

Web ページ制作入門において利用可能な 関連情報を提示するための GUI コンポーネント

佐藤 信¹

概要: 本稿では、関連づけられた情報を Web ページにおいて提示することを目的としてカスタマイズした GUI コンポーネントを提案する。提案のコンポーネントを用いると、HTML 標準規格に定義されるビルトイン要素と同様の記述により、複数のコンポーネントを連携させることにより関連情報を提示することが可能である。それにより、Web ページ制作の入門コースなどにおいても提案のコンポーネントの利用が容易である。どのように複数のコンポーネントを連携させると効果的であるのかを検討することは、情報を提示する場合の適切な順序について考える機会としても重要である。提案のコンポーネントは、汎用的な利用を目的として設計したものであるが、プログラミングの入門コースでの利用にも適しているといえる。

GUI Components for Providing Associated Information Usable for Introduction to Web Page Creation

MAKOTO SATOH¹

Abstract: This paper presents GUI components customized for providing associated information on Web pages. Using the components, associated information can be provided with the cooperation of multiple components by employing the same notations as built-in elements defined in the HTML standard. It makes that the components can be used with ease in the introductory course to Web page creation. Considering the effective way to cooperate multiple components is important as the opportunity to think about the suitable sequences to provide information. In addition to the fact that the components are designed for general use, the components are suitable for use in the introductory course to computer programming.

1. はじめに

本稿では、関連づけられた情報を Web ページにおいて提示することを目的として、カスタマイズした GUI コンポーネント (本稿では、Associated Components と呼ぶ。) を提案する*¹。特徴は、次のとおりである。

- HTML プログラムにおいて、HTML 標準規格 [10] に定義されるビルトイン要素 (HTML ビルトイン要素) と同様の記述により、複数の GUI コンポーネントを

連携させて関連情報を提示する Web ページの制作が可能である。

- コンポーネント間で受け渡される情報の直接的な記述により、簡潔なプログラミングが可能である。

提案のコンポーネントは、一般的な利用を目的として設計したものであるが、Web ページ制作の入門コースなどにおいても利用が容易である。複数のコンポーネントの効果的な連携について検討することは、情報を提示する場合の適切な順序について考える機会としても重要である。

これ以降の構成について簡単に説明する。2 節では、提案手法と関連研究との比較をおこなう。関連づけられた情報を提示するための GUI コンポーネントである Associated Components の設計を、3 節において提案する。4 節では

¹ 岩手大学

Iwate University, Ueda, Iwate 020-8551, Japan

*¹提案手法の一部分を実装したものを、次の URL で公開予定である。

<https://blue0.an.cis.iwate-u.ac.jp/AssociatedComponents>

Associated Components の実装を用いた GUI の例を示し、検討をおこなう。最後の 5 節において本稿のまとめと今後について述べる。

2. 関連研究との比較

2.1 GUI による関連情報の提示

Web ページにおいて提示される情報には、相互に関連をもつものが多いといえる。例えば、ボタン要素にカーソルを移動した場合に表示されるツールチップ*2 [2] [8] は、そのボタンを押下した場合におこなわれる動作についての説明を動的に提示するためなどに用いられる。そのようにユーザの操作に対応して GUI 要素が反応するという一種の対話は、マイクロインタラクション (microinteractions) [9] であり、GUI 要素の機能を予測するためなどに重要な役割を担っている。

Associated Components は、複数の HTML 要素の動作を連携させたマイクロインタラクションをおこなうための GUI 要素であるといえる。Associated Components を用いると、HTML ビルトイン要素と同様の記述方法によりプログラムを作成するだけで、複数の HTML 要素の動作に関連をもたせることが可能である。例えば、ボタン要素に簡単な説明が表示されている場合に、そのボタン要素をカーソルによりホバーすると、関連づけられた別の HTML 要素に補足のための情報を表示することが可能である。HTML ビルトイン要素に標準的に備わっているツールチップの機能では、カーソルをホバーした HTML 要素そのものに情報が関連づけられるが、Associated Components では、カーソルをホバーした HTML 要素と他の HTML 要素の動作を関連づけることが可能であり、多様なパターンによる関連情報の提示が可能である。

