

透明度可変 Web コンテンツビューアを用いたマッシュアップ環境の実現

Realizing a Mashup Environment based on a Web Contents Viewer with Variable Opacity and Event-Transparency

丹羽 佑輔^{*1} 白松 俊^{*1} 大園 忠親^{*1} 新谷 虎松^{*1}
Yusuke NIWA Shun SHIRAMATSU Tadachika OZONO Toramatsu SHINTANI

^{*1} 名古屋工業大学大学院情報工学専攻

Department of Computer Science and Engineering, Graduate School of Engineering, Nagoya Institute of Technology

We implemented a Web Contents Viewer that provides users a mashup environment. The viewer has one or more layers, which display web pages or web applications. Users can change order of layers on a desktop screen and can set a layer order to specified positions, top-most, above-desktop, and under-desktop positions. Users can also change the transparency of a background of layer. Moreover, users can blend contents and events among multiple distributed layers. The system provides Web APIs to control such properties.

1. はじめに

Webコンテンツの背景を透過しデスクトップ画面上に表示しマッシュアップを可能とするビューアを試作した。本ビューアを応用することで、独立したWebアプリケーションやデスクトップ画面上のアプリケーションの描画内容を重ねて表示可能となる。また、本ビューアが提供するAPIをWebアプリケーションから呼び出すことで、他アプリケーションのデスクトップ画面上での表示位置や重なり順序を制御することが可能となる。

本ビューアを応用することで、既存のアプリケーションにはない機能を追加することが可能である。具体的には、既存のプレゼンテーションツールを用いてプレゼンテーションを行う場合、遠隔同期可能なポイントを実現するWebアプリケーションを本ビューアによって表示することで、リモートポイント機能として拡張することが可能となる。本研究では、既存アプリケーションには存在しない機能を、Webアプリケーションによって追加することをマッシュアップと呼ぶ。Webアプリケーションは、HTML5によって記述可能であり、<canvas>要素や<video>要素などを利用し、ビデオ動画の再生やWebGLによる3D CGの描画を行える。また、既存のJavaScriptに関するライブラリ群を利用することで、Webアプリケーションの簡単に作成できることが期待できる。

2. レイヤーに基づくマッシュアップ

本研究では、三つのマッシュアップについて考える。一つ目は、視覚的情報の重ね合わせによるビジュアルマッシュアップである。二つ目は、マルチモーダル環境での複数の入力デバイスにより入力されたユーザの操作イベントに関するイベントマッシュアップである。三つ目は、デスクトップ上で表示中の独立した複数アプリケーションの機能をマッシュアップさせる、アプリケーションマッシュアップである。

2.1 ビジュアルマッシュアップ

紙媒体上に赤色で書かれた文字を、プラスチック製の赤色の半透明シートを利用して隠すことで記憶学習を行うための教材などがある。また、OHPを用いたプレゼンテーションで、黒色の

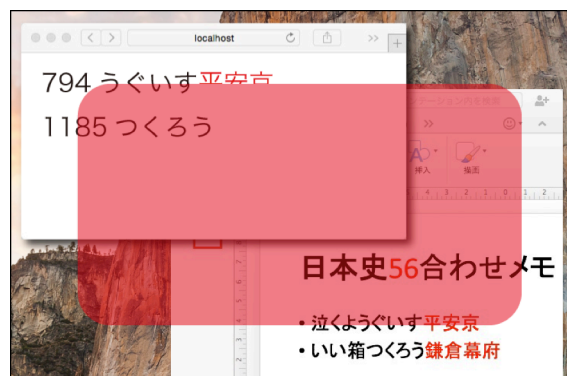


図1 透明度可変 Web コンテンツビューアを用いた赤シートの利用例

シートを用いて、部分的にスライドの内容を隠す表現がある。これらは、重ね合わせにより、コンテンツの視覚的情報を合成あるいは遮断する例である。図1は、PCのデスクトップスクリーン上で、Webブラウザで開かれたWebページに赤色の半透明レイヤーを重ねて閲覧している状態の例である。Webページ上に記述された赤い文字は、赤色の半透明レイヤーを重ねた部分のみが消えており、背後のアプリケーション上の赤色の時はそのまま表示されている状況である。Web上には、文字による情報資源が豊富であり、これらの情報を教材として利用し、学習する際、赤色の半透明シートと同様に半透明レイヤーを重ねた部分だけ消えると便利である。実際に紙媒体上に印刷する必要もなく、端末やディスプレイ上で記憶学習を行うことができる。タブレット端末上で実現できれば持ち運びにも対応する。

