマッピングされなければならない。

サーバが複数のクライアントと関係し、同時に複数のトランザクションを処理しているとき、トランザクションの直列可能性（serialisability）を保証するのは、サーバの責務である。この直列可能性を達成するためのメカニズムである同時実行制御（concurrency control）は、RDAでは規定していない。

#### （4）データベース操作

データ操作オペレーションは、データを検索したり、データの値を更新したりするデータ操作言語（DML）文を転送し、その処理結果を返却する。このDML文が特定モデルにあたり、現RDAではSQLに基づいて規定される。

データベース操作オペレーションは4つあり、次の2つに分類される。

- ・DML文の直接実行に対するオペレーション  
1つのDML文を指定した回数だけ繰り返して実行することが、R-ExecutedDMLで行える。
- ・DML文の蓄積と実行に対するオペレーション  
1つのDML文をチェックしてサーバに蓄積することが R-DefineDMLで行える。このとき蓄積されたDML文を後で参照するためにコマンドハンドルが返却される。コマンドハンドルを呼出し、指定した回数だけ繰り返して実行することが R-InvokeDMLで行え、その蓄積を無効にすることが R-DropDMLで行える。このコマンドハンドルの有

効期間は、R-DropDMLで指定されないならばアソシエーションが解放されるまでの間である。現仕様は、最初に簡潔性と簡易性をねらっているため複雑なサービスは将来課題としている。従って、コマンドハンドルのアソシエーションを越えた永続性や、複数の異なるDML文の実行のようなマクロ機能は、将来課題として検討していく必要がある。

いずれの実行においても実行の結果、複数のデータ（SQLでは行（row））が得られた場合には、1つの転送内でこれらを返却することができる。

データ操作オペレーションでは、DML文の実行に必要な入力となる引数（argument）の渡し方と、出力となる実行の結果（result）の受け取り方についての表現が定義されている。

引数と結果は、それぞれ型リストと値リストから定義される。引数型リスト（ArgTypeList）は、サービスへの入力となるすべてのデータを解釈するのに必要な型情報を含み、結果型リスト（ArgValueList）は、サービスから受け取るための値の型情報を提供する。これらの型リストは各フィールドの型を記述する型記述子（TypeDescriptor）からなる。

引数値リスト（ArgValueList）と結果値リスト（ResultValueList）は、それぞれすべての入力データとすべての出力データからなる。これらは構造情報をもたない OCTET STRING からなり、[APPLICATION1]、[APPLICATION2]とタグ付けられている。

DML文の実行に際し、DML内で実際の引数を参照するためには、引数参照（ArgReference）パラメタと結果参照（ResultReference）パラメタが使用される。これらのパラメタは、型参照（TypeReference）と値参照（ValueReference）の対からなる。型参照は型リスト中の型記述子のエントリ番号を、値参照は値リスト中の参照する値のエントリ番号を示す。図2に、入力データの参照の例を示す。

これらの定義は、特定モデルに依存しない柔軟性の高いメカニズムを提供するものとして提案されている。

### 3.2 データ表現

RDA応用サービス要素間で情報を交換するために、RDA規格では、プロトコル構文とメッセージのデータ内容を定義するのにISO抽象構文表記法（Abstract Syntax Notation One: ASN.1）を使用して表現している。このことにより、プレゼンテーション層で抽象構文の符号化（encoding）が行われ、転送構文に変換される。特定モデルであるDML文、即ちSQL文に対する構文もこのASN.1の表記法で規定される。

### 3.3 サービスの状態遷移

RDAでは、アソシエーションを通してデータベースになされる仕事を、状態の遷移としてモデル化している。オペレーションはこの状態間の遷移を意味し、適切な状態にあるとき正当であり、状態に違反するときエラーとなり規定される。

図3はこのRDAの状態遷移を示す。このモデルは、コミットメントレベル2と3のものであり、障害時の状態は示していない。レベル1については現仕様では規定がなく、次版で追加される予定である。

## 4. サービスマッピング

ISOの応用層の応用サービス要素には、アソシエーション制御を行うACSE（Association Control Service Element）、アトミックアクションに対する2相コミット

