

GUI制御部品の開発

4W-1

大原茂 栗根達志 鵜澤亨

(株)日立製作所 公共情報事業部

1. はじめに

日立製作所 公共情報事業部では WS-PC の C/S システム開発に、4GL 開発環境として APPGALLERY や PowerBuilder (Powersoft 社製) を使用している。4GL は、GUI 画面の開発とビジュアルな DB 定義、オブジェクト指向による保守性・再利用性などの機能から、大規模な C/S システムの開発に適している。しかし、4GL での開発では、4GL 開発環境の多機能（選択肢の多さ）に翻弄され、プロトタイプ開発時の設計項目が増大したり、イベント駆動型プログラミングに起因する不良の作り込みがどのプロジェクトにも共通してみられた。

そこで、4GL 標準開発手順書による設計項目、手順、留意点を取り込んだ部品を開発することで、開発効率向上と開発方法統一を試みた。

本論文では、この GUI 制御部品の開発作業とその適用効果について報告する。

2. GUI 制御部品の概要

GUI 制御部品は、画面遷移や画面内の状態を統括的に管理する機能を持つ。また、画面遷移、画面内の状態遷移は、表形式で定義した画面遷移表や項目状態遷移表に従って自動的に処理を実行する。つまり、画面遷移、状態遷移はスクリプトとは独立して定義する。表定義による制御の定義は、スクリプトとして埋め込まれた制御に比べて、関連した項目が把握しやすく、また、その修正も容易である。

システム開発時は、4GL 開発環境自身が提供する GUI オブジェクトに GUI 制御機能を追加したオブジェクトである GUI 制御部品を使用する。プロトタイプ開発フェーズではシステム全体から、類似画面を抽出し、いくつかのパターンに分類する。この画面部品はのちの全体工程において雛形となる部品である。プロトタイプ開発フェーズにおいては、画面遷移や、画面内の項目が、頻繁に変更される。この時、画面遷移や画面内の項目の状態の遷移は、表の定義を変更することで修正可能であり、スクリプトの修正が発生しないため、スクリプト修正による品質の低下を抑えることができる。また、この画面遷移表、状態遷移表から直接、画面遷移図や状態遷移図を生成することができるため、画面遷移や画面内の項目の修正が発生しても、常に仕様書とプログラムの整合性を保つことが可能となる。

パラレル開発フェーズでは、「画面部品」を使用して開発を行う。「画面部品」は画面遷移、状態遷移、イベント制御に関する処理を画面部品開発時に定義済みであるため、差分の業務処理のみを記述する。

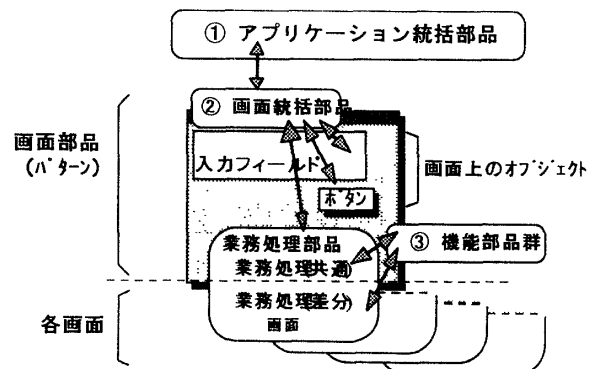


図1 GUI 制御部品の構成

The Development Of GUI Control Class Library.
 Shigeru Oohara, Satoshi Awane, Toru Uzawa
 Government & Public Corporation Information
 Systems Division, Hitachi Ltd.
 Shinsunaplaza 6-27, Shinsuna 1-Chome, Koto-ku,
 Tokyo 136, Japan

3. GUI 制御部品構成

GUI 制御部品は、大きく下記の3つの部品から構成されている。(図1 GUI 制御部品構成参照)

(1) アプリケーション統括部品

画面遷移表を基に画面遷移を行う。

(2) 画面統括部品

画面内項目のイベントを制御する部品である。処理実行の可否を判定し処理を実行する。

(3) 機能部品群・画面状態遷移

状態遷移表を基に項目の状態(入力可/不可、表示/非表示など)を変更する。

4. GUI 制御部品による開発手順

GUI 制御部品を使用して、RAD 開発手順に従い開発を行う方法を述べる。(図2 GUI 制御部品を使用した開発手順参照)

(1) プロトタイプ開発フェーズ

プロトタイプの開発においては、4GL 開発手順書に基づき、顧客要求仕様が発散しないよう努める。

続いて、自システムの共通仕様の検討を行う。画面部品(パターン)の自プロジェクト用カスタマイズ GUI 制御部品を組み合わせ、業務処理における共通部分を記述する。この段階では、業務固有となる処理も1パターン作成し、早期段階で品質を確保する。

(2) パラレル開発フェーズ

プロトタイプ開発フェーズで用意された画面部品(パターン)を用い、各画面固有の業務仕様の作成を行う。イベント制御、画面内項目制御に関するスクリプト記述は部品内に取り込まれているため、業務処理に専念して開発を行うことができる。

(3) 総合テストフェーズ

総合テストにおいて発生しやすい、イベント制御と画面内項目状態の制御から発生する不良が、GUI 制御部品で処理され、品質が確保されているため、不良対策が別の不良を誘発するような二次

的な不良発生を防ぎやすくなる。そのため、テストに要する工数が削減される。

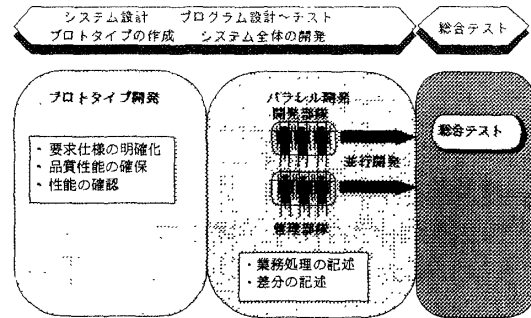


図2 GUI 制御部品を使用した開発手順

(4) GUI 制御部品の適用効果

図3は、Aシステムの見積値と実推である。従来の開発方法で開発を行った場合の見積値を100%とすると、GUI 制御部品を適用した開発では、その約70%の開発工数で開発することができた。

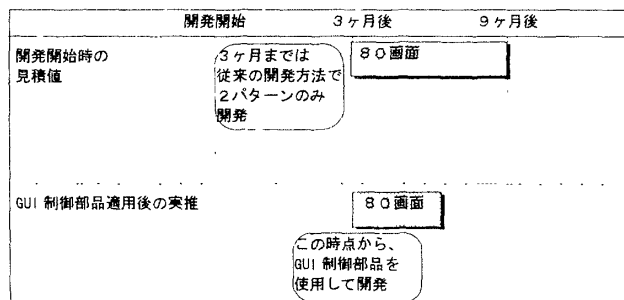


図3 Aシステムの見積値と実推

5. おわりに

GUI 制御部品を用いてシステム開発を行うことにより以下の目的を達成することができた。

- プロトタイプ開発フェーズにおける作業効率の向上。
- イベント駆動型プログラミングに起因する不良の事前回避。
- 画面部品の作成によるコーディング量の削減。

今後のエンハンス項目として、項目チェック表、DB定義表等あり、そのサポート方法と生産性を検討中である。