

パターン・フレームワークを用いた組み込み GUI 開発効率化のための実践的アプローチ

久代 紀之^{††} 大澤 幸生^{††}

操作性向上を目的とし、民生機器への組み込み型 GUI(Graphical User Interface) の搭載要求が高まっている。市場の多様な要求に対応するため、民生機器は、複数機種を同時並行的に開発することが必須とされ、再利用技術を用いた開発プロセスの効率化が強く要求される。しかし、いざこれら技術を実務適用しようとするとき非常に困難が伴う。あいまいで抽象度の高い要求をフレームワーク上の具体的設計にどのようにマッピングするかという変換のギャップと組み込み GUI 設計に関わる専門技術者のコミュニケーションギャップという 2 つのギャップがあるためだ。本研究では、要求と実装間のギャップを補間する中間仕様記述と技術者間の協調を促進する共通知識フレームと左記フレームをベースとする開発支援ツールの整備によりこれらの課題の解決を試行した。本手法を複数種類の民生機器用操作器開発に適用し、手法の可用性と GUI 設計・評価プロセスへのインパクトについて評価した。

Practical Approach for improving GUI design efficiency with Pattern and Framework

NORIYUKI KUSHIRO^{††} and YUKIO OHSAWA^{††}

There is a growing interest to introduce graphical user interface (GUI) into consumer products for improving their usability. Design reuse technologies like pattern and framework are important for consumer products to satisfy various market requirements and short-term development demands in the consumer products domain. However, both transfer gap from requirements to their implementations and communication gap among engineers with different specialities, hinder the penetration of design reuse technologies to use. In this paper, we propose an integrated GUI design and evaluation knowledge frame to support cooperative design activities among specialists, and a intermediate document to compensate the gap between requirements and implementation to solve these issues. We confirmed their availability and impact for design and evaluation process of embedded GUI by applying them for plural of products design.

1. はじめに

民生機器メーカーにとって、多様な市場に柔軟に対応する製品群を短期に実現する設計再利用技術の導入が技術戦略上大きな位置を占めるようになってきている。設計再利用技術というと、ソフトウェア工学分野では、一見確立した技術として考えられている。1970 年代初頭より構造化分析・設計¹⁾、オブジェクト指向分析・設計²⁾を始めとして、パターン³⁾やフレームワーク⁴⁾⁶⁾、さらにはこれらの技術を統合した方法論であるソフトウェアプロダクトライン⁵⁾に至る多くの取り組みが

なされ、成果を挙げてきている⁷⁾。しかし、いざこれら技術を実務に適用しようとするとき困難が伴う。多様な要求をフレームワーク上の具体的設計にどのようにマッピングするかという変換ギャップと組み込み GUI 設計に関わる複数技術者間のコミュニケーションギャップがあるためだ。本研究では、特に設計再利用の要求が高まっている家電・設備機器など民生機器組み込み GUI (Graphical User Interface) を対象とし、要求と具体的 GUI 設計のマッピングを補間する中間仕様の導入と GUI 設計評価に関係する技術者間の共通知識フレームに基づく設計支援ツールの開発により、これらの課題の解決を試行した。さらに、本手法を複数の民生機器用 GUI 開発実務に適用し、その可用性と設計評価プロセスへのインパクトについて評価した。

[†] 三菱電機株式会社環境研究開発センター
Living Env. Systems Lab., Mitsubishi Electric Corp.
^{††} 東京大学大学院工学系研究科
School of Engineering, The University of Tokyo

2. パターン・フレームワーク適用開発の課題と解決策

2.1 課題

GUI 開発への再利用技術の適用に関しては、すでに種々の取組みがなされている¹³⁾。コンポーネント化されたソフトウェア部品とその動作環境（GUI フレームワーク）を利用し画面設計・実装を効率化するアプローチ¹¹⁾¹²⁾ や分析・設計・実装再利用のための各種パターンを統合した開発方法論⁸⁾ などがある。代表的なものである。筆者らも、民生機器用組込み GUI フレームワーク開発に取組み⁹⁾、いくつかの製品設計に適用してきた¹⁰⁾。これら適用を通じ、下記の2つのギャップが、パターンフレームワークを適用した GUI 設計・評価の開発効率を阻害する大きな要因として認識された。

