

HTML によるデジタルサイネージの実現

Digital Signage System in HTML

瀬戸 博幸

Hiroyuki Seto

鹿児島女子短期大学

高精細な大型ディスプレイを使用したデジタルサイネージシステムが随所にみられるようになってきたがその制御に専用ソフトウェアを用いる場合が多い。そこで本稿ではブラウザと HTML による簡易で汎用なサイネージシステムについて実験考察し、実現できることを確認した。

キーワード：HTML，デジタルサイネージ，ブラウザ，JavaScript

1. はじめに

2003年から2011年の地上放送デジタル化により液晶大型ディスプレイが身近な存在となっている。また、LED バックライトで省電力化が進み、駅中などにも大型ディスプレイを使ったデジタルサイネージが使われ始めている。さらに、ノートパソコンに HDMI インターフェースが普及し、HDMI ケーブルで簡単に大型ディスプレイと接続できるようになってきた。しかし、デジタルサイネージシステムを導入するために調査すると、コンテンツ作成や実行がメーカー専用のソフト⁽¹⁾等に依存している場合が多く、現有している他のシステムとの連携を図るなど幅広い応用を考えると自由度が低いといえる。

そこで本稿では、パソコンとそれに附属する一般的なブラウザによるデジタルサイネージの制御⁽²⁾を試み、その実現可能性を考察する。

2. HTML によるデジタルサイネージシステムの概要

HTML はホームページを記述する言語であり、Windows など一般的な OS でメモ帳など一般的なテキストエディタを使って簡単にホームページを作成することができる。

しかし、ホームページは PULL 型、つまり、利用者がブラウザを起動し、ブラウザに URL を与えるか、ホームページ上のリンクをクリックするなど積極的なアクセスを通してホームページ作成者の提供するコンテンツを引き出すシステムである。

一方、デジタルサイネージは PUSH 型、つまり、利用者が望むか否かにかかわらず自動的にディスプレイにコンテンツを表示し、利用者に情報を提供するシステムといえる。

また、ホームページは不特定多数の利用者からの不定期の PULL を待ち受けなければならないが、デジタルサイネージはシステム作成者が表示対象のディスプレイを事前に特定でき、表示シーケンスを計画できるなどの違いがある。

よって、今回のシステムは図1のように計画した。

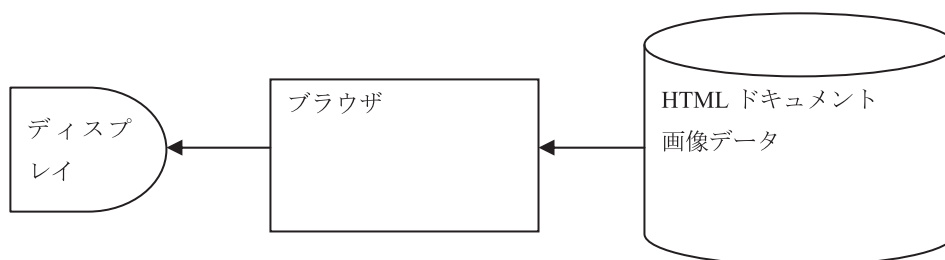


図1. HTML によるデジタルサイネージシステムの概要

3. HTMLによるデジタルサイネージシステムの実現

HTMLによるデジタルサイネージを実現するために、OSとしてWindows Vista、ブラウザにInternet Explorer9（以下IEと記す）を用いた。旧式のシステムでも実現できる可能性を実験するためでもある。

3.1 デジタルサイネージシステムの起動

コンピュータを起動したらデジタルサイネージシステムがスタートできるようにIEをOSのスタートメニューに入れ、IEのホームページを起動用のHTMLドキュメントに設定した。（付録1）

図2に今回の実験システムの状態遷移を示す。

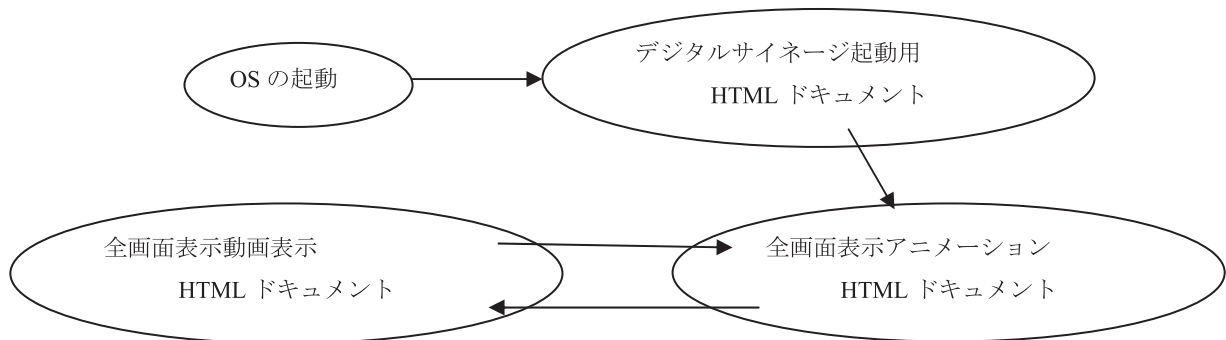


図2. 実験システムの状態遷移

OSによって起動されたIEは起動用のHTMLドキュメントを表示し、JavaScriptで記述されたwindow.open()ファンクションにより、全画面表示状態でアニメーションを実行するHTMLドキュメントを開く。

3.2 全画面表示アニメーションの実験

デジタルサイネージ以外のデスクトップ上のタスクの表示を覆い隠すように全画面表示されたアニメーション実験用HTMLドキュメントは、背景色の制御と画像(e1.gif)が全画面を跳ねる簡単なアニメーションをJavaScriptで実現している。（付録2）

