

OSI CCRの実装

小花貞夫 杉山敬三 西山智

国際電信電話株式会社

筆者らは、OSI(開放型システム間相互接続)のCCR(コミットメント,同時性及び回復制御)プロトコルのソフトウェアを実装した。本ソフトウェアでは、応用層構造(ALS)に基づいたソフトウェア構成、CCRユーザとCCR ASEの明確な機能の分担及び障害への柔軟な対応を実現している。本稿では、まず実装の基本的な考え方を議論し、次いでこの結果に基づいて実装したCCR ASEのソフトウェア構成および仕組み、さらにRDAトランザクション機能を実現するCCRユーザについて詳細に報告し、その機能等について考察している。

Implementation of OSI CCR

Sadao OBANA Keizo SUGIYAMA Satoshi NISHIYAMA
KDD R & D Labs. 2-1-15, Ohara, Kamifukuoka-shi, Saitama, 356

Software for CCR (Commitment, Concurrency and Recovery) Protocol of OSI (Open Systems Interconnection) has been implemented. It has an architecture based on Application Layer Structure (ALS), and achieves the clear functional distinction between CCR service-user and CCR ASE, and provides powerful function for recovery from failures such as communication failure and application failure. In this paper, design strategies for the software are firstly discussed. Then detailed specifications of the software for CCR ASE and CCR service-user, which realizes RDA (Remote Database Access) transaction, are discussed. Furthermore, facilities of the software are evaluated.

1.はじめに

近年、OSI(開放型システム間相互接続)の標準化が進捗し、応用層においてもMHS(メッセージ通信処理システム)やFTAM(ファイル転送、アクセスと管理)などの各種の標準が出揃いはじめ、応用も多様化してきている。このような応用のうち、分散処理の性格が強い応用を実現する上で重要なASE(応用サービス要素)にCCR^[1](コミットメント、同時性及び回復制御)がある。CCRは、分散された処理プロセスの同期をとるための機能であり、応用層で共通的に使用されるプロトコルの一つである。現在、CCRはRDA(遠隔データベースアクセス)やTP(分散トランザクション処理)などで使用されている。

筆者らは、RDAトランザクション機能^[2]を実現するためCCRを実装した。CCRは、RDAを用いた分散データベース・アクセスを実現するのに不可欠であり、今後分散処理を指向した応用を開発する場合にも基礎となるプロトコルである。本稿では、CCRのためのソフトウェア構成やCCRユーザとCCR ASEの機能分担及び障害への対応などCCR実装のための基本検討と、それに基づいたCCR ASEとCCRユーザの実装結果の詳細について論じる。

2. OSI CCRの概要^[1]

CCRは、複数のシステムにあるファイルやデータベースを同時に変更する際に、システム間の処理の同期をとるためのプロトコルであり、すべてのシステムで変更が正しく実行されたり(コミットメント制御)、変更の途中で他の利用者からのアクセスがあっても資源の一貫性を保証したり(同時性制御)、障害が発生しても正しい処理の進行を保証する(回復制御)ためのメカニズムを提供している。

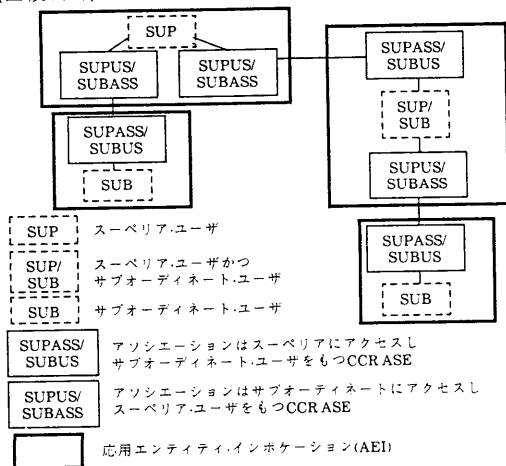


図1 CCRにおける原子動作木

CCRを用いる応用では、一つのまとまった処理の単位を原子動作(アトミック・アクション)と呼ぶ。原子動作では、協調動作する各AEI(応用エンティティ・インボケーション)は、原子動作木と呼ばれる木構造の形で、他のAEIと関係を持つ。原子動作木の木の枝の上位をスーパーリア、下位をサブオーディネートと呼ぶ。また最上位はマスタと呼ばれ、原子動作木における処理は基本的にマスタにより制御される。ただし、CCR自体はスーパーリアとサブオーディネート間の1対1のプロトコルである。原子動作木とCCR ASE及びCCRユーザの関係を図1に示す。

