Vol.2020-CG-180 No.8
Vol.2020-DCC-26 No.8
Vol.2020-CVIM-223 No.8
2020/11/5

データ駆動型 Associated Components を用いた Ambient Web Design

佐藤 信[1]

概要：本稿では，データ駆動型 Associated Components を提案し，そのコンポーネントを用いた Ambient Web Design について述べる．既提案の Associated Components は，Web ページにおいて関連づけられた情報を提示するためのコンポーネントである．HTML 標準規格に定義される文法のみを用いて，Associated Components と標準の HTML 要素が混在した Web ページを制作可能である．データ駆動型コンポーネントが用いるデータの定義には，CSS のサブセットの構文を採用した．これにより，要素属性を変更した複数のコンポーネントの生成が容易になり，コンポーネントの動作および外観の洗練化が容易になる．Ambient Web Design は，落ち着いた雰囲気をもつ Web ページを制作するための手法である．そのような繊細なデザインでは，データ駆動型 Associated Components を用いた洗練化が有効である．

Ambient Web Design with Data-driven Associated Components

Makoto Satoh[1]

Abstract: This paper proposes data-driven Associated Components and describes Ambient Web Design using the components. The components are the GUI components for providing associated information on Web pages. The Web pages contain simultaneously the components and built-in HTML elements can be created using only the syntax that is defined in the HTML standards. The subset of the CSS syntax is utilized to define the data used by the components. It facilitates the creation of the multiple components of which some attribute values are different, and the easy refinement of the actions and appearances of the components. Ambient Web Design is a method for creating relaxing ambient Web pages. In such a nuanced design, the components are useful for refinements.

1. はじめに

本稿では，データ駆動型 Associated Components を提案する．Associated Components [*1] [7], [8] は，複数の HTML 要素の連携動作により，情報の関連を表現するためのグラフィカル・コンポーネントである．データ駆動型コンポーネントとすることにより，次の特徴がえられる．

- 複数の HTML 要素の関連を定義するための要素属性を，head 部にデータとして記述可能である．局所的な定義により，要素の関連の把握が容易である．
- ある要素属性の値のみが異なる複数の HTML 要素を簡潔に記述可能である．例えば，カーソルに対応する動作のみが異なりデザインは共通である多数のボタン要素をレイアウトするような場合に有効である．

データ駆動型 Associated Components を用いると，外観または動作を少しずつ変更した HTML 要素を多数生成することが容易である．ここでは，落ち着いた雰囲気をもつ Web ページを制作するための手法である，Ambient Web Design による繊細な Web ページの洗練化のためにデータ駆動型 Associated Components が有効であることを示す．

これ以降の構成について説明する．2 節では，関連研究について説明する．Associated Components をデータ駆動

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

[*1] Associated Components の機能の一部分の実装を公開している．
https://blue0.an.cis.iwate-u.ac.jp/AssociatedComponents

型とするための設計について，3 節において述べる．4 節では実装を用いた 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

2.1 Associated Components による関連情報の提示

Web ページで提示される情報には相互に関連をもつものが多く，それらの関連を視覚的な情報として分かりやすく表現するためにグラフィカル・インタフェースの動作を用いた情報の関連づけがおこなわれる．関連づけには，次のような役割がある．

- 情報構造の明示
- 補足情報または追加情報の提示
- 操作の誘導

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，グラフィカル・インタフェースを構築するための研究が重要であるといえる．

例えば，ボタン要素にカーソルをホバーした場合に表示されるツールチップ[*2] [2] [13] は，そのボタンを押下した場合におこなわれる動作の説明を動的に提示するためなどに用いられる．ツールチップのような，ユーザ操作に対応するグラフィカル・コンポーネントの反応は一種の対話であり，マイクロインタラクション (microinteractions) [14] と呼ばれる．対話的操作を含んだ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．

