Vol.2019-GN-106 No.8
Vol.2019-CDS-24 No.8
Vol.2019-DCC-21 No.8
2019/1/24

動画のながら視聴に特化したウェブブラウザの開発に関する研究

井上弘仲

概要：近年、ウェブブラウザで動画を視聴することは一般的となった。その際、テレビの垂れ流し視聴と同様に、Web サイト上の動画を見ながら他のソフトウェアも同時に起動・作業を行う“ながら視聴”は需要として考えられる。本研究では、ユーザが任意のソフトウェアと容易にながら視聴を実現できるソフトウェアを開発する。本システムには大きく分け、既存のブラウザと同様の機能を持つウェブブラウザ機能と、本システム独自の“ながら視聴”機能に分かれる。“ながら視聴”のシステムは、大手動画視聴サイトを閲覧し、その中から動画を取得し、任意のサイズ・レイアウトで閲覧するなどがある。そして、本研究の”ながら視聴”の需要の調査、開発したシステムの性能評価を行った。その結果既存のブラウザに比べ性能が低い、本システム独自の“ながら視聴”機能には改善点がある。今後は更なる性能の向上行いフリーソフトとして配布を予定している。

キーワード：ソフトウェア構築、ウェブブラウザ，動画，ながら視聴 [**]

1. はじめに

1.1 背景

近年、YouTube やニコニコ動画といった動画視聴サイトの出現と普及により、ウェブブラウザで動画を視聴することは一般的となった。映画館で見る映画などとは違い、プライベートな空間内での動画視聴はそれだけに集中して視聴する必要もなく、テレビにおける垂れ流し視聴などと同様に、他のことをやりながら視聴するスタイルが考えられる。PC 上で視聴する際、動画の流れる場所だけ表示すれば、それ以外の場所で他の Web サイトの閲覧や word による書類作成など、他の作業を行うことができるスペースができる。

以上のことから、Web サイト上の動画を見ながら他のソフトウェアも同時に起動・作業を行う“ながら視聴”（図 1）は需要として考えられる。

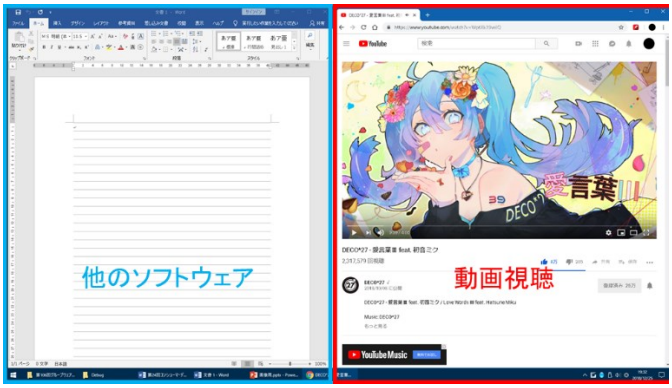


図 1　既存ウェブブラウザでながら視聴（他のソフトウェアと動画視聴を同時に行う）を行う例

1.2 目的

図 1 を例とすると、現存のウェブブラウザでながら視聴を行う場合、以下のような問題点がある。

- 動画視聴ウィンドウと他の作業ウィンドウを分ける作業が毎回必要。
- 動画領域に合わせた動画視聴用ウィンドウの位置とサイズの調整が毎回必要

本研究ではこれらの問題を解決し、ユーザが任意のソフトウェアと容易にながら視聴を実現できるソフトウェアを開発する。

2. 研究手法

2.1 事前調査

ながら視聴が可能なソフトウェアを開発することにあたって、まず動画抽出する機能を持つ現存のシステムについての調査を行った。

(1) 既存システム 1

Floating YouTub

- **概要**

Floating YouTub はブラウザのアドオンである。このアドオンはウィンドウのツールバーにボタンが追加される。YouTube の動画再生の画面で追加されたボタンを押すことで別のウィンドウが開き、そこに動画が表示される。そのウィンドウのサイズ、配置を自由にできる。

- **問題点**

１つの動画しか視聴、取得することはできない。また YouTube の動画しか取得できない。そして、YouTube の関連動画をクリックしたときに反応しない。

(2) 既存システム 2

Window Expander For YouTube

- **概要**

Window Expander For YouTube はブラウザのアドオンである。このアドオンはウィンドウのツールバーにボタンが追加される。YouTube の動画再生の画面で追加されたボタンを押すことで、動画部分だけをブラウザウィンドウサイズに合わせ最大化してくれる。

- **問題点**

１つの動画しか視聴、取得することはできない。YouTube の動画しか取得できない。関連動画を押すと新しいタブで動画が開くため、もう一度ボタンを押さないといけない。

(3) 既存システム 3

ゲーム実況動画における動画多画面視聴支援システム提案 *[1]