CCRでは、原子動作のコミットメント制御を行うため、2相コミットメント・プロトコルを用いる。2相コミットメントでは、図2に示すように、原子動作の処理が終了したときに、すべてのノードに処理が正常に終了したかどうかを尋ね、すべてが肯定応答ならコミットメント、一つでも否定応答が含まれるならロールバックの指示を出して、すべてのノードが同一の決定を行う。

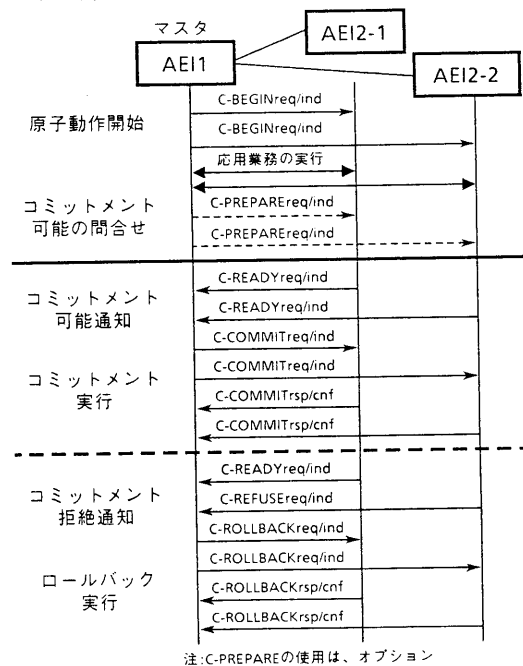


図2 2相コミットメント

また、各ノードでは、応用の処理により自システム内の資源にアクセスし、これらの資源に変化をもたらす。そのため、同一資源にアクセスする複数の原子動作に対しては、ロックなどの手法を用いて原子動作を単位として同時性が制御される。

さらに、原子動作に関与するノードでは、障害が発生しても、原子動作を維持しなければならない。

CCRでは、障害から回復した後にも正常に利用できる資源を保証データ(セキュアデータ)と呼んでおり、原子動作の状態や制御情報である原子動作データ(アトミック・アクション・データ)及び制御対象データ(バウンド・データ)を保証データにする必要がある。この保証データとCCRの再始動メカニズム(C-RESTART)を用いることで、障害から正しく復旧することができる。CCRで対応している障害は、アソシエーションの切断等の通信障害と、CCRの動作するAEIのプロセスの障害である応用エンティティ障害である。

3.基本検討^[3]

CCRの実装では、ソフトウェア構成、CCRユーザとCCR ASEの機能分担及び障害への対応等が重要である。

3.1 ALSに基づくソフトウェア構成

ASEの部品化や今後の他のASEの効率的な実装を考慮し、応用層構造(ALS)に基づいたソフトウェア構成^[4]にする必要がある。

(1)AEIの構成^[5]

プロセス間通信のオーバヘッドやASEモジュールの独立性等を考慮し、応用に共通なASEとSACF(単一アソシエーション制御機能)に対応するモジュールは1つのプロセスとし、応用に固有なASEは別プロセスとして実現することとする。ユーザプロセスやプレゼンテーションプロセスとAEIプロセス間のプリミティブの受信や発行は、プロセス間通信を用いて行う。また、SACFに対応するモジュールを明示的に導入し、複数のSAO(単一アソシエーション・オブジェクト)とそれに伴う個々のアソシエーションを管理する機能や、AEタイトルからプレゼンテーションアドレスを導出する応用ディレクトリ機能をSACFに持たせる。図3に、筆者らの提案するALSの実現方法に基づくAEIのソフトウェア構成を示す。