2.2 イベントマッシュアップ

マルチモーダルな入力インタフェースを柔軟に組み合わせて新たなアプリケーションを簡便に作成するための支援環境が必要である。例えば、図2の例では、モーションセンサーによって得られたユーザの手の位置情報によって作成した手のモデルを、プレゼンテーションソフトの上に重ねて表示している。ここでは、複数の入力デバイスを用いた環境で、入力デバイスに応じて利用するデバイスの情報を扱えると好ましい。現実世界で紙



図2 透明度可変 Web コンテンツビューアを用いたモーションセンサーの利用例

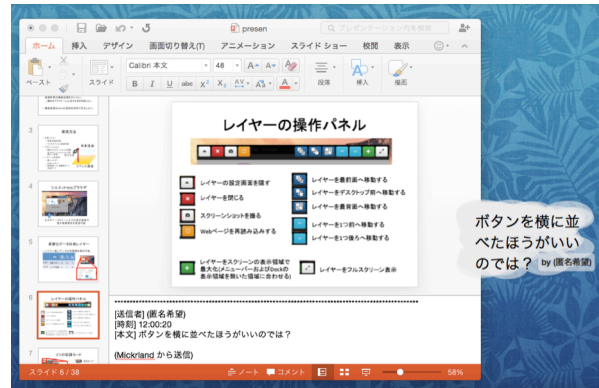


図3 Web アプリケーションとプレゼンテーションソフトの連携例

を重ねた場合、背後の紙を直接触ることはできない。PC のデスクトップスクリーン上においても、アプリケーションウィンドウを重ねた場合、ユーザの操作イベントは、手前にあるアプリケーションに伝播させ、背後のアプリケーションを操作させないようにすることが一般的である。しかしながら、多数のコンテンツおよびアプリケーションを重ね表示することで新たなシステムを構築することを考えると、そのような画一的なイベント処理では不十分である。

ここで重要な点は、コンテンツおよびアプリケーションの部分領域毎に必要な入力イベントが異なる点である。また、複数のコンテンツ等が重畳表示されている場合に、前面にあるコンテンツがイベントを透過して、背面にあるコンテンツにイベントを渡したい場合等が考えられる。すなわち、コンテンツおよびアプリケーションは、部分領域毎に特定のイベントが透過させる条件を詳細に設定することが可能であることが重要である。本研究では、イベントの透過性を簡便な記述によって制御可能にすることで、上記目的の実現を目指している。入力インタフェースとして、マウスおよびキーボードに加え、Web カメラ、音声、およびモーションセンサー (Kinect, Leap Motion 等) を扱う。

ここでは、単にイベントを透過させるのみならず、イベントに処理を加えるための汎用的な層を積層することで、多様なイベント処理を実現することを目指す。例えば、簡単な例としては座標系を変換するための層 (3次元から2次元への変換など) を複数層間に挟むことで、積層されたウィンドウ間における柔軟なイベント処理が容易に可能になる。