また、Web ページ制作の入門コースにおいて Associated Components を用いると、提示する情報をどのように関連づけるのが適切であるのかについて、Web ページを実際に制作しながら試行錯誤することが容易である。

2.2 HTML 要素のカスタマイズ

ソフトウェアの専門家がプログラムを作成する場合には、ソフトウェア・コンポーネント (software components) を再利用するという考え方は一般的である [4] [5]。

アプリケーション・ソフトウェアなどの GUI 要素については、再利用可能なソフトウェア・コンポーネントとするための手法が多数発表されている [1]。HTML 要素は、小規模なソフトウェア・コンポーネントであり、HTML 要素を組み合わせることにより Web ページの制作がおこなわれる。しかし、HTML 要素そのものをカスタマイズして新

*2 ツールチップは、ポップヒントなどとも呼ばれる。バルーンは、吹き出しにより同様の機能を実現したものである。

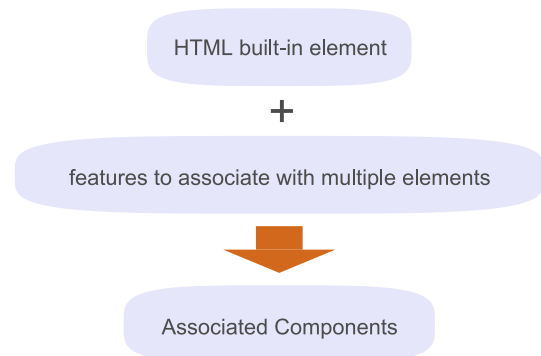


図 1: Associated Components の基本的なコンセプト
Fig. 1 Basic Concept of Associated Components.

たなコンポーネントを作成するための機能が HTML 標準規格に含まれたのは最近のことである。

Associated Components では、HTML 標準規格に定義されるカスタム要素*3 *4 を作成するための機能を用いることにより、カスタマイズされた GUI コンポーネントを作成する*5。

3. Associated Components の設計

3.1 設計の概要

Associated Components の設計の基本的なコンセプトを、図 1 に示す。HTML ビルトイン要素がもつ機能を維持しながら、複数の HTML 要素が連携することにより関連情報を提示するための機能を追加することによりコンポーネントを構築している。

コンポーネントの設計では、HTML 標準規格に含まれる自律的カスタム要素*6 [7], [10] を作成するための機能を用いて HTML ビルトイン要素の機能を継承している。それにより、HTML 標準規格の文法を用いることにより、HTML ビルトイン要素と Associated Components を共存させて Web ページを制作することが可能になっている。

例えば、ボタン要素のための Associated Components を用いる場合には、HTML 標準規格に含まれるボタン要素のためのタグ名に加えて Associated Components 用のタグ名を使用可能である。また、Associated Components どうしを連携させた動作が可能であるばかりではなく、Associated

*3 HTML 標準規格に定義されるカスタム要素 (custom elements) を用いると、カスタマイズした GUI コンポーネントを作成できる。カスタム要素のことを、Web コンポーネント (Web Components) と呼ぶことも多い [7]。

*4 カスタム要素に関連する研究である [6] では、GUI コンポーネントを定義するための DSL (Domain Specific Languages) を用いて GUI コンポーネントを生成している。

*5 Polymer [3] を用いると、HTML 標準規格のカスタム要素の機能を用いる JavaScript のプログラムを簡略化できる。

*6 HTML 標準規格のカスタム要素には、自律的カスタム要素 (autonomous custom element) とカスタマイズしたビルトイン要素 (customized built-in element) がある。

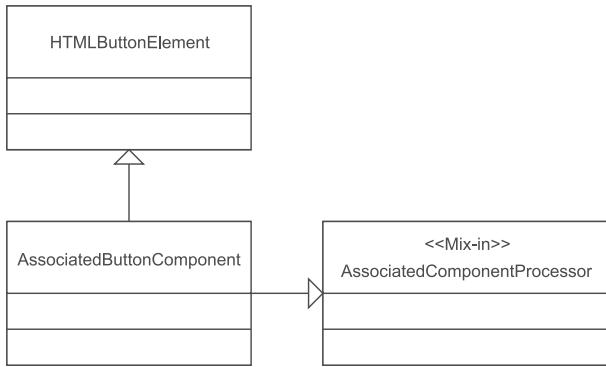


図 2: Associated Components のクラス構造の例

Fig. 2 An example of Associated Components class structure.