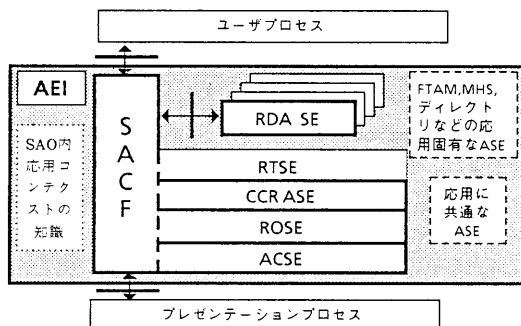


図3 AEIの構成

(2)応用コンテキストの管理

SAOに含まれる各ASE間の相互動作規則や下位サービスマッピング等の応用コンテキストの管理は、SACFで行うこととする。また、CCR ASEはスーパーリアとサブオーディネートの状態遷移が異なるため、SACFは各SAOがどちらの役割で動作するかをCCR ASEに通知することとする。

3.2 CCRユーザとCCR ASEの機能分担

CCRの標準では、CCRユーザとCCR ASEの機能分担が明確ではないので、明確化する必要がある。

(1)スーパーリア及びサブオーディネートの調停

CCRユーザには、単一アソシエーション上でCCRサービスを利用して応用の処理を実行するスーパーリア(スーパーリア・ユーザ)及びサブオーディネート(サブオーディネート・ユーザ)の役割を果たす機能が存在する。また、原子動作木のマスタや中間ノードのように、ひとつのAEI内に複数のSAOが存在する場合、その間のプリミティブの伝搬やコミットメントの調停等を行うため、スーパーリア及びサブオーディネートの調停機能が必要となる。このためCCRのプロトコ仕様^[1]には、AEIにおける複数のSAO全体としての状態を示すコミットメント状態変数が定義されている。しかし、CCRのプロトコルは単一アソシエーション上のプロトコルを規定したものであるため、複数アソシエーションに関するコミットメント状態変数をCCR ASEのモジュール内部に持たせるのは好ましくない。従って、スーパーリア及びサブオーディネートの調停を行うのはCCRユーザの役割とし、コミットメント状態変数はCCRユーザに原子動作データとして持たせて、コミットメント状態変数の設定やアクションの決定を行わせることとする。また、単一アソシエーションに関する原子動作データ、例えばCCR ASEの状態等は、CCR ASEが管理することとする。

(2)ローカル資源の管理

ローカル資源の管理方法は応用や実装方法によって異なるため、この機能をCCR ASEやCCR ASEと組み合わせて使用する応用固有のASEに持たせることは、ASEの部品化の観点から好ましくない。そこで、ローカル資源の管理はユーザプロセスに行わせ、応用固有のASEで扱うデータである制御対象データはユーザプロセスが管理することとする。

以上から、CCRユーザとCCR ASEの機能分担をまとめたものを表1に示す。

3.3 障害への対応

障害回復の実現方法等については標準で規定されていないため、詳細に検討する必要がある。

表1 CCRユーザとCCR ASEの機能分担

CCRユーザ	スーパーリア・ユーザ	ローカル資源の管理 (制御対象データの管理)
	サブオーディネート・ユーザ	
CCR ASE	スーパーリア及びサブオーディネートの調停 (複数アソシエーションに関する原子動作データの管理)	
	単一アソシエーション上のプロトコルの実行 (単一アソシエーションに関する原子動作データの管理)	

(1)通信障害

CCR ASEは、通信障害後にアソシエーションが再確立されたことにより、状態遷移を変えて動作を継続するので、アソシエーションの状態を知る必要がある。そこで、SACFが必要に応じてアソシエーションの状態をCCR ASEに知らせるようにする。また、CCR ASEは、アソシエーション切断時にアソシエーション及びCCR ASEに関する情報(すなわち原子動作データ)を退避し、再確立後に退避された原子動作データと再確立されたアソシエーションを関連付けるようにする。これは、再確立前と再確立後における原子動作識別子及び分岐識別子を対応させることで行う。

また、通信障害によりアソシエーションが切断されてもAEIは存続しているため、スーパーリア・ユーザはアソシエーションを再確立する時に相手のAEIを一意に識別しなければならない。従って、スーパーリア・ユーザは、A-ASSOCIATE要求プリミティブの利用者オプションのパラメタのうち、プレゼンテーションアドレス/AEタイトル(APタイトルとAE修飾子)/AEI識別子/API識別子を指定し、切断前のアソシエーションの応答側のこれらの値をそれぞれ再確立時の着呼側のパラメタに設定することとする。図4に、これらのパラメタを使用したアソシエーションの再確立に関する概念を示す。

