

B-28 プラットフォームシステム連携環境におけるユーザ情報管理手法の一考察 Study on user management method in Platform Integration Environment

西村 徹
Toru Nishimura

堀川 桂太郎†
Keitaro Horikawa

1. はじめに

様々な適用領域に向けたプラットフォーム (PF) 製品を統合するアプリケーション (AP) 開発において、開発工程の短期化のためには、PF 製品で共通な機能のインタフェース (IF) 統合が有効である。例えば、データベース (DB) の IF を統合化した JDBC [1] により、DB 製品ごとの差分を隠蔽することで、利用する製品に依らない AP 開発が可能になる。

一般ユーザを対象とした Web-AP において、ユーザ情報の管理は必須機能の一つであり、情報の一貫性確保が求められる。PF 製品はそれ自体にユーザ管理機能を持つ場合が多い。このとき、ユーザ情報の登録や変更機能の実装において、トランザクション区間が内部に閉じる場合、情報の一貫性を確保したまま、PF 製品を統合し一つの Web-AP とするためには、各 PF 製品のトランザクション区間の統合を含めたユーザ管理機能の統合が課題となる。

そこで本稿では、PF 製品のユーザ管理機能の統合に関して、ロングタームトランザクション [2] の概念を適用して一貫性を確保する統合ユーザ管理機能の実現方式を提案し、そのために PF 製品に必要な条件を示す。

2. 要求条件

Web-AP がユーザ管理を行う上での要件は、(1) Web-AP におけるユーザ情報の操作(登録・更新・削除)は、統一した IF により行われる、(2) ユーザ情報の操作において、PF 製品間での情報の一貫性が確保されることである。要件(1) は Web-AP の短期開発における基本的要件であり、要件(2) はシステムがユーザ情報を扱う際の基本的要件である。

3. 前提条件

対象とする PF 製品の前提条件は、(1) PF 製品内で管理するユーザ情報の登録・変更・削除の機能を API として提供している、(2) トランザクション制御は、(1) で示した機能の内部に閉じていることである。前提(1) は、ユーザ管理機能統合のために、最低限必要な PF 製品の外部 IF として妥当である。以降、{登録, 変更, 削除} 機能を呼び出すための API を {register, update, delete} = OPERATION とする。前提(2) は制約範囲を広げる意味がある。なぜなら、外部からのトランザクション制御が可能な場合 (制御が閉じていない場合)、それを利用して、ユーザ情報の一貫性確保が実現可能である。

4. 要求条件の分析

要件(1) に対して、各 PF 製品のユーザ情報が、PF 製品一統合ユーザ管理機能間で定められたデータオブジェクト (Data Object, DO) で表現できる必要がある。換言すると、対象となる任意の PF 製品 (PF_i) に対応する UDO_i と全単射 $m_i: UDO_i \rightarrow PDO_i$ が存在する。ここで、UDO_i は、統合ユーザ管理機能が扱う引数集合 DO (以下、Unified DO, UDO) の部分集合であり、PDO_i (PF_i dependent DO) は、PF_i 製品が扱う引数集合である。UDO の生成については、全 PF 製品のユーザ情報属性の和集合の、属性の意味上の等しさ (≈) での同値分割である統合ユーザ情報 (すなわち $UDO = (U_i \in PF PDO_i) / \approx$, ここで、PF は PF 製品の集合) を与える方法を文献[3] で提案している。

要件(2) に対して、一貫性が保てなくなる要因としては、一貫性 (Consistency) に関する要因と、隔離性 (Isolation) に関する要因、耐久性 (Durability) に関する要因がある。以上の要因に対して、従来の DBMS による手法は、前提条件(2) により、トランザクション制御の区間が PF 製品の内部に閉じているため、利用できない。以下に、各要因の具体的な説明と、従来の DBMS による手法、今回提案の eCo-Flow を用いた実現手法について示す。

(a) 一貫性要因: 一部 PF 製品に対するユーザ情報操作に失敗した場合、操作に成功した PF 製品との間にユーザ情報の不整合が生じる。

[DBMS による手法] 一部の PF 製品への操作が失敗した段階で、DBMS のロールバック機能により、操作に成功している PF 製品のユーザ情報の更新処理を取り消すことができる。

[提案手法] 一部 PF 製品に対するユーザ情報操作に失敗した段階で、操作に成功している PF 製品に対して、逆操作 (後述) を実行する。

(b) 隔離性要因: ユーザ情報操作中は、全体としてユーザ情報が不整合の状態となる。

[DBMS による手法] 処理実施中は、他の処理から更新中のユーザ情報を参照できないように制御する。

[提案手法] ユーザ情報へのアクセスを統合 API に制限するとともに、API の内部でアクセス制御を行う。

(c) 耐久性要因: ユーザ情報操作中に、処理要求側システムが停止状態になった場合、既に処理が終了した PF 製品と、未終了のそれとの間にユーザ情報の不整合が生じる。

[DBMS による手法] 障害が発生する以前の整合性を保つ状態へ復旧させることができる。

†日本電信電話株式会社 NTT 情報流通プラットフォーム研究所
NTT Information Sharing Platform Laboratories, NTT Cooperation

