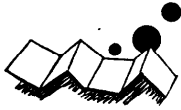


解 説

5. 標準化動向



5.4 トランザクション処理の標準化動向†

佐 藤 健††

1. はじめに

OSI (Open Systems Interconnection) の標準化作業は、応用層のプロトコル規格の開発に焦点が移り、OSI サポートの製品開発のための検討も、ここ 1~2 年急速に進展しつつある。

応用層プロトコルの規格としては、今まで FTAM (File Transfer, Access and Management), VT (Virtual Terminal) 基本クラス, JTM (Job Transfer and Manipulation) などが開発され、現在 DIS (Draft International Standard) の段階にある。

応用層の国際標準規格の中で、現在ユーザのニーズの高いものの一つに、トランザクション処理用の通信プロトコルがある。

ここでは、ISO/TC 97/SC 21/WG 5 (特定応用サービス) で検討を進めているトランザクション処理用の応用層プロトコルに関して、その概要、標準化状況及び今後の課題などについて述べる。

2. トランザクション処理標準化の概要とその経緯

(1) トランザクション処理について

ネットワーク処理の対象となる業務の処理形態において、一般にトランザクション処理といわれる業務の割合は、7~8割を占めるといわれている。

「トランザクション処理」と一口にいっても、そのスコープは広く、ISO の場でもその定義について検討している。

トランザクション処理プロトコルは、一般にプロセス・ツー・プロセスの接続において、比較的短いデータの交換を行うためのものとして利用される。したがって、ファイル転送、ジョブ転送、電子メールなど多量のデータを一方向に転送するものとは、その必要と

する機能も異なってくる。

データベースの参照、更新などに使用されるリモート・データベース・アクセス・プロトコル (RDA) があるが、これとはトランザクションのコミット機能などにおいて共通な機能をもっている。

ISO におけるトランザクション処理プロトコルの標準化作業は、TC 97/SC 21/WG 5 に割り当てられており、現在緊急なテーマとして検討中である。

TC 97 に提案された標準化の作業項目のタイトルは「Transaction-Mode Application Service Element and Protocol for OSI」となっており、現在 TP と略称されている。ここではこれをトランザクション処理プロトコルと呼ぶ。

トランザクション処理プロトコルは、応用層の特定目的向けプロトコルの一つとして位置付けられているが、特定目的向けというよりは、汎用性の高い応用層プロトコルとして利用されることになると思われる。

(2) トランザクション処理プロトコル標準化の必要性

ファイル転送、ジョブ転送、電子メールなどのプロトコルがあっても、問い合わせ/応答が自由にできないため、トランザクションの処理を行うには不都合がある。また、複数のシステムに関連した処理において、トランザクションの同期性を保証するための機能が必要になる。

ユーザのニーズとしては、たとえばファイル転送を行うと同時に、問い合わせ/応答の処理を行いたいという形態が多い。たとえば、全銀協ベーシック手順を使用して、1本の回線でこれら二つのことが同時に実現可能かという不可能である。

OSI プロトコルを使用した場合、同一回線上でファイル転送とトランザクション処理は、同時に使用可能となる。ファイル転送は、OSI の FTAM を使用できるが、トランザクション処理プロトコルはない。この標準化が待ち望まれているわけである。

トランザクション処理用のプロトコルとしては、各

† Standardization Trend of Transaction Processing by Takeshi SATO (Software Works, Hitachi, Ltd.).