メント制御を行うCCR (Commitment, Concurrency and Recovery)、応用サービス要素間のリモートオペレーションを行うROS (Remote Operation Service) がある。

RDAは、この応用サービス要素ACSE、CCR、ROSを利用し、プレゼンテーション層以下の下位層へマッピングする。すなわち、RDAのアソシエーション管理はACSEを、トランザクション管理はCCRを、資源ハンドリング及びデータ操作はROSを利用する。

## 5. 検討課題

現状のWDは、RDAの汎用モデルアーキテクチャや、機能面ではかなり見通せるようになったが、仕様面ではISOとの文書形式への整合、仕様の記述の詳細化などまだ問題が残されている。これらの多くは次版で解決されると見込まれる。最初は簡潔で簡易なものにしようという方針に沿った仕様になっているが、最初から必要なものについてはまだ提案の余地がある。現在あげられている課題を以下に列挙する。

### (1) コミットメントレベルの問題

ISO標準データベース言語を用いる場合には、トランザクション概念が必須であるので、レベル2か3を使用するのが妥当であるが、参照のみの処理の効率化がはかれるようにする必要がある。レベル1については、状態遷移とエラー条件がまだ規定されていないし、マッピング方法が明確でない。

CCRは2相コミットメントのみを提供しており、レベル2の1相コミットメントと整合がはかれていない。さらに、パラメタにCCRと整合がとれていないところがある。

### (2) ASN. 1によるSQL簡易表記法の必要性

SQL文の構文は、ASN. 1でタグ付けられた構文で表現しなければならないが、標準SQL文で記述されている場合には、変換効率を考慮してそのまま転送できるような簡易な表記法が必要である。

### (3) 応用層構造との不整合

O/S Iの用語・概念として最新の応用層構造と整合していないところがある。

### (4) 将来課題

以下のものは将来の拡張課題としてあげられる。

- ・データ操作のマクロ機能
- ・大容量データの効率的な転送方法
- ・データベースアクセスの非同期処理方法
- ・分散データベースへの適用方法
- ・トランザクション処理プロトコル (TP) との関連

## [ 参考文献 ]

- (1) ISO/TC97/SC21 N1646, Second Working Draft of ISO Remote Database Access (Dec. 1986).
- (2) ISO 7498 Open Systems Interconnection - Basic Reference Model (1984).
- (3) JIS X 5003 開放型システム間相互接続の基本参照モデル (1985).
- (4) ISO 9075 Database Language SQL (Nov. 1986).