- (1) 専門技術者間のコミュニケーションギャップ
GUI 開発は、システム設計者、画面設計者、プログラム設計者など多くの異なる専門技術者の協調により達成される。現在提供されているパターン・フレームワーク技術の多くは、それぞれの専門技術に特化したもの⁷⁾であり、専門が異なる技術者の協調作業を支援できる環境がない。
- (2) 要求仕様と画面設計・プログラム実装間の変換ギャップ

GUI 設計は、大きく要求・システム分析と画面・プログラム実装の2つのフェーズからなる¹¹⁾が、抽象度の高い要求・システム分析と具体的画面やプログラム実装間の大きな抽象度のギャップ、および文字を中心として表現された要求・システム仕様から絵や音を中心として表現される画面へのモダリティ変換という2つの大きな変換ギャップがある。これら変換ギャップにより、設計・評価のプロセスの連続性が失われ設計結果のトレースが難しくなるほか、工程間の仕様漏れや誤解から開発の手戻りが発生する。

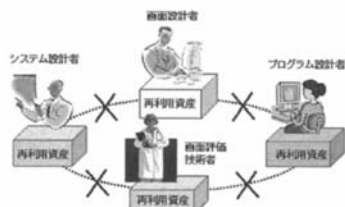


図1 技術者間のコミュニケーションギャップ
Fig.1 Communication gap between Engineers

2.2 解決策

本研究では、前述の2つのギャップを解決することを目標に、GUI 設計・評価に関する技術者協調のための共通知識フレーム（共通場）と共通場の構造に基づいた設計支援ツールおよび左記設計支援ツールの一部として要求・実装仕様間のギャップを補間する中間仕様（画面論理仕様書/記述手法）を開発した（図2）。図2に示す GUI 開発モデル上で、ユーザの多様な要求を逐次取り入れつつショートループでシステム設計・画面設計・プログラム実装・評価サイクルを繰り返すアジャイルな製品開発プロセス（図3）の実現を目標とする。



図2 共通場をベースとする GUI 開発モデル
Fig.2 GUI development model based on common field

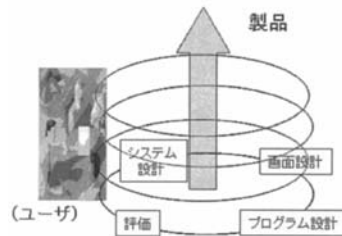


図3 共通場をベースとする GUI 開発プロセス
Fig.3 GUI development Process based on common field

3. 共通場

本研究では、GUI 設計に携わる専門家として、下記4種の技術者を想定する。

1. システム設計者: 要求からシステム仕様を設計
2. 画面設計者: システム仕様から画面仕様を設計
3. プログラム設計者: システム・画面仕様をベースにプログラムを実装
4. 評価技術者: 実装された画面の評価

Malone ら¹⁵⁾ らは、Coordination Process Model の中で、技術者間の協調成立の要件として、共通知識構造と共通言語の存在あげている。既存のパターン・フレームワークの多くは、分析²⁴⁾¹⁷⁾¹⁸⁾・設計³⁾¹⁹⁾・

実装²⁰⁾など、それぞれの作業領域毎に独自の構造と記述形式で開発されており、専門の異なる技術者間でこれら知識を直接伝達することは困難である。本研究では、技術者間コミュニケーションを介在する中間的な共通知識構造を設置することで上記課題を解決を目指す。各技術者は、自分の専門知識を共通知識構造上にマッピングし、中間的な共通知識構造を介在して他の技術者と協調する。この技術者間協調のためのゲートウェイの役割を行う共通知識構造を共通場と呼ぶ。

これら技術者による協調作業の成果は、絵を中心に表現された GUI 画面である。従って、これら GUI 設計に携わる設計者の“共通場”として、設計の最終ターゲットである“絵を描くという行為”に必要な知識構造を適用することが自然であると考えられる。出来栄を別にすれば、“絵を描くという行為”自体は、一般的にはそれほど難しいこととは考えられない。しかし、具体的にどんな知識を用いて“絵を描いているか”を説明できるかという問いに対しては完全に無力となるレベルに、“絵を描くという行為”は高度な創造的作業²¹⁾である。“絵を描くという行為”に必要な知識を説明する研究は管見では見当たらない。本研究では、“絵を描くこと”と類似の創造行為として捉えることが可能で、研究が先行する言語構造²³⁾を転用し、“絵を描く行為”に必要な知識構造を整理する。言語では、文章は、要素“文”から構成され、文は、要素“単語”から構成され、単語は要素“形態素”から構成され、形態素は、要素“音素”から構成されるとする。さらに、各要素間に、単語と形態素間、文と単語間の関係、文章と文との関係などを規定する統語関係や構文などの規則（文法）が存在するものとしている。この言語学のフレームを用いて GUI の各設計要素を下記のようにマッピングする（図 4）。

