Vol.2020-HCI-187 No.33
2020/3/17

関連づけられた情報の提示のためにカスタマイズしたGUIコンポーネントの設計

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設計されたカスタマイズした GUI コンポーネントである Associated Components について述べる．そのコンポーネントを用いると，関連づけられた情報を局所的に定義でき，そして，簡潔に表現することが可能である．Associated Components 間のみではなく，HTML 標準規格に定義されるビルトイン要素と連携させることによっても関連づけられた情報の提示が可能である．HTML 標準規格を用いた Web コンポーネントとしてコンポーネントを設計することにより，ビルトイン要素と同様の記述を用いて使用することを可能にした．そして，コンポーネントのグループを導入することにより，コンポーネントの部分集合の関連づけを簡潔に記述することを可能にした．これらにより，コンポーネントの再利用が容易となっている．

Designing GUI Components Customized for Providing Associated Information

Makoto Satoh[1]

Abstract: This paper describes Associated Components, GUI components customized for providing associated information on Web pages. Associated information can be defined locally and can be represented concisely using the components. Not only with associating the components, but also with associating built-in HTML elements with the components, associated information can be provided. By designing the components as Web Components using the features defined in the HTML standard, the components can be used with the almost same notation as used by built-in HTML components. Introducing the group of the components enables concise representation of the association of component subsets. From the above, the components can be reused easily.

1. はじめに

本稿では，Associated Components [*1] の設計について述べる．Associated Components 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された GUI コンポーネントである．特徴は，次のとおりである．

- 複数の GUI コンポーネントを連携させて動作させることにより，関連づけられた情報を提示する．
- HTML 標準規格 [18] において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，コンポーネント間で受け渡される情報を簡潔に記述することが可能である．

Associated Components を用いると，文書構造を動的に変化させながら関連づけられた情報の提示をおこなう Web ページを，HTML のみにより簡潔に記述することが可能である．そのため，関連づけられた情報の提示方法を試行錯誤により検討するようなプロトタイプの制作に適しているといえる．また，HTML のみにより動的な文書構造の変化

[1] 岩手大学
Iwate University, Ueda, Iwate 020–8551, Japan

[*1] Associated Components の機能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

を記述できることから，解析が容易な動的 Web ページを作成することが可能である．

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 節において説明する．4 節では 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] [3] [15] 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) [16] 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．ユーザ・インタフェースによる計算機のユーザ体験は，マイクロインタラクションにより大きく印象づけられるといえる．

Web ページにおいて情報を分かりやすく提示するためには，ページのデザイン・スタイルにあわせた関連情報の提示方法の検討が必要である．デザイン・スタイルの選択では，従来のスキューモーフィック・デザインにかわりフラット・デザインが選択される機会が増加している[*3] [2], [14], [17] ．フラット・デザインを用いると簡潔なデザインが得られるが，提示される情報が不足する場合もありえるので，補足情報を提示するための手法が必要である．

Associated Components では，複数の HTML 要素の動作を連携させることにより，補足などのための関連づけられた情報の提示が可能である．

2.2 再利用可能な GUI コンポーネント

プログラムの設計においては，ソフトウエア・コンポーネントの再利用がよくおこなわれる [6] [7] [10] ．

Web ページの制作では，小規模なソフトウエア・コンポーネントである HTML 要素などが用いられる．Web ページの設計過程ではデザイナとプログラマによる共同作業がおこなわれる場合が多いが，デザイナからプログラマへのデザインの意図の伝達では，外観 (appearance) と比較して動作 (behavior) については伝達が難しいといわれている [13]．そのため，[1] などのプロトタイプ制作用ソフトウエアを用いることによりデザインを具体的に示すことがよくおこなわれる．動作を含むプロトタイプを制作する場合には，HTML 要素が単体でおこなう動作については，プロトタイプ制作用ソフトウエアを用いて GUI 要素をカスタマイズして再利用することが可能であるが，複数の HTML 要素の動作を連携させることにより関連づけられた情報を提示するような場合には，イベント処理を含むプログラムを JavaScript などにより作成する必要がある．