- **概要**

この研究では、1 つの Web ブラウザ上で、動画投稿サイト（YouTube、ニコニコ動画など）に投稿された動画、配

Vol.2019-GN-106 No.8
Vol.2019-CDS-24 No.8
Vol.2019-DCC-21 No.8
2019/1/24

信動画（Ustream tv、ニコニコ生放送など）を複数同時に視聴することのできる Web アプリケーションを提案する。

- **問題点**

複数同時に視聴する際、投稿された動画を一度自分でアップロードしないといけない。

(4) 事前調査の考察

以上いくつか類似の事例があるが、アップロード等の事前準備が必要なもの､複数動画のストックができないもの、Youtube にのみ対応しているものなど問題点がある｡また、これらは特定のブラウザ上、あるいは開発ソフトウェア上のみでの動作を前提としており、他の Word などのソフトウェアとの連携に関しては考えられていない。そこで、本研究では、上記問題を解決しまた他のソフトウェアとの連携も容易ながら視聴ソフトウェアを開発する。

2.2 本研究システムの要件

2.1 の調査結果から、本研究において開発するながら視聴システムは事前準備の必要がなく、大手動画視聴サイト（YouTube、ニコニコ動画など）から動画のみを取得し、さらに他のソフトウェアとの連携もできる必要がある。したがって、大手動画視聴サイトなどを通常のブラウザと同程度の機能をもって閲覧できる「ウェブブラウザの基本機能」､またそこから動画情報を取得して他のソフトウェアと同時に閲覧するための｢ながら視聴の機能｣が必要である。

以下は、本研究ソフトウェアの要件を大きく上記 2 つの機能で分けて列挙したものである。

(1) ウェブブラウザの基本機能

- 戻る、進む、更新などの基本機能
- タブ機能

(2) ながら視聴の機能

- 動画の取得機能
- 取得した動画のサイズ制御・削除機能
- ウィンドウレイアウトの制御機能

2.3 ソフトウェア設計

本研究ソフトウェアは、2.2 の要件を満たさなければならない。表 1 および表 2 は、要件から更に分けたソフトウェアの詳細設計一覧である。

表１　ウェブブラウザの基本機能一覧

基本機能
「戻る」「進む」「更新」「ホーム」「URL」「設定」
タブ機能
「新しいタブを追加」「タブ選択」「タブを削除」「アクティブなタブの制御」

表２　ながら視聴の機能一覧

動画取得機能
「ムービービューアーの生成」「動画の取得」
取得した動画の制御・削除機能
「動画サイズの自動調整」「動画の削除」「他のソフトとのながら視聴」
ウィンドウレイアウトの制御機能
「レイアウトの保存」「レイアウトの呼び出し」「保存したレイアウトの削除」

2.4 開発手順

本研究ソフトウェアは､以下１）～４）手順で開発した。

(1) ベース・ブラウザの開発

表１の機能を持つソフトウェア（以下、ベース・ブラウザ）を開発する。このベース・ブラウザに、表２のながら視聴の機能を追加する。

(2) 動画の取得機能の開発

・動画を取得して表示する為の動画視聴用ウィンドウ（以下、ムービービューアー）を生成する。
・ムービービューアーとベース・ブラウザが連携し、容易に動画を取得・表示する機能を開発する。

(3) 取得した動画の制御・削除機能の開発

・ムービービューアー内に（２）で取得した動画サイズを自動調整できる機能を開発する。
・ムービービューアー内に（２）で取得した動画を削除できる機能を開発する。
・ムービービューアー内に（２）で取得した動画のサイズを他のソフトに合わせて変更できる機能を開発する。

(4) ウィンドウレイアウトの制御機能の開発

・ベース・ブラウザ、ムービービューアーのレイアウトを保存できる機能を開発する。
・保存したレイアウトの呼び出し機能を開発する。
・保存したレイアウトを削除できる機能を開発する。

3. ソフトウェアの構築

3.1 開発環境

2.研究手法で示したように、本研究ソフトウェアはウェブブラウザとしての機能だけではなく、他のソフトウェアを含むウィンドウサイズの制御など、OS の基本機能とも密接に連携し､任意の拡張が可能な開発環境が必要である。そのような開発環境として代表的なものは２つあり、以下は情報取得のしやすさに着目してそれぞれの特徴を列挙したものである。