また，PDA(Personal Digital Assistance) の普及に伴い，フラット・デザイン[*3] のように簡潔なデザイン・スタイルにより Web ページを制作する機会が増加している [1], [12], [15] ．簡潔なデザインを選択することにより，画面サイズが小さく計算性能の低い計算機においても快適なブラウズが可能となるが，一方では，デザインを簡潔にすることにより情報が不足する場合もあることから，それを補うための手法が重要であるといえる．

既提案の Associated Components [7], [8] では，複数の HTML 要素の動作を連携させることにより提示される情報の関連を明確にすることが可能である．一般的には，動作を含む Web ページの制作では HTML,CSS および JavaScript を使用するが，Associated Components を用いると HTML 標準規格 [16] に定義されるビルトイン要素 (HTML ビルトイン要素) と同様の文法により動作を含む Web ページを制作可能である．Web ページ制作でよく用いられる言語の設計では，HTML により文書構造を記述，CSS により外観を記述，そして，JavaScript により動作を記述するというように基本設計がおこなわれている．対話的な操作に応じて文書構造を変化させる動作については，JavaScript を用いて HTML の文書構造を変更するように記述されることから，その場合には，文書構造に関連するプログラムが HTML と JavaScript に分散して記述されることになる．Associated Components を用いると，HTML のみを用いて対話的な操作に応じた文書構造の変化を記述可能となる．

本稿では，Associated Components の HTML 要素属性を，head 部に局所的に記述するための手法を示す．それにより，HTML 要素の動作の関連づけの把握が容易になり，Web ページにおいて提示される情報の関連づけについてのプログラムの可読性が向上する．

2.2 データ駆動型 Web ページ

サーバに格納されたデータに基づき Web ページを生成するというデータ駆動型 Web ページ制作のための手法は，Web ページ制作のための標準的な手法のひとつとして確立されている．データ駆動型 Web ページの制作では，PHP などのプログラミング言語が用いられる．

また，D3 (Data Driven Documents) は，データ可視化のための JavaScript ライブラリであり，データにあわせて表示形式を変更するような高度にカスタマイズしたグラフを作成可能である．そのようなグラフを含む Web ページもデータ駆動型である．

Web ページ制作において HTML とともに用いられる CSS には，ブラウザ上に文書がレンダリングされる外観に関する情報が記述される．CSS に記述されたデータによりブラウザ上に表示されるグラフィカル・インタフェースの外観が決まることから，これもデータ駆動型 Web ページの身近な例であるといえる．

提案のデータ駆動型 Associated Components では，CSS と同様の書式の要素属性データシート (3.2 節) を用いる．それにより，要素属性の定義，変更，および，一部の要素属性の値のみが異なる複数のコンポーネントの生成が容易になる．なお，HTML ビルトイン要素の標準的な要素属性のみを用いる場合であっても，Associated Components とするための `is` 属性を指定することにより，データ駆動の機能のみを使用することが可能である．データ駆動型コンポーネントとすることにより，次のようなコーディングが可能となる．

- head 部において文書構造の動的な変化に関する情報 (Associated Components の要素属性) を局所的に記述し，body 部において静的な文書構造を記述する．
- head 部において，ほとんど全ての HTML 要素属性を記述する．これにより，Associated Components のみではなく，HTML ビルトイン要素についても局所性の高いプログラムの作成が可能となる．