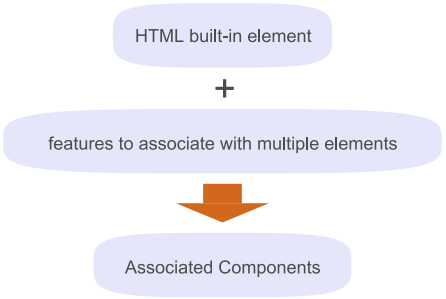


図 1: Associated Components の基本的なコンセプト
Fig. 1 Basic concept of Associated Components.

Associated Components により，複数の HTML 要素が連携して動作する GUI コンポーネントを HTML のみを用いて記述することが可能である．そのため，プロトタイプの作成にも適しているといえる．

3. Associated Components の設計

3.1 設計のコンセプト

Associated Components の設計の基本的なコンセプトを，図 1 に示す．複数の HTML 要素の動作を関連づけるための機能を HTML 標準規格に含まれる HTML ビルトイン要素に追加することにより，Associated Components を構築する．HTML 標準規格に含まれるカスタマイズしたビルトイン要素のための機能[*4] [*5] [12], [18] を用いて機能の追加をおこなうことにより，Associated Components では HTML 標準規格の HTML ビルトイン要素のための文法の全てを変更せずに使用可能である．

HTML 標準規格に含まれるカスタム要素のための機能は比較的新しい機能であるが，Chrome, Firefox および Edge

[*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．

[*3] マテリアル・デザイン [4] が選択される機会も増加している．フラット・デザインと同様に簡潔な表現のデザイン・スタイルであるが，マテリアル・デザインは，紙に印刷した文書と同様の印象を得られるようにデザインを工夫することにより，印刷物のもつ長所をデジタル・メディアに生かそうとしている．

[*4] 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 [12] ．

[*5] 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．本稿では，後者を用いる．

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．モダンブラウザ以外の IE11 などにおいても，HTML プログラムのヘッダ部などでポリフィル (Polyfill) を指定することによりブラウザ側での変更は必要とせずにカスタム要素のための機能を使用可能である．

3.2 クラス構造[*6]

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[*7] から派生させたクラスである．HBE クラスの派生クラスとすることにより，ABC クラスは HBE クラスのもつ HTML 標準規格に含まれる HTML ビルトイン要素のボタン要素のための機能を継承することができる．そして，ABC クラスに AssociatedComponentProcessor(ACP) クラスを Mix-in することにより，ACP クラスに定義された機能が ABC クラスに追加される．ACP クラスには，複数の HTML 要素を関連づけて動作させるための機能が含まれている．

このようなクラス構造により HTML ビルトイン要素と Associated Components の機能をあわせもつ HTML 要素を構成することができる．そして，ABC クラスをさらに継承して派生させた新しいクラスの作成することも容易である．

図 2 のクラス定義は，Associated Components として使用する HTML 要素の種類ごとに必要である．必要となる変更は基底となるクラス名などの名前の変更のみであることから，4.1 節で述べるように，配列に格納した定義データを用いて繰り返し処理をおこなうことにより簡潔な実装が可能である．複数の HTML 要素の動作を連携させるための機能を含む ACP クラスについては，変更することなく全ての HTML 要素について共通のものを使用可能である．なお，同じ種類の HTML のための Associated Components を複数個使用する場合には，その種類の HTML 要素のための Associated Components のクラス定義は 1 回おこなうのみでよい．

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[*8] ．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

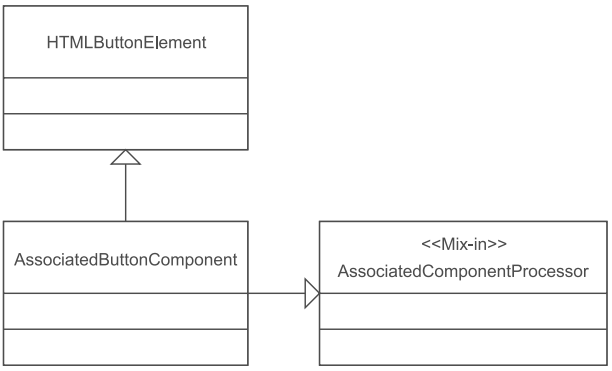


図 2: Associated Components のクラス構造の例
Fig. 2 An example of Associated Components class structure.

