

## C#における冗長なイベントハンドラの合成ツール

谷川郁太†、石井貴大†、鈴木智明†、渡辺晴美†

C#の統合開発環境では、Form や UserControl にイベントハンドラを追加していくうちに、冗長なイベントハンドラが多く出来てしまう問題がある。組込み Windows 向けのシステムを開発する際も、統合開発環境から画面設計を行うとこのような問題に陥る可能性が有る。この問題を解決するために、イベントハンドラを統合する機能を持ったリファクタリングツールを開発した。評価は、デバイスとクラウドを使用するシステムのコンテスト作品である募金箱と Twitter との連動システム「ぶたツイ」で行った。本展示では、リファクタリングツールを紹介し、「ぶたツイ」システムの適用デモンストレーションを行う。

## A Refactoring tool for Reducing C# Event Handlers

Ikuta Tanigawa†, Takahiro Ishii†, Tomoaki Suzuki†, Harumi Watanabe†

Many similar event handlers are created as constructing FORM or UserControl on IDE of C#. In screen design of Embedded system, the event handler problems happen with IDE. In order to solve this problem, we developed the refactoring tool with the function which unifies event handlers. To evaluate the refactoring tool, it was performed by ButaTwi. ButaTwi is a system of the contest which proposes the service that combines equipment with Cloud, and also cooperates a collection box and Twitter. In this exhibition, we introduce a refactoring tool and perform the application demonstration of ButaTwi.

### 1. はじめに

ソフトウェアの保守は、一般的に冗長な箇所が増えるほど困難になっていくことが知られている。ソースコード上の冗長な箇所を取り除くリファクタリング手法は数多く考えられてきた[1][2]。

C#の統合開発環境では、Form や UserControl にイベントハンドラを追加していくうちに、冗長なイベントハンドラが多く出来てしまう問題がある。冗長なイベントハンドラは、メソッドの抽出により冗長な箇所をまとめても、イベントハンドラ自体は残ってしまうため、新たなリファクタリング手法が必要である。本展示では、上記の問題を解決するための冗長なイベントハンドラを統合するリファクタリング手法とツールを紹介し、適用のデモンストレーションを行う。

### 2. 冗長なイベントハンドラの例

統合開発環境からイベントハンドラを追加していくと、そのたびに新たなイベントハンドラの定義が追加されるため、まったく同じ、またはリテラルや参照などが異なる

だけのイベントハンドラが複数できる問題がある。冗長なイベントハンドラの詳細例を以下に示す。

- (1) 設定ウインドウにおいて、内容が変わった際に適用ボタンを有効にするイベントハンドラがコントロールごとに行えるケース。
- (2) 電卓において、ボタンごとに異なる数値を入力するイベントハンドラが行えるケース。
- (3) ダイアログから指定したファイルのパスをテキストボックスに表示するボタンが複数ある場合、ボタンごとにイベントハンドラが行えるケース。

### 3. 冗長なイベントハンドラの統合手順

イベントハンドラの統合により、冗長なイベントハンドラを取り除くことができる。上記例において、ケース(1)では適用ボタンを有効にするイベントハンドラがコントロールごとに行える。これらの内容はまったく同じである。このように、同じ内容のイベントハンドラを統合する手順を以下に示す。

- i. イベントハンドラを渡している箇所を確認する。
- ii. 手順 i で渡されていたイベントハンドラの中身を全て比較し、統合できそうなイベントハンドラ(デリゲートが同じで、中身がコメントとローカル

†東海大学情報通信学部, Tokai University School of Information and Telecommunication Engineering

- iii. 変数名以外同じイベントハンドラ)を調べる。
- iv. 手順 ii で見つけた統合できそうなイベントハンドラのうち、適当なものをコピーして新たなイベントハンドラを作る。
- v. 統合対象のイベントハンドラを渡す箇所を手順 iii のイベントハンドラを渡すよう書き換える。
- vi. テストし、動作が変わってないことを確認する。
- vii. 古いイベントハンドラを全て削除する。