Vol.2019-GN-106 No.8
Vol.2019-CDS-24 No.8
Vol.2019-DCC-21 No.8
2019/1/24

(1) **Web browser コンポーネント（Internet Explorer の素となっている）**

- 機能詳細に関する Microsoft 社の公式サイトがある
- 新しい情報と古い情報が混ざっていることがある
- 個人サイト含め情報量が多い

(2) **Web Kit（元 Google Chrome、Safari の素になっている）**

- (1) のような機能詳細のサイトが見つからない
- 一部の機能に重点を置いて紹介する個人サイトは散見される
- 基本的に英語であり日本語の情報量は少ない

本研究では、機能詳細の公式サイトがあり情報量が多いことから Web browser コンポーネントを使用した。

(3) **開発言語、開発ソフトウェア**

開発言語は「C#」を使用し、開発ソフトウェアとして「Visual Studio 2017(Windows Form Application)」を使用した。

3.2 Windows From Application 内で使用したツール

以下は開発に当たって Windows From Application 内で使用した各ツールのグラフィックである。

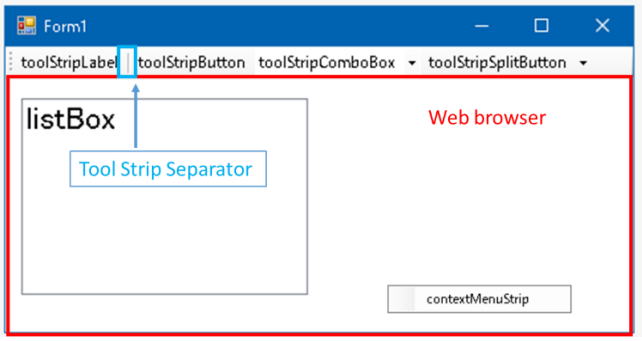


図 2　Windows From Application 内で使用したツールグラフィック

3.3 ベース・ブラウザ機能の構築

表 1 で提示したソフトウェア設計に従い、ベース・ブラウザ機能の構築を行った。機能は大きく「基本機能」「タブ機能」に分けられる。

(1) **基本機能**

以下は「基本機能」のユーザインタフェースを記述する。

- **ユーザインタフェース**

図 3 はベース・ブラウザのユーザインタフェースである。「基本機能」に関してはボタンやテキストボックス等の配置を既存のブラウザと違和感がないよう酷似させている（図 3、図 4 赤線部）。これは、同じような使用感にすることで、ユーザが本ソフトウェアを容易に導入できるようにするためである。基本機能の配置は、ウィンドウ内の最上部、左から順に「戻る」「進む」「更新」「ホーム」「動画の取得」「レイアウトを保存する」「URL」「設定」となっている。（図 3 赤線部）

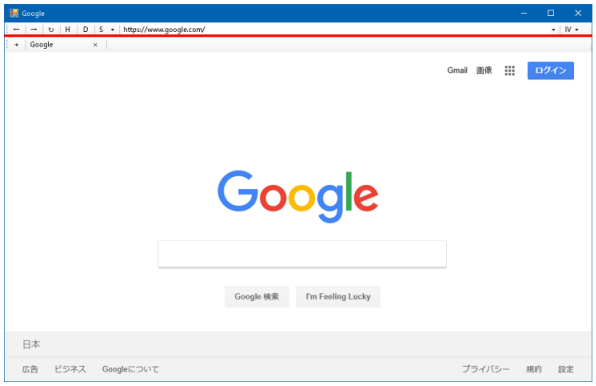


図 3　ベース・ブラウザ

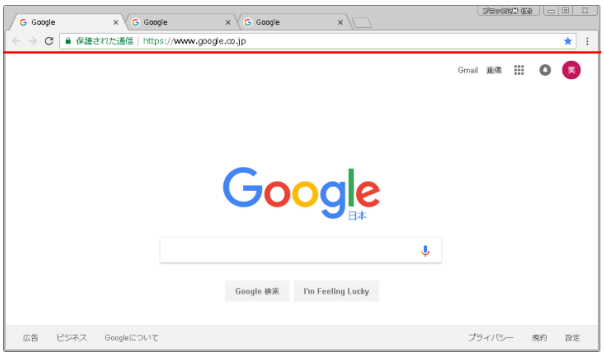


図 4　Google Chrome

(2) **タブ機能**

以下は「タブ機能」のユーザインタフェースを記述する。

- **ユーザインタフェース**

「タブ機能」のユーザインタフェースについても、基本機能の構築と同じ理由で酷似させている（図 5）。タブ機能の配置はウィンドウ内の最上部から 2 段目、左から順に「新しいタブを追加」「タブ選択」「タブを削除」となっている

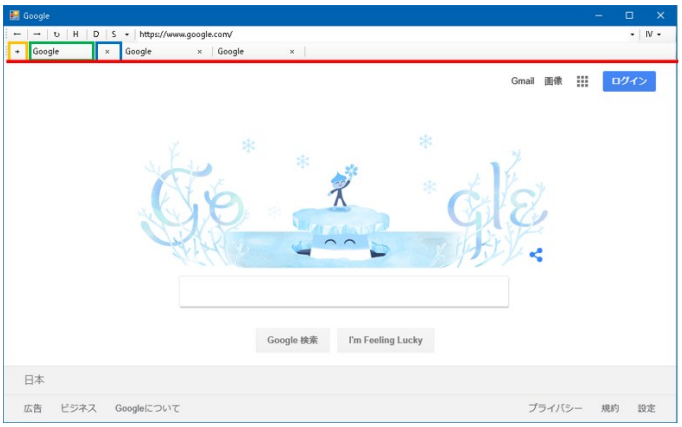


図 5　ベース・ブラウザ

3.4 ながら視聴機能の構築

表 2 で提示したソフトウェア設計に従い、ながら視聴機能の構築を行った。機能は大きく「動画の取得機能」「取得した動画の制御・削除機能」「レイアウトの保存・呼び出し・削除機能」に分けられる。

(1) **動画の取得機能**

以下は動画の取得機能のユーザインタフェースと実装機能の詳細に分けて記述する。

Vol.2019-GN-106 No.8
Vol.2019-CDS-24 No.8
Vol.2019-DCC-21 No.8
2019/1/24

● **ユーザインタフェース**

図６は「動画の取得機能」のユーザインタフェースであり、右のウィンドウが動画を取得して表示する為の動画視聴用ウィンドウ“ムービービューアー”である。

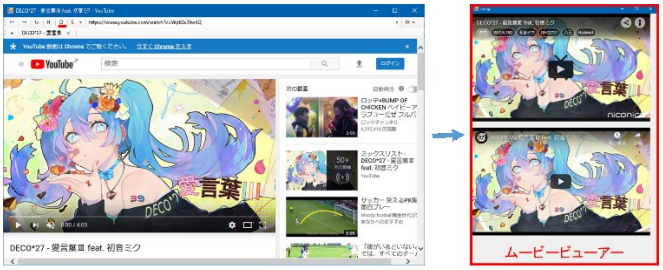


図６　動画の取得

「動画の取得（D)」（図６左赤線）ボタンを押すことムービービューアーが別ウィンドウとして表示される。ボタンを押したときに YouTube・ニコニコ動画の動画視聴サイトであった場合、表示している動画をムービービューアーに取得する。

