

## 1 はじめに

近年、携帯電話を始めとした無線携帯端末の利用者が増加している一方、無線携帯端末による電子商取引の需要も高まりつつある。しかし、電子商取引において無線という通信媒体が原因で、高い通信切断頻度、低い安全性、低い通信速度等の問題が生じてしまい、有線環境下に比べ全電子商取引システムの核であるトランザクション処理の性能が劣る事が考えられる。

特に通信切断頻度の高い無線環境下でのトランザクションの場合、原子性を補完しなくてはならないため、トランザクションが中断される度にクライアントはそのトランザクションを最初からやりなおさなくてはならず非常に非効率になる。

本稿では通信切断頻度の高い無線環境下において、通信の切断が原因で生じるトランザクションにおけるインタラクションの失敗と、長時間の通信切断によるトランザクションの終了の双方に対して、通信復活時に通信切断時の状態からトランザクションを再度実行できる 2 Phase Recovery Transaction を提案する。

## 2 無線通信におけるコネクションの切断

無線通信環境において無線基地局からの距離や周囲の電波障害物等が原因で通信が切断される特性がある。また、コネクションの切断により中断されたトランザクションは切断時間によって大きく 2 種類に分類される。

### 2.1 無線通信の短時間切断

一般にトランザクション実行中に中断が生じた時、原子性を補完するためにシステムはトランザクションを最初からやり直す必要があり、コネクションの切断発生率の高さやトランザクション処理に必要な時間の長さに比例して、トランザクションを成功させるまでのトランザクション実行期待値が高まる。この原子性の補完により生じる問題を解決したトランザクション・システムとして、Compensatable Transaction[1] や Split Transaction[2] があり、本稿ではこの問題に関して Compensatable Transaction を拡張することにより解決する。

### 2.2 無線通信の長時間切断

無線通信の短時間切断との違いは、無線通信環境が原因で通信不可能な状態になり、一定時間以上経過するとサーバ側でタイムアウト処理がなされ、トランザクションが終了してしまう点であり、この問題に対して前記の 2 つのトランザクション・システムは未解決である。本稿において提案する 2 Phase Recovery Transaction では、この様な無線通信の長時間切断の後もトランザクション処理を復帰させることができる。

## 3 2 Phase Recovery Transaction の概要

2 Phase Recovery Transaction のシステム構成を図 1 に示す。通常のトランザクション・システムでは Transaction Client(以下 TC) と Transaction Server(以下 TS) が直接通信するが、本システムでは TC と TS の間に Transaction Agent(以下 TA) を設置し、全てのインタラクションを仲介する。また、本システムは Common Transaction(以下 CT) Layer と Recovery Transaction(以下 RT) Layer より構成される。CT Layer は既存のトランザクションの機能を提供する。本稿においては無線通信の短時間切断問題を解決するために、CT Layer に Compensatable Transaction を適用する。尚、TC と TA は無線環境下で通信を行い、TA と TS は有線環境上で接続されている。

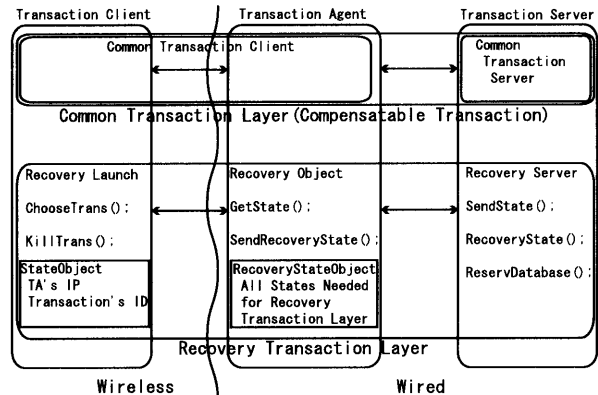


図 1: 2 Phase Recovery Transaction の構成

以下に TA・TC・TS が果たす主な役割と、CT Layer・RT Layer 上での重要な動作について簡単に記述する。

### • Transaction Agent

TA が果たす役割は TC と TS の負荷を可能な限りかけずに RT Layer を実現することである。CT Layer において TA は CT のクライアントの一部であり、TC が入力した情報を一度有線環境上に蓄えてから TS に送信する。この動作により無線環境にある TC を TS から抽象化させ、より信頼性の高いインタラクションを可能とする。RT Layer では、途中終了したトランザクションを復帰するために必要なステートを管理する。

### • Transaction Client

TC が果たす役割は TS が要求した情報を入力・送信することと、ユーザの意思を反映させることである。CT Layer において TC は CT のクライアントの一部であり、特に TS から要求された情報を入力が主な動作である。RT Layer において TC はユーザにトランザクションを復帰させるか否かの意思決定を要求し、実行する。

### • Transaction Server

TS が果たす役割は CT と RT の一方から、もう一方を起動させることとである。CT Layer では通常は TC のサーバ部分を担当するが、TC とのコネクションが切断され一定時間以上が過ぎた場合、CT を終了させると同時に RT を起動させる。また、RT Layer では TA から送られたトランザクションに必要なステートを CT Layer に渡す。

## 4 Recovery Transaction Layer

RT Layerにて、無線通信の長時間切断により終了したトランザクションを復活させることが可能となる。RT Layerはトランザクションの終了命令を受け取る時に実行されるステート保存機構と、TCが接続を再確立した時に起動するトランザクション復帰機構に分かれる。以下に2つの機構の動作について解説する。

