

Associated Componentsを用いた webベースのプレゼンテーション

佐藤 信¹

概要: 本稿では, Associated Components を用いた web ベースのプレゼンテーションについて述べる. 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既提案のコンポーネントである. 例えば, HTML プログラムにおいて HTML 標準規格に定義される文法をそのまま用いて, ボタン要素に Associated Components のための属性を指定することにより複数のボタン要素の動作の関連づけが可能である. 本稿では, Associated Components を用いて, 説明している内容をグラフィカルにポインティングするための手法を提案する. 説明にあわせた対話的なポインティングが可能であることから, プレゼンテーションの内容の理解を容易にするための有効な手法であるといえる.

Web-based Presentation with Associated Components

MAKOTO SATOH¹

Abstract: This paper describes a web-based presentation method using Associated Components. The components are proposed for providing associated information on web pages. For example the behaviors of multiple button elements can be associated by adding the attributes customized for Associated Components to the elements, using the syntax defined in the HTML standard for writing the HTML code. This paper proposes a method for pointing presentation contents graphically using Associated Components. The method is effective for easy understanding of presentations because it enables interactive pointing in line with presentations.

1. はじめに

本稿では, Associated Components ^{*1} [6], [8] を用いた web ベースのプレゼンテーションについて述べる. Associated Components は, web ページ制作のためのグラフィカル・コンポーネントであり, 次の特徴をもつ.

- 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.
- HTML 標準規格 [15] に定義される文法により記述した HTML プログラムにより, Associated Components

と HTML 標準規格に定義される HTML 要素 (HTML ビルトイン要素) が混在した web ページを制作可能である.

これ以降では, web ページを用いたプレゼンテーションについて述べる. Associated Components によりコンポーネントの動作を関連づけることにより, 説明している部分をグラフィカルにポインティングするための手法を提案する. そして, そこでのポインティングのパターンについて検討する. 視覚的に明瞭なポインティングが可能であることから, 説明内容の理解を容易にするための有効な手法であるといえる.

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 を用いたプレゼンテーションについて, 3 節において述べる. 4 節では

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>

プレゼンテーションのためのコンポーネントの構成例を示し、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
- プレゼンテーションでのポインティング

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、情報の関連を提示するためのグラフィカル・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} [3] [12] 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) [13] 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 [2], [11], [14]。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 [6], [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 [15] 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

^{*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。

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よる文書構造の変化の記述が可能となる。

本稿では、web ページを使用したプレゼンテーションを対象として、Associated Components により複数のコンポーネントの動作を関連づけることにより、グラフィカルな動作により説明している内容をポインティングするための手法を提案する。

2.2 エンドユーザ・プログラミング

計算機の普及に伴い、計算機を専門としない一般の方々がアプリケーション・ソフトウェアの機能をカスタマイズする機会が増加している。例えば、表計算ソフトウェアを使用する場合に、自分の仕事にあわせて専用のスクリプトを作成することによりソフトウェアの機能をカスタマイズすることなどがおこなわれる。アプリケーション・ソフトウェアのユーザがおこなうこのようなプログラミングは、エンドユーザ・プログラミング (EUP) [1] と呼ばれる。[1] において報告されているように、多くの分野において EUP がおこなわれ、その形態および技術レベルについては様々なものが存在する。

EUP において用いられる手法は、コンポーネント・ベースの手法が多数を占めるという調査結果が報告されている [1]。HTML プログラムの作成において用いられる HTML 要素は GUI (Graphical User Interface) のためのソフトウェア・コンポーネントであり、専門家でない方々が簡単な web ページを自分で制作する、あるいは、専門家に制作していただいた web ページの HTML プログラムに目的にあわせて簡単な変更を加えるというような場合も EUP である。