†† (株)日立製作所ソフトウェア工場

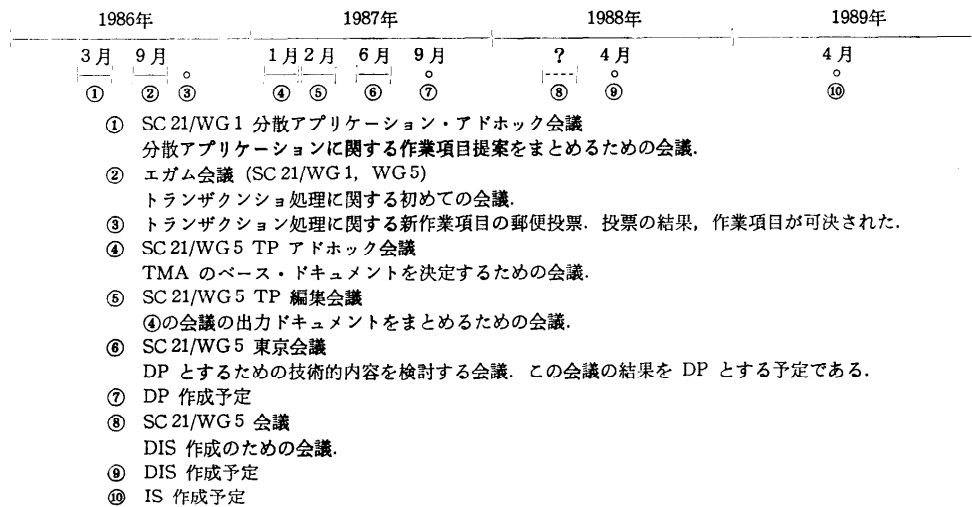


図-1 トランザクション処理に関する国際会議などの経過及び予定

メーカ固有のもの、業界で取り決めたものなどがあるが、マルチベンダ・システムにおいて、このプロトコルをどのように設定するかは、ユーザのオンラインシステム構築上における大きな問題の一つであった。

プロトコルの標準化は、情報処理システムの相互運用性（インターオペラビリティ）の確保の一つの重要な手段であるが、これさえもできないのが現状である。

高度情報化社会の基盤技術として、OSI の推進は大きな意味があり、その中で利用分野の広いトランザクション処理プロトコルの標準化は、重要な意味をもつ。

(3) トランザクション処理プロトコル検討の経緯

分散処理の分野における標準化の必要性は、ISO においても早くから認識されており、1985年2月の SC 21 パリ会議において話題となり、引き続いて行われた TC 97 会議へ提案された。

その後 SC 21 において検討を進め、次のことが合意された。

- 分散処理分野における標準化のためのアーキテクチャ・フレームワークをまとめることが必要である。
- トランザクション処理に関して、早急にプロトコルをまとめることが必要である。

前者は、ODP (Open Distributed Processing: 開放型分散処理) 基本参照モデルを開発すべく、作業項目として提案され、郵便投票中である。この作業は、分散処理の全体にかかわるフレームを設定するものであり、1989年に DP (Draft Proposal) 作成を目標と

している。

後者は、作業項目の提案が合意され、現在 SC 21/WG 5 にて検討が進められている TP プロトコルである。

トランザクション処理プロトコルの作業項目提案は、1986年3月の SC 21/WG 1 アドホック会議においてまとめられ、フランスより TC 97 へ提案され、郵便投票が行われた³⁾。

トランザクション処理に関する国際会議の、今までの経過及び今後の予定は、おおむね 図-1 に示すとおりである。トランザクション処理プロトコルの開発スケジュールは、DP (1987年9月)、DIS (1988年4月)、IS (1989年4月) と設定されている。

トランザクション処理に関する標準化の作業項目の郵便投票の結果は、Pメンバのうち投票を行った14カ国すべてが賛成であった。(2カ国が内容に関して反対)

今後の開発スケジュールも、図-1 に示したように今までは例をみないスピードで標準化を進めることを目標としている。

3. トランザクション処理の機能要件

トランザクション処理サービス及びプロトコルに必要な機能として、どのようなものがあるか。ISO の作業項目の提案書で示された機能要件 (Functional Requirement) としては、次のものがある。