2.3 アプリケーションマッシュアップ

ファイルやクリップボードを用いたアプリケーション間連携は、既存アプリケーションの機能を拡張することが困難である。Apple 社の AppleScript など OS に特化したアプリケーション間通信機能は古くから実現されており、プログラミングなしでアプリケーション間を可能としている例も挙げられる。ここでは、事前に定められたプロトコルに従い、プログラムされたアプリケーション間での柔軟な連携が可能である。しかしながら、近年では Web アプリケーションが広範囲に利用されており、デスクトップアプリケーションおよび Web アプリケーション間での連携に関しても考慮する必要がある。例えば、InterTwine [Fourney 2014] では、既存アプリケーションを改変することで、アプリケーション間連携を実現しているが、非専門家には容易ではない。本研究では、非専門家がアプリケーション間連携を例えば、図3のように、図中の右のテキスト情報は Web アプリケーションにより表示しており、このテキスト情報をプレゼンテーションソフトのノート機能に

共有することが可能となる。複数のアプリケーションを協調的に利用することで、それぞれのアプリケーションを単純に組み合わせるのみでは得ることができない機能を実現することが可能である。

3. 透明度可変 Web コンテンツビューア

3.1 Web コンテンツビューア

本ビューアは、WebKit ベースの Web コンテンツビューアである。HTML5 によって記述されたコンテンツを表示することが可能であるため、JavaScript による動的なコンテンツを表示することも可能である。適切な部分を CSS によって透明度を設定することで自由な形状の UI を作成でき、また透過部分に関してはマウスイベントも透過される。適切な部分を CSS によって透明度を設定することで自由な形状 UI を作成でき、具体的には、Web ページ中の <body>要素に“background-color: transparent;”の宣言を設定し、<body>要素内に、“background-color:blue; width:50px; height:50px; border-radius:25px;”の宣言を持つ<div>要素を作成することで、半径 25px の青色の円領域を作成することができる。本ビューアにより、この Web ページを表示すると、半径が 25px 青色の円が表示され、円の領域外のクリックイベントは透過され背後のアプリケーションで処理される。ウィンドウの表示領域がアルファチャンネルをサポートすることで、ウィンドウ全体の不透明度を設定できるようになった。ウィンドウリージョンによるウィンドウの表示領域の各ピクセルのアルファ値の設定も可能になり、矩形領域に縛られないウィンドウが実現できるようになった。これらの機能を用いる場合は、OS のウィンドウに関する API の知識が必要であった。

3.2 透明度制御

本ビューアは、PC のデスクトップ上に表示されたアプリケーションの描画領域上に Web コンテンツを重ねて表示することが可能である。重ねた Web コンテンツの透明度は可変であり、ユーザにより設定可能であるため、ユーザが適切に設定することで、Web コンテンツの背景にある描画内容をユーザが見やすいと思う透明度に調整可能である。透明度は 0%から 100%の値で設定する。透明度が 0%の時は完全に透明であり、100%の時は完全に不透明である。図4は、3つのレイヤーを表示している例であり、最前面のレイヤーおよび中間のレイヤーは透明度制御により、不透明でない状態であり、最背面のレイヤーは不透明の状態である。このように、透明度を設定することで、複数のレイヤーの内容を合成することが可能となる。