表 1: Associated Components のカスタム属性

Table 1 Custom attributes of Associated Components.

attributes	abbreviations	description
data-receiverId	data-rId	Specify receiver identifiers.
data-groupId	data-gId	Specify group identifiers.

Components と HTML ビルトイン要素を連携させた動作も可能である。後者の場合には、HTML ビルトイン要素については HTML 標準規格の仕様による記述のみでよい。

3.2 クラス構造*7

図 2 は、ボタン要素のための Associated Components のクラス構造である。AssociatedButtonComponent(ABC) クラスは、HTMLButtonElement(HBE) クラス*8 から派生させたクラスであり、HBE クラスからボタン要素のための機能を継承する。そして、ABC に AssociatedComponentProcessor(ACP) クラスを Mix-in している。それにより、ACP クラスに実装されている関連づけのための機能が ABC クラスに追加される。

3.3 カスタム属性およびイベント・ハンドラ

表 1 に、Associated Components のためのカスタム属性を示す*9。data-receiverId 属性には、そのコンポーネントと動作を関連づける receiver のコンポーネント・パスを指定する。最も簡単なコンポーネント・パスの例は、HTML ビルトイン要素および Associated Components の id 属性の値である。data-groupId 属性は、そのコンポーネントが

*7 HTML 標準規格ではカスタム要素の説明において、JavaScript のクラス構文を用いている。本稿でもクラス構文を用いる。

*8 HTMLButtonElement(HBE) は、HTML 標準規格に含まれるボタン要素のための DOM interface である。JavaScript のクラス構文を用いることにより、HBE を継承した派生クラスを作成できる。

*9 カスタム属性名は data- で開始するように HTML 標準規格に規定されている。

表 2: Associated Components のカスタム・イベント・ハンドラ

Table 2 Custom event handlers of Associated Components.

event handlers (abbreviations)	description
appendReceiverChild (appendRChild)	Append child nodes to receivers.
removeReceiverChild (removeRChild)	Remove child nodes from receivers.
setReceiverInnerHTML (setRInnerHTML)	Set HTML codes to receivers.
setReceiverStyle (setRStyle)	Set CSS styles to receivers.

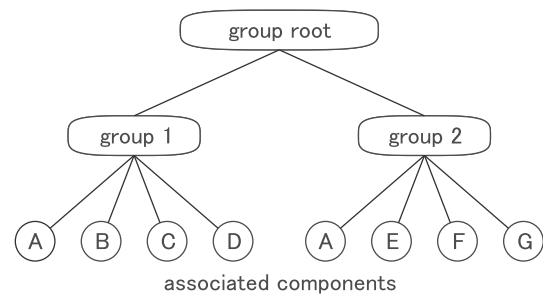


図 3: Associated Components のグループ木の例

Fig. 3 An example of an Associated Components group tree.

所属するグループのグループ・パスを指定するために用いられる。これらのパスは、スペースを区切り文字として複数指定できる。

Associated Components のためのカスタム・イベント・ハンドラを、表 2 に示す。イベント・ハンドラの名前は、その処理に関連する JavaScript の関数の名前を基に決定した。

3.4 Associated Components グループ

複数の Associated Components の動作を簡潔に記述するなどの目的のために、Associated Components グループを導入する。グループは階層構造をもち、グループ木により管理される。図 3 にグループ木の例を示す。

グループ木では、ルート (group root) の下に group または component ノードが存在する。group ノードには、対応するグループの group ID (Group 1 および Group 2) が格納され、component ノードには、対応する Associated Components の ID (A, B, ..., G) が格納される。グループの階層は複数も可能であり、同一の Associated Components が複数のグループに所属することも可能である。図 3 の例では、Associated Components A は、group 1 および 2 に所属している。

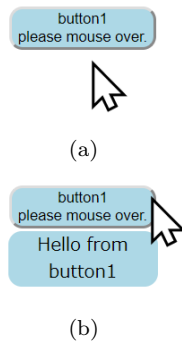


図 4: ボタン要素とテーブル・セル要素の関連づけ

Fig. 4 Associating a button element with a table cell element.