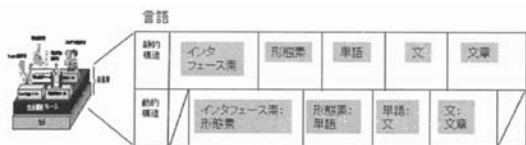


図 4 言語をベースとする GUI 設計・評価のための知識フレーム
Fig.4 Knowledge Frame for GUI design on linguistics

インタフェース素＝ ピクトグラム、効果音、文字等の基本部品

形態素＝ ボタンやアイコンなどの単機能 GUI 部品

単語＝ ラジオボタンやスライドスイッチなど単機能 GUI 部品を組合せた複合 GUI 部品

文＝ 単機能 GUI 部品あるいは複合 GUI 部品で構成されるある機能を実現するための画面

文章＝ あるシナリオを実現する複数の画面群

さらに、上記 GUI の各構成要素間の動的規則を下記のようにマッピングする。

インタフェース素：形態素＝ ボタン操作時のピクトグラム反転（ON 状態から OFF 状態を表示するピクトへの変更）などの単機能 GUI 部品の動作規則

形態素：単語＝ スライドスイッチを構成する単機能部品（スライドバー、インディケータ）におけるスライドバー操作時のインディケータへの値表示など複合 GUI 部品の動作規則

単語：文＝ ボタン操作時の画面ポップアップやコピー・ペーストなどの画面自身に対する操作に関する動作規則

文：文章関係＝ 一連のユースケースを実現する画面遷移の動作規則

4. 設計支援ツール

設計支援ツールとは、技術者の作業を支援しアウトプットされ仕様を前述の共通場（図 4）の上にマッピングするあるいは共通場を介在し受け取った仕様に基き技術者の作業を支援するツールをいう。設計支援ツールとして、表 1 に示すツールを開発した。

表 1 GUI 設計・評価支援ツール
Table 1 Supporting methods for GUI design and evaluation

対象設計者	設計支援ツール
システム技術者	画面論理仕様（中間仕様書）/記述手法
画面設計者	デザインガイドライン/GUI 部品カタログ
プログラム設計者	組込み型 GUI プラットホーム
画面評価者	デザインガイドライン/GUI 部品カタログ

4.1 システム設計者のための設計支援ツール

システム設計者の作業を図 5 に示す。システム設計者は、要求仕様から、システム設計書と画面論理仕様書（中間仕様書）を作成する。本研究では、システム設計者が、システム仕様から画面論理仕様を生成するプロセスとオブジェクト指向分析手法をベースとした画面論理仕様記述手法を定義した。

従来は、画面設計者は、システム設計者が作成したシステム仕様書を用いて直接、画面設計をしていた。

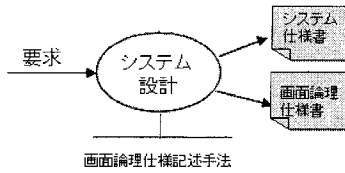


図5 システム設計プロセス
Fig.5 System Design Process

しかし、前述のギャップから、仕様漏れや誤解を生じ、本工程で多くの開発手戻りが発生していた。これらギャップを補完するために、本研究では、システム仕様書と画面仕様書間に、画面論理仕様書という中間仕様を置く。システム設計者は、画面設計者・評価者と協調し、システム仕様に基づき画面設計に必要な情報（一度に表示すべきオブジェクトやプロパティ、オブジェクト・プロパティ間の表示優先度等）を追加した画面論理仕様書を作成し、これを画面設計者との仕様受け渡しに用いることでコミュニケーション・変換の2つのギャップを補完する。

