

トランザクション環境下でのGUIのソフトウェア構成

4S-4

菅沼 毅 竹内 宏典 稗田 隆
NTT情報システム本部

1. はじめに

HUI(ヒューマンインタフェース)には一般的に、次の特性が存在する。

- 技術進歩が早く技術動向/標準化動向が十分には定まっていない。またユーザ要求が流動的で変化しやすい。

さらに、トランザクション処理環境におけるHUIは、対話処理環境と比較して次の特性も有する。

- 処理の流れが定型的であるため、ユーザとのインタラクションが単純、かつその発生頻度が少ない。

また上記の諸特性に加え、トランザクション処理はメインフレーム系のホスト計算機を中心に発達して来たため、キャラクタベースのHUIが主流である。さらにHUI系のAP開発支援製品にはベンダ毎の独自色が強く、業務APのポーティングが困難であった。

しかし近年、WS上でのトランザクション処理の利用に際し、WS上で既に一般的となっているOPEN LOOKやMotifなどのGUI(グラフィカルユーザインタフェース)の利用に対する要求が強まっている。加えて、APの生産性向上やポーティングも不可欠な要求である。

本稿では、トランザクション処理環境下でGUIを使用する際の、APポーティングを可能とするソフトウェア構成を示す。

2. 目的

本稿の目的は、トランザクション処理環境への適用を想定し、GUIを活用可能でなおかつ複数ベンダのマシン間で業務APがポータブルなソフトウェア構成を実現する事である。

なお、検討に際し次の点に留意するものとする。

- GUIとしてはトランザクション処理環境にふさわしいものでなおかつ必要十分な機能とする。例えば、トランザクション処理中にユーザのウィンドウ操作に起因するガーベージコレクションが発生しスループットが低下するような事は避けなくてはならない。
- 現時点の技術や現実的なコストで実現可能な構成とする。したがって将来性や開発コスト低減などため、標準化動向や市販製品を積極的に活用する。
- トランザクションのコミット/ロールバックと画面表示との親和性を保つメカニズムに関しては本稿では言及しない。

3. ソフトウェア構成の規定

実現方式に対する要件は次の通りである。

- ①画面制御処理を業務処理のトランザクション境界内に含めない様にする等、HUIがスループットへ与える影

響をプログラミングで制御出来るようにする必要がある。このためには、イベントドリブンタイプのプログラム構造ではなく、ユーザとのインタラクションはAP側で制御するような構造が不可欠である。

②トランザクション境界やコミット/ロールバックの設定と画面処理とを同一の人間が設計/コーディングするのは負担が大きい。したがって、業務処理と画面制御処理を分離して記述可能とする事が必要である。

③ウィンドウシステムを直接使用する画面処理のプログラミングには高いスキルが要求されるため、様々なAP開発支援パッケージが市販されている。開発コストの低減や優れた開発環境利用の利点などのため、こういった市販のHUIパッケージを利用可能でなくてはならない。ただしこの場合でも、マルチベンダ性を保ち業務APのポーティングを可能とする構造である事。

④HUIとそれを実現するAPIは、企業や部門によりカスタマイズやローカルな標準化が行われる事が多い。よって、このような特定の処理を一元的に開発し、共通的なAPIにて使用可能な構成でなくてはならない。

これらより、図1の構成とした。

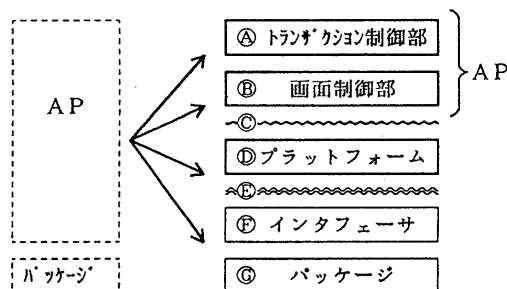


図1 ソフトウェア構成

図1の各部分の位置づけ/所要機能は以下の通りである。

- ④トランザクション制御部
トランザクションの制御と業務処理を行う部分。画面との全ての入出力の発生契機を制御するが、画面の表示状態の処理は含まない。
- ⑤画面制御部
画面表示に関する処理の記述を行う部分。適用環境や業務等に要求される画面の表示状態に応じて作成される。ベンダやハードウェアに依存する情報は、外部定義(画面定義)にて規定する。

APを④、⑤の2つに分割する事により、HUIがトランザクションのスループットに与える影響をAP側で制御可能で、なおかつ業務処理部分と画面処理部分を明確に分離する事が可能となる。