[*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 [3] が選択される機会も増加している．

Vol.2020-CG-180 No.8
Vol.2020-DCC-26 No.8
Vol.2020-CVIM-223 No.8
2020/11/5

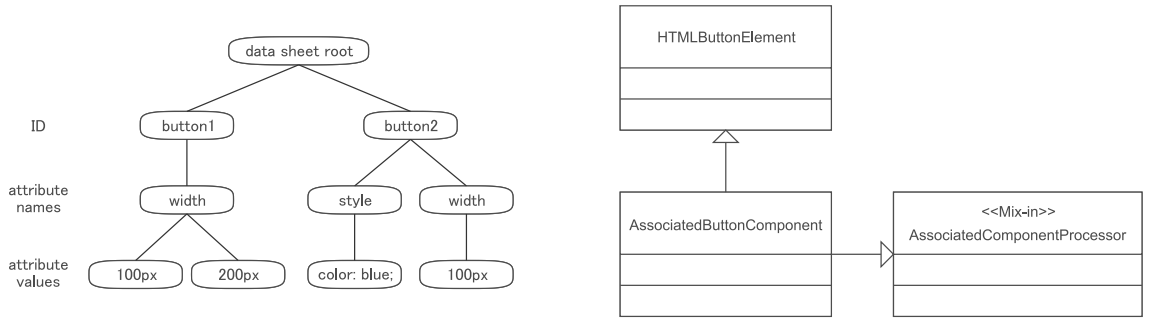


図 1: データ駆動型 Associated Components のソフトウエア設計
Fig. 1 Software design of data-driven Associated Components.

3. データ駆動型 Associated Components

3.1 ソフトエア設計

データ駆動型 Associated Components の設計について，その主要部分を図 1 に示す．クラス図には，Associated Components の構築において中心となるクラス間の関連が示されている[*4]．Associated Components の基本設計については [7], [8] に説明があるので，ここではこれ以降の説明において必要となる内容を中心に説明する．

図 1 のクラス図は，Web ブラウザに標準的に実装されているボタン要素に対して，Associated Components の機能を追加するためのクラス構成である．このように設計することにより，Associated Components では Associated Components の機能のみではなく HTML ビルトイン要素の機能も使用可能となる．そして，HTML 標準規格に定義される文法を用いて，HTML ビルトイン要素と Associated Components を混在させた HTML プログラムを記述可能である．

具体的には，AssociatedButtonComponent(ABC) クラスは，HTMLButtonElement(HBE) クラスから派生させたクラスであり，ABC クラスは HBE クラスのもつ 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 AssociatedComponent-Processor(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る．このようなクラス定義は，その HTML プログラムにおいて Associated Components として使用する HTML 要素の種類ごとに必要となるが，[7], [8] のように簡潔な実装が可能である．

データ駆動型 Associated Components が用いるデータは 3.2 節において提案する要素属性データシートに定義がおこなわれ，それに基づき Associated Components が生成される．図 1 に示す要素属性データシート木には，そのためのデータが格納される．データシートの構築などのデータシートを操作するための機能は，ABC クラスおよび ACP クラスにより実現される．

リスト 1 に，HTML プログラムでのデータ駆動型 Associated Components の使用方法を示す．head 部では，`script` 要素にデータ駆動型 Associated Components モジュールを指定し，`clm-associated-data-sheet` 要素に要素属性データシートの内容を記述している．body 部では，Associated Components として使用するボタン要素の `is` 属性に `clm-associated-button` を指定している．なお，HTML ビルトイン要素のボタン要素のための属性も追加して指定可能である．

リスト 1: データ駆動型 Associated Components の使用方法
Listing 1: Usage of data-driven Associated Components

```
<html>
<head>
  ...
  <script
    type="module"
    src="./DataDrivenAssociatedComponents.js"
  ></script>
  <clm-associated-data-sheet>
    (specifying the contents of this Data Sheet Element)
  </clm-associated-data-sheet>
</head>
<body>
  ...
  <button
    ...
    is='clm-associated-button'
  >
  </buttton>
</body>
</html>
```

[*4] Associated Components では，HTML 標準規格に定義されるカスタム要素のための機能を使用する．HTML 標準規格では，カスタム要素についての説明に JavaScript のクラス構文を用いている．本稿でも同様に，クラスによる表現を用いる．