### 4.1 ステート保存機構

ステート保存機構を図2に示す。TAとTCのコネクションが切断されて一定時間が経過するとTS内のTimeoutメソッドが実行され、SendStateメソッドを呼び出す。SendStateメソッドはトランザクションが終了するまでに正常終了したトランザクション内の各オブジェクトを認証するためのID、入力されたデータ、選択されたパラメータ、ステートを保持する有効期限、その他のトランザクション復帰に必要なステートをTA内のGetStateメソッドに送信する。SendStateメソッドはステートが正常に送信された事を確認した後終了する。またGetStateメソッドは受け取ったステートをRecoveryStateObjectに記憶し終了する。

一方、コネクションが切断されてTA・TS側から孤立状態にあるTCは、同様に一定時間が経過するとTimeoutメソッドが起動され、TAのIPアドレス・TAによる認証のID・トランザクション名をStateObjectに保持し終了する。

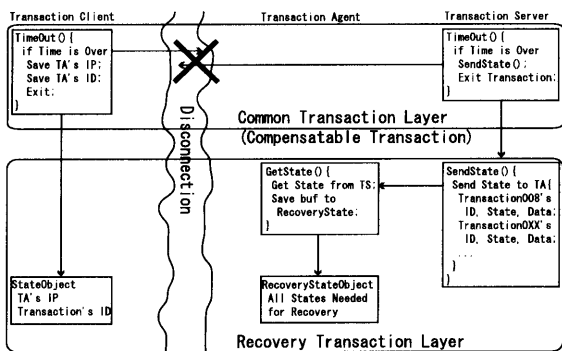


図2: ステート保存機構

### 4.2 トランザクション復帰機構

トランザクション復帰機構を図3に示す。TCとTAのコネクションが回復した時、ChooseTransメソッドが起動される。ChooseTransメソッドはStateObjectの存在を確認し、ユーザにトランザクションを復帰させるか、新規に実行し直すかを問い合わせる。存在しない場合トランザクション復帰機構を終了する。ユーザがトランザクションの復帰を選択したら、StateObjectに保存されていたTAのIPアドレスを元にTAのSendRecoveryStateメソッドと接続し、認証のIDより復帰させるトランザクションのRecoveryStateObjectを特定する。その後TCはTAとの接続を維持したまま、TAからのインタラクションの待ち状態に入る。

SendRecoveryStateメソッドはRecoveryStateObjectに記載されていたTSのIPアドレスとトランザクションの認証IDを元にTSのRecoveryStateメソッドとのコネクションを確立し、RecoveryStateObjectに保持されているトランザクション復帰に必要なステートをTSのRT Layerにて常駐しているRecoveryStateメソッドに送信する。その後SendRecoveryStateメソッドはRecoveryTransメソッドからの接続を待機するWaitConnect

メソッドを起動し終了する。RecoveryStateメソッドはCT Layer内にRecoveryTrans関数を起動させ、同時にステートも受け渡す。RecoveryTransメソッドは復帰させるトランザクションを起動しステートを入力する。一方でWaitConnectメソッドで待ち状態であったTAとのコネクションを確立させる。以上の過程を経て、ステート保存機構が起動される直前の状態からトランザクションを復帰させることが可能となる。

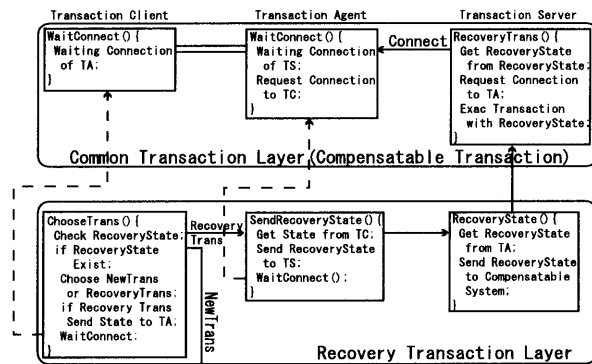


図3: トランザクション復帰機構

### 4.3 データベース予約機構

有限であるデータベースのリソースの予約が完了した直後に無線通信の長時間切断が生じた場合、トランザクションが終了してから復帰するまでの間予約保持をどう扱うかは非常に重要であり、この問題を処理するのがデータベース予約機構である。ステート保存機構の起動と同時にTSにてReservDatabaseメソッドが実行され、データベースの予約の有効期限を設定する。有効期限が経過した場合ReservDatabaseメソッドはデータベースの予約を解放し、TAのKillStateメソッドにRecoveryStateObjectを破棄させるシグナルを送信する。一方ユーザがトランザクション復帰をキャンセルしたい場合KillTransメソッドを実行し、TAのKillTransにRecoveryStateObjectを破棄させるシグナルを送信する。同様にKillStateはReservDatabaseにデータベースの予約を解放させる。

## 5 今後の課題とまとめ

トランザクション処理中に無線通信特性であるコネクションの切断が発生した場合にトランザクションは原子性を補完するため、再度実行し直す必要が生じる。本稿ではこの問題に対しコネクション切断段階を2段階に設定し、各々に対処する2 Phase Recovery Transactionを提唱し、設計について述べた。

今後の課題としては、無線端末の高移動性に適応可能なトランザクション・システムへの拡張(関連研究: An Efficient and Reliable Reservation Algorithm for Mobile Transactions[3]), 及び2 Phase Recovery Transactionを最も効率的に実現するコネクションの短時間切断と長時間切断の時間測定及び設定が挙げられる。

### 参考文献

- [1] H. Tokuda: Compensatable Atomic Objects in Object-oriented Operating Systems, Proceedings of the 1st PCCS, pp.210-219 (1985).
- [2] C. Pu, G. Kaiser and N. Hutchinson: Split-transactions for Open-ended Activities, Proceedings of the 14th VLDB Congerence, pp.26-37 (1998).
- [3] A. Elmagarmid, J. Jing, O. Bukhres: An Efficient and Reliable Reservation Algorithm for Mobile Transactions: Proceedings of the 6th International Conference on Information and Knowledge Management, pp.67-74 (1997).