具体的な画面論理仕様書作成のプロセスを下記に記載する。

- Step1 オブジェクト・オブジェクト間関係の精緻化
 - Step11 サービス、属性値の詳細化
 - Step12 属性の取り得る値の詳細化
 - Step13 オブジェクト間関係の詳細化
- Step2 操作者とオブジェクト間インタラクションの詳細化
 - Step21 シナリオ毎のオブジェクト・属性とピクトとの論理的マッピング
 - Step22 属性の表示優先度と属性の集合（まとめて表示すべき項目）の抽出、
 - Step23 上記属性の集合とアイコン・ラベルとのマッピング
- Step3 論理的な画面設計・手順設計
 - Step31 ダイナミックな操作器設計、オブジェクト間の遷移（画面遷移）仕様の追加
 - Step32 キー属性の設定：インスタンスを識別するために必要となるキー属性を設定

画面論理仕様記述には、システム仕様設計で一般的に利用されているオブジェクト指向分析・設計ツールを拡張する。システム設計から画面論理設計までをオブジェクト指向という一貫した方法論にて設計することを可能とする。オブジェクト指向分析ツールとしては、UML²²⁾が現在主流の手法であるが、本研究では、画面設計者・評価者などシステム工学を専門としない技術者にとっても直感的でわかりやすいことを目標に、

記述手法がもっともシンプルな Coad²⁴⁾ によるオブジェクト記述手法を選択した。

Coad の仕様記述手法と画面論理仕様との対応関係を表2に整理する。

表2 Coad の仕様記述と画面論理仕様との対応関係
Table 2 GUI specification description methods

記述手法	画面論理設計仕様上の概念
シナリオ	GUI 操作シナリオ (文章)
クラス図	
オブジェクト間関係	画面間遷移情報 (文:文章関係)
オブジェクト	表示・操作対象 (文)
属性	表示・操作対象属性 (単語)
属性間関係	まとめて表示する情報 (単語・文)
オブジェクト・属性の優先度	表示優先度の情報 (文)
メソッド	表示・操作メニューに関する情報 (単語・文)
イベントトレース図	画面上のオブジェクトと人とのインタラクションあるいは画面の遷移情報 (単語:文, 文:文章関係)
データディクショナリ	操作・表示提示属性の定義 (インタフェース素・形態素・単語)

Coad の記述手法にて記述ができない属性間の関係、オブジェクト・画面の優先度等の画面論理仕様記述特有の記述内容に関して、これを記述できるように記法を拡張した。拡張した記述例を図6に示す。最終的に、システム仕様書上のオブジェクト、オブジェクト間の関係、属性、属性間の関係などの各概念は、本プロセスにより詳細化され図4上に示す共通場上の各概念に分解されそのマッピング関係が、画面論理仕様書上に記載されることとなる。

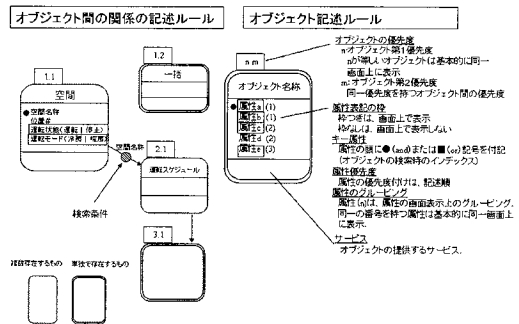


図6 画面論理仕様の記述例
Fig.6 Example of GUI specification description

4.2 画面設計者のための設計支援ツール

画面設計者の作業を図7に示す。画面設計者は、画

面論仕様書に基づき、すぐに利用可能な具体的部品で構成された GUI 部品カタログおよび左記 GUI 部品カタログを設計するためのメタなデザインルール（例えば使用する色や効果音の種類、ボタン形状やボタン操作時の動作、画面上への GUI 部品配置規則など人間・認知工学的な知見に基づくデザインルール）で構成されたデザインガイドラインの 2 種の再利用資産を利用して画面設計を行う。



図 7 画面設計プロセス
Fig. 7 Panel Design Process

GUI 部品カタログおよびデザインガイドラインは、共通場構造に従って、体系化され整理されている。デザインガイドラインの一例を図 3 に示す。新規 GUI 部品の設計が必要となった場合は、画面設計者は、デザインガイドラインに沿って GUI 部品を設計する。さらに、プログラム設計者・画面評価者と協調し、新規 GUI 部品の操作性の評価を実施後に GUI 部品カタログとして登録し、これを利用する。