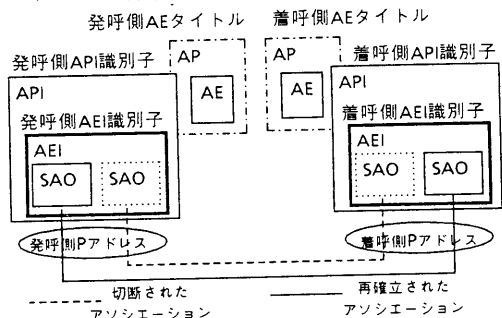


図4 アソシエーションの再確立

(2)応用エンティティ障害

応用エンティティ障害の場合には、AEIのプロセスを再起動する必要が生じるので、CCR ASEは原子

動作中に原子動作データを二次記憶に退避し、プロセスが再起動された時にそのデータを読み出すものとする。そして、通信障害と同様、原子動作識別子と分岐識別子を比較することにより、再確立されたアソシエーションと原子動作データを関連付ける。

また、スーパーリア・ユーザは、障害発生以前に確立されていたすべてのアソシエーションを再確立する。この際、AEIは新しく起動されるので、A-ASSOCIATE要求プリミティブのパラメタのうち、プレゼンテーションアドレスとAEタイトルは障害発生以前と同じ値に設定しなければならないが、AEI識別子とAPI識別子は同じ値を設定する必要はない。しかし、スーパーリア・ユーザは、サブオーディネート・ユーザで発生した障害が通信障害なのか応用エンティティ障害なのか知ることはできないので、AEI識別子とAPI識別子についても同じ値に設定することとする。また、障害から回復不能の時には、CCRユーザが管理している資源をロールバックし、他のユーザに資源を解放するようにする。

4.実装の基本方針

今回のCCRの実装においては、以下のような方針をたてた。

(1)CCRプロトコルとして、2ndDIS^[11]に基づき、通常の2相コミットメントや再始動などを実装する。Presumed Abortや、ヒューリスティック判断などの拡張機能については、今回対象外とする。今回実装の対象とするCCRプロトコルに対応するサーベスプリミティブとパラメタの一覧を表2に示す。

表2 実装するCCR機能

プリミティブ	パラメタ
C-BEGIN	原子動作識別子,分岐識別子,原子動作タイム,利用者データ
C-PREPARE	利用者データ
C-READY	利用者データ
C-REFUSE	利用者データ
C-COMMIT	
C-ROLLBACK	
C-RESTART	再開点(要求指示ではACTION, COMMIT, ROLLBACK, 応答確認ではDONE, RETRY-LATER, REFUSE, ACTION), 原子動作識別子,分岐識別子,再開タイム,利用者データ

(2)CCRユーザとしてRDAを実現し、RDAトランザクション機能をCCRコンテキスト^[7]により実装する。今回、RDAサーバ側のデータベース機能実現のために汎用のRDB(関係型データベース管理システム)であるORACLEを使用する。また、ローカルな資源の管理機能は、RDBが提供する資源管理機能と対応をとることで実現する。

(3)VAX(OS:VMS)上にソフトウェアを作成し、移植性を考慮してC言語でプログラミングする。AEIプ

プロセスとユーザプロセス間等のプロセス間通信にはVMSの提供するメールボックス機能を用いる。

5. CCR ASEの実装

5.1 CCR ASEのソフトウェア構成

CCR ASEは、図5のように、マネージャ、アソシエーション処理モジュール、プリミティブ解析モジュール、プロトコル処理モジュールから構成される。これらのモジュールの間で共有される情報は、CCR管理テーブル(5.2節参照)にセットされる。

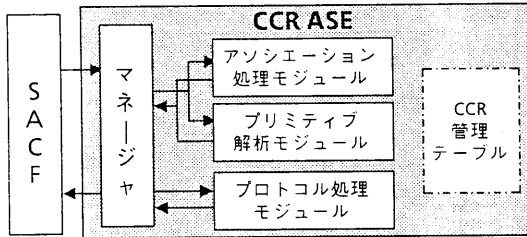


図5 CCR ASEのソフトウェア構成

① マネージャ

SACFから受け取ったイベントの解析を行い、イベントがアソシエーションの確立・解放または切断ならばアソシエーション処理モジュールを、CCRプリミティブやプレゼンテーションプリミティブの受信ならばプリミティブ解析モジュールを呼び出す。その後、プリミティブの受信または原子動作中のアソシエーション再確立の時は、プロトコル処理モジュールを呼び出す。