本稿では、インターネットなどのネットワーク経由での web ページによるプレゼンテーションにおいて、説明している内容をグラフィカルにポインティングするために、Associated Components による複数の HTML 要素の動作の関連づけ機能を活用する手法を提案する。HTML 標準規格に定義される文法により記述した HTML プログラムを作成することにより、視覚的に明瞭なポインティングが可能となる web ページを制作可能であることから、Associated Components を用いるとプレゼンテーションのための web ページ制作のための EUP が容易であるといえる。

3. Web ペースのプレゼンテーション

3.1 Associated Components の使用方法

これ以降での説明のために、Associated Components の使用方法について説明する。なお、コンポーネントの設計については、[6], [7], [8] に説明がある。

リスト 1 は、Associated Components を使用する HTML プログラムの例である。このように、HTML 標準規格の文法により記述した HTML プログラムに Associated Components のための記述を追加することにより Associated Components の機能を使用可能である。具体的には、次のとおりである。

- head 部において、script 要素を用いて Associated Components モジュールを指定する (茶色) *4。
- body 部において、Associated Components の機能を用いる HTML 要素 (ここではボタン要素) の is 属性に clm-associated-button を指定する (緑色)。

以上により、Associated Components を使用するための準備が完了である。なお、Associated Components をデータ駆動型 [7] とする場合には、要素属性を head 部の clm-associated-data-sheet 要素に記述する (青色)。

次に、リスト 2 のように Associated Components のためのカスタム属性およびカスタム・イベント・ハンドラを用いて、ボタン要素と div 要素の動作を関連づける。

- ボタン要素において、data-rId 属性により動作を関連づける要素 (receiver) の ID を、informationArea と指定する (茶色)。informationArea は、div 要素の ID である。
- ボタン要素の onmouseover 属性において、カーソルがホバーした場合の動作として、カスタム・イベント・ハンドラ setInnerHTML を指定する。この例では、ホバーにより文字列 'Relaxing design' が receiver である div 要素に設定される。なお、文字のエスケープなどを適宜おこなうことにより任意の HTML コードを設定可能である。設定されたコードを HTML プログラムとして実行した結果がレンダリングされる。

*4リスト 1 において Associated Components モジュールのパスを指定している部分 (./AssociatedComponents.js) を、インターネットに公開しているモジュールの URI に変更した HTML プログラムを作成すると、インターネット経由で Associated Components の動作を確認可能である。モジュールには、Associated Components の一部分の機能を実装しているが、IPJS の研究会などで示したプログラム例については動作確認が可能である。詳細については、<https://blue0.an.cis.iwate-u.ac.jp/AssociatedComponents> に説明の資料を掲載している。

リスト 1: Associated Components の使用方法 (1)

Listing 1: Usage of Associated Components (1)

```
<html>
<head>
  ...
  <script
    type="module"
    src="./AssociatedComponents.js"
  ></script>
  <clm-associated-data-sheet>
    Specifying the contents of Data Sheet element.
  </clm-associated-data-sheet>
</head>
<body>
  ...
  <button ...
    is='clm-associated-button'
  >
  </button>
</body>
</html>
```

リスト 2: Associated Components の使用方法 (2)

Listing 2: Usage of Associated Components (2)

```
...
<div id='informationArea' ... ></div>
...
<button
  is='clm-associated-button'
  data-rId='informationArea'
  onmouseover="setInnerHTML('Relaxing design')"
  ...
>Ambient Web Design</button>
...
```

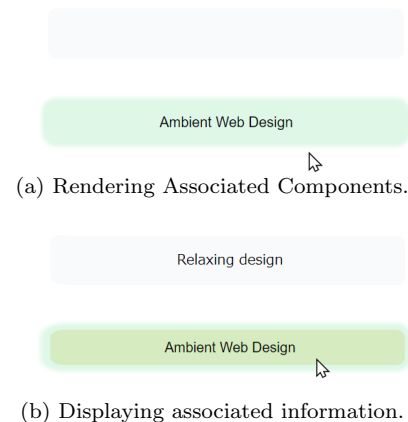


図 1: Listing 1, 2 の動作
Fig. 1 Behavior of Listing 1 and 2

表 1: ポインティング・パターンの分類

Table 1 Classification of Pointing Patterns

type	cursor position	interesting position
type 1	element A	element A
type 2	element A	element A and B
type 3	element A	element B