表 3 デザインガイドラインの一例
Table 3 Example of design guideline

分類	デザインガイドライン
インタフェース素	
アイコン・ボタン	(1) 画面グリッド単位 ボタンの大きさ：5～8 グリッド角 アクションボタン：表示文字数で変化 (2) 16色パレット指定 ボタン：グレー基調 ウィンドウ：スカイブルー基調 ヘルプ、ガイド：緑色基調
素と単機能 GUI 部品間の関係	
操作確定	操作確定：ナビゲーションボタン 確定：○印 OK ボタン 取り消し：U ターンボタン

4.3 プログラム設計者のための設計支援ツール

プログラム設計者の作業を図 8 に示す。プログラム設計者は、システム仕様書と画面論仕様書を入力とし、GUI プラットホームを利用して画面を実装する。GUI プラットホームは、GUI クラスライブラリと

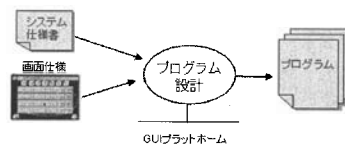


図 8 プログラム設計プロセス
Fig. 8 Program Design Process

クラスライブラリが動作する動作環境（ハードウェア・ソフトウェア）で構成されている。GUI クラスライブラリは、共通場と同一構成・粒度で構成されているため、プログラム設計者は、容易に画面論仕様をプログラムとして実装することが可能となる。

一般に GUI 方式操作系の実現には、高速 CPU や大容量メモリの搭載が必要とされ、低コスト化が要求される民生機器の操作系への適用が難しいという課題がある。本課題を解決するために、本研究では、省メモリでプログラム実行効率の高いソフトウェア・ハードウェアプラットフォーム（以下、GUI プラットホームと呼ぶ）を開発した⁹⁾¹⁰⁾。GUI プラットホームは、以下の技術で構成される。

- (1) 省リソース型オブジェクト指向 GUI-OS (Operating System)
- (2) GUI クラスライブラリ
- (3) 低コスト・高性能ハードウェアプラットフォーム

4.3.1 省リソース型オブジェクト指向 GUI-OS

GUI 開発には、インクリメンタルな開発プロセスの実現を目的に C++等のオブジェクト指向プログラミング言語の適用が一般的である。しかし、これら汎用的プログラミング言語は、言語の自体のオーバーヘッドが大きく、メモリ・CPU 制約の強い組み込み機器の実装には不向きである。省リソース型オブジェクト指向 GUI-OS では、オブジェクト指向プログラミング言語に必須の要素であるオブジェクトの管理機構とオブジェクト間のメッセージ伝達機構の 2 つの機構を、オーバーヘッドの小さい C 言語を用いて実現した。これにより、従来の C++等のオブジェクト指向言語での実装に比べ、1/100 程度の CPU 性能とメモリ容量という制約下においても、十分な操作性を実現できる GUI 動作環境を実現した (図 9)。

4.3.2 GUI クラスライブラリ

GUI クラスライブラリは、共通場と同一構成・粒度の約 100 種のライブラリで構成される (図 10)。結果的に、言語というアナロジーを適用することにより、プログラム設計者にとってクラスライブラリ構成の理解が直感的にできるようになった。また、言語持つ柔軟な構造が継承されることで、部品改定や部品追加などの改