● **実装機能の詳細**

「ムービービューアーの生成」

「D」ボタンが押されると、ベース・ブラウザとは別の Form をムービービューアーとして表示する。この時「D」を押す毎にウィンドウが増えていかないよう、表示の制御を行った。ムービービューアーが既に表示されているときは Form のタイトルに文字列の値が入っており、表示されてないときには、タイトルには null が入っている。タイトルに値が入っている場合は既に表示されていると判断し、Form を表示する処理を省く。これにより、ムービービューアーの複数ウィンドウ生成を制御している。

「動画の取得」

「D」ボタンには Tool Strip Button を使用した。Click イベントハンドラ（ボタンを押したときに動く処理）が起動すると、一番手前に表示している Web サイトの URL を取得する。取得した URL によって視聴サイトの種類や個々の動画を判別する。YouTube ならサイトの URL に「youtube」という文字列が入っており、動画視聴ページではさらに「v=」という文字列が入っている。ニコニコ動画ならばサイトの URL に「nicovideo.jp」という文字列が入っており、動画視聴ページではさらに「watch」という文字列が入っている。上記の条件を満たしていると動画の取得処理に入る。動画の取得処理は、まず Web browser をムービービューアー上に生成する。

次にこの Web browser に表示する Web ページの URL を生成する。以下は動画視聴サイトの URL と、そこから生成される URL の例である。

<動画視聴サイトの URL>

https://www.youtube.com/watch?v=WptXk39wiIQ

URL 最後の「WptXk39wiIQ」が動画の番号である。

<生成される URL>

https://www.youtube.com/embed/WptXk39wiIQ

「WptXk39wiIQ」を動画視聴サイトの URL から取得し、動画埋め込み機能の URL の最後に追加して生成する。

最後に、Navigate 関数（指定した URL のページを Web browser に表示する）を使用して指定した URL の Web ページに移動する。

(2)　取得した動画の制御・削除機能

この機能は、大きく「取得した動画サイズの自動調整」「取得した動画の削除」「他のソフトとながら視聴」の機能に分けられる。以下、各機能のユーザインタフェースと機能実装の詳細に分けて記述する。

● **ユーザインタフェース**

「動画サイズの自動調整」

取得した動画は図６右のように、ムービービューアーのウィンドウ幅に合わせて自動的に大きさが変更される。

「動画の削除」

右クリックすることでメニューが表示される、その中の「削除」を押すことで取得した動画を削除できる。

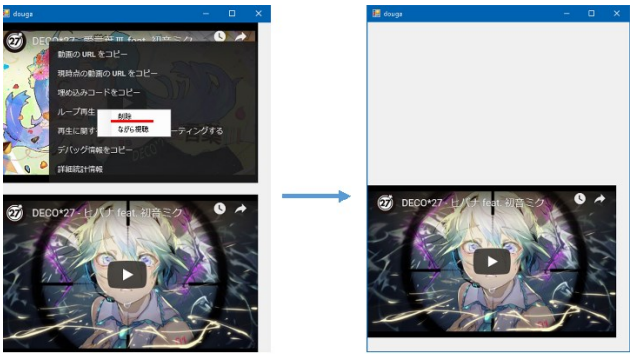
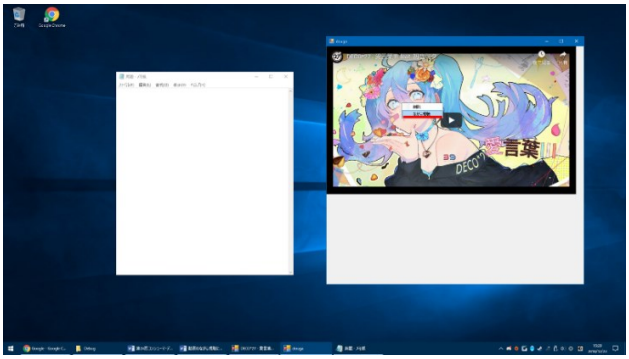


