

# Web マッシュアップに基づくスマートサイネージシステムについて

今井 翔太<sup>†</sup>   鈴木 亮詞<sup>††</sup>   白松 俊<sup>††</sup>   大園 忠親<sup>††</sup>   新谷 虎松<sup>††</sup>  
 名古屋工業大学工学部情報工学科<sup>†</sup> 名古屋工業大学大学院工学研究科情報工学専攻<sup>††</sup>

## 1 はじめに

本研究室では、タブレット端末を利用した新たなサイネージシステムとしてスマートサイネージシステムを開発している。ここでのスマートサイネージシステムの開発において、(1) 多様なネットワーク環境への対応、(2) 柔軟なコンテンツの作成・配信、(3) 簡便な運用、の3点の実現を目指している [1]。

本稿では、(2) 柔軟なコンテンツの作成・配信、に関して Web からの情報を用いたコンテンツを容易に作成、配信することを目的としている。従来のサイネージシステムでは、事前に準備されたコンテンツを用いた情報配信が主である。コンテンツの内容にソーシャルメディアなどの Web 上のコンテンツが利用できることが望ましいが、サイネージシステム上で、それらのコンテンツを自在に利用するには一般的に Web API から自分の目的とする情報の取得、抽出、さらにその結果をもとにコンテンツの構築、といったことをプログラミング言語を用いて行わなければならない、ユーザにとって手間のかかる作業となる。

本研究では、これらの問題の解決のために、Web および PC 上のコンテンツやサービス・アプリケーションをマッシュアップ可能なビジュアルプログラミング環境を実現する。本環境では、コンテンツ取得、コンテンツ処理、コンテンツ配信を直感的に設計可能なプログラミング環境の実現を目指す。

## 2 マッシュアップのためのビジュアルプログラミング環境

プログラミングの非専門家がプログラムを行うことに関しては、幅広い研究が行われている [2]。本研究では、子供やプログラミングの非専門家が利用に対する事例の豊富な、ビジュアルプログラミングに着目した。ビジュアルプログラミングでは、テキスト形式のプログラムの作成ではなく、画面上の図形の操作や配置などにより、グラフィカルに処理を記述することを可能にする。これには、プログラムの構造やデータの流れを比較的容易に把握できるという利点があるため、子供のプログラミング入門向け、ロボットの制御用、および、アナリストのためのデータ処理向けなど、多様なビジュアルプログラミング環境が提案されている。

本システムではマッシュアップしたコンテンツの作成や配信を行うために、図 1 のようなエディタで処理を作成する。本システムでは必要な処理がパーツというモジュールとして左のリストに用意されている。ユーザはこれらのパーツを中央のキャンバスに配置し、このパーツ同士を線（以降、リンクと呼ぶ）で接続することで処理を作成する。この処理の流れをワークフローと定義する。作成したワークフローは図 1 上部の開始ボタンを押すと実行される。システムはキャンバス上のワークフローを読み取り、適切な順番で処理を実行してコンテンツを作成する。