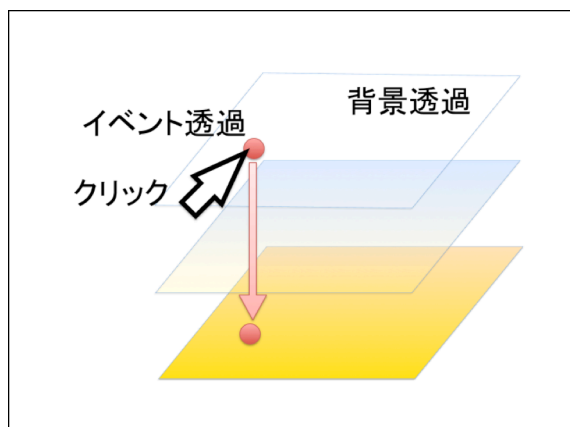


図4 背景透過とマウスイベント透過の例

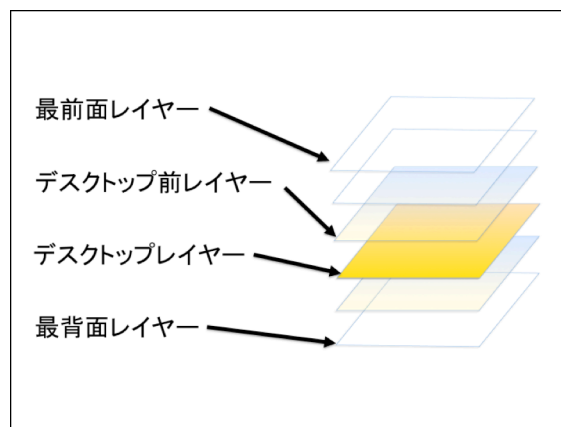


図5 レイヤー順序の構造

3.3 マウスイベント透過制御

PC のデスクトップ上において、アプリケーションウィンドウを重ねた場合、ユーザの操作イベントは、手前にあるウィンドウに送られるのが一般的であり、背後のアプリケーションを直接操作することはできない。本ビューアは、手前に重ねて描画した Web コンテンツへの操作イベントに関しても、背後に透過することが可能であり、これにより、既存のアプリケーション上に重ねて描画し、操作イベントを透過することで、描画内容のみを拡張するという利用方法が可能になる。透明度制御で、透明度を 0% に設定した場合は、常にマウスイベントが透過される。図 4 では、最前面のレイヤーおよび中間のレイヤーはマウスイベント透過制御により、マウスイベントを透過する状態であり、最背面のレイヤーはマウスイベントを透過しない状態である。このように、マウスイベント透過制御の状態を設定することで、最前面レイヤーをクリックした場合でも、最背面のレイヤーにマウスイベントが透過され、最背面レイヤーを操作することが可能となる。

3.4 レイヤー順序制御

本ビューアは 1 台の PC 上で複数起動して表示することができる。レイヤー順序制御機能により、表示中のビューアをレイヤー構造として管理し、デスクトップ画面上での表示順序を制御することができる。常にデスクトップの最前面もしくは最背面に表示することも可能である。図 5 は、レイヤー順序の構造を示す。最前面レイヤーは、デスクトップ画面上で最も手前のレイヤーである。デスクトップ前レイヤーは、デスクトップスクリーンのレイヤーよりも 1 つ手前のレイヤーである。デスクトップレイヤーは、デスクトップスクリーンのレイヤーであり、OS ファイルシステムで管理されているファイル等のアイコンが表示される。最背面レイヤーは、デスクトップレイヤーよりも奥で、本ビューアで管理するレイヤーの最背面のレイヤーである。最前面レイヤーとデスクトップレイヤー間、デスクトップレイヤーと最背面レイヤー間には、複数のレイヤーが存在できる。本ビューアのレイヤー制御機能は、5 つのレイヤー操作を提供する。最前面へ移動、最背面へ移動、デスクトップの手前へ移動、1 つ前へ移動、1 つ後ろへ移動である。最前面へ移動は、複数のレイヤーの最前面にレイヤーを移動する。最背面へ移動は、複数レイヤーの最背面レイヤーを移動する。1 つ前へ移動、1 つ後ろへ移動はそれぞれ、1 つ前のレイヤーの前、1 つ後ろのレイヤーの後ろに移動する。

3.5 スクリーンショット撮影・共有

本ビューアは、ビューアの表示領域のスクリーンショットを撮影する機能を持つ。また、撮影したスクリーンショットをリモートの PC と共有する機能を持つ。これにより、例えば、プレゼンテーションソフトを用いてプレゼンテーションを行う場合、リモートの PC とスライド資料の内容をスクリーンショットによって共有することが可能となる。スクリーンショットの共有に関して、3 つの共有モードを持ち、同一のトークンが設定されたビューア間でスクリーンショットの共有が行われる。3 つの共有モードは、自由モード、聴講者モード、発表者モードである。自由モードは、他の PC 上で表示しているビューアとスクリーンショットを共有しないモードである。聴講者モードは、発表者モードのビューア表示領域のスクリーンショットを表示するモードである。発表者モードのビューアでスクリーンショット撮影ボタンを押した場合、自ビューアの表示領域のスクリーンショットを撮影し、他の PC 上で起動している同一トークンを持つ聴講者モードのビューアに対して撮影したスクリーンショットを送信する。聴講者モードのビューアでスクリーンショット撮影ボタンを押した場合、他の PC 上で起動している同一トークンを持つ発表者モードのビューアに対して、スクリーンショットの撮影要求を送信する。スクリーンショット撮影要求を受け取った、発表者モードのビューアは、スクリーンショットを撮影し、要求を送信したビューアに対してスクリーンショットを返信する。聴講者モードのビューアは、返信されたスクリーンショットを表示する。

