

.Net Framework 上における独創的な GUI コンポーネントの開発

Rich GUI components in originality on the .NET Framework

湯川 洋平¹⁾ 松本 曜²⁾
Yohei YUKAWA You MATSHMOTO

- 1) 京都大学理学部 4 回生 (〒606-8502 京都市左京区北白川追分町
E-mail:yukawa@physics.mbox.media.kyoto-u.ac.jp)
- 2) 京都大学工学部情報学科 3 回生 (〒606-8501 京都市左京区吉田本町)

ABSTRACT. These days the GUI (Graphical User Interface) of the computers is widely used. The expanding applications of computer need new style GUI components. The Microsoft .NET Framework makes it easy to develop GUI components which is high portability. We have developed two GUI Components on the .NET Framework. The former is "ZoomBar" that helps you to select the range you are interested in when you deal with the scrollable data. The later is "FindApplet" which consists of flexible find framework and rich visual component.

1. 背景

パーソナルコンピュータ(PC)で広く用いられているグラフィカルユーザインターフェイス(GUI)アプリケーションの開発は、GUI コンポーネントと呼ばれる部品を機能的に配置することによって実現されることが多い。一般にこれらのコンポーネントはライブラリとして提供され、再利用と共通化により操作体系の統一と開発コストの低下が図られている。

新規に GUI コンポーネントを作成するときには、再利用と共通化が容易なように配慮することが重要である。これを実現するために、従来の Microsoft Windows では、Active テクノロジーが利用されてきた。Active テクノロジーの特徴は、COM(Component Object Model)によって言語に依存しない GUI コンポーネントの作成を可能にする点である。これは例えば Microsoft Visual Basic で作成された GUI コンポーネントが、Borland C++ Builder で利用できるといったことを意味する。このような特徴を持った Active テクノロジーを利用した GUI コンポーネントは広く普及したが、開発にある程度の技術が必要なこと、セキュリティの確保や作成したコンポーネントの配布において制限が大きいことといった問題があった。

現在 Microsoft は、Active テクノロジーの問題点を解消し、より広範囲なソフトウェア開発基盤となる .NET Framework の普及を進めている。 .NET Framework の上で作成された GUI コンポーネントは、今までよりも容易に開発・再利用が可能となっている。また、ネットワークの利用や XML などの連携が少ないコードの記述で行えるようになってきている点も重要である。

一方で近年の WWW の爆発的普及は、Web サイトのデザインという形で多くの人々をサービスの利用者からサービスの提供者に変えた。HTML と ECMA Script、及びサーバサイドのプログラムを適切に結びつけることにより、実質的に GUI のアプリケーションを誰でも作ることが可能な時代が到来している。興味深いのは、より多くの人々がデザインを行うようになった結果、Web においては必ずしも GUI のルックアンドフィールの統一がなされなくなった点である。 Cascading Style Sheets や Macromedia Flashなどを駆使した新しい GUI が次々と生まれるようになってきた現在、新しいスタイルの GUI が

受け入れられる下地は整ってきていると考えられる。

2. 目的

本研究開発では、.NET Framework 上での新しい GUI 部品の開発と普及活動を行うことを目的とする。前章に挙げた状況をふまえ、以下の 2 つのコンポーネントの開発を行った。

- (1) ZoomBar
- (2) FindApplet

3. ZoomBar

(1) 概要

一画面に収まらないデータを扱う際にはスクロールバーがしばしば用いられる。しかしながら非常に大規模な情報を通常のスクロールバーで操作することは困難を伴う。Cakewalk(*)など、このような場合一般には拡大率を変化させるボタンやショートカットなどを併用する方法がよく使われている。今回開発したコンポーネントは、スクロールバーのつまみの両端を持って移動させることで、注目している表示位置と拡大率の変更を同時に行うことが出来るのが特徴である。

(*)Cakewalk 社の MIDI/オーディオシーケンズソフト

(2) 設計

GUI コンポーネントとして広く使われているスクロールバーを使用しているアプリケーションなら、簡単に導入できることを目標とした .NET Framework で使用されているスクロールバーとほぼ同じプロパティを持ち、アプリケーション開発者は少ない修正で ZoomBar を導入することが出来る。つまみのサイズが変更されたことはイベントによりアプリケーションに通知される。アプリケーションは、つまみのサイズに応じて表示内容を更新すればよい。またつまみサイズが小さくなりすぎたり大きくなりすぎたりしないように、サイズの上限下限を設定できるようになっている。

(3) 実装

実装にはスクロールバーを継承するのではなく、新規

にコントロール継承コンポーネントとして作成した。これはマウス操作の細かい制御を行いたかったためであるが、そのことによってスクロールバーとの互換性があまり損なわれないように特に注意した。

4 . FindApplet

(1) 概要

アプリケーション内の文字の検索に用いられる検索ダイアログには、検索する文字列と、上方向か下方向を選択するラジオボタン、検索を実行するボタンとキャンセルボタンを備えるスタイルのものがある。このようなダイアログは文章中から特定の一つの単語を検索するには適しているが、一度に前後ひとつ分の検索・移動しかできないため、同じ単語が複数使われているような長い文章中を調べまわるのは向いていない。今回開発したコンポーネントは縦断的な検索を Web の検索サービスのような手軽さで提供する。そのため、なるべく対象の資料を邪魔しないような見た目と、複数の検索結果を分かりやすく表示選択できることを目指した。

(2) 設計

初期状態では文字を入力して検索を実行するために必要十分な要素のみを表示し、必要に応じて検索の細かい設定が可能にできるようにする。デフォルトでは単語の指定に Web と同じ書式を用い、検索サイトと同じ感覚で使用できるようにする。またあいまい検索も可能にする。検索結果は見つけ次第リストに追加・表示されてゆき、全体の中でどれくらいヒットし、文章中での候補の相対的位置関係が容易に分かるようにする。また検索は非同期的に行われるようにしユーザが操作不能な状態が無いようにする。アプリケーション実装者が検索をコンポーネントに任せるか検索・評価アルゴリズムを独自に実装するか選択できるようにし、応用性を高めてある。

(3) 実装

ダイアログとしてではなく埋め込んだ形でも利用できるように、コントロール継承コンポーネントとして作成した。検索・評価関数はマルチスレッドを用いてバックグラウンドで実行され、マシンパワーが十分にある場合は並列実行を行う。検索・評価アルゴリズムは、コールバック関数を指定することでカスタマイズ可能となっている。スペース区切りの検索結果に対するユーザのアクションはイベントとしてアプリケーションに通知するようにした。

5 . 今後の課題

ユーザビリティ調査が不十分なことから、今後ユーザのフィードバックを得て完成度を高めてゆきたい。また、アイデア段階で実装に至らなかったコンポーネントがいくつかあるので、それらについても順次開発を行ってゆきたい。

6 . まとめ

本研究開発では .NET Framework を利用し上記のような特徴を持った GUI コンポーネントを作成することが出来た。今回得られたノウハウを元に今後も新しい GUI コンポーネントを提案していくことが可能になったと考えている。

7 . 参加企業

株式会社 TNT (プロジェクト管理組織)

8 . 参考文献

[1] Microsoft : .NET Framework Ver. 1.0 (2001)
<http://msdn.microsoft.com/library/default.asp?url=/nhp/Default.asp?contentid=28000451>

[2] Ian Griffiths : Give Your .NET-based Application a Fast and Responsive UI with Multiple Threads, MSDN Magazine (2003)
<http://msdn.microsoft.com/msdnmag/issues/03/02/Multithreading/default.aspx>