パーツには、ワークフロー中で扱われているデータを表すオブジェクトパーツ、Web API からの情報の取得やファイル、

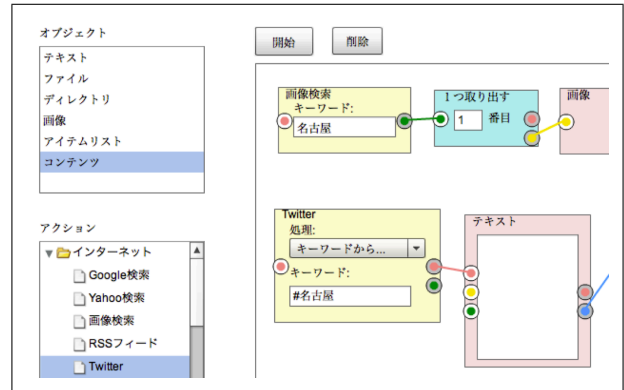


図 1: コンテンツ作成例

アプリケーションの処理、サイネージへの配信などを行うアクションパーツ、処理の分岐や結合などワークフローのデータの流れの制御を行う制御パーツの3つの種類がある。これらのパーツはそれぞれの処理に必要な細かい処理を内包している。そのため、ワークフロー上でユーザが扱うデータのタイプは、文字列を表すテキスト、ファイルの場所や URL を表すためのパス、出力されるコンテンツを Web コンテンツの形式で扱うコンテンツ、複数の要素を格納している集合、の4つで済むようになっている。パーツが扱うデータタイプは、パーツについているマーカの色で識別できる。左側のマーカに入力、右側のマーカに出力のリンクを接続するようにになっていて、マーカの色が一致しているパーツ同士でないと接続することができない。これによって、ユーザが接続することのできるパーツが視覚的に分かるようになっている。

## 3 ソースコード生成機構

ワークフローを作るとき、あらかじめ用意されたパーツ以外の処理をワークフロー中で利用したい、以前作ったワークフローを再利用したい、ということが考えられる。そこで、本システムではソースコード生成機構が組み込まれている。これはユーザが作成したワークフローからソースコードを生成したり、逆にソースコードからシステム上で用いるパーツを生成するというものである。

今回、ソースコードを ActionScript として生成した。ソースコードの生成を行うと、このソースコードとこのコードを実行するときに必要な XML ファイル、実行可能ファイル、そして、入出力などシステム上でパーツとして使うために必要な情報が記入されたファイルが1つのフォルダにまとめて生成される。ユーザは生成されたコードを編集することで、出力されたワークフローの処理を編集することができる。また、このソースコードは本システムの実行環境がない他の PC でも、ソースコードの実行環境があれば実行することができる。本ビジュアルプログラミング環境上でこれを再利用するときは、作成時に生成されたファイルの入ったフォルダを読み込ませ、キャンバス上にユーザの独自パーツとして配置して用いる。

## 4 システム構成

図 2 にシステム構成図を示す。システムは、エディタモジュール、スケジューラモジュール、評価器モジュール、処

Smart Signage System Based on Web Mash-up

Shota IMAI, Ryoji SUZUKI, Shun SHIRAMATSU, Tadachika OZONO, and Toramatsu SHINTANI

<sup>†</sup>Dept. of Computer Science, Nagoya Institute of Technology.

<sup>††</sup>Dept. of Computer Science and Engineering, Graduate School of Engineering, Nagoya Institute of Technology.

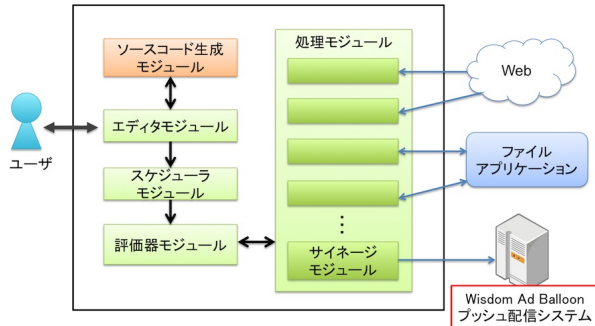


図 2: システム構成図

理モジュール、および、ソースコード生成モジュールから成り立っている。ユーザは、GUI上のエディタモジュールによりワークフローを作る。エディタモジュールはエディタ上で作成されたワークフローのパーツ、リンクの情報を保持して、スケジューラモジュールにその情報を提供する。スケジューラモジュールは、このエディタモジュールからの情報をもとにワークフローの構造を読み取り、適切な順番に処理が行われるようにスケジューリングされたデータを作成する。評価器モジュールはこのスケジューリングされたデータに基づいて処理モジュールを順番に呼び出すことでワークフローを実行する。処理モジュールは、評価器モジュールからの命令により、Web APIからのデータの読み取りや、サイネージ端末へのコンテンツの配信などの適切な処理を行うモジュールを呼び出す。ソースコード生成モジュールではエディタモジュールで作成したワークフローからソースコードを作成したり、作成されたソースコードから情報を読み取ってエディタモジュールに適切なパーツを配置する。