② アソシエーション処理モジュール

アソシエーション確立時及び解放時にCCR管理テーブルの作成及び削除を行う。また、通信障害発生時にCCR管理テーブルのCCR ASE内のリストへの退避を行ったり、CCR管理テーブルにプロトコル処理モジュールに対する障害発生イベント(CRASH/FAILURE)の設定を行う。

③ プリミティブ解析モジュール

受信したプリミティブやPDU(プロトコルデータ単位)の解析を行って、CCR管理テーブルに必要な情報やプロトコル処理モジュールに対するイベントを設定する。なお、このモジュールでは、連続した複数のCCRプリミティブを受信したらPDUの連結要求と判断し、プロトコル処理モジュールに対して対応するイベント、例えばC-COMMITreq+C-BEGINreqの受信イベントを生成する。

④ プロトコル処理モジュール

プロトコル処理モジュールに対する入力イベントと現在の状態から状態遷移を行うとともに、PDUやプリミティブの作成を行う。作成したプリミティブ

は、ユーザプロセスやプレゼンテーションプロセスへの発行をSACFに委託する。さらに、応用エンティティ障害に対処するため、原子動作中にCCR管理テーブルを二次記憶へ退避し、障害回復時にはCCR管理テーブルと再確立されたアソシエーションの関連付けを行う。

5.2 SACFとCCR ASEにおける管理情報

(1)SACFでは、アソシエーション確立時にそのアソシエーションに関する情報を格納するためのテーブル(コネクション情報テーブル)を作成する。コネクション情報テーブルは、アプリケーションに関する知識やアソシエーションを管理するための識別子等の情報及び各ASEの扱うテーブルに対するポインタの情報等を持っており、アソシエーション解放時に削除される。各ASEの扱うテーブルがコネクション情報テーブルに接続されることにより、各ASEは活性化されSACFの管理下に置かれる。また、コネクション情報テーブル全体を管理するテーブル(コネクション情報管理テーブル)があり、コネクション情報テーブルのリストや送受信するプリミティブ等を管理する(図6)。このテーブルは、プロセス起動時に生成される。

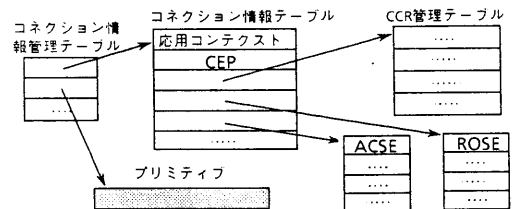


図6 SACFとCCRのテーブルの関係

(2)CCR ASEでは、CCRプロトコル処理モジュールの状態やプリミティブ作成に必要な情報及び障害からの回復に必要なアソシエーションに関する情報を格納するテーブル(CCR管理テーブル)を持つ。CCR管理テーブルは、アソシエーションが確立された時に作成され、原子動作後にアソシエーションが解放された時や障害回復後不要になった時に削除される。CCR管理テーブルへのポインタは、図6に示すように、SACFのコネクション情報テーブルに格納されているので、対応するアソシエーションに関してCCR ASEで必要となる情報は、SACFから渡されたコネクション情報管理テーブルへのポインタをたどることで得られる。表2にCCR管理テーブルの主な内容を示す。

5.3 障害の処理

(1)通信障害

通信障害により原子動作中にアソシエーションが切断された場合、コネクション情報テーブルが解放

表2 CCR管理テーブルの主な内容

他のCCR管理テーブルへのポインタ
アソシエーション識別子
PCEP識別子
プレゼンテーションコンテキスト識別
同期点通し番号
トークン
スーベリア/サブオーディネート
状態*
イベント
原子動作識別子*
分岐識別子*
原子動作タイム
再開点
再開タイム
利用者データ

注：*は原子動作データの項目

されるので、アソシエーションが再確立されるまで対応するCCR管理テーブルの情報を保存する必要がある。そこで、SACFからアソシエーション切断のイベントを受け取ると、コネクション情報テーブルに接続されているCCR管理テーブルをCCR ASE内部のリスト(アソシエーション再確立待ちリスト)の一番最後に接続してSACFの管理から外し、プロトコル処理モジュールに対する入力イベントとしてFAILUREを設定する(図7)。

