

GUIアプリケーションの品質に関する考察

5N-9

黒川 裕彦、徳丸 浩二、佐藤 信春

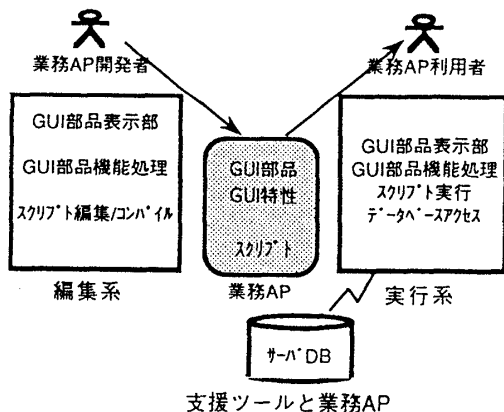
NTT ソフトウェア研究所

1. はじめに

作業の生産性向上、より使い易い操作性を目的に、グラフィカルユーザインタフェース（GUI）による対話操作でアプリケーション（AP）を開発／実行できる構築支援ツールの研究・開発が進んでいる。生産性、操作性の評価を目的としてマルチプラットフォームで動作可能なGUI構築支援ツール（以下、支援ツールと略す）の試作を行った。支援ツールは、オペレータのシミュレーションを前提に多様な視点と、対話制御、データベース（DB）とのインタフェースなど様々な構成要素の集合体であるため、品質要素は多岐に渡る。本稿では、試作した支援ツールの概要と支援ツールで考慮すべき品質要素を明確にし、支援ツールを適用したAPに関する不具合の分析について述べる。

2. 支援ツールの概要と業務AP

支援ツールは、クライアント／サーバ（C/S）環境でのデータベースアクセス機能を持つツールであり編集系機能と実行系機能を具備している。APの開発は、ツールウィンドウからのGUI部品の選択・配置と部品のアクションを規定するスクリプトの記述による画面の作成と、画面間の遷移の記述により行われる。



3. 支援ツール開発における品質

3.1 発見不具合の分類

検査において発見された不具合について、分析したところ78%が実行系機能で発生し、作り込み原因としては64%が詳細設計工程であった。

3.2 原因分析

原因1：設計書の良否が大きく影響

支援ツールの開発において、開発メンバによる設計・検討の結果、以下のユーザインタフェース（UI）要素と機能要素を記述した設計書を作成した。

- (1)静的UI：編集系と実行系オペレータを想定ウィンドウの構成、GUI部品の形状・デザイン等
- (2)動的UI：実行系オペレータを想定キーボードやマウスからの各種イベントに対するGUI部品の動作と制御
- (3)編集用UI：編集系オペレータを想定GUI部品の特性設定ダイアログ、部品の配置・設定・移動およびスクリプト編集
- (4)スクリプト実行機能：実行系
- (5)DBアクセス処理機能：実行系

以上のように編集系と実行系のオペレータについてシミュレーションする必要があり、また機能要素の設計はUI要素に影響を及ぼす。特に(2)動的UIは、(3)GUI部品の特定設定と(4)スクリプト実行の設計に多く影響される。

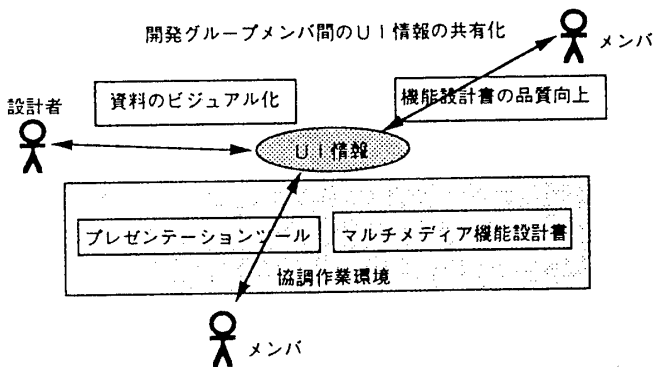
以上のUI要素設計の複雑さに加えて、設計書におけるUI要素の表現方法も問題となる。実行時のUI情報（UI要素の集合：画面レイアウト、操作手順等）は設計者の頭の中のイメージとしては形成されるが、その情報の資料化に時間がかかり情報の欠落も発生しやすいため、UI設計書をベースとした詳細設計工程の品質に影響を与える。