4. 実験と結果の検討

4.1 実装

3 節において提案した Associated Components の実装をおこない、動作の確認をおこなった。実装には、JavaScript を用いた。

グループ木 (3.4 節を参照) は、DOM ツリー (グループ DOM ツリー) として実装した。HTML 標準規格に定義される shadow DOM のための機能を用いることにより、Web ページを表示するために用いられる HTML document の DOM ツリーとは独立した DOM ツリーとしてグループ DOM ツリーを生成した。

4.2 ボタン要素とテーブル・セル要素の関連づけ

Associated Components を用いた簡単なプログラム例を、リスト 1 に示す。図 4 は、Web ブラウザにより動作を確認した結果である。ボタン要素をカーソルがホバーすると、下側にあるテーブル・セル要素に関連情報が表示されることが確認できる。ここで、ボタン要素は Associated Components であり、テーブル・セル要素は HTML ビルトイン要素である。

プログラムの head 部では、script 要素において AssociatedButton.js を指定している。これにより、Associated Components の定義などがおこなわれる。

body 部では、テーブル・セル要素にボタン要素を配置している。属性の指定では、is 属性に clm-associated-button を指定している。

```
is='clm-associated-button'
```

これにより、このボタン要素を Associated Components として使用することが可能になる。clm-associated-button は、ボタン要素としての Associated Components のタグ名であり、AssociatedButton.js において定義などがおこなわれている。また、data-rId 属性 (表 1 を参照) により、動作

リスト 1: ボタン要素とテーブル・セル要素の関連づけ

Listing 1: Associating a button element with a table cell element.

```
<!DOCTYPE html>
<html>
<head>
  <title>Associating a button with a table cell.</title>
  <script
    type="module"
    src="./AssociatedButton.js"
  ></script>
</head>
<body>
  <table align='center'>
    <tr>
      <td>
        <button
          id='button1'
          style='background:lightblue;
            border-radius:12px;'
          is='clm-associated-button'
          data-rId='table1Element'
          onmouseover="setRStyle('background:lightblue;\
            border-radius:12px;');"
          setInnerHTML('Hello from<br>button1')"
          onmouseout="setRStyle('background:white;');"
          setInnerHTML('')"
        >
          button1<br>please mouse over.
        </button>
      </td>
    </tr>
    <tr>
      <td
        style='background:white;'
        id='table1Element'
        align='center'
      >
    </td>
    </tr>
  </table>
</body>
</html>
```

を関連づける HTML 要素の ID を指定している。

```
data-rId='table1Element'
```

table1Element は、ボタン要素と動作を関連づけるテーブル・セル要素 (receiver) の ID である。

onmouseover および onmouseout 属性には、カーソルの

*10 Associated Components では、継承したビルトイン要素がもつ機能を使用可能である。

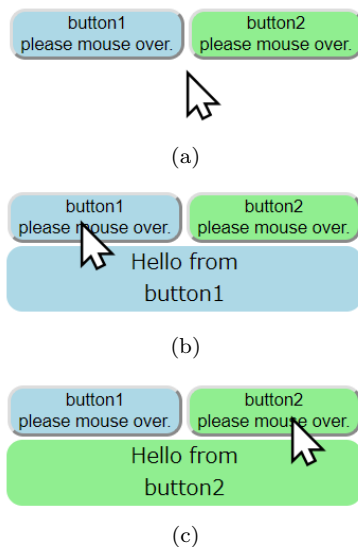


図 5: 2 つの要素に関連づけられた共通の要素

Fig. 5 Shared element associated with two elements.