3.6 Web アプリケーションによるレイヤー制御

本ビューアで表示中のレイヤーの透明度制御、マウスイベント透過制御、レイヤー順序制御、スクリーンショット撮影・共有は、Web アプリケーションから制御可能である。制御について二つの方法を提供する。一つ目は、HTML に記述した `<meta>` タグを用いる方法である。二つ目は、JavaScript を用いる方法である。

本ビューアは、`<head>` タグ内に記述された、`<meta>` タグを抽出し、`name` 属性値に応じてレイヤーの設定を行う。表 1 は、設定の一例である。レイヤー位置の設定では `name` 属性に“`swb-position`”を指定し、`left`、`top`、`width`、`height` 属性はそれぞれ、スクリーン上での X 座標位置、Y 座標位置、レイヤーの横幅、レイヤーの高さを指定する。レイヤーの順序の設定では、`name` 属性に“`swb-level`”を指定し、`level` 属性に、レイヤー順序の番号を指定する。数字が小さいほど前面に表示される。レイヤーの透明度の設定では、`name` 属性に“`swb-background-alpha`”を設定し、

表1 HTMLの<meta>タグによる制御

設定項目	記述方法
位置	<code><meta name="swb-position" left="" top="" width="" height="" /></code>
順序	<code><meta name="swb-level" level="" /></code>
透明度	<code><meta name="swb-background-alpha" alpha="" /></code>
フルスクリーン	<code><meta name="swb-fullscreen" fullscreen="" /></code>

alpha 属性に透明度を 0.0 から 1.0 の範囲の実数を指定する。0.0 の場合は、完全に不透明で、1.0 の場合は完全に透明である。レイヤーをフルスクリーンの状態で表示したい場合は、*name* 属性に“swb-fullscreen”を指定し、*fullscreen* 属性に“true”を設定する。JavaScript を用いる方法では、本ビューアと通信を行うために開発した、JavaScript 用のライブラリを呼び出して利用する。表2に本ビューアが提供する示す API の一例である。JavaScript によって制御が可能であるため、Web アプリケーション上でのユーザの操作情報に応じて、レイヤーを制御することが可能となる。レイヤーの表示領域内のスクリーンショットの撮影は、*swb.screenshot* 関数を用いる。第一引数の *param* にスクリーンショットのパラメータを Object 型で指定し、第二引数 *callback* にスクリーンショットの撮影が完了した後に呼ばれる関数を指定する。*param* は、*left*, *top*, *width*, *height* のキーによりスクリーンショットの撮影範囲を指定することが可能であり、省略した場合は、レイヤーの表示領域の範囲のスクリーンショットが撮影される。*type* キーは、スクリーンショットの画像フォーマットを MIME タイプで指定する。対応するフォーマットは“image/png”, “image/jpeg”, “image/gif”, “image/tiff”, “image/bmp”である。利用する環境に応じて、画像フォーマットを指定することが可能である。また“image/jpeg”の場合は、圧縮率を *compressionFactor* キーによって指定可能であり、0.0 から 1.0 の範囲の実数で指定する。*callback* は `function (result) {}` の形式であり、*result* は、撮影したスクリーンショットのデータと関連情報が Object 型のオブジェクトとして設定される。*result* は、*left*, *top*, *width*, *height*, *type*, *data* のキーを持ち、それぞれ、撮影範囲の矩形領域を示す X 座標、Y 座標、横幅、高さ、データの形式、スクリーンショットのデータが設定される。*data* に設定される値は、Data URI スキームに従う文字列であり、スクリーンショットのデータを *type* で指定したファイル形式でエンコードしたバイナリデータの Base64 表現を含むものである。Web アプリケーション側では、`<image>` 要素の *src* 要素に *result.data* を設定するだけで、スクリーンショットを表示することが可能である。Web アプリケーション側からデスクトップ上にファイルを保存する場合は、*swb.saveFile* 関数を利用する。第一引数の *param* に保存に関するパラメータを Object 型で指定する。*param* は *name* と *blob* をキーにより、ファイル名と、ファイルのデータを指定する。*blob* は Blob 型で指定する。