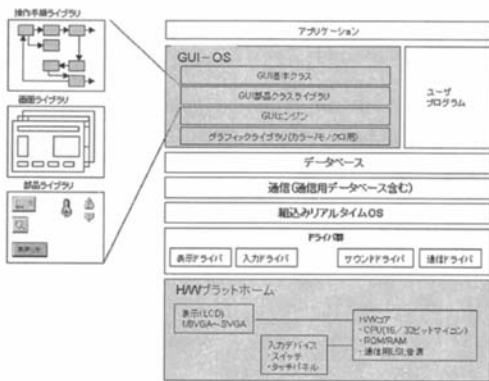


図 9 省リソース型オブジェクト指向 GUI-OS
Fig.9 Memory Reduced GUI OS

定や追加の影響がきわめて局所的で小さい再利用性の高いクラスライブラリを構成することができた。

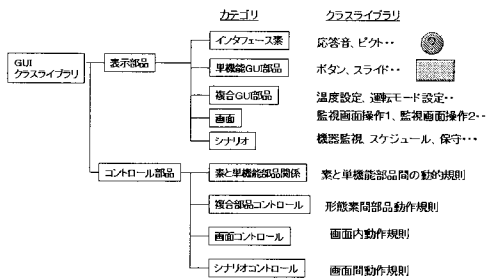


図 10 GUI クラスライブラリ
Fig.10 GUI class library

4.3.3 ハードウェアプラットフォーム

汎用的なマイコン、メモリ (ROM, RAM)、周辺ゲートアレイの4チップで構成された小形低コストな組込み型 GUI ハードウェアを開発した。小型化、部品としての扱いやすさを実現するために、LCD コントローラや通信インタフェースなどの周辺回路をゲートアレイ化した。また、音声出力やタブレット入力制御は、マイコンに内蔵された PWM (Pulse Width Modulation) 回路や A/D コンバータ回路を利用して構成し、周辺回路を極力削減することで、低コスト化を達成した (図 11)。

4.4 画面評価者のための設計支援メソッド

画面評価技術者の作業を図 12 に示す。

画面評価者は、画面設計者・プログラム設計者により実装された画面を、共通場上に整理されたデザインガイドラインを前提に操作性評価を行う。設計は、すべて共通場をベースとしてなされているため、実装された画面が、どういう要求・仕様に基づき、どの GUI

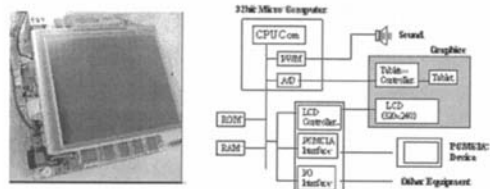


図 11 GUI ハードウェアプラットフォーム
Fig.11 GUI Hardware platform



図 12 画面設計技術者の作業
Fig.12 Panel Evaluation Process

部品カタログ/デザインガイドライン、GUI クラスライブラリを用いて具体化されたかという一連の設計プロセスのトレースが可能である。つまり、評価結果が、システム仕様上の問題なのか、画面設計の問題なのか、実装の問題なのかを迅速に特定することが可能であり、評価結果の反映のための手戻りを最小化することができる。実装上の問題であれば、課題はプログラム設計者に戻され、プログラム設計者はプログラムを改定する。画面設計上の問題の場合には、課題は画面設計者に戻される。画面設計者は、評価結果に基づいてデザインガイドラインを見直し、必要に応じて新規に GUI 部品を設計し、GUI 部品カタログをアップデートする。さらに新規 GUI 部品は、プログラム設計者により実装され、GUI プラットフォームに登録される。画面評価者による評価結果が、共通場の継続的なブラッシュアップを牽引する役割を担う。

5. 提案手法を用いた製品開発

対象・用途・規模が異なる4種類の民生機器用組込み型 GUI 図 13 操作器に、提案手法を適用し、その可能性と GUI 設計・評価プロセスのインパクトについて評価した。適用対象は、ビルに設置される設備用操作器1機種、住宅に設置される家電機器用操作器2機種¹⁰⁾、さらに家電用リモコン1種類の操作器である。

6. 評価

可用性に化しては、4種類の異なる対象・用途・規模の製品に適用できたことから満足していると考えられる。以下に4種の操作器適用を通じて獲得した設計・評価・実装へのインパクトに関し述べる。従来の GUI 設計・




名称	用途	機能	画像
ビル空調	ビル空調用集中コントローラ	・GUI表示 ・50台の空調機器のリアルタイムコントローラ ・インターネット遠隔監視機能	
ホームシステム	ホームシステム用集中コントローラ	・GUI表示 ・各種家電機器の集中コントローラ ・インターネットブロードウェイ機能 ・プログラムダウンロード	
リモコン	リモコン	・マルチメディア(表示、音、振動) ・数種類の家電機器	