図７　取得した動画の削除

仮にムービービューアー内の動画をすべて削除したいときは、ムービービューアーのウィンドウを閉じるとすべての動画を削除できる。

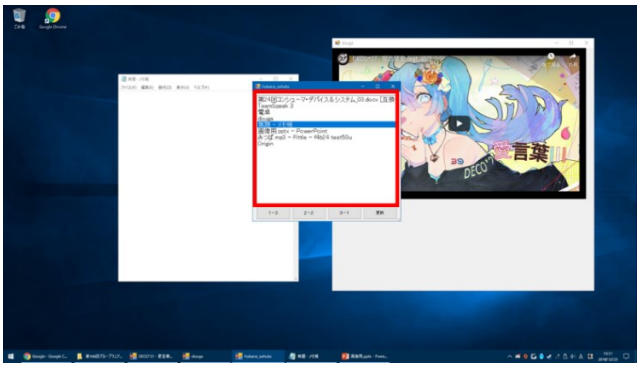
「他のソフトとのながら視聴」

この機能に関してはユーザ操作が煩雑なので、流れ図で説明する。今回他のソフトの例としてメモ帳を使用した。

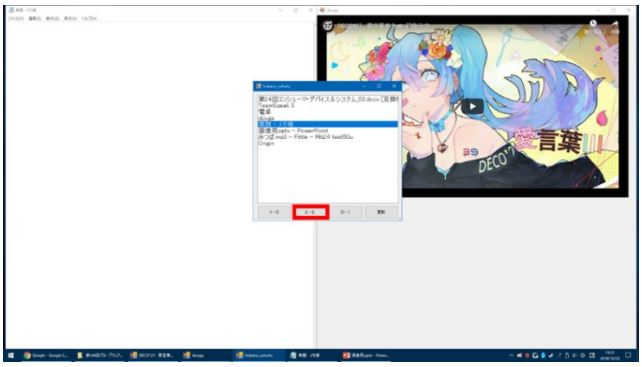


手順１：削除同様右クリックしたときにメニューが表示される、その中の「ながら視聴」をクリック。

Vol.2019-GN-106 No.8
Vol.2019-CDS-24 No.8
Vol.2019-DCC-21 No.8
2019/1/24



手順２：他のソフトの一覧が表示される。



手順３：赤枠の比率のボタンを押すとそのソフトウェアとムービービューアーのサイズが比率に応じて変化する。

- **実装機能の詳細**

「動画サイズの自動調整」

Form の Resize イベントハンドラ（サイズが変更されたときに動く処理）を使用した。ムービービューアー内の Web browser をすべて取得し、ムービービューアーの横幅と同じサイズに変更する。縦幅に関しては、高さ 1 に対して横が 1.618 という比率を使用した。

「動画の削除」

ムービービューアーでマウスを右クリックしたときにムービービューアー用に作成した Context Menu Strip を表示する。「削除」を押されたら、クリックされたことでアクティブになっている Web browser の Name を取得する。ムービービューアー内から同名の Web browser を見つけ、HTML 要素を白紙にした後に Web browser を削除する。

「他のソフトとのながら視聴」

ムービービューアーでマウスを右クリックしたときにムービービューアー用に作成した Context Menu Strip を表示する。「ながら視聴」が押されたら、他のソフトの一覧を表示する Form を表示する。Form が表示されたら、List Box に他のソフトのタイトルを追加していく。以下はパソコン上で起動している全プロセスを取得するプログラムである。

```
foreach ( System.Diagnostics.Process eachP
in System.Diagnostics.Process.GetProcesses())
```

上記 foreach 内では、「eachP」に各プロセスの情報が入る。以下のプログラムにより、List Box に起動しているソフトのタイトル（MainWindowTitle）をすべて追加する。

```
ListBox.Items.Add(eachP.MainWindowTitle);
```

手順３の比率ボタンが押された時は、再度全てのプロセスを取得し、List Box で選択されている文字列と一致したらサイズ変更処理に移行する。以下は 1:1 で他のソフトのサイズ・位置を変更する簡略化プログラムである。User32.dll の MoveWindow 関数を使用しており、各プロセスのハンドル、ウィンドウ左上の位置、横幅、縦幅、ウィンドウ内の再描画、の順で指定する。

```
MoveWindow(eachP.MainWindowHandle, 0, 0, ScreenWidth / 2, ScreenHeight, 1);
```

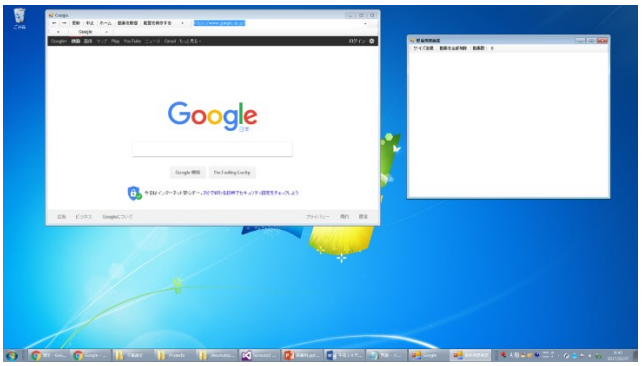
ムービービューアー側は Form の Left,Top,Height,Width プロパティの値を変更することでサイズと位置を変更する。

(3) ウィンドウレイアウトの制御機能

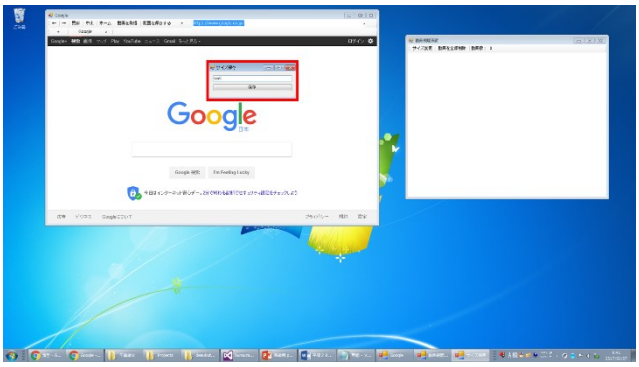
この機能は、大きく「レイアウトの保存」「レイアウトの呼び出し」「保存したレイアウトの削除」の３つに分かれるが、ユーザ操作が煩雑である。したがって一連の流れを図で各インタフェースを外観する。その後、実装機能の詳細に関して記述する。