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.

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.

ビルトイン要素の id 属性の値である．data-groupId 属性は，そのコンポーネントが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を指定できる．パスについては，3.4 節において詳しく説明する．

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，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．

[*6] 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．
[*7] 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．
[*8] 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．

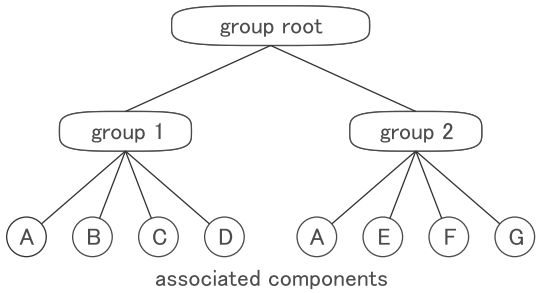


図 3: Associated Components のグループ木の例
Fig. 3 An example of an Associated Components group tree.

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．

Associated Components のカスタム属性でのグループの指定 (3.3 節を参照) では，グループ・パス (グループ木でのパス) を用いる．記法は，次のとおりである．

```
(/)[group ID/]
```

`/` はセパレータである[*9] ．`(*)` は，`()` で囲まれた内容 `*` を省略可能であることを示す．`[*]` は，`[]` で囲まれた内容 `*` の 0 回以上の繰り返しである．OS のインタフェースとして用いられるシェルのパスとは異なり，グループ木には現在の位置に相当する概念が存在しないので，グループ・パスの先頭が `/` であるかどうかにかかわらず，グループ・パスはグループ木のルートからの絶対パスとして取り扱われる．Associated Components の所属グループは data-groupId 属性により指定される．指定がない場合には所属グループを `AssociatedComponents` とする．

Associated Components と関連づけるコンポーネントは，コンポーネント・パスを用いて指定する．コンポーネント・パスの記法は，次のとおりである．

```
(/)[group ID/](component ID)
```

コンポーネント・パスの最後が `/` の場合には，グループ木でのそのパス以下の部分に所属する全てのコンポーネントの指定になる．component ID のみを指定した場合には，HTML document の DOM ツリーに含まれる HTML 要素が対象となる[*10] ．

4. 実験と結果の検討

4.1 実装

Associated Components を，JavaScript により実装した．実装した Associated Components の動作確認には，Chrome および Edge を用いた．

Associated Components のクラス定義は，Associated Components として用いる HTML ビルトイン要素の種類ごとに必要であるが，定義に必要なデータを対応する HTML ビルトイン要素の種類ごとに配列に格納し，繰り返し処理を用いて定義をおこなうことにより簡潔な実装をおこなった．ボタン要素のための配列データを，リスト 1 に示す．

グループ木 (3.4 節を参照) は，HTML document の DOM ツリーとは独立した DOM ツリーとして，HTML 標準規格に含まれる shadow DOM のための機能を用いて生成した．図 4 に，グループ DOM ツリーの例を示す[*11] ．

リスト 1: ボタン要素のための定義
Listing 1: Definition for button elements.

```
[
  {
    // AssociatedButton
    superClass : HTMLButtonElement,
    supterTagName : 'button',
    derivedTagName : 'clm-associated-button'
  },
  ...
]
```

```
<div id='group root'>
  <div name='group1' attribute='group'>
    <div name='A' attribute='component'>
    <div name='B' attribute='component'>
  <\div>
 <div name='group2' attribute='group'>
    <div name='A' attribute='component'>
    <div name='C' attribute='component'>
  <\div>
<\div>
```

図 4: グループ DOM ツリーの例
Fig. 4 An example of a group DOM tree.

