

トランザクション環境下でのオペレータ主導の画面制御方式

5K-10

甲斐 こずえ 菅沼 毅 稔田 隆

NTT情報システム本部

1. はじめに

端末系APの形態として、APの規定するデータフロー上必要な時にデータの入出力を行うAP主導型と、オペレータの画面操作を契機に該当する処理が走りだすオペレータ主導型の2つに分類できる。

AP主導型ではオペレータの入出力に制約がつくるので操作性は良いとは言えないが、リソースの管理が容易なため、現状ではトランザクション処理環境で広く使われている。

近年、ホスト系計算機で利用していたトランザクション処理を、マルチベンダのWSで利用しようという要求が大きくなっている。また、WSではGUIが一般的になっている。従ってユーザはトランザクション処理環境において、GUIによるより良い操作性を要求し、その実現のためにオペレータ主導を必要とする。

本稿ではオペレータ主導型の画面制御方式を、マルチベンダのトランザクション処理環境で実現する方式に関して述べる。

2. 目的

オペレータ主導による操作性の良いAPを、トランザクション処理環境においてベンダに依存しない方式で提供することを目的とする。

これを実現するにあたり、トランザクション処理の特性と操作性の向上のバランスを取ることが課題となる。

3. 実現方式

目的の画面制御方式の実現方式について以下に述べる。

3-1 ソフトウェア構成

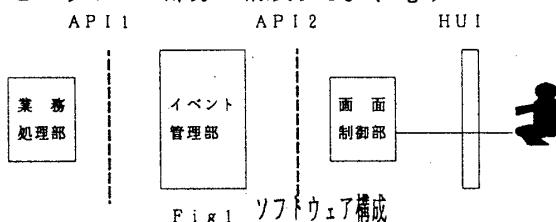
- ・画面制御部

The Control Method of Graphical User Interface in Transaction Processing Environment
Kozue Kai, Takeshi Saganuma and Takashi Hieda
NTT Information Systems Headquarters

・業務処理部

・イベント管理部

という3つの部分で構成する。(Fig1)



(1) 画面制御部

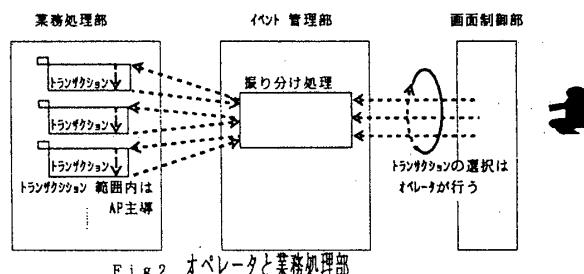
画面の表示を制御する部分を呼ぶ。業務処理毎にベンダ毎のGUI構築ツール等を利用して作成する。

オペレータからの入力によって業務処理部へ処理を要求する。

(2) 業務処理部

業務処理を行うタスクの集合体を呼ぶ。業務処理毎に開発する部分である。

業務処理部全体をひとつのタスクとして扱うと実質的にAP主導となってしまい、操作性は良くない。このためタスクを分割する。分割したタスクをどのような順番で実施するかは、操作性を良くするためにオペレータが自由に決めることとする。しかし、タスクをトランザクションの範囲を越えて分割した場合には、ひとつのトランザクションが制御を独占して、他のトランザクション処理を阻害してしまう。また、トランザクションの範囲内をオペレータ主導とした場合にも、同様の障害が起こる。これを防ぐため、トランザクションの範囲を越えてタスクを分割してはならず、各タスクの範囲内の処理はAP主



導とする必要がある。つまり、Fig2に示すような、タスクの選択順番はオペレータが決めるが、タスク内ではAP主導である構成とする。

(3) イベント管理部

業務処理部と画面制御部間の入出力イベントを送受信する機能を持つ部分。共通モジュールとして一元開発する。

画面制御部からの入力によって要求を受けた該当のタスクを起動し、該当のタスクと引数を利用してデータの交換を行う。

ウィンドウシステムの特性によって、イベント処理で待ち状態になると、処理を受け付けられなくなったり(PM)、画面の再描写が出来なくなったり(Motif/OPEN LOOK)するため、オペレータ主導型のGUI環境を提供するためには送受信形態を非同期に行う必要がある。次に業務処理部の起動と、業務処理部と画面制御部との会話を非同期に可能にする機能について述べる。

A. 業務処理部非同期起動機能

イベント管理部は画面制御部からタスクの起動を要求されると、業務処理部からの返答を待たずに制御を戻す。トランザクションが終了すると、タスク終了ハンドラに制御を戻すことにより、画面制御部にトランザクションの終了を通知する。(Fig3)

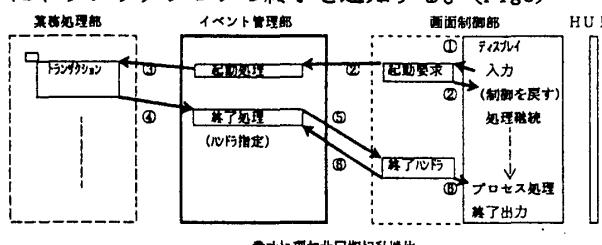


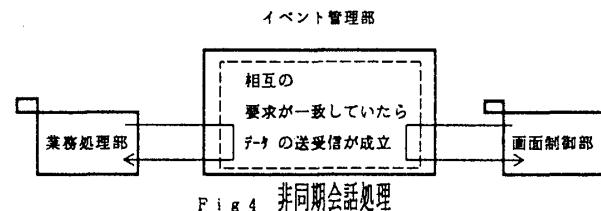
Fig3 業務処理部非同期起動機能

B. 非同期会話処理

業務処理部と画面制御部との会話を非同期に行う機能。

業務処理部はイベント管理部と、それとは非同期に画面制御部もイベント管理部とデータの送受信等の要求を送り、それぞれの要求が一致したときにデータの受渡しを実行する。画面制御部からの送信要求が先にあった場合、イベント管理部は一旦データを受取ると制御を戻す。このデータへの業務処理

部からの受信要求を受けると、保管していた該当のデータを受け渡す。逆に業務処理部の受信要求が先であり、画面制御部からの送信要求が掛からない場合にはタイムアウトする。(Fig4)



これらの機能により、APが要求するデータがいつまでも入力されない、若しくはオペレータの操作にAPの処理が伴わずにレスポンスが保証されないなどの、オペレータとAPのステータスの不一致を無くす。

4. 試作概要

これまで述べてきた実現方法の評価の為に、以下の条件で試作している。

言語 : C, STDL^[1]

使用OS : OS/2, UNIX

Windowシステム : PM, Motif

使用GUI構築ツール : VPE, CUART/2

対象GUI機能 : ボタン, キャンバス, テキスト

5. 評価

GUI構築ツール毎に異なるイベント処理機能を利用せず、トランザクション特性に適したイベント管理メカニズムを一元開発することにより、ベンダ依存部の局所化が可能となる。

本方式により操作性向上が期待できるが、トランザクション処理環境での応答速度を含めた評価は、試作システムの完成後に行う予定である。

6. まとめ

本稿では、業務処理部と画面制御部の間に、イベント管理部を設置することによって、マルチベンダでのトランザクション処理環境におけるオペレータ主導の画面制御方式について述べた。

参考文献

- [1] MULTIVENDOR INTEGRATION ARCHITECTURE
VERSION1.3 テクニカルワイヤメント 第2編アリケーション・プログラム
・インターフェース仕様 TR50001-1-6J-2 Apr.1992