- (a) 複数の開放型エンドシステムを含むトランザクションのための通信機能

- (b) 高い負荷, 低い負荷, またはバースト状態での負荷条件において, トランザクションを効率的に通信する機能
- (c) 短いデータの効率的な処理
- (d) 複数の開放型エンドシステムが, 効率的, 効果的にタスクを同期する手段
- (e) リソースの相互一貫性を保証するのに, コミットメント機能のすべてを必要としないトランザクションの効率的サポート
- (f) トランザクション処理のユーザに対する適切な応答時間
- (g) リソースの最適な使用
- (h) 障害に対処するための機能

これらの機能要件を考慮して, サービス及びプロトコルの開発を行うこととしている。

現在, 検討しているのは, コネクション・オリエンテッドな通信である。ある利用分野においては, コネクションレス通信も有効な方法と考えられ, 今後の検討課題である。

4. 国際における標準化状況

トランザクション処理プロトコルに関する実質的な技術検討が開始されたのは, 1986年9月のエガム(イギリス)会議である。

この会議においては, まだ標準化の作業項目が正式に郵便投票にかけられておらず, 各国の意見の説明及びそれらから抽出された問題点の整理を行った。WG5会議では, チュートリアル・セッションという位置付けで, 米国, ECMA, 西ドイツ, 米国エキスパート(ベルコア社)の4件のドキュメントに基づいた説明が行われた。

主要な提案としては, 一つは ECMA の OSI TP モデルがあり, これはその中心に ROS を置き, またコミットメント制御の単位がトランザクションであるとしている。

エガム会議の結果は, 以上の状況であり, まだ方向付けは見えていなかった。

続く会議は, 1987年1月にパリで行われた WG5 TP アドホック会議である。

この会議への出席者は, 合計 28 名であり, その数から見ても, 各国のこの標準化への関心度がわかる。

各国の提案ドキュメントの主要なものとして, 次のものがある。

米国は, 前回のベルコア社案をベースとしてサー

ビス定義案を提案した。この案の主な特徴としては, CCR (Commitment, Concurrency and Recovery) を直接的に使用する方式, request/reply というサービスを提供するなどである。

フランスは, フランスの BULL 社, 西ドイツの SIEMENS 社, 及び米国の IBM 社の3社共同提案をベースとした案を提案した。

ECMA は, OSI TP の第3版を提出した。これは前回の案を改訂したものである。

この会議の結果は, おおむね次のとおりである。

(a) トランザクション処理モデルに関する主な結果

- トランザクションの定義は, CCR で扱う狭義の意味に限定しないこととする。
- アソシエーション (Association) は, アプリケーション・エンティティ (AE) 間の関係であり, ダイアログ (Dialogue) は, トランザクション処理サービス・ユーザ (TPSU) 間の関係である。

(b) トランザクション処理サービスに関する主な結果

- 機能ユニットの設定が行われた。半二重サービス及び全二重サービスが定義された。
- サービス定義のためのサービスプリミティブとして, 30種がリストアップされた。

今回の結果は, 必ずしもすべて合意されたものではなく, CCR など一部は, 未整理のまま記載されているものもある。

● 同期のレベル

同期のレベルとしては, CCR あり, 同期あり, どちらもなしの三つのレベルが設定された。CCR の使い方としては, CCR の複雑さをユーザに意識させない案(フランス)と, ユーザが CCR を直接使用する案(米国)があり, 一本化には至らなかった。

- トランザクション処理サービスと ROS の関係
この問題についての議論は進展せず, 今後の課題とされた。

(c) トランザクション処理プロトコルに関する主な結果

- 下位層へのマッピングについての案が作成された。今回の会議ではまだ十分な検討は行われていない。

5. 国際標準規格案の解説

5.1 トランザクション処理プロトコル

ここで示す内容は、1987年1月のアドホック会議の結果であるが、詳細内容についてはこれから検討されるので、変更される可能性がある。現在WD(Working Draft)である^{1),2)}。