◎業務AP用API

AP作成者に対し提供される、各環境で使用し易いAPI（例えば、社内標準等）。

⑩プラットフォーム

各適用環境でのHUIとしてのプリミティブ機能を、◎のAPIにて提供する。この部分には業務処理を含まず、汎用性を持たせる。

⑪ベンダ独立インタフェース

多様なHUIパッケージの仮想化を行い、上位のポーティングを可能とするためのインタフェース。◎と比較すると、機能は豊富だが低レベルである。例えば、単独フィールド入出力/ウィンドウ制御/円、直線などのグラフィックス描画等、HUIとして最もプリミティブな機能が該当する。

⑫インタフェーサ

ベンダ独立インタフェースと、市販製品であるHUIパッケージの提供するベンダ固有インタフェースとの交換を行う。

⑪、⑫により、複数ベンダのパッケージ上でAPのポーティングが可能となる。

◎パッケージ

◎のベンダ独立インタフェースを実現可能な市販製品を使用する。パッケージはGUI対応になっている事が要件である。なお、ベンダやハードウェアに依存する情報を定義するための外部定義情報はベンダ毎となる。

本構成ではAPを④トランザクション制御部と⑤画面制御部という2つの部分に分けるが、両者の協調動作は次の様にモデル化される。

④はトランザクション処理の中で画面との入出力が必要な場合に、入出力を行うデータ実体とその流れの方向（入力/出力）のみを指定して要求を行う。したがって、④は業務固有処理に専念し、画面の表示様態を極力意識しないため、画面処理の知識が無くても作成可能である。

⑤は、入出力データと画面上への表示様態との対応を管理しており、④からの要求を契機として、指定された通りのHUIを実現するよう、必要に応じ◎のインタフェースで⑩の機能を呼ぶ。

4. 実装上の課題と対策

これまで述べた構成を実現するために不可欠なベンダ独立インタフェースについて述べる。

ベンダ独立インタフェースには、①業務処理と画面処理の分離した記述、②GUIのサポート、が必要となる。①の考え方に適した標準化案としてCODASYL-FIMS仕様[1]が存在する。ただしFIMS仕様はキャラクタベースであり、②を満たさない。よって、FIMS仕様をベースにトランザクション処理環境への適用の範囲内で必要十分なGUIへの拡張を行う事とした。

FIMSでのインタフェースは原則的にデータフローの方向とパケット単位のデータ実体の指定のみであり、

必要に応じタイムアウトなどの要求オプションの指定が可能となっている。今回、APからの要求オプションタイプとしてGUI対応化の拡張を行った(図2、表1参照)。

```
rc=SEND_PACKET(SESSION_ID,PACKET_NAME,
                SEND_PACKET,REQUEST_OPTION)
```

図2 ベンダ独立インタフェースのシンタックス

APより各オプションタイプが指定されると、インタフェーサがHUIパッケージの該当する機能呼び出す事により要求された機能を実現される。本稿で対象とする環境への適用にあたり、FIMS仕様に不足する機能は主に次の2点である。

- (1)ウィンドウ制御機能
- (2)グラフィックス描画機能

表1 拡張したFIMSの要求オプション識別子

(1)ウィンドウ制御機能	(2)グラフィックス描画機能
MENU	LINE
LIST BOX	ARC
WINDOW TITLE	RECTANGLE
WINDOW POSITION	JOIN STYLE
WINDOW SIZE	FILL STYLE
KEY ASSIGN	CAP STYLE
BUTTON	CURSOR
COLOR	BEEP
:	:

5. 評価

本稿で提案した方式により、画面のHUIとしてOPEN LOOK, Motif等のGUI表示が可能である。ただし、元来キャラクタ表示端末を前提に作成されたFIMSの仕様の拡張のため、ウィンドウシステムが元来持つGUI機能の全てに対応する事は不可能である。例えば、本方式ではユーザの操作に起因するマウスやウィンドウ操作などのイベントを取得するインタフェースが存在しない。したがって、これらのイベントを取得するようなAP記述には機能的に不十分な点もある。

しかし、トランザクション処理環境下ではオペレータとのインタラクションが頻繁に行われる事は無く、定型の入出力が大部分である。したがって、トランザクションAPとしてはこれらが必要とは考えにくい。よって、本稿の方式はトランザクション環境下でのHUIとして必要十分な機能の提供が可能である。

なお、インタフェーサ部分の規模、全体に占める割合や性能面での影響に関しては現在、試作による評価を行っている。

6. まとめ

トランザクション処理環境向けのHUI機能要求を満たし、機種依存性が無く業務APのポーティングを可能とするソフトウェア構成の検討を行った。さらに、その構成に必要なベンダ製品の差異を吸収するインタフェースの検討を行った。

今後は、本検討の方式の試作、およびトランザクションのコミット/ロールバックとのHUIの親和性を検討する。

参考文献

[1]Draft CODASYL Standard-I:Form Interface Management System 1991