3.2 Associated Components によるポインティング

3.1 節での説明のように, Associated Components を用いると, 簡潔な記述の HTML プログラムにより複数の HTML 要素の動作の関連づけが可能である. ここでは, 要素の動作の関連づけを, web ベースのプレゼンテーションでのポインティングに用いるための手法を提案する.

ポインティングのパターンを分類するために, HTML 要素の位置を, イベントが発生した要素の位置 (Cursor Position, CP) および説明において着目している要素の位置 (Interesting Position, IP) に分けて考える. イベントの例としては, カーソルがその要素をホバーした場合に発生するイベントなどがある. 基本的なポインティングのパターンを CP と IP について分類すると, 表 1 のようになる.

タイプ 1 は, CP と IP が同じ HTML 要素の場合であり, ある HTML 要素に発生したイベントに対応して, その要素自身に動作を関連づける場合である. 例えば, ある HTML 要素をカーソルがホバーすることにより, その要素の背景色を変更する, または, サイズを変更するなどがある.

タイプ 2 は, CP にある HTML 要素およびその要素以外の HTML 要素が IP となる場合である. ある HTML 要素に発生したイベントに対応して, その要素自身およびその要素以外の要素に動作を関連づける場合である.

タイプ 3 は, CP と IP が異なる HTML 要素の場合である. ある HTML 要素に発生したイベントに対応して, その要素以外の要素に動作を関連づける場合である.

表 1 では, 1 つのイベントが発生した場合について, CP および IP に着目することによりポインティングの分類をおこなった. これらの基本パターンを組み合わせることにより, 多様なポインティング・パターンを構成可能である. 例えば, カーソルがある要素をホバーしてからその要素を離れる場合には 2 種類のイベントが発生することになり, それぞれのイベントにあわせた動作の関連づけが可能である. 例えば, カーソルがある要素をホバーするとその要素の背景色を変更し, カーソルがその要素を離れると背景色を元に戻すなどがある.

なお, Associated Components では, 1 つのイベントに

対して複数の HTML 要素の動作の関連づけが可能である. HTML 要素のグループを定義することにより, グループに含まれる各要素を対象として動作を関連づけることも可能である.

4. 実験と結果の検討

4.1 実装

インターネットなどのネットワーク経由での web ページを用いたプレゼンテーションにおいて説明内容をグラフィカルにポインティングすることを目的として, Associated Components による HTML 要素の動作の関連づけをおこなった. web ページの HTML プログラムをプレゼンテーションの内容にあわせて作成するというプログラムの作成形態は, エンドユーザ・プログラミング (EUP) [1] (2.2 節) である. そのためここでは, HTML 標準規格に定義される文法を用いて HTML プログラムを記述するだけで HTML 要素の動作の関連づけが可能になっている仕組みを中心として, Associated Components の実装について説明する (詳細は, [6], [7], [8] を参照).

Associated Components の実装では, HTML 標準規格に定義されるカスタム要素のための機能^{*5} および shadow DOM のための機能を用いた^{*6} [10], [15]. カスタマイズしたビルトイン要素^{*7} のための機能を用いることにより, HTML ビルトイン要素が備える標準的な機能を Associated Components に継承しながら Associated Components のための機能の追加をおこなった. これにより, HTML ビルトイン要素において使用可能な標準的な要素属性に加えて Associated Components のための要素属性を使用可能となり, 標準的な文法により記述した HTML プログラムにおいて Associated Components を使用可能となった. また, グループに所属する要素との動作の関連づけ [6] のためにグループ木を用いた. そして, Associated Components をデータ駆動 [7], により動作させるためにデータシート木を用いた. これらの木構造データの取り扱いの実装には, 自律的カスタム要素のための機能および shadow DOM ^{*8} を用いた. なお, 実装のためのプログラミング言語には JavaScript を用いた.