(1) トランザクション処理のモデル

「トランザクション」とは、次の4つの特性をもつ仕事の単位 (a unit of work) と定義された⁶⁾⁻⁸⁾。

4つの特性とは、次のものである。これをACIDという。

- アトミシティ (Atomicity)
トランザクションは、全体として成功するか失敗するかである。
- コンシステンシー (Consistency)
成功したトランザクションは、データを新たなコンシステントな状態に移させる。
- アイソレーション (Isolation)
トランザクションの途中の効果を他のトランザクションから分離する。
- デュアラビリティ (Durability)
障害からの回復が可能である。

トランザクションの定義を、コミットメント機能の対象となるものだけとするか否かについては、この意味に限定しないものとした。

トランザクションは、複数の開放型システムにわたって処理されることがある。これを説明するためにトランザクション・ツリー概念が導入されている。複数システムにわたるトランザクションを、分散トランザクションと呼ぶ。

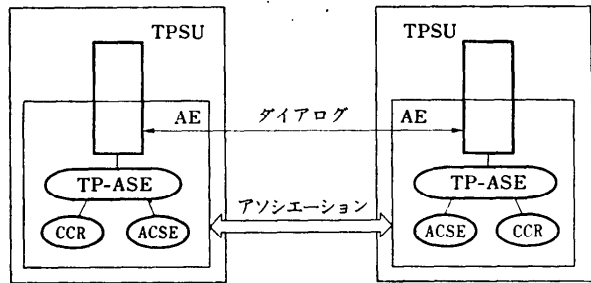
トランザクション処理プロトコルのモデルを、図-2に示す。

アソシエーションは、二つのAE間の関係であり、ダイアログは、二つのTPSU間の関係である。一つのAEには、複数のTPSUが存在できる。

一つのアソシエーション上では、一時には一つのダイアログが実行される。

(2) トランザクション処理のサービス

トランザクション処理のサービス機能を分類するものとして、機能単位が導入された。これは表-1に示すものであるが、アプリケーションの必要性に応じ



AE: Application Entity
 TPSU: Transaction Processing Service User
 TP-ASE: Transaction Processing-Application Service Element
 ACSE: Association Control Service Element
 CCR: Commitment, Concurrency and Recovery

図-2 トランザクション処理プロトコルのモデル

て、この機能単位を選択することができる。

サービス機能の概念を示すサービス・プリミティブとして、30種がリストアップされた。これを表-2に示す。

必要なパラメタについても記載されているが、今後詳細な検討が必要である。

同期処理のレベルとしては、次の三つが設定された。

- 同期なし—同期及びコミットメントは、使用しない。
- 同期—同期だけが使用できる。

表-1 トランザクション処理サービスの機能単位 (WDにおける案)

機能単位	説 明
カーネル	ダイアログの開始、終了、異常終了、データ転送、例外報告などの必須機能
全二重	トランザクション処理ユーザ間における両方向同時データ転送をサポートする機能
半二重	トランザクション処理ユーザ間における両方向交互データ転送をサポートする機能
単独 (Individual)	トランザクション処理のユーザに、直接 CCR サービスを提供して、分散システムでのデータ転送をコミットする機能
複合 (Combined) コミット	トランザクション処理のユーザに、CCR サービスを組み合わせた使用しやすい形のプリミティブを提供して、分散システムでのデータ転送をコミットする機能
自律 (Autonomous) トランザクション	"decouple", "inserted", "remote" と呼ぶ、通常のトランザクションとは独立なトランザクションをサポートする機能
同期 (Synchronization)	アソシエーションの障害などにおいて、回復を行うために同期点を取る機能
リスタート	以前に取られた同期点から、再開するための機能

表-2 トランザクション処理のサービス・プリミティブ (WD における案)