動作によりボタン要素に該当のイベントが発生した場合に用いられるカスタム・イベント・ハンドラ (表 2 を参照) を指定している*¹⁰。該当のイベントが発生すると, `setRStyle` の引数で指定した CSS スタイルが receiver に設定される。そして, `setInnerHTML` の引数で指定した HTML プログラムが receiver に設定される。これらの引数には, HTML ビルトイン要素と同様の記述を用いることができる。

id および style 属性については, HTML ビルトイン要素で用いる属性と同じである*¹⁰。

なお, data-rId 属性にそのコンポーネント自身の id 属性の値を指定することにより, そのコンポーネントを receiver とすることも可能である。

4.3 2 つの要素に関連づけられた共通の要素

図 5 では, 2 つのボタン要素に共通のテーブル・セル要素を関連づけている。カーソルをホバーしたボタン要素から receiver であるテーブル・セル要素に, receiver に表示する文字列およびスタイルのデータが送信され, それらの値が receiver に設定される。リスト 2 は, その HTML プログラムの body 部である。head 部はリスト 1 と同様である。

ボタン 1 をホバーした場合には, ボタン 1 の `onmouseover` 属性に設定されている Associated Components のカスタム・イベント・ハンドラ `setRStyle` および `setInnerHTML` が実行される。そして, カーソルが Associated Components から離れると, `onmouseout` 属性に設定されている Associated Components のカスタム・イベント・ハンドラ `setRStyle` および `setInnerHTML` が実行される。動作の詳細については, 4.2 節と同様である。ボタン 2 の動作については, receiver に関連づけるデータが異なる点を除いてはボタン 1 と同様である。このように, カーソルをホバー

リスト 2: 2 つの要素に関連づけられた共通の要素

Listing 2: Shared element associated with two elements.

```
<body>
<table align='center'>
  <tr>
    <td>
      <button
        id='button1'
        style='background:lightblue;
          border-radius:12px;'
        is='clm-associated-button'
        data-rId='table1Element'
        onmouseover="setRStyle('background:lightblue;\
          border-radius:12px;');"
          setInnerHTML('Hello from<br>button1')"
        onmouseout="setRStyle('background:white;');"
          setInnerHTML('')"
      >
        button1<br>please mouse over.
      </button>
    </td>
    <td>
      <button
        id='button1'
        style='background:lightgreen;
          border-radius:12px;'
        is='clm-associated-button'
        data-rId='table1Element'
        onmouseover="setRStyle('background:lightgreen;\
          border-radius:12px;');"
          setInnerHTML('Hello from<br>button2')"
        onmouseout="setRStyle('background:white;');"
          setInnerHTML('')"
      >
        button2<br>please mouse over.
      </button>
    </td>
  </tr>
  <tr>
    <td
      style='background:white;'
      id='table1Element'
      align='center'
      colspan=2
    >
  </td>
</tr>
</teble>
</body>
```

したボタン要素から共通のテーブル・セル要素に関連情報を動的に設定することにより, ボタン要素とテーブル・セル要素の動作を関連づけ関連情報を提示している。

ここでは、2つのボタン要素は Associated Components であり、テーブル・セル要素は HTML ビルトイン要素である。

4.4 応用例 1: 段階的な行列の積の計算

Associated Components を用いた応用例として、行列の積を計算するプログラムについて説明をする Web ページを制作する場合を想定し、そこで用いるユーザ・インタフェースを作成した。行列の積の計算過程を、1) ベクトルの内積の計算、2) 行列とベクトルの積の計算、3) 行列の積の計算の3ステップに分けて考えた。そして、それぞれのステップについての説明を記載したページを用意し、各ページに移動するためのボタンを作成した。

説明を分かり易くするために、次の行列 A と B を用いて具体的に説明する。

$$A = \begin{pmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{pmatrix}, B = \begin{pmatrix} b_{11} & b_{12} \\ b_{21} & b_{22} \end{pmatrix}$$

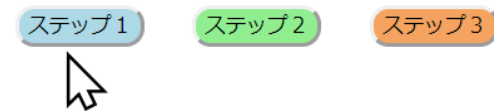
これらの積 $C = AB$:

$$C = \begin{pmatrix} c_{11} & c_{12} \\ c_{21} & c_{22} \end{pmatrix}$$

を計算するためのプログラムについて説明する場合に、次のような複数のステップのページに分けることを考える。ステップ1では、 c_{11} を求めることを考える。 c_{11} は、行列 A の第1行の各要素を成分とする行ベクトルと行列 B の第1列の各要素を成分とする列ベクトルの内積の計算により得られる。そこで、1重の for 文を用いた繰り返し処理によりこれらの内積を計算するプログラムについて説明する。ステップ2では、 c_{11} および c_{21} を求めることを考える。これらは、行列 A と行列 B の第1列の各要素を成分とする列ベクトルの積により得られる。そこで、ステップ1で説明した for 文の入れ子を1つ増やした2重の for 文を用いて、これらの積を計算するプログラムについて説明する。ステップ3では同様に、ステップ2で説明したプログラムの for 文の入れ子を1つ増やして、3重の for 文を用いた繰り返し処理により行列 C の各要素を計算するプログラムについて説明する。これらの説明の他に、計算値を格納するための変数についての説明が必要であるが、基本的な考え方についてはこのような説明をする各ページを用意するという想定である。

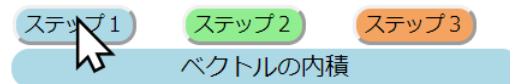
図6に、説明の掲載されたページへの移動ボタンをカーソルがホバーした場合の動作を示す。ボタン要素の下側にあるテーブル・セル要素に、移動したページで説明されている内容のポイントが表示されている。図6のプログラムは、リスト2と同様であるので、ステップ1のページに移動するためのボタン要素のための Associated Components の例をリスト3に示す。ここでは、Associated Components の

行列の積を、3ステップで考えてみましょう。



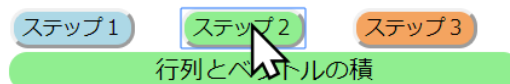
(a)

行列の積を、3ステップで考えてみましょう。



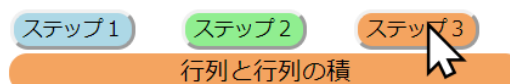
(b)

行列の積を、3ステップで考えてみましょう。



(c)

行列の積を、3ステップで考えてみましょう。



(d)

図6: 段階的な行列の積の計算

Fig. 6 Matrix multiplication in a step-by-step manner.

リスト3: 段階的な行列の積の計算

Listing 3: Matrix multiplication in a step-by-step manner.

```
<td>
  <button
    id='Step1'
    style='background:lightblue;
      border-radius:12px;'
    is='clm-associated-button'
    data-rId='table1Element'
    onmouseover="setRStyle('background:lightblue;\
      border-radius:12px;');"
    setInnerHTML('ベクトルの内積')"
    onmouseout="setRStyle('background:white;');"
    setInnerHTML('')"
  >
    ステップ1
  </button>
</td>
```

ために必要な属性を中心に示している。ボタンを押下した場合に該当のページに移動するためには、ページを移動するためのコードが必要になる。それらについては、ボタン要素のための Associated Components についても、HTML 標準規格に定義される機能をそのまま使用可能である。例えば、HTML ビルトイン要素のボタン要素に指定可能なり

リンクのための属性などを、HTML 標準規格に定義されているとおりにそのまま使用可能である。また、Associated Components を用いたクライアントからボタン要素が押下されたことをサーバにポストすることにより、PHP 言語などを用いたサーバサイド・プログラムと連携をする場合についても、Associated Components を用いることによる変更は生じない。

4.5 応用例 2: テクスチャ画像の対話的な比較

図 7 は、イメージ要素のための Associated Components の例であり、イベント・ハンドラ `appendReceiverChild` の使用例にもなっている。

表示されているテクスチャ画像 (a) の中から、比較する画像をクリックにより選択すると選択した画像がその下側に並んで表示される (b,c)。選択した画像をカーソルによりホバーすることにより、どの画像を選択したのかを確認できる (d)。Reset ボタンにより選択した画像をクリアしてから、画像を再選択することもできる (e)。

選択された画像が下側に表示される動作は次の方法により実現した。

Step 1 選択された画像が表示される領域を、HTML コンテナとして用意しておく。

Step 2 画像が選択されると、その画像を格納しているイメージ要素のための Associated Components は、同じ画像を格納したイメージ要素のための Associated Components を HTML コンテナに追加する。