- **ユーザインタフェース**

＜保存の流れ＞



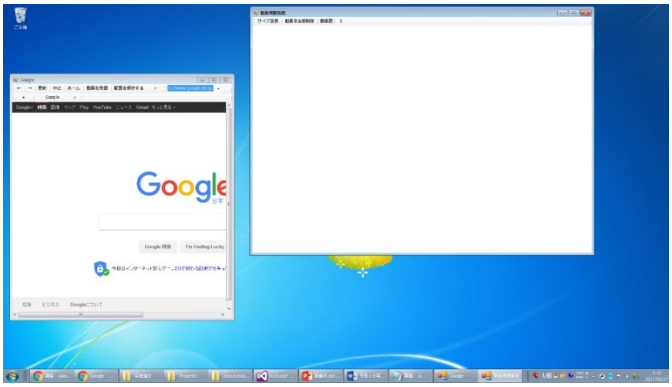
手順１：各ウィンドウを好きなサイズ、配置にする。



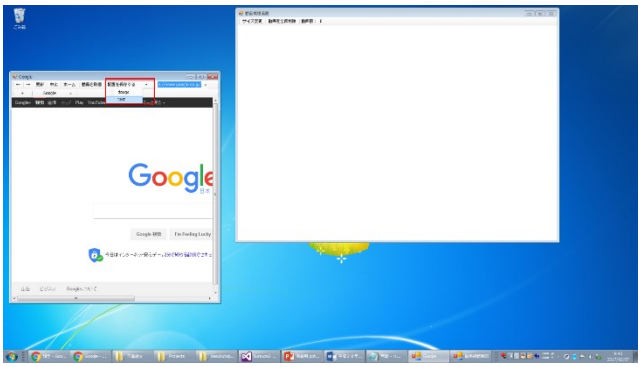
手順２：「配置を保存する」ボタンを押すとダイアログが出て、この配置を「test」と名前をつけ保存する。

Vol.2019-GN-106 No.8
Vol.2019-CDS-24 No.8
Vol.2019-DCC-21 No.8
2019/1/24

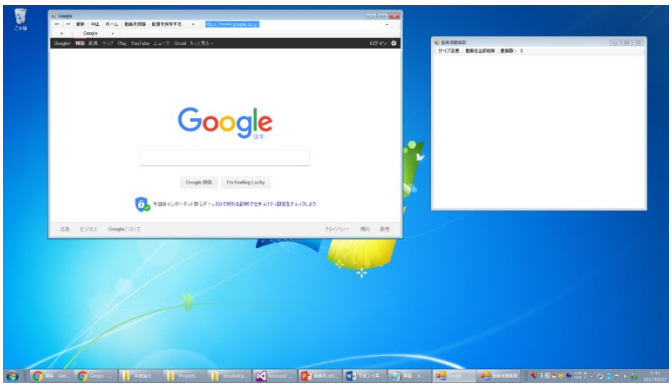
＜呼び出しの流れ＞



手順１：サイズ、配置を変える。

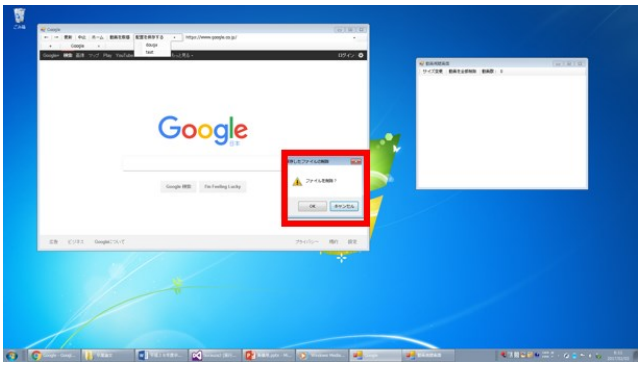


手順２：「配置を保存する」ボタンの隣の▼を押す。
先ほど保存した「test」があるのでクリックする。

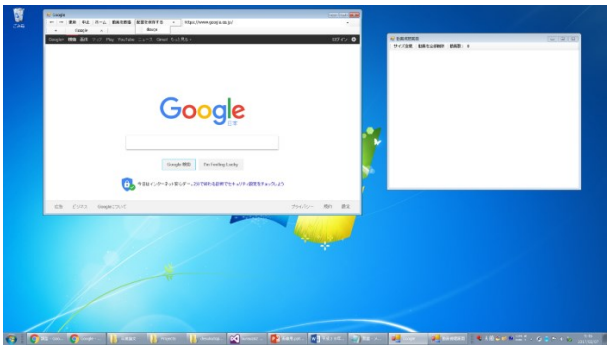


手順３：「test」で保存したサイズ、配置に変更される。

＜削除の流れ＞



手順１：「配置を保存する」ボタンの隣の▼を押して、「test」を右クリックするとダイアルログが出る。



手順２：「OK」を押すことで保存した「test」が削除される。

- **実装機能の詳細**

「レイアウトの保存」

レイアウト保存ボタンには Tool Strip Split Button の左側のボタンを使用した。Click イベントハンドラが起動すると「サイズ保存」ダイアログを表示する。ダイアログ中で保存が押されたら、「テキストボックスに入力された文字列.txt」というテキストファイルを生成する。さらにテキストファイルの内容に、そのとき表示しているベース・ブラウザとムービービューアーの幅と高さ、またディスプレイ上の左上からの相対位置をピクセル単位の数値として書き込み、保存する動作をする。