No.	機能単位	サービス・プリミティブ	タイプ*	説明
1	カーネル	TP-BEGIN-DIALOGUE	RI(RC)	TPSU 間のダイアログを確立する。
2		TP-END-DIALOGUE	RI(RC)	ダイアログを正常終了する。
3		TP-U-ABORT	RI	TPSU がダイアログを異常終了する。
4		TP-P-ABORT	II	TP サービス提供者がダイアログを異常終了する。
5		TP-DATA	RI	相手の TPSU にデータを転送する。
6		TP-U-EXCEPTION-REPORT	RI	TPSU が異常発生を相手 TPSU に通知する。
7		TP-P-EXCEPTION-REPORT	II	TP サービス提供者が異常発生を通知する。
8	半二重	TP-GIVE-TOKENS	RI	送信権 (トークン) を相手 TPSU へ譲渡する。
9		TP-PLEASE-TOKENS	RI	送信権 (トークン) を相手 TPSU へ要求する。
10	単独コミット	TP-BEGIN	RI	アトミックユニット**を開始する。
11		TP-PREPARE	RI	アトミックユニットの処理の終了を宣言し、コミットの可否の通知を要求する。
12		TP-READY	RI	コミット可能であることを通知する。
13		TP-REFUSE	RI	コミット不可であることを通知する。
14		TP-COMMIT	RIRC	コミットを指示する。また、コミットの完了を応答する。
15		TP-ROLLBACK	RIRC	アトミックユニットの破棄を指示する。また、その指示に対して応答する。
16	複合コミット	TP-CHAIN-COMMIT	RI	TPSU がマスタの場合に発行するコミット要求。
17		TP-CHAIN-COMMIT-AND-GIVE-TOKENS	I	TP-CHAIN-COMMIT に対して、送信権の譲渡を付加したもの。
18		TP-CHAIN-COMMIT-CONTINUE	R	TPSU がマスタでない場合に発行するコミット要求。
19		TP-CHAIN-COMMIT-COMplete	I	TP-CHAIN-COMMIT の確認を行う。
20		TP-CHAIN-COMMIT-AND-END	I	コミット後、ダイアログを終了する。
21		TP-PREPARE-TO-GIVE-TOKENS	R	送信権譲渡付の PREPARE 要求を行う。
22		TP-PREPARE-TO-END	R	ダイアログ終了付の PREPARE 要求を行う。
23	TP-CHAIN-ROLLBACK	RI C	状態を以前の状態に回復させる。	
24	自律トランザクション	TP-DECOUPLE	RI	応答が必要のない独立したデータを転送する。
25		TP-REQUEST	RI	独立したデータを転送する。
26		TP-REPLY	RI	TP-REQUEST に対する応答を行う。
27	同期	TP-SYNCHRONIZE	RIRC	同期点を設定する。
28		TP-SYNCHRONIZE-AND-GIVE-TOKENS	RIRC	同期点を設定した後、送信権を相手へ譲渡する。
29		TP-SYNCHRONIZE-AND-END	RIRC	同期点を設定した後、ダイアログを終了する。
30	リスタート (今後検討)	TP-RESTART	RIRC	障害後、同期点からの再開を行う。

* サービス・プリミティブのタイプは、次のとおりである。

R (Request)…要求, I (Indication)…指示 R (Response)…応答, C (Confirm)…確認

** コミットメントの対象となる原子的なトランザクション

- コミットメント同期及びコミットメントが使用できる。

(3) トランザクション処理のプロトコル

プロトコル仕様としては、下位層へのマッピングについて記述されている。アソシエーション制御 (ACSE), CCR 及びプレゼンテーション層プロトコルにマッピングしている。

5.2 ROS

ROS (Remote Operations Service) の現在のトランザクション処理における位置付けは、まだ明確になっていないが、関連が深いので、次にその概要を述べる。

(1) ROS の標準化状況

トランザクション処理プロトコルのベースとして、

表-3 ROS のサービス・プリミティブ (DP における案)