^{*5}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 [10].

^{*6}これらの機能は, Chrome, Firefox および Edge などのモダンブラウザに標準機能として実装されている.

^{*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.

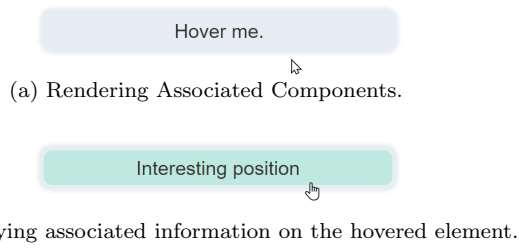


図 2: ポインティング・パターン type 1 (cf. Table 1)

Fig. 2 Pointing pattern type 1 (表 1 参照)

リスト 3: 図 2 のコード

Listing 3: Code of Fig. 2.

```
<button
  id='IPAndCP'
  is='clm-associated-button'
  data-rId='IPAndCP'
  onmouseover="setInnerHTML('Interesting position');
               setStyle('background: rgba( ... );');"
  onmouseout="setInnerHTML('Hover me. ');
              setStyle('background: rgba( ... );');"
  ...
>
  Hover me.
</button>
```

4.2 ポインティング・パターン type 1

図 2 は、表 1 のポインティング・パターンの type 1 の例である。この場合の動作は次のとおりである。

Step 1 ボタン要素をカーソルがホバーすることにより、ボタン要素にイベントが発生する。そのボタン要素の位置が、Cursor Position(CP) である。

Step 2 Step 1 において発生したイベントにあわせて、そのボタン要素は説明のための文字列をその要素に再設定する。その要素が説明において着目している要素となるので、その要素の位置が、Interesting Position(IP) である。

リスト 3 は、図 2 の実験に用いた HTML プログラムのボタン要素の記述である。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 (茶色)。そして、その要素と動作を関連づける要素 (receiver) の ID として、data-rId 属性に IPAndCP を指定している (緑色)。IPAndCP はその要素の ID なので、自分自身に対して関連づけた動作をおこなうように設定している。onmouseover 属性には、その要素をホバーした場合に receiver に対しておこなう動作として、文字列および背景

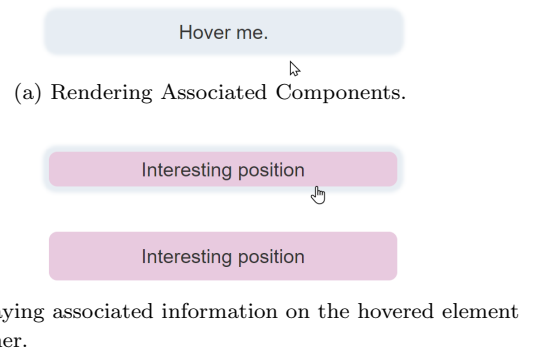


図 3: ポインティング・パターン type 2 (cf. Table 1)

Fig. 3 Pointing pattern type 2 (表 1 参照)

リスト 4: 図 3 のコード

Listing 4: Code of Fig. 3.

```
<button
  id='CP'
  is='clm-associated-button'
  data-rId='CP IP'
  onmouseover="setInnerHTML('Interesting position');
               setStyle('background: rgba( ... );');"
  onmouseout="setRid( 'CP' );
              setInnerHTML('Hover me. ');
              setStyle('background: rgba( ... );');"
              setRid( 'IP' );
              setInnerHTML('');
              setStyle('background: white');"
              setRid( 'CP IP' );"
  ...
>
  Hover me.
</button>
...
<div id='IP' ... ></div>
```