[提案手法] 実行履歴を管理し、障害発生時は、発生前に起動されていた処理を再実行する。

5. 統合ユーザ管理機能

一貫性要因に対する対応として、逆操作実行により整合性を保つために、実処理に対する逆操作を定義する必要がある。逆操作は、実処理で行われた処理内容を打ち消す意味を持ち、その処理に必要な情報は、実処理で利用する情報や、実処理の結果得られた情報、環境から取得可能な情報に限定される。ユーザ情報の操作を行う IF においては、表 1 に示すように、{ register, update, delete } に対する逆処理が、それに閉じている。と同時に、逆操作実行に必要な情報 (操作前の情報) をあらかじめ保持しておく必要がある。

表 1 操作と逆操作の対応

操作	逆操作	逆操作の方法
register	delete	登録時に利用したユーザ情報を削除
update	update	更新前のユーザ情報で、再度更新
delete	register	削除前のユーザ情報で、再度登録

図 1 は、登録・更新・削除操作の実装の概略である。図中の $operation \in OPERATION$ およびその入力・出力が、統合ユーザ管理機能の IF になる。前出の UDO を引数とし、フロー内部の処理(4)において、連携の対象となる PF 製品に個別に操作 (委譲) を行うことで、IF 統合を実現する。

(1) 入力データチェック

UDO を本フローの引数とし、その引数の実行前チェックを行う。具体的には、PF 製品に操作可能な情報であるか、正当な処理実行者が要求した処理かなどを確認する。

(2) 操作前データ保存

逆操作のために必要な操作前データの保存を行う。具体的には、登録時はユーザ ID (userID)、更新・削除時は更新・削除前情報 (preUserInfo) である。本データは処理(7)の PF 製品逆操作時に利用される。

(3) 操作対象 PF 製品選択

操作が必要とされる PF 製品群を選択する。本処理は、操作対象となるユーザに利用を許可する PF 群をあらかじめ規定しておくことにより、静的に実現できる。本処理により選択された PF 製品の IF オブジェクトは、配列型の変数 PFList[] に格納されることとする。また処理(4), (5) で利用されるカウ

ンタ変数 $i \in [0..(N-1)]$ の初期化 ($i=0;$) を行う。

(4) PF 製品操作

PF 製品の IF オブジェクト PFList[i] に対して、PF 製品ユーザ情報操作 API を利用して操作を行う。メソッド起動時の引数は、PF 製品ごとの射像 (m_i) を用いて変換を行った PDO_i とする。すなわち、

$PFList[i].operation(m_i(UDO));$

ここで、operation は、フロー実行時に指定されたユーザ情報操作 API のメソッドである。

(5) 操作完了確認

処理(3)により選択された PF 製品群に対する操作が、全て終了したか否かを確認する。具体的には、カウンタ変数 (i) に 1 加算 ($i++$)、その結果が、PFList の要素数 N に達する ($i > N$) まで、処理(4) に遷移 (ループ) する。

(6) 結果返却

処理結果を返却する。

(7) PF 製品逆操作

処理(4) で、PF 製品への操作に失敗した場合、それまで成功した PF 製品に対する操作を、表 1 に基づく逆操作によりキャンセルする。具体的には、実処理が register の場合は、処理(2)で保存された userID をキーに delete を、実処理が update (delete) の場合は、処理(2)で保存された preUserInfo を引数に、update (register) を行う。

隔離性要因に対する対応として、あるユーザに対するフロー実行中は、同一のユーザに対するフロー実行を許可しないことにより対応可能である。また、耐久性要因は、[2]の再実行機能を利用する。

6. まとめ

ユーザ情報の登録、変更、削除機能を持ち、トランザクション制御の範囲が上記機能の内部に閉じている PF 製品に対して、ロングタームトランザクションの概念を用いてユーザ情報の一貫性を保つ仕組みを提案した。

参考文献

[1] <http://java.sun.com/j2ee/ja/jdbc/index.html>
 [2] T. Hatashima, et al., WebService Processing Platform - eCo-Flow, SAINT 2002, Workshop, Jan. 2002, pp. 186-195.
 [3] T. Nishimura, et al., OSS collaboration system using OSS-independent information, APNOMS 2000, pp. 493-494.

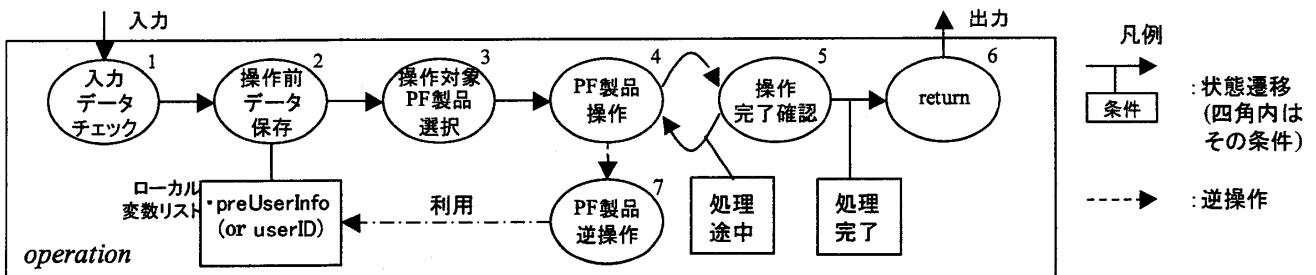


図 1 処理フローの概略図