[*9] id および name 属性の値として使用可能な文字の種類は，HTML 標準規格のバージョンにより相違がある．Associated Components では，`/` をグループ・パスおよびコンポーネント・パスでのノードのセパレータとして用いる．

[*10] Associated Components も，HTML document の DOM ツリーに含まれる HTML 要素であるので，HTML 標準規格で定義されるビルトイン要素，および Associated Components が対象となる．

[*11] グループ・ノードおよびコンポーネント・ノードには，同じ ID が格納される場合があるので，name 属性を用いて ID を格納した．

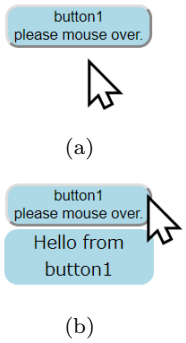


図 5: ボタン要素とテーブル・セル要素の関連づけ
Fig. 5 Associating a button element with a table cell element.

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 HTML プログラムの動作を，図 5 に示す．リスト 2 は，そのプログラムである．ボタン要素をカーソルがホバーすると，下側に配置されたテーブル・セル要素に関連情報を表示するプログラムである．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．

head 部では，script タグにおいて AssociatedButton.js を指定している．この指定により，Associated Components のクラス定義などがおこなわれる．AssociatedButton.js はボタン要素のための Associated Components を実装したプログラムである．実際には，ボタン要素以外のいくつかの HTML 要素のための Associated Components も実装されている．

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．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，動作を関連づける HTML 要素の ID を指定している．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[*12]．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，`setRInnerHTML` 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様の指定が可能である[*12]．

なお，data-rId 属性にそのコンポーネントの id 属性の値を指定することにより，そのコンポーネント自身を receiver とすることも可能である．その場合にも，カスタム・イベント・ハンドラを使用可能であるが，標準の HTML ビルトイン要素と同様の動作をおこなうことになる．

[*12] 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．

リスト 2: ボタン要素とテーブル・セル要素の関連づけ
Listing 2: Associating a button element with a table cell element.

```
<!DOCTYPE html>
<html>
<head>
  <title>Associating Button with Table.</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;');
            setRInnerHTML('Hello from<br>button1')"
          onmouseout="setRStyle('background:white;');
            setRInnerHTML('')"
        >
          button1<br>please mouse over.
        </buttton>
      </td>
    </tr>
    <tr>
      <td
        style='background:white;'
        id='table1Element'
        align='center'
      >
      </td>
    </tr>
  </teble>
</body>
</html>
```

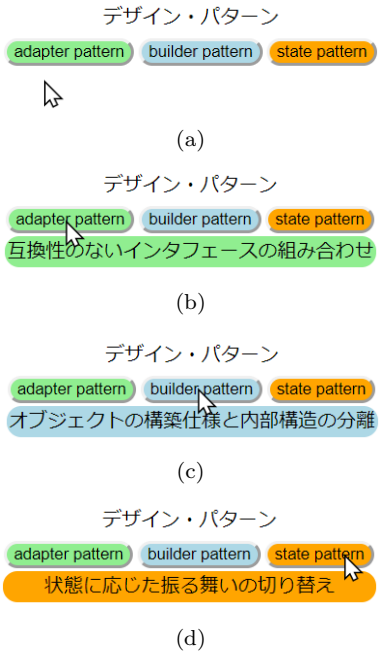


図 6: Web ページの内容の簡単な説明
Fig. 6 Brief description of Web page contents.