色の設定を指定している (青色)。onmouseout 属性では、カーソルが離れた場合に receiver をホバーする前の状態に復元する操作を指定している。

4.3 ポインティング・パターン type 2

図 3 は、表 1 のポインティング・パターンの type 2 の例である。ボタン要素のホバーにあわせて、そのボタン要素およびその下側に配置された div 要素に文字列および背景色を設定している。ボタン要素の位置が CP であり、ボタン要素および div 要素の位置が IP である。

リスト 4 は、図 3 の実験に用いた HTML プログラムのボタン要素および div 要素の記述である。ボタン要素では、data-rId 属性により receiver の ID に CP および IP を指

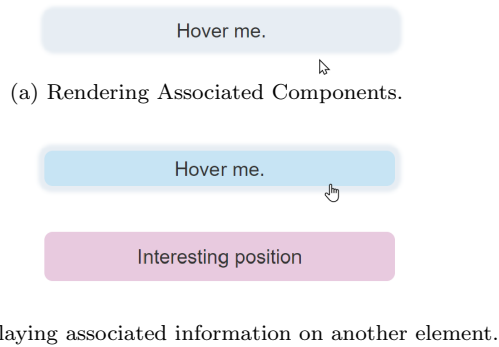


図 4: ポインティング・パターン type 3 (cf. Table 1)

Fig. 4 Pointing pattern type 3 (表 1 参照)

リスト 5: 図 4 のコード

Listing 5: Code of Fig. 4.

```
<button
  id='CP'
  is='clm-associated-button'
  data-rId='IP'
  onmouseover="setInnerHTML('Interesting position');
               setRStyle('background: rgba( ... );');"
  onmouseout="setInnerHTML('');
              setRStyle('background: white');"
  ...
>
  Hover me.
</button>
...
<div id='IP' ... ></div>
```

リスト 6: リスト 5 のデータ駆動版

Listing 6: Data-driven version of Listing 5.

```
<clm-associated-data-sheet>
  #CP{
    data-rId: "IP";
    onmouseover:"setInnerHTML('Interesting position');
                 setRStyle('background: rgba( ... );');"";
    onmouseout: "setInnerHTML('');
                setRStyle('background: white');"";
  }
</clm-associated-data-sheet>
```

定している (緑色). CP および IP は, それぞれボタン要素および div 要素の ID である. 自分自身を含めた複数の要素との動作の関連づけを設定している. `onmouseout` 属性では, カスタム・イベント・ハンドラ `setRId` により receiver を変更しながら, ホバーする前の状態に復元している.

4.4 ポインティング・パターン type 3

図 4 は, 表 1 のポインティング・パターンの type 3 の例である. ボタン要素のホバーにあわせて, ボタン要素の下側に配置された div 要素に文字列および背景色を設定している. ボタン要素の位置が CP であり, div 要素の位置が IP である.

リスト 5 は, 図 4 の実験に用いた HTML プログラムのボタン要素および div 要素の記述である. ボタン要素では, `data-rId` 属性により receiver の ID に IP を指定している (緑色). IP は, div 要素の ID である. 他の要素との動作の関連づけを設定している.

リスト 6 では, リスト 5 のデータ駆動版である. head 部の `clm-associated-data-sheet` 要素に, カスタム属性などを記述している (3.1 節を参照).

4.5 プレゼンテーションのコンテキストの反映

図 5 は, プレゼンテーションのコンテキストを反映したポインティングの例である. ボタン要素とその下側に配置した td 要素の動作を, 次のように関連づけた.

- ボタン要素および td 要素をレンダリングする.
- ボタン要素をカーソルがホバーすることにより, td 要素での表示を拡大する. ボタン要素の位置が CP であり, td 要素の位置が IP である.
- カーソルを移動して, td 要素をホバーする. td 要素をグラフィカルにハイライトする. td 要素の位置が CP および IP である.
- カーソルが td 要素から離れると, (a) の状態を復元する.

ここで想定されるプレゼンテーションのコンテキストは, 例えば次のような場合である.