画像は、跳ねる高さがだんだん小さくなり、画像が止まった後、次の動画表示の全画面HTMLドキュメントを開き、アニメーションのHTMLドキュメントを閉じるシーケンスをsetTimeout()ファンクションで実現している。

3.3 全画面表示動画再生の実験

これは全画面表示アニメーション実験用のHTMLドキュメントによって起動される動画自動再生用のHTMLドキュメントである。（付録3）

動画表示は<object>タグにより実現している。動画コンテンツをMediaPlayerで再生しているのだが、付録に示すように、URLを指定するだけでネットワーク上の動画コンテンツを簡単に表示再生することができる。

この実験では、全画面表示動画再生のHTMLドキュメントが表示されてから19秒後に全画面表示アニメーションのHTMLドキュメントを再び起動し、その1秒後に動画再生のHTMLドキュメント自身を閉じるシーケンスがJavaScriptで記述されている。

このようにJavaScriptでシーケンスを制御できることを実験で確認した。

4. 考察

図1に示すように、HTMLドキュメントや画像データはブラウザと独立したファイルである。よって、ブラウザによってデジタルサイネージが稼働中であっても、HTMLドキュメントや画像データを他のシステムから更新することができ、更新による中断がおきないサイネージシステムが容易に実現できると考えられる。また、システムがネットワークに繋がっていれば、ローカルにデータが存在しなくてもURLを指定するだけで容易にネットワーク上のデータを表示することができる。今回の実験では動画データの表示がこの場合に該当する。

さらに、今回の実験ではOSにWindows Vista、ブラウザにInternet Explorer9を用いたが、最新のWindows 8.1と

InternetExplorer11 の環境でも実験に使用したドキュメントが同様に動作することを確認した。

しかし、JavaScript の様々なファンクションについて、実行が可能か否かは現状では統一されておらず、ブラウザや OS の仕様に依存している。図 1 の構成のデジタルサイネージであれば、一つの環境に特化すれば実現できるので問題はないが、多くの利用者が HTML を用いてサイネージシステムを実現することを考えると、仕様の統一が望まれる。

さらに HTML5 によって、よりダイナミックな表示も可能になるのではないかと考える。

5. おわりに

今回は HTML による PUSH 型のデジタルサイネージについて考察したが、利用者の ID などを得ることにより、PHP やデータベースを利用した PULL-PUSH 型の情報提供システムへと繋げて行きたい。

参考文献

- 1) e-Signage デジタルサイネージソフトウェア シャープ株式会社
- 2) 鈴木亮詞, 今井翔太, 白松俊, 大園忠親, 新谷虎松: “スマートサイネージシステムにおけるオフライン Web コンテンツ再生機構の実現とその応用”, 情報処理学会第74回全国大会講演論文集 2012(1), 157-159, 2012-03-06

付録 実験に使用したドキュメント

付録 1. デジタルサイネージ起動用ドキュメント

```
<html>
<head>
<title>サブウィンドウを全画面で開く</title>
<script Language="JavaScript"><!--
window.open("subwin.html","sample","fullscreen=yes");
// --></script>
</head>
<body>
</body>
</html>
```

付録 2. 背景色および簡単なアニメーション用ドキュメント

```
<html>
<head>
<title>subwin.html</title>
<script Language="JavaScript"><!--
w = 420;
h = 380;
ww = screen.width;
hh = screen.height;
step = count = 10;
function resizeWin()
{
    count=0;
    setTimeout("fade('0123456789abcdef')",1000);
}

function fade(str)
{
```

```

    c = str.charAt(count++);
    document.bgColor = "#"+c+c+c+c+c+c+c+c;
    if (count < 16) setTimeout("fade("+ str + ")",500);
    else {

        setTimeout("Set()",1000);
    }
}

var x = 0;
var y = 400;
var v0 = -15;
var a = 0.2;
var s = 1.305;
var vx = 3;
var haneiti = hh;
var kabeiti = ww;
function Set() {
v = v0;
if(vx>0) { vx = 3; } else { vx = -3; }
document.E1.style.left = x;
document.E1.style.top = y;
setTimeout("moveE1()",10);
}
function moveE1() {
    x = x + vx;
    document.E1.style.left = x;
    if (x>kabeiti-72) {
        vx = -vx;
        x = kabeiti-72;
    }
    if (x<0) {
        vx = -vx;
        x = 0;
    }
    y = y + v;
    v = v + a;
    document.E1.style.top = y;
    if (y>haneiti-71) {
        vx = vx*0.9;
        v = -v* 0.8;
        y = haneiti-71;
    }
    if(v>s || v<-s) {
        setTimeout("moveE1()",10);
    }
    else {

```

```

        setTimeout("wmv()",100);
    }
}
else {
    setTimeout("moveE1()",10);
}
}

function wmv() {
window.open("wmv.html","wmv","fullscreen=yes");
setTimeout("window.close()",1000);
}
// --></script>
</head>
<body onLoad="resizeWin()" bgcolor="black">
背景色の変更と簡単なアニメーション

</body>
</html>

```

付録3. 動画表示用ドキュメント

```

<html>
<head>
<script Language="JavaScript"><!--
setTimeout("opnsub()",19000);
setTimeout("window.close()",20000);

function opnsub() {
window.open("subwin.html","sample","fullscreen=yes");
}
// --></script>
</head>
<body>
<div class="section">


動画の表示<br />
<p>
<object classid="clsid:22D6F312-B0F6-11D0-94AB-0080C74C7E95" id="MediaPlayer1" width=470 height=264>
<param name="AutoStart" value="1">
<param name="Filename" value="http://www.jkajyo.ac.jp/information//upload/CM/Suisen2.wmv">
</object>
</p>
</div>
</body>
</html>


```