4.3 応用例 1: Web ページの内容の簡単な説明

図 6 は，複数の Associated Components の動作を共通の HTML 要素に関連づける例である．Web ページへの移動ボタンをカーソルがホバーすると，そのボタンを選択した場合に移動する Web ページに掲載されている内容のポイントが，下側に配置されたテーブル・セル要素に表示されるようになっている．それぞれのボタンに対応する説明は，共通のテーブル・セル要素に表示される．ここで，各ボタン要素は 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．そして，それらを格納しているテーブル要素も HTML ビルトイン要素である．

プログラムの中の 2 つのボタン要素に関連する部分を，リスト 3 に示す．head 部およびテーブル・セル要素については，リスト 2 と同様である．

ボタン要素で用いられている属性の仕様上の機能については，リスト 2 と同様である．ここでは，どちらのボタン要素においても，

```
data-rId='Description'
```

と指定している．これにより，それらのボタン要素をカーソルがホバーした場合には，ID が'Description' である共通の HTML 要素に動作が関連づけられるようになる．次のように ID を'Description' と設定したテーブル・セル要素が定義されているので，複数のボタン要素とテーブル・セル要素の動作が関連づけられることになる．

リスト 3: Web ページの内容の簡単な説明
Listing 3: Brief description of Web page contents.

```
<td>
  <button
    id='adapterPattern'
    style='background:lightgreen;
          border-radius:12px;'
    is='clm-associated-button'
    data-rId='Description'
    onmouseover="setRStyle('background:lightgreen;\
                          border-radius:12px;');
                setRInnerHTML(' 互換性の...')"
    onmouseout="setRStyle('background:white;');
      setRInnerHTML('')"
  >
    adapter pattern
  </buttton>
</td>
<td>
  <button
    id='builderPattern'
    style='background:lightblue;
          border-radius:12px;'
    is='clm-associated-button'
    data-rId='Description'
    onmouseover="setRStyle('background:lightblue;\
                          border-radius:12px;');
                setRInnerHTML(' オブジェクトの...')"
    onmouseout="setRStyle('background:white;');
      setRInnerHTML('')"
  >
    builder pattern
  </buttton>
</td>
```

```
  <tr>
    <td
      style='background:white;'
      id='Description'
      align='center'
      colspan=3
    >
    </td>
  </tr>
```

4.4 応用例 2: 深層学習モデルの対話的な確認

図 7 は，Associated Components グループの例である．画像要素を配置するためのテーブル・セル要素およびボタン要素は Associated Components であり，画像要素などのそれ以外の HTML 要素は HTML ビルトイン要素である．画像要素を配置したテーブル・セル要素では，それぞれが

複数の Associated Components グループに所属するように data-gId 属性を指定した．

図 7 では，深層学習により手書き数字画像 [9] を学習したモデルについて対話的な確認をおこなっている．学習では，LeNet5 [8] に変更をおこなったネットワーク・モデル[*13] を用いた．学習の各エポックでの学習パラメータを保存し，それを用いてクラス分類のテストをおこなった．

図では，テストに用いた画像が上側に並んでいる．各エポックに対応するボタンをカーソルでホバーすると，そのエポックの学習パラメータを用いて正しく分類できた画像をオレンジ色の枠でハイライトしている．数字画像が配置されたテーブル・セルの背景色を変更することによりハイライトをおこなっている．図 7(a) から，Epoch 1 では 7 および 2 を正しく分類できたことが分かる．図 7 (b, . . . , e) からは，Epoch 2 から 5 へと学習が進むにつれて正しく分類できるようになったことが分かる．

Epoch 2 ボタンをホバーした場合について，左から 3 番目の画像がハイライトされる動作を説明する．

Step 1 左から 3 番目の画像の配置されたテーブル・セル要素の所属グループを，次のように指定する．

```
data-gId= '/Data/Epoch-1-False/ \
          /Data/Epoch-2-True/ ... '
```

これにより，グループ/Data/Epoch-1-False/および/Data/Epoch-2-True/など (. . .) に所属する．

Step 2 Epoch 2 ボタンと動作を関連づける HTML 要素を次のように指定する．

```
data-rId='/Data/Epoch-2-True/'
```

