

## 柔軟にデザインを変更できる WebベースGUI部品ライブラリ

### 設計自由度を向上させ、多様なGUIデザインを容易に実現

GUI (グラフィカルユーザーインターフェース) は、パソコンやスマートフォンなどのデバイスの入出力手段として欠かすことができません。Webブラウザ上で表示されるWebベースGUIを搭載するデバイスの魅力を向上させるには、優れた操作性だけでなくGUIのデザインにも差異化が求められています。しかし、WebベースGUIの開発で使用できる既存の部品ライブラリでは、提供される部品の色や大きさは変更できても、枠などの飾りを追加するようなデザインの変更はできません。

東芝は、従来よりもデザインを自由に変更できるWebベースGUIの部品ライブラリを実現できる技術を開発しました。これを用いることでWebベースGUIでも多様なデザインを容易に実現できるようになります。

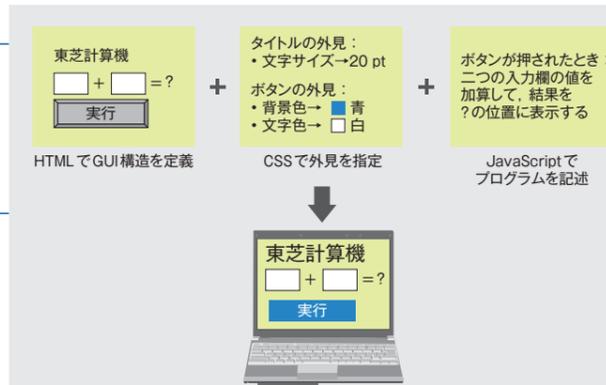


図1. WebベースGUIの概要 — WebベースGUIは、HTML、CSS、及びJavaScriptを組み合わせて開発されます。

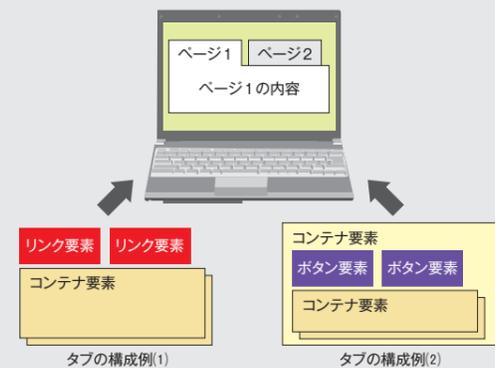


図2. 異なる要素の組合せで作られたタブの例 — 要素の組合せや構造が異なっても、タブの外見や機能は変わりません。

例えば、マウス操作に応じてHTMLの定義を書き換え、表示を切り替えるプログラムを記述できます(図1)。

#### 部品ライブラリの必要性

HTMLは本来Webページを作成する言語であるため、GUIでよく見られるタブやダイアログボックスなどの部品を直接表現するHTMLの要素はありません。足りない部品は要素を組み合わせで作ります。その際、部品の表示を変化させるプログラムも併せて作らなければなりません。

また、外見や機能は同じ部品でも異なる要素の組合せで作られる可能性もあります(図2)。部品の構成が異なると、部品の表示を変化させるプログラムを部品の構成ごとに作らなければなりません。

ません。

毎回プログラムを作成することを避けるため、GUIでよく使用される部品をあらかじめ作成して提供しているものがWebベースGUIの部品ライブラリです。これまでに商用からオープンソースまで様々な部品ライブラリが開発されています。しかし、既存の部品ライブラリが提供する部品は外見を自由に変更できず、デザインを妥協しなければなりません。

#### 既存の部品ライブラリの課題

部品ライブラリが提供する部品の外見は、通常、CSSで指定します。しかしCSSでは、飾りの追加や複雑な形状の指定はできません。これを実現するには、部品の要素構造を変える必要があ