本研究では、ビジュアルプログラミング環境、および、ソースコード生成機構は Adobe AIR を用いて実装した。PC 上のデータおよびアプリケーションの操作には、Apple 社の OS である Mac OS X における Apple Script を利用することで、プログラムからアプリケーションの操作を行なっている。

図 2 中のプッシュ配信システム Wisdom Ad Balloon (以降、WAB) [3] は、本研究室で開発しているシステムで、Web ページ内のユーザが指定した配信領域に対してコンテンツをプッシュ配信するシステムである。WAB は、プッシュ配信に必要な技術を効果的に隠蔽しているため、Web 技術の経験が少ないユーザでも簡単にプッシュ配信を利用したシステムを実現することができる。WAB によって配信されたコンテンツは、Web ブラウザ相当の機能を持つタブレット端末で閲覧ができる。ビジュアルプログラミング環境内には、この WAB を用いるパーツが用意されていて、ユーザは作成したコンテンツをこのパーツを用いて配信する。

## 5 実行例と考察

本システムの実行例として、キーワードから写真や地図、そして Twitter の情報を得て、コンテンツを構築し、サイネージ端末に配信した例を図 3 に示す。観光地に関する広告を配信する利用状況を仮定し、「名古屋城」というキーワードを用いた。この例では、それぞれの Web API からの情報を 3 つのパーツから取得して結合し、それを WAB のパーツを用いてプッシュ配信している。このように、ユーザが直接 Web API を使ってデータを抽出したり、HTML などでのコンテンツの編集をしたりすることなく、マッシュアップして作成したコンテンツをサイネージへ配信することが可能である。さらに、プログラミングスキルのある人は処理を拡張したり、Web コンテンツ作成に慣れた人は JavaScript によりインタラクティブなコンテンツを作成したりすることもできる。

ただし、サイネージシステムのユーザは店主や観光関係者



図 3: 実行例

など必ずしもプログラミングスキルがあるとは限らない。そうしたユーザに対しても利用しやすいシステムにするためには、まだ改善の余地があると考えられる。プログラミング経験のないユーザの場合、まず、どのパーツとどのパーツがつながり合わせることができるか、というイメージがつきにくいと考えられる。そのため、よく使われるようなワークフローをあらかじめテンプレートとして用意しておき、それを編集する形で利用する、といったようなユーザの意図する処理の作成を支援できる工夫が必要である。また、ワークフロー上で用いるパーツは、粒度が大きければ少なく済むためにワークフローも分かりやすくでき、粒度が小さければ、パーツの量が増えるためワークフローは分かりにくくなるが、自由度が高い処理が実現しやすいと考えられる。これより、プログラミング知識のないユーザには粒度の高い部品を、プログラミングに習熟したユーザには粒度の低い部品を、というように、ユーザの知識レベルによってワークフローのパーツの粒度を替える、といった機構があることが望ましいと考えられる。さらに、広告などのサイネージ用コンテンツの作成では、マッシュアップ時のレイアウトの調整もできた方が望ましい。本稿のビジュアルプログラミング環境では、この点で改良の余地があり、これも今後の課題である。

## 6 終わりに

本稿では、タブレット端末を利用したサイネージシステムである、スマートサイネージシステムにおいて、Web マッシュアップを用いてコンテンツを作成、配信可能なシステムをビジュアルプログラミング環境によって実現した。本システムでは視覚的な操作によってコンテンツの作成が可能で、ユーザがコンテンツを作成する時間や手間を減らすことが期待できる。今後、行える処理の拡張や、条件分岐などの細かいワークフローの制御をできるようにすることが課題である。

## 参考文献

- [1] 鈴木亮詞, 村瀬隆拓, 白松俊, 大園忠親, 新谷虎松: タブレット端末を用いたスマートサイネージシステムの実現とその応用, 日本ソフトウェア科学会第 28 回大会, 2011.
- [2] KO, A.J., et al.: The State of the Art in End-User Software Engineering, ACM Computing Surveys, Vol.43, No.3, 2011.
- [3] 向井康人, 大園忠親, 伊藤孝行, 新谷虎松: Wisdom Ad Balloon: プッシュ型配信技術に基づく動的ページ構成システム, 第 4 回情報技術フォーラム, 2005.