Step 2 では、イベント・ハンドラ `appendReceiverChild` を用いた。イベント・ハンドラでの指定の概要を、次に示す。なお、引用符のレベルを考慮してエスケープなどをおこなう必要がある。

```
onclick="appendRChild( \
    '<img src=\'./path/to/image\' \
        is=\'clm-associated-img\' \
        data-rId=\'texture01TD\' \
        onmouseover= ... \
        onmouseout= ... > \
    </img>' \
    )"
```

選択されて下側に並べられた画像をカーソルによりホバーした場合に、上側に並んでいる基の画像をオレンジ色の枠でハイライトする機能は、イメージ要素のための Associated Components どうしを関連づけることにより実現した。

4.6 検討

リスト 1, 2 および 3 からは HTML 標準規格に定義される HTML ビルトイン要素と同様の記述により、HTML プ

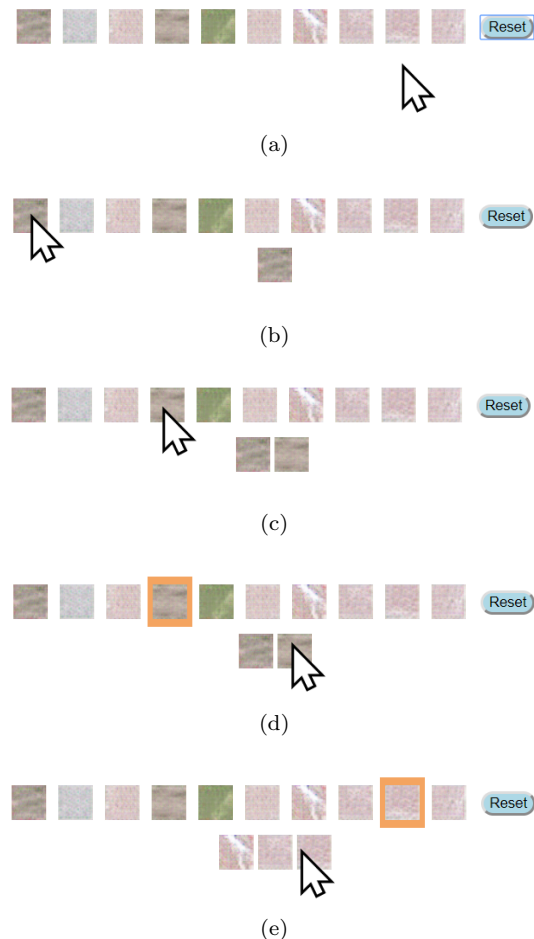


図 7: テクスチャ画像の対話的な比較

Fig. 7 Interactive comparison of texture images.

ログラムにおいて Associated Components を使用可能であることが分かる。Associated Components と HTML ビルトイン要素をシームレスに記述することができ、それらの動作の関連づけが可能であることが分かる。従来手法により同様の動作をおこなうプログラムを作成するためには HTML および JavaScript を用いる必要があるが、Associated Components を用いると HTML と同様の文法のみを用いることにより、複数の GUI コンポーネントを連携させる動作を記述できることが分かる。

4.2, 4.3, 4.4 および 4.5 節に示す Associated Components の動作からは、複数の GUI コンポーネントを用いて、関連づけられた情報を提示することが可能であることが分かる。

4.4 節の応用例 1 からは、行列の積を求めるプログラムを作成する場合の考え方のポイントとなる情報を、プログラムを構成するステップに沿って提示できることが分かる。動作を伴うユーザ・インタフェースを用いて関連する情報を提示することは、ユーザの興味を引くという利点のみではなく、情報を整理して順番に提示することが可能であるという利点がある。ユーザの操作に対応して関連する情報を提示するというユーザ・インタフェースにより、ユーザ

に一呼吸して考える間を与えるということは、教育のためのソフトウェアには重要であるといえる。

4.5 節の応用例 2 からは、対話的に選択したテキストを並べて比較を繰り返すためのユーザ・インタフェースを構築可能であることが分かる。Associated Components を用いた HTML のみによる記述のプログラムにより、ある程度複雑なユーザ・インタフェースを構築可能であることが分かる。