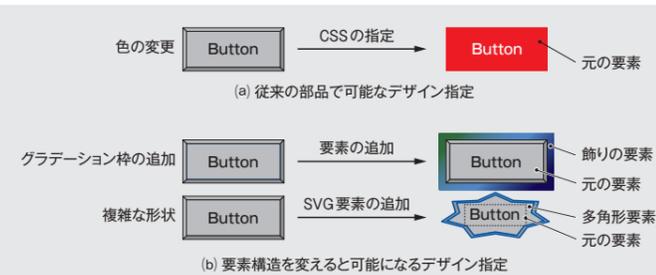


図3. 要素構造を変えると可能になるデザイン指定の例 — 部品の要素構造を変えると、グラデーション枠を追加したり、複雑な形状にしたりできるようになります。

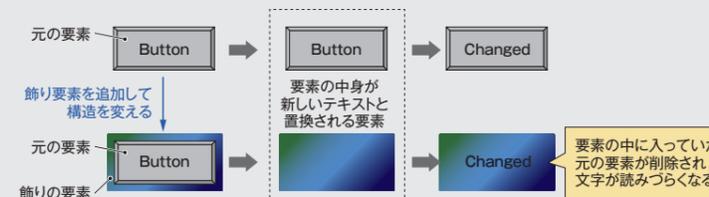


図4. 構造を変えた既存の部品でテキスト変更操作を実行した例 — 既存の部品ライブラリが提供する部品では、要素の構造を変えると部品の操作が期待どおりに動かなくなります。



図5. 役割を明示的に指定した部品でテキスト変更操作を実行した例 — テキストを表示する要素を明示的に指定したため、部品の操作が期待どおりに動きます。

ります。例えば、飾りを表現するための要素や、ベクトル画像の規格であるSVG (Scalable Vector Graphics) で定義された要素を追加します(図3)。

既存の部品ライブラリが提供する部品では、前述したようなデザインを指定できません。要素構造が変わらない前提で部品のプログラムが作成されており、飾り要素を追加できないからです。例えば、図4のようにボタンのテキストを変更する操作が「もっとも外側にある文字列要素の内容を変更する」ように作られているとします。このボタンに飾り枠の要素を追加すると、テキストの変更によって外見が崩れてしまいます。

東芝は、要素構造を変えても外見が崩れない技術を開発し、この問題を解消した部品ライブラリを作成しました。

#### 多彩なデザインを指定できる 部品ライブラリの実現

まず、部品に「役割」という概念を導入しました。全ての部品は一つ以上の役割を持ち、部品を構成する要素がその役割を担います。例えばボタンなら、「クリックを受け付ける役割」と「テキストを表示する役割」が部品の役割として挙げられ、一つの要素がこの二つの役割を担います。これと同様にして、他の部品でも役割を抽出し体系化しました。

次に、HTMLの作成者が部品を構成する各要素に任意の役割を指定できるようにしました。図4に示したボタンの例では、外側の要素に「クリックを受け付ける役割」を、内側の要素に「テキストを表示する役割」を指定すれば、テキ

スト変更時に外見が崩れません(図5)。

要素への役割の指定には、要素の属性を指定するクラス属性というHTMLの仕様を使います。クラス属性に任意のクラス名を設定して要素を分類するものです。ここでは、各役割に対応するクラスをあらかじめ定義しています。要素への役割の指定は、この定義済みクラスの名前を要素のクラス属性に設定します。例えば、「クリックを受け付ける役割」を指定するのであれば、「button-clickable」というクラス名を要素のクラス属性に設定します。

今回開発した技術では、HTML作成者も要素構造の変化を伴うデザインを部品に指定でき、既存の部品ライブラリでは実現できなかった様々なデザインの部品が容易に実現できるようになります。タブのつまみを台形にすることも、ボタンの選択状態をグラデーションのかかった枠で表現することも可能です。この技術を採用した部品ライブラリにより、その利便性を保ったまま、デザイナーが作成したGUIデザインを妥協することなく忠実に再現できます。

#### 今後の展望

当社は、役割という概念を導入することで、プログラムを修正せずに柔軟なデザイン指定ができるWebベースGUIの部品ライブラリを開発しました。今後はこのライブラリを多くのデバイスで使えるように、部品の種類を増やすとともに、役割のいっそうの細分化も進めていきます。また、HTMLを使用した外見も、CSSのように手軽に差し替えられるツールの開発も予定しています。

日野 克哉

ソフトウェア技術センター  
先端ソフトウェア開発担当

(注1) JavaScriptは、Oracle Corporation及びその子会社、関連会社の米国及びその他の国における登録商標。