- ボタン要素にはキーワードが表示され, td 要素にはキーワードの説明が表示されている. 説明を開始する.
- これから説明するキーワードが何かを述べる.
- キーワードの説明をする.
- 説明を終了する.

この例では, (b) から (c) へのプレゼンテーションの進行にあわせて CP が変更される. また, (b) と (c) では IP のハイライトの方法を変更することにより, (b) から (c) への移行をグラフィカルに印象づけている.

なお, HTML ビルトイン要素に Associated Components のための機能を追加するためには, ボタン要素では `is` 属性に `clm-associated-button` を指定するが, td 要素では `is` 属性に `clm-associated-td` を指定する.

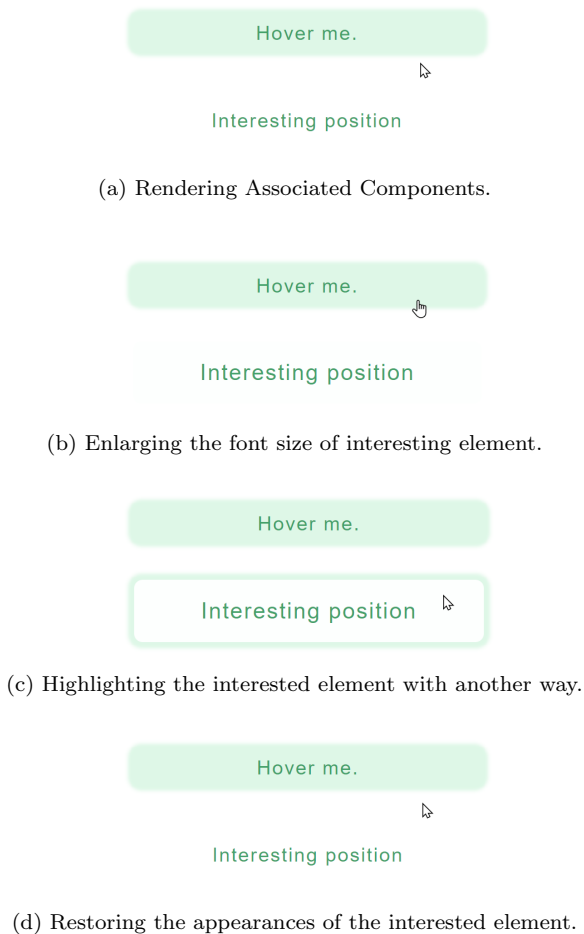


図 5: プレゼンテーションのコンテキストの反映
Fig. 5 Reflecting the context of presentation

リスト 7 は、図 5 の実験に用いた HTML プログラムでの動作の関連づけの記述の部分である。ここでは、head 部の `clm-associated-data-sheet` 要素に、関連づけを定義している。要素 ID が CP であるボタン要素については、`data-rId` 属性により receiver の ID に IP を指定している (緑色)。`onmouseover` 属性により、カーソルでホバーした場合の動作として、背景色およびフォントサイズの設定をおこなっている。なお、背景色には非常に薄い色を用いている。以上により、(b) の動作の関連づけの設定がおこなわれる。要素 ID が IP である td 要素については、`data-rId` 属性により receiver の ID に IP を指定している (緑色)。`onmouseover` 属性では、ボタン要素と同様の定義をおこなっている。これにより、(c) の動作の関連づけのうちの背景色およびフォントサイズの設定が定義される。縁取りによりハイライトする動作は、td 要素が使用している CSS の `class` で定義されている。Associated Components を使用する HTML プログラムでは、HTML 標準規格に定義される文法を自由に使用できる。そのため、CSS についても、Associated Components と組み合わせ使用可能である。