「レイアウトの呼び出し」

▼を押されたときに表示される一覧には Tool Strip Menu Item を使用した。まずレイアウト保存用のテキストファイルを全て読み込み、そのファイル名を一覧に表示する。ユーザに選択されて各項目の Click イベントハンドラが起動すると、一覧で押されたものと同じ名前のテキストファイルを探して内容を読み取り、幅と高さ、相対位置として各ウィンドウプロパティに適用する。

「保存したレイアウトの削除」

▼を押されたときに表示される一覧には Tool Strip Menu Item を使用している。Click イベントハンドラ中でクリックボタンの判断を行い、右クリックの場合には削除ダイアルログを表示する。OK が押されると、選択した名前と一致するテキストファイルを削除し、同時に Tool Strip Menu Item の一覧のデータからも削除する動作をする。

4. 結果と考察

4.1 需要調査

(1) 目的：本研究システムの対象となるユーザがどの程度存在するのかを見出すことを目的とする。

(2) 概要：日常的に PC で動画を視聴しているユーザ、その中でながら視聴を行っているユーザの割合を調査する。また動画を視聴しないユーザ、その中でもテレビでながら視聴を行っているユーザの割合を調査する。またアドオンに関する認識調査も行った。以下は質問項目の一覧である。

Vol.2019-GN-106 No.8
Vol.2019-CDS-24 No.8
Vol.2019-DCC-21 No.8
2019/1/24

＜基本の質問＞
- 年齢
- パソコンで動画を視聴しますか？

＜動画を視聴しているユーザへの質問＞
- １週間でどのくらい視聴していますか？
- 動画視聴中に他の事を行っていますか？

＜動画を視聴してないと答えたユーザへの質問＞
- テレビは視聴していますか？
- テレビ視聴中に他の事を行っていますか？

＜アドオンの認識調査＞
- ブラウザのアドオンは使用していますか？
- 使うと答えた方は、どの程度使っていますか？

(3) 対象：本大学の情報科６０名

(4) 結果

年齢

１０代	１．７％
２０代	９５％
３０代	３．３％

パソコンで動画を視聴しますか？

はい	８６．９％
いいえ	１３．１％

１週間でどのくらい視聴していますか？

１時間以下	７．８％
１時間	５．９
２時間	１７．６％
３時間	７．８
４時間以上	６０．８％

動画視聴中に他の事を行っていますか？

はい	６７．９％
いいえ	３２．１％

テレビは視聴していますか？

はい	６２．５
いいえ	３７．５

テレビ視聴中に他の事を行っていますか？

はい	６０％
いいえ	４０％

ブラウザのアドオンは使用していますか？

はい	３１．１％
いいえ	８．２％
分からない	６０．７％

使うと答えた方は、どの程度使っていますか？

よく使う	３６．８％
あまり使わない	６３．２％

(5) 考察

アンケートの結果から、動画をパソコンで見ているユーザは 86.9%、その中でも視聴時間が 1 週間で 4 時間以上のユーザが 60.8%存在した。ここから、全体の 52.3%のユーザが日常的に動画を閲覧するヘビーユーザであると考えられる。また、パソコンで動画視聴を行うユーザの中で、動画視聴中に他のことを行うユーザは 67.9%存在した。したがって全体の 59.0%がながら視聴を行っていることになる。以上のことから、動画視聴に特化し、さらにながら視聴機能を持つ本開発システムは､少なくとも 20 代のユーザに需要はあると考えられる。

さらに、動画は見てないがテレビを見ながら他のことを行うユーザは 60.0%存在した。このユーザは今後パソコンで動画を視聴するようになった際にながら視聴を行うと考えられ、潜在的な需要もあることが分かった。

アドオンの調査に関しては、研究を行っていく中で「アドオンで開発しないのか」という意見がしばしば見られたため、意識調査を行った。結果として、使うと答えたユーザは 31.1%であり、その中でも「よく使う」と答えたユーザは 36.8%であった。これは全体の 11.4%である。ここから、アドオンはブラウザの基本機能が既存のものであるため性能が高い、また導入が楽といった利点があるものの、10%ほどのユーザにしか機能を使ってもらえない可能性や、他のソフトウェアとの連携が難しいといった欠点も持つことから、今回独自のソフトウェアとして開発を行ったのは適切な判断であったと考える。