原因2：詳細設計工程で多い仕様変更

機能設計工程で決定された上記UI要素に対し操作性向上が期待できる変更要求は優先的に仕様変更を行う。しかし、この変更が機能設計書に即座に反映されないあるいは情報の欠落のため認識誤り、工数増加・工程圧迫による詳細設計の検討漏れが発生する。

機能的不具合の防止対策だけでなく、使いやすさ

の品質（効率、効果、満足）をグループワーク作り込めるように、機能設計・検討の段階から設計者だけではなく開発グループメンバーの多くが設計に参加することが望ましく、開発メンバー間でのUI情報の共有化が重要であると考えます。このためには、UI情報を開発メンバーが直感的に理解できるように検討資料を工夫すると共に、UI情報を視覚化してプレゼンテーションできるツールの利用が有効である。さらに、設計変更からの手戻りを削減するために、作成した視覚化したUI情報から設計書に変換、あるいはUI情報自体を設計書としてグループで議論・レビューできる協調作業環境の利用も設計品質を向上させる手段であると考えます。

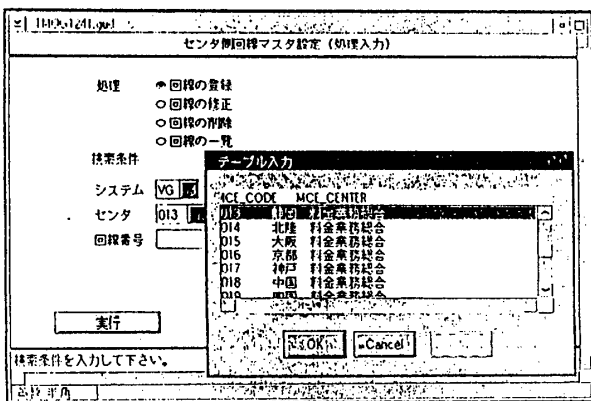


4. 支援ツールを利用した業務APの品質

4.1 開発の背景

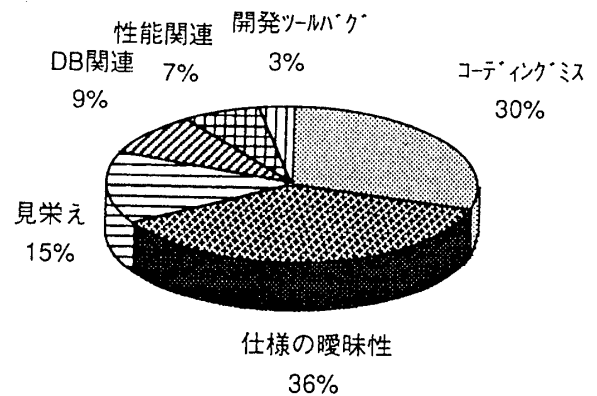
本支援ツールを、大規模システムに接続するPC情報をサーバDB上で管理するC/S環境の業務システムの開発に適用した。

- 開発規模 画面数：約100 GUI部品数：約4000
スクリプト規模：約70KL（下に画面例）



4.2 検査時発見された不具合

検査時に発見された不具合を原因別に分類する。



4.3 考察

仕様の曖昧性、見栄えについてはユーザと開発者の認識のズレから発生したものである。画面表示とデータの入力チェック関連するものが多い。開発当初から支援ツールを用いて、画面設計段階からプロトタイプの上でユーザが仕様を確認する、さらに、GUI部品の特性を理解し特性値リストを作成し機能仕様書に反映する対策を取った場合、不具合の発生はかなり減少できる。

開発工程に余裕があれば、この種の不具合は次ステップの改善項目と見なすこともできる。逆にユーザと共に検査と改善を繰り返すことによりUIの品質を向上させることが可能であるため、AP開発においてはUI検査・改善の工程が必要である。その場合、不具合を修正・改造するための工数が問題となるが、今回の場合1件の不具合の分析および修正にかかった工数は0.53日、認識ズレに起因する不具合修正の期間は平均3.2日であり、改善にかかる工数は少ないと考える。

さらに、品質向上と生産性の向上のために、プロトタイプ作成工程も含め、どのような工程を進めるか、どの工程で何を決定し生産物として残すか、使いやすいUI設計等、支援ツールを利用しAPを開発する上でのガイドラインが有効であると考えます。

5. まとめ

GUI構築支援ツールとそれを利用し開発した業務APの品質および開発方法についての考察を述べた。GUIの品質向上には機能設計工程での開発メンバー全員のUI情報の共有化が重要であり、設計ツールおよび開発環境の改善が必要である。AP開発には品質に向けたガイドラインの提供が有効である。