図 13 適用アプリケーション
Fig. 13 Example of Application

評価の典型的なプロセスを図 14 に示す。技術者間協調の枠組みがなく、実装された画面以外に各技術者の設計結果を評価する有効な方法がないことから、一連の開発ループが大きく、開発の手戻りが大きかった。画面仕様は、要求の変化が大きく、実装しこれを見せた後に真の要求が判明することも多いため、どうしてもユーザを交えたインタラクティブな開発スタイルが必須とされ、この大きな開発ループが GUI 開発効率を大きく悪化させていた。

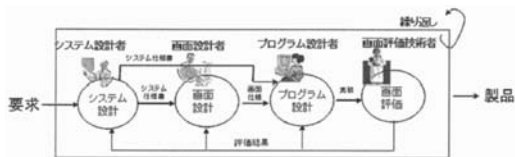


図 14 従来の GUI 設計・評価プロセス
Fig. 14 Legacy GUI design and evaluation process

本手法の適用により、図 15 に示すような、システム設計・画面設計・プログラム設計・評価の各ステップを手戻り少なく迅速にまわすこと（横のアジャイル）が可能で、同時に各ステップ毎に階層的に評価が可能（縦のアジャイル）な GUI プロセスを実現した。

- (1) 共通場を介して、システム設計から、画面設計・実装・評価までシームレスな開発プロセスを構築することで、仕様の漏れ・誤解による開発の手戻りを少なくした。さらに、各ステップでは、共通場上に整理された設計再利用資産を最大限に活用し、新規開発部分を最小化し、一連の開発プロセスを短期間にまわすことを可能とした（横のアジャイル）。
- (2) 各設計者が各設計ステップのアウトプット仕様の意味を解釈できるため、設計ステップ毎に階

層的な評価が可能となった。必要に応じて、各開発ステップの評価にユーザを参画させることで、ユーザ要求の変化を適宜取り込むことが可能となり開発の効率化を達成した（縦のアジャイル）。本研究では、評価方法として画面論理仕様書には、認知的ワークスルー²⁵⁾、画面仕様には、ペーパープロトタイプ²⁶⁾、実機を対象として、ユーザビリティ評価²⁷⁾を適用した。

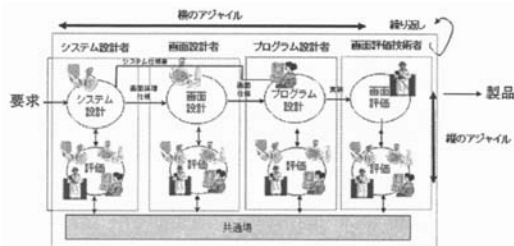


図 15 統合フレームワーク適用時の GUI 設計・評価プロセス
Fig. 15 Improved GUI design and evaluation process

設計・評価プロセスの効率改善を定量的に評価するために、ほぼ同等機能の操作器を対象とした従来プロセスによる開発と本手法による開発プロセスによる開発の際の開発工数の比較をした。結果を図 16 に示す。開発工数は、本提案手法適用によりほぼ半減できることが確認された。

- (1) システム設計：システム仕様の再利用により、システム設計フェーズの工数が半減した。
- (2) 画面設計：GUI 部品カタログからの部品再利用により大幅に工数を削減したが、デザインガイドラインを用いた新規 GUI 部品設計および評価が 10 件程度発生したため、全体としては、従来の開発と比較し、30 %程度の工数削減に留まった。
- (3) 画面実装：GUI クラスライブラリの利用により、実装・評価工数は、従来比 40 %程度に削減した。
- (4) 評価：予め評価された GUI 部品を再利用することで、操作性上の致命的な問題点が無くなった。階層的评价により、手戻りの工数を大幅に削減した。

7. おわりに

本研究では、要求と実装間のギャップを補間する中間仕様記述と技術者間の協調を促進する共通場および設計支援ツールの開発により、あいまいで抽象度の高

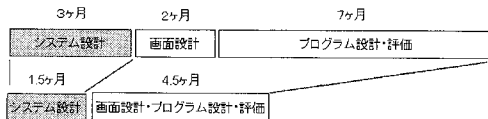


図 16 設計・評価プロセスの効率化
Fig. 16 Improvment of design and evaluation process

い要求をフレームワーク上の具体的設計にどのようにマッピングするかという変換ギャップと技術者間のコミュニケーションギャップという2つのギャップの解決を試みた。本提案手法を民生機器用操作器開発に適用し、手法の可用性と GUI 設計・評価プロセスへのインパクトについて評価し、下記を確認した。