Vol.2020-CG-180 No.8
Vol.2020-DCC-26 No.8
Vol.2020-CVIM-223 No.8
2020/11/5

リスト 2: 要素属性データシートの例
Listing 2: Example of an element attribute data sheet.

```
<clm-associated-data-sheet>
  #button01
  {
    style: "width: 100px;"
    data-RId: "button02"
  },
  ...
</clm-associated-data-sheet>
```

リスト 3: 要素属性値の多重定義の例
Listing 3: Example of multiple defined element attibute values.

```
<clm-associated-data-sheet>
  #button01
  {
    style: "width: 100px;",
           "width: 200px;",
           "width: 300px;"
  },
  ...
</clm-associated-data-sheet>
```

3.2 要素属性データシート

データ駆動型 Associated Components では，HTML 要素を生成するためのデータを，HTML プログラムの head 部の要素属性データシート要素に記述する．要素属性データシート要素は，Associated Components のためにカスタマイズした HTML 要素であり，要素名は `clm-associated-data-sheet` である．要素属性データシートでは CSS のサブセットの構文を用いる．HTML 要素の ID を指定してその要素に追加する属性を記述する (リスト 2)．HTML ビルトイン要素の属性 (緑色) および Associated Components のカスタム属性 (茶色) [7], [8] を記述可能である．

要素属性データシートの書式を，次に示す．

(♯*id* {
(*attribute name* : " *attribute value* ";)
})

(*) は，() で囲んだ内容 * の 0 回以上の繰り返しである．同一の属性について属性名と値の組を複数記述することにより多重定義が可能である．多重定義がおこなわれると，その id 属性値をもつ要素に加えて，その id 属性値に $-n$ を付加した ID をもつ要素がクローン生成され，それらに多重定義した属性が追加される．ここで，n は $[1, \ldots, \mathtt{N}-1]$ の整数であり，N は多重定義した要素属性の個数である．

次に示す簡略化した書式による多重定義も可能である．

(♯*id* {
(*attribute name* : (" *attribute value* " ,)
" *attribute value* ")
})

リスト 3 は，`style` 属性について多重定義をおこなう例である．この場合には，$\mathtt{N} = 3$ である．

4. 実験と結果の検討

4.1 実装

提案のデータ駆動型 Associated Components を実装し，Ambient Web Design [9] のためのグラフィカル・コンポーネントの制作をおこなった．提案のコンポーネントでは，HTML 標準規格に定義される新機能[*5] を使用するので，動作の確認には Chrome、Firefox および Edge のモダンブラウザを使用した．プログラミング言語には HTML, CSS, および JavaScript を用いた．

データ駆動型 Associated Components の実装において HTML 標準規格に定義される機能をどのように使用しているかについて簡単に説明する．データ駆動型 Associated Components のための機能を担う部分については、HTML 標準規格に定義されるカスタム要素[*6] [*7] [11], [16] のための機能および shadow DOM [*8] のための機能を JavaScript のプログラムから使用した．データシート木およびグループ木 [7], [8] は，shadow DOM として実装した．shadow DOM は標準の DOM ツリーとは独立した DOM ツリーであるので，DOM ツリーに含まれるデータの役割にあわせたデータ構造の明確化が可能である．また，shadow DOM を用いることにより，プログラムの可読性およびセキュリティについても向上するという利点がえられる．

4.2 Associated Components による関連情報の提示

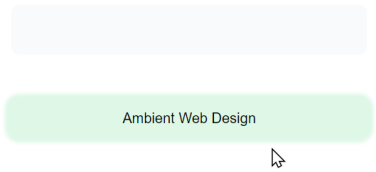
Associated Components を用いた HTML プログラムの動作を，図 2 に示す．カーソルがボタン要素をホバーすると，その上側に配置された `div` 要素に関連づけられた情報が表示される例である．ボタン要素および `div` 要素は Associated Components であり，ボタン要素がホバーされるとボタン要素から `div` 要素に情報が送信され，`div` 要素にその情報が表示されている．

[*5] カスタム要素および shadow DOM のための機能を用いる．これらの機能は，標準的なモダンブラウザに実装されている．

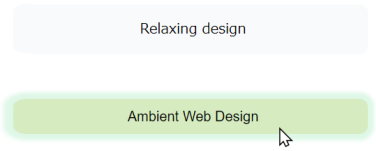
[*6] HTML 標準規格に定義されるカスタム要素 (custom elements) を用いると，カスタマイズした GUI コンポーネントを作成できる．カスタム要素のことを，Web コンポーネント (Web Components) と呼ぶことも多い [11] .