4.2 性能評価

(1) 目的：本研究で開発したシステムが、ユーザにとって使いやすいものなのか評価を行う。

(2) 概要：実際に開発したシステムを使用してもらい、ベース・ブラウザの基本機能、ながら視聴のしやすさに関して既存のブラウザと比較を行った。また、独自に開発した他のソフトウェアとのながら視聴機能、レイアウト保存機能の評価、そして最後に総評を評価してもらった。比較する質問には、1 点～4 点で評価点をつけている。以下は質問項目の一覧である。
- 本システムのベース・ブラウザの基本機能
- 既存のブラウザの基本機能
- 本システムのながら視聴のしやすさ
- 既存のブラウザのながら視聴のしやすさ
- 独自の他ソフトとながら視聴機能の便利さ
- 独自のウィンドウレイアウトの制御機能の便利さ
- 総評

(3) 対象：日常的に動画を視聴する 4 名

今回 4 名に評価してもらった。4.1 の調査を行い、4 名とも「4 時間以上視聴する」と答えたユーザである。実際に使用してもらい評価を行った。

(4) 結果

Vol.2019-GN-106 No.8
Vol.2019-CDS-24 No.8
Vol.2019-DCC-21 No.8
2019/1/24

本システムのベース・ブラウザの基本機能

十分な機能がある	4	２５％
概ね機能が足りている	3	２５％
少し機能が足りない	2	５０％
全く機能が足りない	1	０％

既存のブラウザの基本機能

十分な機能がある	4	２５％
概ね機能が足りている	3	７５％
少し機能が足りない	2	０％
全く機能が足りない	1	０％

本システムのながら視聴のしやすさ

十分に視聴しやすい	4	２５％
概ね視聴しやすい	3	７５％
少し視聴しにくい	2	０％
視聴が難しい	1	０％

既存のブラウザにおける動画の視聴のしやすさ

十分に視聴しやすい	4	５０％
概ね視聴しやすい	3	２５％
少し視聴しにくい	2	２５％
視聴が難しい	1	０％

独自の他ソフトとながら視聴機能の便利さ

ブラウザの機能として是非あって欲しい	０％
ブラウザの機能としてあれば便利である	７５％
あまり便利とは思えない	０％
不便である	２５％

独自のウィンドウレイアウトの制御機能の便利さ

ブラウザの機能として是非あって欲しい	５０％
ブラウザの機能としてあれば便利である	５０％
あまり便利とは思えない	０％
不便である	０％

総評

今後ブラウザとして使いたい	０％
動画を見るときは使いたい	１００％
使いたくない	０％

(5)　考察

アンケートの結果から、ベース・ブラウザの性能に関しては評価点の合計が自作システムは 11、既存のブラウザは 13 となり、既存のものと比べて性能が低いという結果となった。主な要因として「中クリックでリンク先を新しいタブで開く機能がない」「お気に入り機能がない」など、コメントを頂いた。

次にながら視聴のしやすさは、自作システムが 13、既存のブラウザは 13 であり総合点として同点であった。内訳をみると、自作システムの方が視聴しやすいと答えたユーザが 1 名、既存のものの方が視聴しやすいと答えたユーザが 1 名おり、後者の理由として「Youtube の動画配信への対応を考慮して欲しい」とのコメントを頂いた。また独自の機能については、他のソフトとのながら視聴の機能に関しては 75%が、レイアウト保存機能は 100%が「是非あって欲しい」または「あれば便利である」と答えており、ブラウザの機能として好評価であることが分かった。だが、他のソフトとのながら視聴の機能については、「他のソフトのレイアウトを変えずに余っているところを埋めてほしい」「比率を 9 分割して選択できるようにしてほしい」などのコメントを頂いた。総評として、動画を見る際使いたいと評価を受けた。

5. 結論と今後

5.1 結論

本研究では、動画エリアのみを抽出して表示することで、１つのディスプレイ内で動画を視聴しながら他のスペースで作業をする、“ながら視聴” を容易に行うことができるソフトウェアの開発を行った。本研究システムは、通常のブラウザの基本機能を持つベース・ブラウザと、動画視聴用ウィンドウであるムービービューアーを持ち、動画の取得、サイズの自動調整、レイアウトの保存、他ソフトとのながら視聴といったながら視聴に特化した機能を実装した。本研究システムは既存のシステムと比べて事前の動画アップロードなどの事前準備を必要とせず、動画視聴サイトから直接動画を取得することができる。また複数動画のストックや、他ソフトとの連携という独自性も持つ。需要調査と性能評価により、本システムは需要があると結論でき、また性能に関して動画視聴時に使いたいという高評価を得た。

5.2 現状の問題、今後の展開

現状の本研究ソフトウェアの問題点を以下に列挙する。

- まだ基本的なウェブブラウザとしての機能が既製品（Internet Explorer ,Google Chrome など）に比べ劣るので基本機能を充実させる。「お気に入り」「中クリックでリンクを新しいタブで開く」など。
- ながら視聴の動画配置に関して、使いやすさを向上させる。余っているエリアを動画で埋める機能、画面を 9 分割して選択できる機能など。

上記の問題を解決しフリーソフトとして配布を予定している。

参考文献

[1]”ゲーム実況動画における動画多画面視聴支援システム提案”, 著者,中野裕太,服部哲,速水治夫,収録されている論文集,マルチメディア、分散協調とモバイルシンポジウム 2011 論文集, 収録ページ pp.370-373, 発行日 2011-06-30