リスト 7: 図 5 のコード

Listing 7: Code of Fig. 5.

```
<clm-associated-data-sheet>
#CP{
  data-rId: "IP";
  onmouseover: "setRStyle( 'background-color: ... ; \
                font-size: 32px; ' );";
}
#IP{
  data-rId: "IP";
  onmouseover: "setRStyle( 'background-color: ... ; \
                font-size: 32px;' );"
  onmouseout: "setRStyle('background: white; \
                font-size: 28px;' );";
}
</clm-associated-data-sheet>
```

4.6 検討

4.2, 4.3, 4.4 節の実験結果から、Associated Components を用いて HTML 要素の動作を関連づけることにより、表 1 に示す各ポインティング・パターンを実現できることが分かる。図 2, 3, 4 からは、各パターンを用いた具体的な動作を確認できる。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、HTML 要素の動作を連携させることが可能であることを、リスト 3, 4, 5 から確認できる。また、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。リスト 5 のデータ駆動版であるリスト 6 からは、複数の HTML 要素の動作の関連づけのためのカスタム属性を head 部に局所的に記述可能であることが分かる。

4.5 節からは、表 1 の基本的ポインティング・パターンを組み合わせることにより、プレゼンテーションのコンテキストにあわせたポインティングのためのコンポーネントを構成可能であることを確認できる (図 5)。リスト 7 に示すように、複数のパターンを組み合わせたポインティングを実現するための複雑な動作の関連づけを簡潔に記述可能である。そして、ポインティングのための HTML 要素の動作に冗長性をもたせることにより、プレゼンテーションをおこなう場面にあわせてポインティングに用いる HTML 要素の動作を選択可能なような動作の関連づけを構成可能であることが分かる。

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。それを用いたソフトウェアの新しいメカ

*9 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。

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。既存研究には、JavaScript プログラムを簡略化するための手法の開発*⁹，および、独自の DSL(Domain Specific Languages) を用いた研究*¹⁰ などがある。

Associated Components の特徴は、HTML プログラムを記述することにより、複数の HTML 要素の連携動作を用いた関連づけられた情報の提示が可能という点である。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。プレゼンテーションでのポインティング・パターンの試行錯誤を繰り返す場合には、Associated Components が提供する動作の関連づけのための抽象的な記述方法が有効である。

5. おわりに

web ページを用いたプレゼンテーションを目的として、既提案の Associated Components によるポインティング手法を提案した。グラフィカルなポインティングについて基本的パターンの分類をおこない、各パターンでの実装により動作を確認した。また、基本的パターンを組み合わせることにより、プレゼンテーションのコンテキストを反映したポインティングのパターンを示した。視覚的に明瞭なポインティングが可能であることから、説明内容の理解を容易にするための有効な手法であるといえる。

今後の課題には、多様なポインティングのパターンへの対応、コンポーネントの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] Barricelli, B. R., Cassano, F., Fogli, D. and Piccinno, A.: End-user development, end-user programming and end-user software engineering: A systematic mapping study, *Journal of Systems and Software*, Vol. 149, pp. 101 – 137 (online), DOI: <https://doi.org/10.1016/j.jss.2018.11.041> (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 October 2019).
- [5] Google: Polymer Project, <https://www.polymer-project.org/> (Retrieved: 8 October 2019).
- [6] 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).

- [7] 佐藤 信: データ駆動型 Associated Components を用いた Ambient Web Design, 情報処理学会研究報告, Vol. 2020-CG-180, No. 8, 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] 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).
- [10] Mozilla Developer Network: Web Components, https://developer.mozilla.org/en-US/docs/Web/Web_Components (Retrieved: 4 April 2019) (2017).
- [11] 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).
- [12] 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).
- [13] Saffer, D.: *Microinteractions: Full Color Edition Designing with Details*, O'Reilly Media, Inc., 1st edition (2013).
- [14] 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).
- [15] WHATWG: HTML Living Standard — Last Updated 18 July 2019, <https://html.spec.whatwg.org/> (Retrieved: 25 July 2019).

*¹⁰カスタム要素に関連する研究である [9] 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。