[*7] 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．

[*8] shadow DOM は HTML 標準規格において定義される DOM であり，標準の DOM ツリーとは独立した DOM ツリーである．



(a) Rendering Associated Components.



(b) Displaying associated information.

図 2: Associated Components による関連情報の提示
Fig. 2 Providing associated information with Associated Components.

リスト 4: 図 2 のコード
Listing 4: Code of **Fig. 2**.

```
<table ... >
  <tr>
    <td>
      <div
        id='informationArea'
        is='clm-associated-button'
        ...
      >
      </div>
    </td>
  </tr>
  <tr>
    <td>
      <button
        id='button1'
        is='clm-associated-button'
        data-rId='informationArea'
        onmouseover="
          setRInnerHTML('Relaxing design')"
        onmouseout="setRInnerHTML('')"
        ...
      >
        Ambient Web Design
      </button>
    </td>
  </tr>
</table>
```

リスト 4 は，そのプログラムの一部分である．内容を簡単に説明する (詳細は，[7], [8] を参照)．

ボタン要素では，HTML ビルトイン要素のボタン要素に Associated Components のための機能を追加するために，`is` 属性に `clm-associated-button` を指定している．

そして，ボタン要素と動作を関連づける要素 (receiver) の ID として，`data-rId` 属性に `informationArea` というように `div` 要素の ID を指定している．これにより，ボタン要素から `div` 要素への関連づけが設定される．`onmouseover` 属性には，ボタン要素がホバーされた場合に receiver に対しておこなう操作を，`onmouseout` 属性には，カーソルがボタン要素を外れた場合に receiver に対しておこなう操作を指定している．以上のように，Associated Components のためのカスタム要素属性を HTML ビルトイン要素に指定するだけで，ボタン要素と `div` 要素の動作を連携させることが可能である．なお，head 部においては，Associated Components のためのモジュールの指定が必要である (リスト 1 参照)．

このように，Associated Components では HTML 標準規格に定義された文法を使用可能であり，Associated Components のためのカスタム要素属性と HTML ビルトイン要素のための要素属性をともに指定することも可能である．なお，Associated Components と HTML ビルトイン要素を関連づけて動作させることも可能である．

4.3 データ駆動型 Associated Components の例

図 3 は，データ駆動型 Associated Components の例である．複数のボタン要素のなかのあるボタン要素をカーソルがホバーすると，その背景色の設置値を上部に配置された `div` 要素に表示するプログラムである．ボタン要素では，Associated Components の機能を使用するために `is` 属性を指定している．

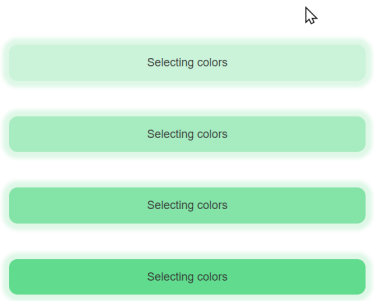
リスト 5 は，図 3 の HTML プログラムの head 部において要素属性データシート要素を記述している部分である．ID が `associatedButtonForSelectingColors` である要素について，`style` および `onmouseover` 属性を多重定義している．`style` 属性については，各要素の背景色を徐々に変更するように多重定義し，`onmouseover` 属性については，その要素がホバーされた場合にその要素の背景色の設定値を関連づけられた要素に表示するように多重定義している．この多重定義に基づいて，複数の要素が生成されることになる．

body 部では 1 個のボタン要素を定義し，その ID を `associatedButtonForSelectingColors` としている．要素属性データシートでは，この ID をもつ要素についての多重定義がおこなわれているので，多重定義された属性をもつ複数のボタン要素が生成される．