- (1) 本手法は、用途・規模の異なる操作器に適用可能な可用性を有する
- (2) 本手法の適用により、ユーザ要求に対応しうるアジャイルな GUI 開発プロセスを達成し、従来の開発プロセスに比較し大幅な設計効率の改善（従来比設計効率約 2 倍）ができる。

今後、種々の操作器への適用を通じブラッシュアップを図るとともに、本設計支援ツールのコンピュータ支援を進めることで、さらなる効率化を図り、民生機器用へ GUI の適用の道を広げていきたいと考える。

参 考 文 献

- 1) E. Yourdon: Structured Design: Fundamentals of a Discipline of Computer Program and Systems Design, Yordon press, (1978).
- 2) J. Rumbaugh, Mi.Blaha and Wi. Premerlani: Object-oriented Modeling and Design, Prentice-Hall, (1991).
- 3) E.Gamma, R.Helm, R.Johnson, and J.Vlissides: Design Patterns: Elements of Reusable Object-Oriented Software, Addison-Wesley, (1995).
- 4) T.Reenskaug: Working with Objects: The OORam Software Engineering Method, Manning, (1996).
- 5) K.Poul, G.Bockle and F.V.Linden: Software Product Line Engineering: Foundations, Principles and Techniques, Springer-Verlag, (2005)
- 6) C.Kruger.: Software reuse, ACM Computing Surveys, 24(2) pp131-183, (1992).
- 7) 中谷 多哉子, 青山 幹雄, 佐藤 啓太: ソフトウェアパターン, bit 別冊 11 月号, 共立出版, (1999).
- 8) C.Stary: TADEUS:seamless development of task-based and user-oriented interface, IEEE Transaction on Systems, Man and Cybernetics, Vol.30,No.5, pp.509-525,(2000).

- 9) N.Kushiro et.al.: Agent Based Graphical User Interface Architecture for Consumer Products, IEEE Transaction on Consumer Electronics, pp.1044-1049(1996).
- 10) N.Kushiro et.al.: Integrated Residential Gateway Controller for Home Energy Management System, IEEE Transaction on Consumer Electronics,pp.629-636 (2003).
- 11) E.Kantrowitz, A.Lyakas and A.Myasqobsky; A User Interface Framework, Proceedings of the IEEE international Conference on Software-Science, Technology and Engineering,(2003)
- 12) T.N.Q.Tram, C.G.T.Tai and D.T.B.Thuy: User Interface Design pattern management system support for Building Information System, IEEE,(2006)
- 13) A.D.Granlund et al.: A pattern supported approach to the user interface design process, 9th International Conference on Human Computer Interaction, (2001)
- 14) S.Srinivasan: Design Patterns in Object-Oriented Frameworks, IEEE Computer, Feb,(1999)
- 15) T.Malone and K.Crowston:The Interdisciplinary Study of Coordination, ACM Computing Surveys, Vol. 26, No.1, March, (1994).
- 16) P.Coad, D.North and M.Mayfield:Object Models: Strategies, Pattern and Applications, Prentice Hall,(1995)
- 17) D.Hay:Data Model Patterns: Conventions of Thought, Dorset House,(1996).
- 18) M.Fowler: Analysis Patterns:Reusable Object Models, Addison-Wesley,(1996).
- 19) B.P.Douglass:Real-Time UML, Addison Wesley, (1998).
- 20) F.Buschmann et al.:Pattern Oriented Software Architecture: A System of Patterns, John Wiley and Sons, (1996).
- 21) R.A. Finke, B. Thomas and S.M. Smith : Creative Cognition, MIT press, (1992).
- 22) UML: <http://www.uml.org/>.
- 23) F.D. Saussure: Writings in General Linguistics, Oxford Univ. press, (2006).
- 24) P.Coad: Object Oriented Analysis Yourdon press, (1990).
- 25) J. Nielsen et al.: Usability Inspection Methods, John Wiley and Sons, (1994).
- 26) C. Snyder: Paper Prototyping: The fast and Easy Way to Design and Refine User Interfaces, Morgan Kaufman, (2003)
- 27) 黒須正明: ユーザビリティテスト, 共立出版, (2003).