アソシエーションが再確立されると、新しいCCR管理テーブルが作成され、コネクション情報テーブルに接続される。その後、C-RESTARTが交換される時に、CCR管理テーブルの切断時と再確立時におけるアソシエーション情報の対応をとる。これは、アソシエーション再確立待ちリストに接続されたCCR管理テーブルとコネクション情報テーブルに接続されているCCR管理テーブルに格納されている原子動作識別子及び分岐識別子をそれぞれ対応させることで行う。つまり、C-BEGINのパラメタとして設定された原子動作識別子及び分岐識別子と、C-RESTARTのパラメタとして設定された原子動作識別子及び分岐識別子と比較する。原子動作識別子はマスタのAEタイトルとサフィックス、分岐識別子はスーベリアのAEタイトルとサフィックスから構成されており、これらが一致するテーブルを発見すると、コネクション情報テーブルに接続されているCCR管理テーブルに、障害発生前のプロトコル処理モジュールの状態等の必要な情報をコピーする。これにより、引続き原子動作が実行可能とな

る。そして、アソシエーション再確立待ちリストからそのCCR管理テーブルを外して削除する。

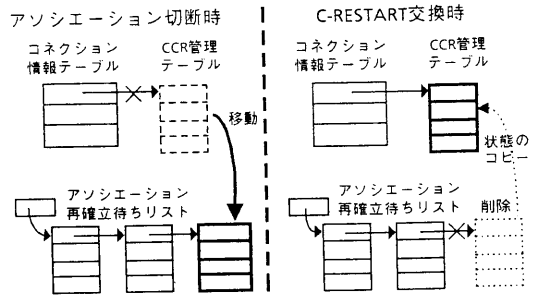


図7 CCR管理テーブルの操作

(2) 応用エンティティ障害

今回の実装では、CCR ASEにおける原子動作データとしてCCR管理テーブルを用い、これをディスク上のファイルとして保存する。このファイルは、図8のようなタイミングで、作成、更新及び削除を行う。応用エンティティ障害の場合、AEIのプロセスを再起動するが、初めて起動されたのか再起動されたのかを判断する必要がある。このため、AEIプロセスの起動後でアソシエーションが最初に確立された時に原子動作データが二次記憶に存在していれば、原子動作中にAEIが異常終了した(すなわちAEIは再起動された)と判断することとした。そして、ファイルから読出した情報をCCR管理テーブルに設定し、プロトコル処理モジュールに対する入力イベントとしてCRASHを設定して、アソシエーション再確立待ちリストに接続する。その後、C-RESTARTが交換されると、通信障害の時と同様に、原子動作識別子及び分岐識別子が一致するテーブルを検索し、コネクション情報テーブルに接続されているCCR管理テーブルに必要な情報をコピーして、アソシエーション再確立待ちリストの対応するテーブルを削除する。

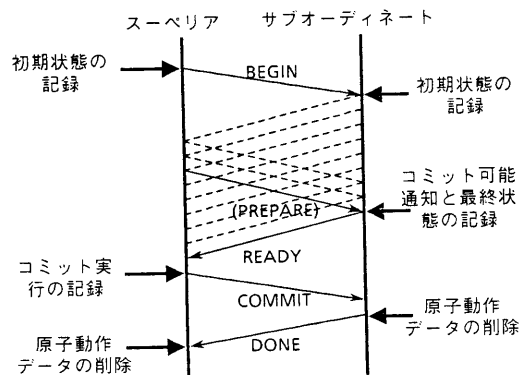


図8 原子動作データの更新

6. CCRユーザの実装(CCRを用いたRDAトランザクション機能の実現)

CCRユーザの機能を実装し、CCR ASEの有効性を実証するため、CCRを用いてRDAトランザクション機能⁶⁾を実現した。

5.1 RDAトランザクション⁷⁾

RDAのトランザクションは、クライアントにより決定される論理的な一つの処理単位であり、遠隔のデータベース処理の一貫性を保証するために使用される。RDAでは、トランザクション機能を実現するため、①基本コンテキスト、②CCRコンテキスト、③TPコンテキスト、の3つの応用コンテキストを定義している。