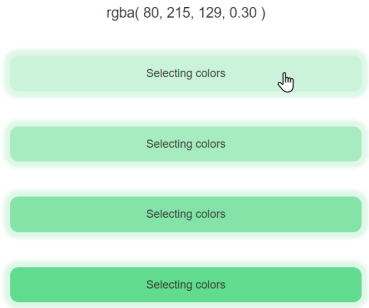
4.4 GUI コンポーネントの洗練化

図 4 に，GUI コンポーネントの洗練化の例を示す．意図にあったボタン要素を制作するために，背景色および形状を少しずつ変更した複数のボタン要素を生成し，それらを表示している．

Vol.2020-CG-180 No.8
Vol.2020-DCC-26 No.8
Vol.2020-CVIM-223 No.8
2020/11/5



(a) Rendering data-driven Associated Components.



(b) Displaying associated information.

図 3: データ駆動型 Associated Components の例
Fig. 3 Example of data-driven Associated Components.

リスト 5: 図 3 のコード
Listing 5: Code of **Fig. 3**.

```
<head>
  ...
  <clm-associated-data-sheet>
    #associatedButtonForSelectingColors{
      style : "background: rgba( ... ); ... ",
              "background: rgba( ... ); ... ",
              "background: rgba( ... ); ... ",
              "background: rgba( ... ); ... "
      onmouseover : "setRInnerHTML( '<label> ... ' )",
                    "setRInnerHTML( '<label> ... ' )",
                    "setRInnerHTML( '<label> ... ' )",
                    "setRInnerHTML( '<label> ... ' )"
    }
  </clm-associated-data-sheet>
  ...
</head>
```

この例では，ボタン要素に `is` 属性を指定することにより，Associated Components の機能を使用している．コンポーネントの関連づけのための機能は使用せずに，要素属性データシートに基づいて複数のコンポーネントを生成するための機能を使用している．

4.3 節ではリスト 5 のように，head 部の要素属性データシート要素の値として要素属性を記述したが，そのように

図 4: GUI コンポーネントの洗練化
Fig. 4 Refinement of GUI components.

リスト 6: 図 4 のコード
Listing 6: Code of **Fig. 4**.

```
$border-radiuses: 0px, 5px, 10px, 15px, 25px;
$blue-step: 10;

#refinement01{
  @each $border-radius in $border-radiuses {
    style : "border-radius : #{$border-radius};"
  }
  @each $border-radius in $border-radiuses {
    $blue-step: $blue-step + 50;
    $blue: 50 + $blue-step;
    ... background-color: rgba( ..., #{$blue}, ...
  }
  ...
}
```

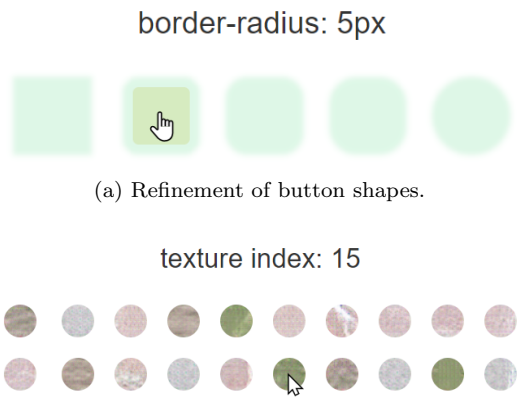
せずに，要素属性データシートの内容を記述したファイルを作成し，そのファイル名を要素属性データシート要素 (`clm-associated-data-sheet`) の `src` 属性に指定することも可能である．