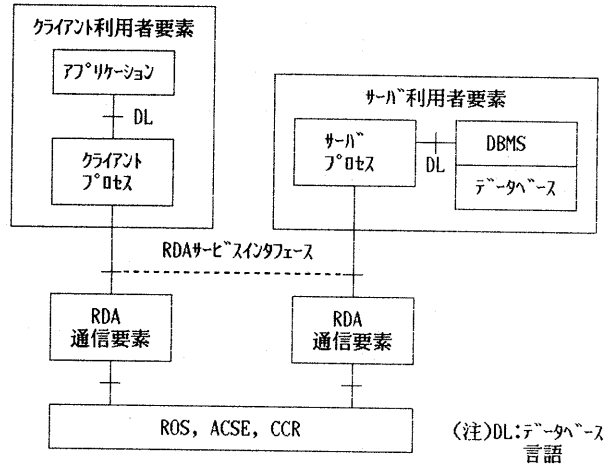


図1 RDAサービス構成<sup>1)</sup>

表1 RDAサービス一覧

サービス	オペレーション	機能概要	入力パラメタ
アソシエーション管理	R-Associate	クライアントとサーバ間のアソシエーションを確立する	プロトコルクラス サービスクラス コミットメントレベル 利用者識別、課金
	R-Release	アソシエーションを解放する	強制解放、中断
資源ハンドリング	R-Open	アクセスする資源の利用を要求する	データ資源名 使用モード
	R-Close	資源の利用を終了する	データ資源名/資源ID
トランザクション管理	R-BeginTransaction	トランザクションを開始する	アトミックアクションID プランチID
	R-PrepareToCommit	2相コミットメントの第1相処理を要求する	—
	R-Commit	トランザクションを終了する	—
	R-Rollback	トランザクションを取り消す	—
	R-Restart	再開のためクライアントとサーバの同期をとる	アトミックアクションID プランチID、再開始点
データ操作	R-ExecuteDML	DML文を実行し結果を返却することを要求する	データ資源ID 引数型リスト 結果型リスト DML文、繰り返し数 引数値リスト
	R-DefineDML	DML文をチェックし蓄積する(コマンドハンドルを返却)	データ資源ID 引数型リスト 結果型リスト、DML文
	R-InvokeDML	蓄積されているコマンドハンドルを実行する	コマンドハンドル 繰り返し数、引数値リスト
	R-DropDML	蓄積されているコマンドハンドルを無効にする	コマンドハンドル

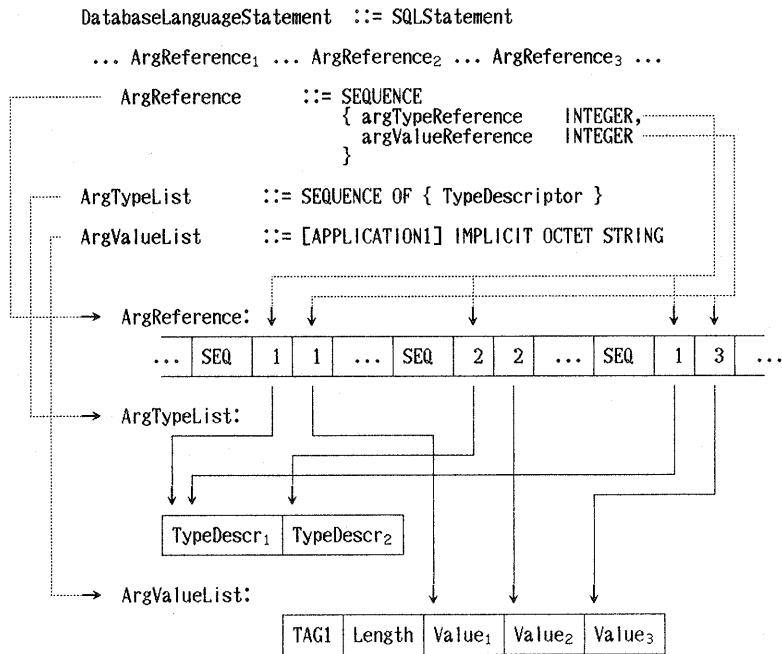


図2 引数参照 (ArgReference) の使用例

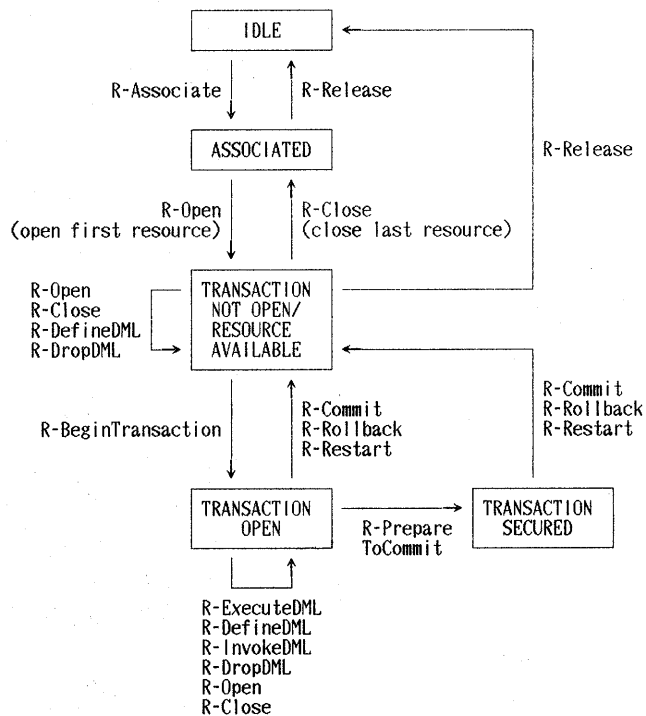


図3 コミットメントレベル2及び3のRDAサービス状態遷移<sup>1)</sup>