5.2 RDAトランザクション機能の実現

今回、CCRコンテキストでRDAトランザクション機能を実装した⁸⁾。この場合、応用のユーザが直接CCRのプリミティブを発行する。図9にRDAトランザクション機能の構成を示す。

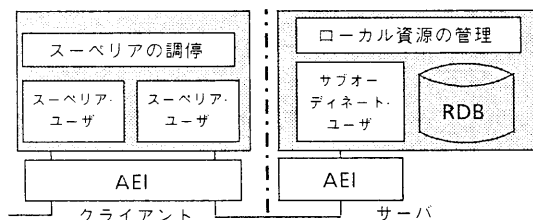


図9 RDAトランザクション機能の構成

(1)スーパーリア・ユーザ

スーパーリアとしてCCRプリミティブの作成/解析及びRDAクライアントとしてRDAプリミティブの作成/解析を行う。今回の実装では、トランザクション開始時に使用する資源を確定するため、ユーザはトランザクションの開始を明示し、同時に資源名を指定することとした。これにより決定されたサイトに対しC-BEGINを発行する。また、RDAではトランザクション中にユーザが複数のSQL文を入力できるため、ユーザはトランザクションの終了を明示し、その際にC-PREPAREを必ず発行することとした。さらに、障害回復後には、原子動作データからC-RESTARTやRDAのプリミティブを発行し、障害回復直前の状態に復旧することとした。

(2)サブオーディネート・ユーザ

サブオーディネートとしてCCRプリミティブの作成/解析及びRDAサーバとしてRDAプリミティブの作成/解析を行う。また、受信したプリミティブ

の種類をローカル資源の管理部に通知し、RDBに対するコマンドの発行等の必要な処理を委託する。

(3)スーパーリアの調停

RDAクライアント側で、複数のスーパーリア・ユーザ間の調停を行い、原子動作データを管理する。ここでは、原子動作識別子、分岐識別子、AEタイトル、API識別子、AEI識別子、コミットメント状態変数及びユーザが発行したRDAプリミティブのログを原子動作データとし、ファイルに格納する。なお、原子動作データは、トランザクション終了時に削除する。

(4)ローカル資源の管理

RDAサーバ側のローカル資源の管理は、RDBが提供する資源管理機能と対応をとることで以下のように実現する。なお、制御対象データはRDBで保証することとした。

1)コミットメント制御

C-COMMITまたはC-ROLLBACKの受信時に、RDBに対しコミットまたはロールバックコマンド(ORACLEでは各々COMMITとROLLBACK)を発行する。通常RDBでは、1つのSQL文を実行するたびにコミットする自動コミットモードで動作している。しかし、ここでは複数のSQL文を1トランザクションとして扱うため、C-COMMITかC-ROLLBACKが発行されるまで実際のデータベースの更新を行わないよう、トランザクションの開始時に自動コミットモードを解除するコマンド(ORACLEではSET AUTOCOMMIT OFF)を発行し、終了時に自動コミットモードを設定するコマンド(ORACLEではSET AUTOCOMMIT ON)を発行する。

2)同時性制御

RDAユーザ間の同時性制御は、RDAサーバ(つまりサブオーディネート・ユーザ)内で行うが、RDAユーザによるデータベースアクセスとローカルユーザによるデータベースアクセスが混在する可能性があるため、RDAのユーザとローカルユーザとの間での資源の同時性制御も管理する必要がある。そこで、RDAのトランザクション中に使用する資源に対しRDBに排他ロックコマンド(ORACLEではLOCK TABLE IN EXCLUSIVE MODE)を発行することとした。

3)回復制御

コミットメントの第1相で発生した通信障害及び応用エンティティ障害の場合、障害回復後にRDBにロールバックコマンドを発行し、資源をトランザ

クシヨソ開始直後の状態にする。第2相で発生した障害の場合にはRDBに対し通常コマンドは発行しないが、回復不能の時には資源をロールバックさせる。図10に、コミットメントの第1相で障害が発生した後に回復し正常終了した場合のトランザクション処理のシーケンス例を示す。