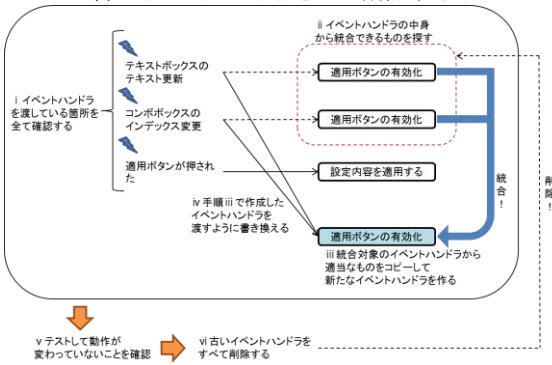


図1 冗長なイベントハンドラの統合手順

一方、ケース(2)~(3)ではそれぞれ、リテラル、参照がイベントの発生元によって異なる。このようなイベントハンドラを統合するには、リテラル、参照が異なる箇所をイベント発生元に応じた値を返す処理に置き換える。図2にケース(2), (3)のイベントハンドラ統合例を示す。

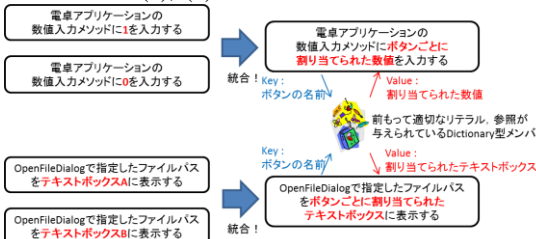


図2 ケース(2), (3)のイベントハンドラ統合例

#### 4. 評価

我々は3章で述べたイベントハンドラ統合を行う機能を持つリファクタリングツールを開発した。本ツールの評価を、D2C コンテストの作品である募金箱と Twitter の連動システム「ぶたツイ」で行った。D2C コンテストはデバイスとクラウドを用いるシステムのコンテストである。

「ぶたツイ」は入金を検知するための通過センサを搭載しており、入金が確認されたら、Twitter に入金されたことをつぶやく。さらに、木が成長するアニメーションの表示と募金額推移を表すグラフの更新を行う。表1に、「ぶたツイ」の構成、図3に動作の様子を示す。

本展示では、この「ぶたツイ」への適用デモを行う。適用対象は画面遷移を行うイベントハンドラと、設定画面中のイベントハンドラとする。前者は画面遷移メソッドの引数である遷移先画面名が違っただけで、後者は2章の例のように、内容が変わった際に適用ボタンを有効にするイベントハンドラがコントロールごとになってきている。

表1 ぶたツイの構成

|                 |                            |
|-----------------|----------------------------|
| CPUボード          | Armadillo-440              |
| OS              | Windows Embedded Compact 7 |
| 開発ツール           | Visual Studio 2008         |
| 言語              | C#                         |
| .NET            | .NET Compact Framework 3.5 |
| 行数(空行, コメント行除く) | 6718                       |



図3 ぶたツイの動作の様子

#### 5. おわりに

本展示では、C#における冗長なイベントハンドラを統合する機能を持ったリファクタリングツールを紹介し、「ぶたツイ」への適用デモンストレーションを行う。

組込み Windows 向けのシステムを開発する場合、画面設計に統合開発環境を用いると効率的に作れる反面、冗長なイベントハンドラが出来てしまう。提案ツールを用いることで、2章で述べた問題点を解決できる。

#### 参考文献

- [1] M. Fowler, K. Beck, J. Brant, W. Opdyke, D. Roberts, Refactoring: Improving the Design of Existing Code, Addison-Wesley Professional, 1999  
(児玉 公信, 友野 晶夫, 平澤 章, 梅沢 真史(訳), リファクタリング プログラムの体質改善テクニック, 2000)
- [2] 肥後 芳樹, 神谷 年洋, 楠本 真二, 井上 克郎, コードクローンを対象としたリファクタリング支援環境, 電子情報通信学会論文誌 Vol.J88-D-I, No.2, pp.186-195, 2005