ここでは，要素属性データシート要素の `src` 属性を使用したが，Sass [5] [*9] を用いて `src` 属性に指定するファイルを生成した．データ駆動型 Associated Components の要素属性データシートの構文には CSS のサブセットの構文を採用していることから，Sass を使用して要素属性データシートの内容を生成可能である．

リスト 6 に示すファイルを Sass の入力として，要素属性データシートの内容をファイルに生成した．そして，そのファイルを HTML プログラムの head 部の要素属性データシート要素の `src` 属性に指定することにより，図 4 の各ボタン要素を生成した．なお，HTML プログラムの body 部において記述したボタン要素の個数は 1 個である．

[*9] Sass は，CSS を生成するためのフロントエンドプロセッサとして使用される言語である．変数，繰り返し文などを使用でき，演算も可能である．簡潔な可読性の高い記述を基に CSS を生成可能である．

Vol.2020-CG-180 No.8
Vol.2020-DCC-26 No.8
Vol.2020-CVIM-223 No.8
2020/11/5



(a) Refinement of button shapes.

(b) Selecting Wash texture images.

図 5: データ駆動型 Associated Components の多重定義

Fig. 5 Multiply defined Associated Components: refinement of GUI componemts (a) and listing the Washi texture images generated using a deep generative model (b).

4.5 データ駆動型 Associated Components の多重定義

図 5 は，多重定義したデータ駆動型 Associated Components の動作を関連づけた例である．図 5 (a) では，4.4 節と同様にボタン要素の洗練化をおこなっているが，ここでは各ボタン要素とその上側に配置された `div` 要素の動作が関連づけられている．カーソルがボタン要素をホバーすると，そのボタン要素の属性の値が `div` 要素に表示される．図 5 (b) では，データ駆動型 Associated Components の多重定義により複数の `img` 要素を生成している．`img` 要素には，深層生成モデルにより学習した和紙のテクスチャモデル [6] から生成したテクスチャ画像を表示している．`img` 要素とその上側に配置された `div` 要素の動作が関連づけられていて，画像をカーソルでホバーするとそのテクスチャのインデックスを `div` 要素に表示するようになっている．

図 5 では，多重定義した要素属性データシートの内容を Sass を使用してファイルに生成し，それを要素属性データシート要素の `src` 属性に指定した．リスト 7 は，図 5 (b) において Sass の入力としたファイルの内容である．

4.6 検討

図 2 およびリスト 4 からは，Associated Components を用いると，HTML 標準規格に定義された文法のみを用いて HTML 要素の動作の関連づけが可能であることが分かる．コンポーネントのデザインでは，落ち着いた雰囲気の Web ページを制作することを目的とした Ambient Web Design の特徴である，コントラストが小さい淡い色調および形状の柔らかさを基本としたデザインをおこなっている．GUI コンポーネントなので，選択されたボタンのコントラストは大きくするようにしている．

図 3 およびリスト 5 からは，データ駆動型 Associated Components の要素属性データシート要素の値として記述した要素属性の多重定義に基づいて，複数のコンポーネントを生成可能であることが分かる．head 部での要素属性の多重定義により複数のコンポーネントの要素属性を機能にあわせてまとめて簡潔に記述可能であることが分かる．

図 4 およびリスト 6 からは，簡潔な記述により，背景色および形状を少しずつ変更した複数の HTML 要素を生成可能であることが分かる．パラメータを微調整しながら多くのデザインを試行錯誤することが容易であることから，Ambient Web Design のような繊細なデザインを洗練化するような場合に適しているといえる．

図 5 およびリスト 7 からは，データ駆動型 Associated Components を用いると簡潔な記述により複数の HTML 要素の動作の関連づけが可能であることが分かる．要素属性データシートの生成では，CSS のフロントエンドである Sass を利用可能であり，明瞭で簡潔な記述を基に要素属性データシートを生成可能であることが分かる．