そして，onmouseover 属性に，カスタム・イベント・ハンドラ setRStyle を背景色をオレンジ色に設定するように指定する．

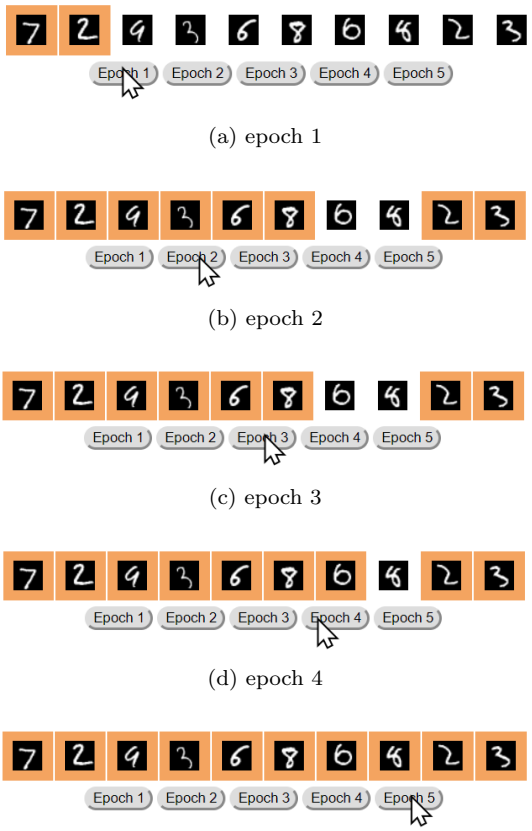
Step 3 **Step 2** により，Epoch 2 ボタンをカーソルでホバーすると，Epoch 2 ボタンはグループ木の/Data/Epoch-2-True/に所属する Associated Components に対してカスタム・イベント・ハンドラ setRStyle を実行する．

Step 4 左から 3 番目の画像の配置されたテーブル・セル要素の所属グループには/Data/Epoch-2-True/が含まれる (**Step 1**) ので，枠がオレンジ色に変化する．

4.5 検討

リスト 2, 3 からは，Associated Components を用いると，HTML プログラムにおいて HTML ビルトイン要素と同様の記述をおこなうことにより，GUI コンポーネントの動作の関連づけを記述できることが分かる．

[*13] ReLU などを用いるように変更した．



(a) epoch 1

(b) epoch 2

(c) epoch 3

(d) epoch 4

(e) epoch 5

図 7: Associated Components グループを用いた深層学習モデルの対話的な確認

Fig. 7 Interactive verification of a deep learning model with Associated Components group.

図 5, 6, 7 からは，Associated Components と HTML ビルトイン要素，および，Associated Components どうしの動作を連携させることにより，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．

図 6 でのボタン要素が選択された場合に移動する Web ページについての簡潔な情報を提示する動作は，ユーザ・インタフェースのためのメカニズムを用いることにより，インタラクションにおいて一呼吸おいて確認などをするための間を提供しているともいえる．

図 7 からは，Associated Components グループを用いることにより，同一の Associated Components が Epoch ごとにグループが変更される様子を表現可能であることが分かる．

3.1 節において述べたように HTML 標準規格に含まれるカスタム要素のための機能は比較的新しい機能であり，その研究・開発は始まったばかりである．それを用いて，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．例えば，

[*14] Polymer[5] 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．

JavaScript プログラムを簡略化するための手法の開発[*14]，および，独自の DSL(Domain Specific Languages) を用いた研究[*15] などがある．

Associated Components の特徴は，複数の GUI コンポーネントの動作を連携させることにより関連づけられた情報を提示する点である．HTML ビルトイン要素と同様の記述が可能であること，および，Associated Components グループの導入などにより，コンポーネントの依存関係が小さく可読性および保守性の高いプログラムを作成可能である．それにより，Associated Components は再利用の容易な GUI コンポーネントであるといえる．

そして，プログラミング言語の設計の点では，Associated Components は動作に伴う HTML の文書構造の変化を HTML プログラム内に記述可能であるということが重要である．JavaScript などにより HTML の文書構造を変化させる場合と比較するとプログラムの解析が容易であるといえる．