No.	サービス・プリミティブ	タイプ	説明
1	BIND*	RIRC	ROS のアソシエーションを確立する。
2	UNBIND*	RIRC	ROS のアソシエーションを解放する。
3	INVOKE	RI	ROS ユーザが相手にオペレーションの実行を要求する。
4	RESULT	RI	実行したオペレーションが成功した場合、その結果を転送する。
5	ERROR	RI	実行したオペレーションが失敗した場合、その結果を転送する。
6	REJECT-U	RI	ROS ユーザが検出した問題点を相手へ知らせる。
7	REJECT-P	II	ROS プロバイダが検出した問題点を知らせる。

* CCITT では、ROS ユーザが直接 A-ASSOCIATE, A-RELEASE へマッピングすることになっており、定義していない。

ROS を使用したいという意見がある。

ROS は、もともとオブジェクト・オリエンテッドなリモート・オペレーション向けのサービスとして、CCITT の MHS で作成されたものである。

これが近年では、汎用性があることから各種の問い合わせ/応答タイプのプロトコルのベース・プロトコルとして採用されつつある。ROS を使用する応用層プロトコルとしては、現在 RDA, ディレクトリ・サービス (Directory Service), OSI 管理の CMIS (Common Management Information Service), MOTIS (Message Oriented Text Interchange Systems) の P3, P7 プロトコルなどがある。

ISO では、SC 18/WG 3 で ROS の標準化を進めており、現在 DP が作成されている⁴⁾。一方、CCITT の SG VII でも ROS の検討が進められており、Draft Recommendation X. ros 0, 及び X. ros 1 が作成されている。

ROS は汎用性の拡大をねらい、多くの機能拡張が検討されており、各標準化機関での進捗度による一時的な差はあるが、最終的には一本化される予定である。

(2) ROS の機能概要

ROS の主な機能について、次に説明する。

(a) ROS のサービス・プリミティブ

ROS のサービス・プリミティブ (マクロ) を、表-3 に示す。問い合わせ/応答型の基本的なリモート・オペレーションを提供しているだけであり、複雑なサービス機能は特でない。

(b) プロトコル

プロトコルとしては、制御 OPDU (Operation Protocol Data Unit), インボーク OPDU, リザルト OPDU, リターンエラー OPDU, 及びリジェクト OPDU がある。

インボーク OPDU については、次の三つのフィー

ルドがある。

- インボーク ID: 起動するオペレーションに割り当てられる識別子である。ユーザが設定する。
- オペレーション: 実行されるべきオペレーションの種類を指定する。ユーザが設定する。
- アーギュメント: オペレーションのアーギュメントを記述する。ユーザが設定する。

このように、インボーク OPDU は、なんら特定の動作を規定するものではなく、その使い方はすべて ROS ユーザによって設定される。

したがって、ユーザは任意に自分の処理方法を規定して、このプロトコルを利用することが必要である。

下位層へのマッピングとしては、RTS (半二重) と、ACSE (Association Control Service Element), プレゼンテーション層 (全二重) の二つの場合がある。

(c) 機能拡張の要求

ROS は、問い合わせ/応答中心のプロトコルであるが、汎用性をもたせるため、各種の機能拡張要求がでている。

たとえば、SC 21/WG.4 からは、複数のサービス・プリミティブをまとめて送るブロッキング機能のサポート、オペレーションの確認の有無を起動時に指定できるようにすること、一つのインボーク OPDU に対して複数のリザルト OPDU を返せるようにすること、などの拡張の要求が出されている。

5.3 関連するプロトコル

(1) RDA プロトコル

RDA (Remote Database Access) プロトコルは、リモートにあるデータベースを SQL (Structured Query Language) 言語を使用してアクセスするための応用層プロトコルである。

現在 SC 21/WG 3 データベース・グループで検討しており、WD (Working Draft) の状態にある。

このプロトコルは、下位層として ROS, ACSE,

CCR を使用する。

RDA のサービス・プリミティブ (オペレーション) に関しては、文献 9) を参照のこと⁵⁾。

これを見ると、トランザクション管理機能は CCR を使用しており、トランザクション処理のコミットメント機能と同じものである。リソース処理、データ操作は、トランザクション処理のデータ転送の特定目的向け使用と考えられる。