これらの実験で用いたプログラムからは，HTML ビルトイン要素と Associated Components を混在させてシームレスに使用可能であることが分かる．図 1 のクラス設計をおこない HTML 標準規格に定義されるカスタム要素のための機能を活用することにより，そのような使用を可能にしている．

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，JavaScript プログラムを簡略化するための手法の開発[*10]，および，独自の DSL(Domain Specific Languages) を用いた研究[*11] などがある．

リスト 7: 図 5 (b) のコード

Listing 7: Code of **Fig. 5** (b).

```
$texture-indexes01: 0, 1, 2, ...;
...
#textures01{
  @each $texture-index in $texture-indexes01 {
    onmouseover:"setRInnerHTML('... #{$texture-index}')"
  }
  @each $texture-index in $texture-indexes01 {
    src: './path/to/Texture-#{$texture-index}.png'
  }
}
...
```

[*10] `Polymer`[4] 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．

[*11] カスタム要素に関連する研究である [10] では，GUI コンポーネントを生成するための DSL を提案している．

Associated Components の特徴は，HTML ビルトイン要素と同様の記述により関連づけられた情報を提示することである．HTML プログラムでのコンポーネントの記述により動作を伴う Web ページのプログラムを記述できることから，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 GUI コンポーネントであるといえる．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コンポーネントの動的な関連の把握が容易であり，コンポーネントの関連の解析ツールなどの開発も容易である．

5. おわりに

Web ページにおいて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 を，データに基づきコンポーネントを生成するデータ駆動型とするための手法を提案し，実装をおこない手法の有効性を確認した．

データ駆動型とすることにより，既提案のコンポーネントのもつ特徴に加えて，次の特徴を備えた手法となった．1) 複数の HTML 要素の関連の定義を head 部にデータとして局所的に記述可能であることから，HTML 要素の関連の把握が容易である．2) ある HTML 要素属性の値のみが異なる複数の HTML 要素を簡潔に記述可能である．

特に，外観または動作を少しずつ変更した HTML 要素を多数生成することが容易であり，落ち着いた雰囲気をもつ Web ページを制作するための手法である Ambient Web Design のような，繊細な Web ページの制作でのグラフィックス・デザインの洗練化のために有効である．

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機能の改良，関連づけの可視化手法に関する研究などがある．

参考文献

[1] 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).

[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: Material Design, `https://material.io/(Retrieved: 8 Octover 2019)`.

[4] Google: Polymer Project, `https://www.polymer-project.org/(Retrieved: 8 Octover 2019)`.

[5] Jobsen, B.: Sass and Compass Designer's Cookbook, Packt Publishing (2016).

[6] 佐藤　信：深層生成モデルにより学習した潜在空間を用いた和紙テクスチャの補間，情報処理学会研究報告，Vol. 2018-CG-169, No. 2, pp. 1–6 (2018).

[7] 佐藤　信：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，情報処理学会研究報告，Vol. 2020-CE-154, No. 13, pp. 1–8 (2020).

[8] 佐藤　信：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，情報処理学会研究報告，Vol. 2020-HCI-187, No. 33, pp. 1–8 (2020).

[9] 佐藤　信：落ち着いた雰囲気のインフォグラフィックス，2020 年度電気関係学会東北支部連合大会予稿集，p. R06 (2020).

[10] 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).

[11] Mozilla Developer Network: Web Components, `https://developer.mozilla.org/en-US/docs/Web/Web_Components(Retrieved: 4 April 2019)` (2017).

[12] 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).

[13] 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).

[14] Saffer, D.: *Microinteractions: Full Color Edition Designing with Details*, O'Reilly Media, Inc., 1st edition (2013).

[15] 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).

[16] WHATWG: HTML Living Standard — Last Updated 18 July 2019, `https://html.spec.whatwg.org/(Retrieved: 25 July 2019)`.