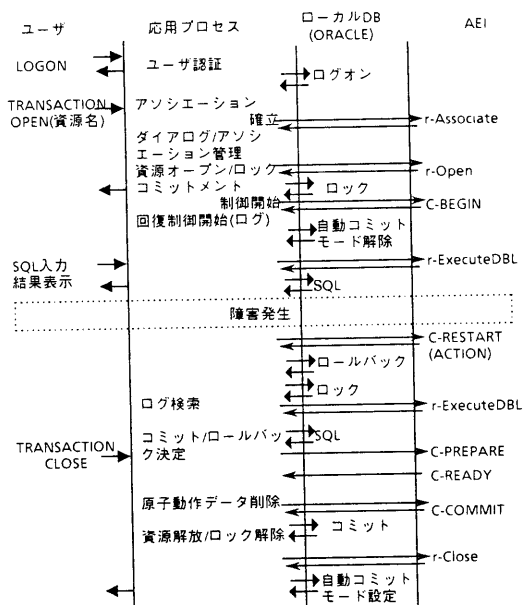


図7 トランザクション処理のシーケンス例

7.考察

- (1)CCR ASE及びRDAトランザクション機能を実現するCCRユーザを実装し、正常に動作することを確認した。
- (2)ALSに基づいたソフトウェア構成とし、SACFに対応するモジュールを明示的に導入してこのモジュールが応用コンテキストを制御するようにした。これにより、各ASEが応用コンテキストの管理を行う必要がなく、ASEモジュールの部品化が可能となった。
- (3)CCRの標準で不明確であったCCR ASEとCCRユーザの機能の分担を明確にし、CCRユーザにスーパーリア及びサブオーディネートの調停とローカルな資源の管理機能を持たせ、CCR ASEは単一アソシエーション上のプロトコルの実行及び原子動作データの管理のみを行うことにより、汎用的なCCR ASEが構築できた。
- (4)通信障害に対しては、CCR管理テーブルをリスト構造で退避することで、アソシエーション再確立

時の効率よい復旧を可能とした。また、応用エンティティ障害に対しては、CCR管理テーブルを原子動作データとしてファイルに書き出し、これをAEIプロセス再起動時に読み出してリストに接続することにより、障害からの効率よい復旧を可能とした。

- (5)CCRではアソシエーションの再確立はユーザに任せているが、C-RESTARTを発行する時にA-ASSOCIATEで使ったAEタイトルが必要となったり、通信障害後のアソシエーション再確立時にAEIを指定する必要があるため、CCRで必要となるACSEのパラメタを明確に規定する等の標準の拡充が望まれる。
- (6)CCRユーザのローカルな資源の管理機能をRDBの機能にマッピングすることにより、RDAトランザクション機能を効率よく実現できた。しかしながら、CCRユーザの実装は、実際の応用やシステムに依存するところが大きい。例えば、コミットメント制御については、RDAの場合ローカルなRDBが提供するコミットメント機能とほぼ対応がとれるが、FTAMの場合実ファイルシステムでは提供していないので、新たに機能を作成する必要がある。

8.おわりに

本稿では、筆者らの実装したOSI CCRのソフトウェアについて報告した。本実装では、ALSに基づいたソフトウェア構成とし、標準で明確に規定されていないCCRユーザとCCR ASEの機能分担や障害への対応方法を明確にした。今後は、Presumed Abortに基づいたCCRやCCRを用いたTPを実装するための検討を進める予定である。最後に日頃御指導頂くKDD上福岡研究所小野所長、浦野次長、鈴木コンピュータ通信研究室長に感謝します。

参考文献

- [1] ISO/IEC DIS 9804.2/9805.2, CCR 2nd DIS, 1988
- [2] 小花、西山、杉山、「OSI RDAを用いた分散データベースアクセスの実現(1)-全体構成-」第38回情処全大
- [3] 杉山、西山、小花、「OSI CCR実装の基本設計」第37回情処全大
- [4] 小花、西山、杉山、鈴木、「OSIにおける応用層構造(ALS)の実現方法の提案」第37回情処全大
- [5] 杉山、西山、小花、「OSIの応用層構造(ALS)におけるSACFの実装」第39回情処全大発表予定
- [6] 杉山、西山、小花、「OSI CCRを用いたRDAトランザクション機能の実現方法の提案」第38回情処全大
- [7] ISO/IEC JTC1/SC21/WG3 Editor's Working Draft for 2nd DP, Sep.,1988
- [8] 西山、杉山、小花、「OSI RDAの実装」情処学会マルチメディア通信と分散処理研究会、43-2,1989