このように考えることにより、プロトコルをできるだけ共通機能の整合を図ることが必要と考える。

(2) その他のプロトコル

このほかに SC 21 以外でも、類似のプロトコルの開発が行われる場合は、できるだけ仕様の共通化を図ることが大切である。

LAN 上のトランザクション処理プロトコルも、WAN との相互接続があるので、共通化することを検討する必要がある。

6. 今後の課題と展望

(1) 今後の検討課題

1987 年 1 月のトランザクション処理アドホック会議は、ベース・ドキュメントの形をまとめたに過ぎず、サービス仕様全般が固まったわけではない。多くの技術的課題が残されている。

プロトコルの構成については、ROS をどう位置付けるかについては、まだ議論が済んだわけではない。ROS をどう位置付けるかについての明確化が必要である。

RDA との関連も調整が必要である。

サービス機能についても、CCR 機能の提供形態、全二重の取り込み、自律トランザクションの扱いなど、さらに検討が必要である。

トランザクション処理は、特に性能面で、厳しいレスポンスタイムなどの実現が要求され、この面での各種の工夫がプロトコル上必要になると思われる。

また、筆者としては、次のような検討が必要と考える。

多くのアプリケーションの共通のプロトコルとなることから、ネゴシエーションやユーザ・データの設定などに拡張性をもたせ、使い勝手のよいものにしていく必要がある。また、ユーザ側からのニーズも反映することが必要である。

トランザクション処理プロトコルは、汎用性の高いプロトコルであり、各アプリケーション分野に固有な

プロトコルのニーズを一つ一つ追加することは困難である。ある程度で線を引き、あとはその上に追加する形での実現が必要となるかも知れない。

ISO で検討されている各種プロトコルとの関連についても、その面からの検討が重要と考える。

ソフトウェアのインプリメントの面から見ても、類似のプロトコルはできるだけ統合化されることが望まれる。

(2) 今後の展望

トランザクション処理プロトコルの標準化の早期実現は、国際的に望まれているものであり、その標準化スケジュールも異例の速さである。

応用層プロトコルの標準化も、プロトコル作成上の具体的な方法が明確になってきたので、その点から考えれば予定どおりのスケジュールでの規格の完成は不可能ではない。

日本はトランザクション処理の国際会議には当初から寄書を提出するとともに、会議にも積極的に参加してきた。

トランザクション処理プロトコルの標準化が予定どおり進めば、1989 年頃には、国際規格に準拠したプロトコルをサポートした製品が市場に出てくるものと予想される。

参 考 文 献

- 1) ISO/TC 97/SC 21 N 1715: Working Draft Transaction Processing Service Definition.
- 2) ISO/TC 97/SC 21 N 1716: Working Draft Transaction Processing Protocol Specification.
- 3) ISO/TC 97 N 1742: Proposal for a New Work Item on Transaction-Mode Application Service Element and Protocol for OSI, AFNOR, 1986-08-26.
- 4) ISO DP 9072/2: Message Oriented Text Interchange Systems Remote Operation Service-Part 2: Basic ROS.
- 5) ISO/TC 97/SC 21/WG 3 N 269: Second Working Draft of ISO Remote Database Access, December 1986.
- 6) Tanenbaum, A. S. and van Renesse, R.: Distributed Operating Systems, Computing Surveys, Vol. 17, No. 4 (Dec. 1985).
- 7) Haerder, T. and Reuter, A.: Principles of Transaction-Oriented Database Recovery, Computing Surveys, Vol. 15, No. 4 (Dec. 1983).
- 8) Kohler, W. H.: A Survey of Techniques for Synchronization and Recovery in Decentralized Computer Systems, Computing Surveys, Vol. 13, No. 2 (June 1981).
- 9) 穂鷹: 分散データベースの標準化動向, 情報処理, Vol. 28, No. 4, pp. 498-504 (1987).

(昭和 62 年 3 月 3 日受付)