5. おわりに

関連づけられた情報を提示するためにカスタマイズした GUI コンポーネントである Associated Components について述べ，実装により動作を確認した．特徴は，HTML ビルトイン要素と同様の記述を用いて，複数の GUI コンポーネントの動作の連携により関連づけられた情報を提示する点である．

関連づけのためのデータの局所的な定義，および，コンポーネントのグループの導入により，情報の関連およびコンポーネントの依存関係を簡潔に記述可能とした．それにより，コンポーネントの再利用が容易となっている．提案のコンポーネントを用いると，関連づけられた情報の提示をおこなう Web ページを容易に制作可能である．そして，プロトタイプの制作などにも適しているといえる．

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機能の改良，コンポーネントのグループの拡張などがある．

参考文献

[1] Adobe: Adobe XD, https://www.adobe.com/jp/products/xd.html(Retrieved: 8 Octover 2019).
[2] Burmistrov, I., Zlokazova, T., Izmalkova, A. and Leonova, A.: Flat Design vs Traditional Design: Comparative Experimental Study, *Human-Computer Interaction – INTERACT 2015* (Abascal, J., Barbosa, S., Fetter, M., Gross, T., Palanque, P. and Winckler, M., eds.), Cham, Springer International Publishing, pp. 106–114 (2015).
[3] 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).
[4] Google: Material Design, https://material.io/(Retrieved: 8 Octover 2019).
[5] Google: Polymer Project, https://www.polymer-project.org/(Retrieved: 8 Octover 2019).
[6] Hopkins, J.: Component Primer, *Commun. ACM*, Vol. 43, No. 10, pp. 27–30 (online), DOI: 10.1145/352183.352198 (2000).
[7] 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).
[8] Lecun, Y., Bottou, L., Bengio, Y. and Haffner, P.: Gradient-based learning applied to document recognition, *Proceedings of the IEEE*, Vol. 86, No. 11, pp. 2278–2324 (online), DOI: 10.1109/5.726791 (1998).
[9] LeCun, Y. and Cortes, C.: MNIST handwritten digit database, (online), available from ⟨http://yann.lecun.com/exdb/mnist/⟩ (2010).
[10] Mahmood, S., Lai, R. and Kim, Y. S.: Survey of component-based software development, *IET Software*, Vol. 1, No. 2, pp. 57–66 (2007).
[11] 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).
[12] Mozilla Developer Network: Web Components, https://developer.mozilla.org/en-US/docs/Web/Web_Components(Retrieved: 4 April 2019) (2017).
[13] Myers, B., Park, S. Y., Nakano, Y., Mueller, G. and Ko, A.: How Designers Design and Program Interactive Behaviors, *Proceedings of the 2008 IEEE Symposium on Visual Languages and Human-Centric Computing*, VLHCC '08, Washington, DC, USA, IEEE Computer Society, pp. 177–184 (online), DOI: 10.1109/VLHCC.2008.4639081 (2008).
[14] Pan, Y. and Stolterman, E.: What if HCI Becomes a Fashion Driven Discipline?, *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems*, CHI '15, New York, NY, USA, ACM, pp. 2565–2568 (online), DOI: 10.1145/2702123.2702544 (2015).
[15] 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).
[16] Saffer, D.: *Microinteractions: Full Color Edition Designing with Details*, O'Reilly Media, Inc., 1st edition (2013).
[17] Schneidermeier, T., Hertlein, F. and Wolff, C.: Changing Paradigm – Changing Experience?, *Design, User Experience, and Usability. Theories, Methods, and Tools for Designing the User Experience* (Marcus, A., ed.), Cham, Springer International Publishing, pp. 371–382 (2014).
[18] WHATWG: HTML Living Standard — Last Updated 18 July 2019, https://html.spec.whatwg.org/(Retrieved: 25 July 2019).

[*15] カスタム要素に関連する研究である [11] では，GUI コンポーネントを定義するための DSL を用いて