また、HTML のみを用いて動作を含むプログラムを記述できることは、プログラミング言語の設計コンセプトの点で重要であるといえる。HTML, CSS および JavaScript の設計コンセプトは、それぞれ、HTML により文書構造を定義、CSS により外観を定義、および JavaScript により動作を定義である。JavaScript によりプログラミングした動作により文書構造が変化するような場合には、HTML プログラム以外のプログラムにより文書構造が変化してしまうことになり、Web ページの文書構造の解析が困難になる可能性がある。一方、Associated Components を用いることにより、動作に関連して動的に変化する文書構造を HTML の文法によるプログラムにより表現することが可能となることから、上記の困難さを緩和するための手法として期待できる。

HTML 標準規格に定義されるカスタム要素は、比較的新しい機能であるが、Chrome および Edge などの主要なモダンブラウザでは標準機能として実装されているので、追加のソフトウェアなどをインストールすることなく使用可能である。モダンブラウザ以外の IE11 などでも、ポリフィルを用いることによりカスタム要素の機能を使用可能である。そのように環境が整ったことから、HTML 標準規格に定義されるカスタム要素のための機能を用いることにより、ユーザ・インタフェースのための新しいメカニズムについて研究することは非常に興味深いことである。

5. おわりに

本稿では、関連づけられた情報を Web ページにおいて提示することを目的としてカスタマイズした GUI コンポーネント (Associated Components) を提案した。特徴は、HTML プログラムにおいて、HTML 標準規格に定義されるビルトイン要素と同様の記述を用いて、複数の GUI コンポーネントを連携させることにより関連情報を提示可能な点である。また、コンポーネント間で受け渡される情報の直接的な記述により、簡潔なプログラミングが可能である。

Associated Components の実装をおこない動作を確認した。提案のコンポーネントは、一般的な利用を目的として設計したものであるが、Web ページ制作の入門コースなどにおいても利用が容易である。複数のコンポーネントの効果的な連携について検討することは、情報を提示する場合

の適切な順序について考える機会としても重要である。

今後の課題には、多様な関連づけのパターンに対応するための改良、コンポーネント間の依存関係の解析ツールに関する研究などがある。

参考文献

- [1] Cardone, R., Brown, A., McDirmid, S. and Lin, C.: Using Mixins to Build Flexible Widgets, *Proceedings of the 1st International Conference on Aspect-oriented Software Development*, AOSD '02, New York, NY, USA, ACM, pp. 76–85 (online), DOI: 10.1145/508386.508395 (2002).
- [2] Farkas, D. K.: The Role of Balloon Help, *SIGDOC Asterisk J. Comput. Doc.*, Vol. 17, No. 2, pp. 3–19 (online), DOI: 10.1145/154425.154426 (1993).
- [3] Google: Polymer Project, <https://www.polymer-project.org/> (Retrieved: 8 October 2019).
- [4] Hopkins, J.: Component Primer, *Commun. ACM*, Vol. 43, No. 10, pp. 27–30 (online), DOI: 10.1145/352183.352198 (2000).
- [5] Jazayeri, M.: Component Programming - a Fresh Look at Software Components, *Proceedings of the 5th European Software Engineering Conference*, London, UK, UK, Springer-Verlag, pp. 457–478 (online), available from (<http://dl.acm.org/citation.cfm?id=645385.651512>) (1995).
- [6] Molina, P. J.: Quid: Prototyping Web Components on the Web, *Proceedings of the ACM SIGCHI Symposium on Engineering Interactive Computing Systems*, EICS '19, New York, NY, USA, ACM, pp. 3:1–3:5 (online), DOI: 10.1145/3319499.3330294 (2019).
- [7] Mozilla Developer Network: Web Components, https://developer.mozilla.org/en-US/docs/Web/Web_Components (Retrieved: 4 April 2019) (2017).
- [8] Price, J.: Comments on Balloon Help, *SIGDOC Asterisk J. Comput. Doc.*, Vol. 17, No. 2, pp. 21–22 (online), DOI: 10.1145/154425.154428 (1993).
- [9] Saffer, D.: *Microinteractions: Full Color Edition Designing with Details*, O'Reilly Media, Inc., 1st edition (2013).
- [10] WHATWG: HTML Living Standard — Last Updated 18 July 2019, <https://html.spec.whatwg.org/> (Retrieved: 25 July 2019).