4. 応用

本ビューアを応用することで、ビジュアルマッシュアップによる論文添削支援環境として応用することが可能である[杉山 2015]。また、Bluetooth ビーコンの近接情報を用いて、ユーザの近くに存在するディスプレイへコンテンツを表示することができる [丹羽

表2 JavaScript による制御

設定項目	記述方法
位置	<code>swb.position(param);</code>
順序	<code>swb.level(param);</code>
透明度	<code>swb.backgroundAlpha(param);</code>
フルスクリーン	<code>swb.fullscreen(param);</code>
スクリーンショット撮影	<code>swb.screenshot(param, callback);</code>
ファイル保存	<code>swb.saveFile(param);</code>

2014]。本ビューアを用いることで、デスクトップスクリーン上の表示順序を制御しつつ Web コンテンツを重ねて表示することが可能であるため、例えば遠隔 PC への情報通知のための Web アプリケーションの利用者が緊急を要する連絡を行いたい場合は、遠隔 PC のデスクトップスクリーン上の最前面へ連絡内容を表示することが可能である。また、マウスイベントの透過制御も設定できるため、連絡を受け取った側の操作を妨害しないために、最前面に連絡内容を表示しつつ、マウスイベント透過制御により、背後のアプリケーションを操作可能な状態にすることも可能である。本ビューアのスクリーンショット撮影・共有機能を用いることで、遠隔 PC との視覚的なマッシュアップも可能となる。文字列によるアノテーションや手書きメモを付加できる機能を持つ Web アプリケーションの背景を透過し、これを既存の文書ビューアで閲覧している論文に重ねることで、論文上にメモを書くことが可能であり、このビジュアルマッシュアップによって得られた様子をスクリーンショット撮影・共有機能により、遠隔 PC 上へ表示することも可能となる。

5. おわりに

本研究では、デスクトップスクリーン上に、Web コンテンツを重ねて表示するツールを試作した。本ツールを用いることで、Web コンテンツの透明度を変更し、Web コンテンツ表示領域の背後にある表示内容と重ね合わせることができる。これにより、視覚的なマッシュアップを実現可能とした。また、Web アプリケーションへのユーザのマウス操作イベントの透過制御を行うことが可能であるため、ユーザの操作イベントを Web アプリケーションの状態に応じて、切り替えることが可能となる。また、本ビューアは、これらの制御を Web アプリケーションから可能とする API を提供しているため、Web アプリケーションの内部から操作を行うことが可能となる。

参考文献

- [杉山 2015] 杉山亮弘, 丹羽佑輔, 白松俊, 大園忠親, 新谷虎松: シルエット Web ブラウザにおける多重透明レイヤー管理機構とその応用, Vol.2015-ICS-179 No.7, SIG-ICS, 2015.
- [丹羽 2014] 丹羽佑輔, 白松俊, 大園忠親, 新谷虎松: リアルタイムな情報通知のための Bluetooth ビーコンを利用した追尾型人物映像移動機構の実現, 電子情報通信学会 人工知能と知識処理研究会, vol.114, no.339, pp.49-54, 2014.
- [Fourney 2014] Fourney, Adam, et al. “InterTwine: creating interapplication information scent to support coordinated use of software.”, Proceedings of the 27th annual ACM symposium on User interface software and technology, ACM, 2014.