

# モデルとビューを分離したプレゼンテーション用ビジュアルエイドシステム

大坪五郎

株式会社 LIFULL

ohtsubogoro@lifull.com

**概要** プレゼンテーションを視覚的に補助するビジュアルエイドは、通常「スライド」として作成される。しかし「スライド」にはプレゼンテーションのストーリーとして喋る内容と、その中で強調するため画面に表示すべき部分が混在している。本来プレゼンテーションで一番重要なのはストーリーでありそれを「モデル」と捉えれば、ビジュアルエイドとして人の目に触れる部分の記述を「ビュー」として分離することが必要なのではないかと、そうした問題意識に基づいたプレゼンテーション用ビジュアルエイドシステムを開発した。まず平易な文章でプレゼンテーションのストーリーをモデルとして作成する。その上でストーリー中の最小限必要な部分を選択し、画面に表示する際の仕様を簡易なビジュアルプログラミング言語でビューとして定義する。これによりプレゼンテーションのストーリーをスライドという形式にとらわれることなく検討することができる。本システムの開発及び実際に使用した結果について述べる。

**キーワード** プレゼンテーション, ビジュアルプログラミング

## 1 はじめに

現状広く使われているプレゼンテーションソフトウェアはいずれもスライドをベースにしている。これはおそらく過去に使われていた物理的実態を持った「スライド」を置き換える目的からスタートしたためと考えられる。スライドをベースにプレゼンテーションの準備を行うことの一番のメリットは画面に映し出す情報を直接構築・編集できるため、手間が省ける点にあると考えられる。しかしながら、スライド（及び補助としてノート）の集合体がプレゼンテーションにおいて最も重要なプレゼンテーションのストーリーを記述するのに最適であるとは限らない。逆にスライドを並べるだけでプレゼンテーションの準備ができたような気がしてしまうため、ストーリーの検討がおろそかになる弊害 [4] があるのではないだろうか。

さらには、昨今プレゼンテーション用スライドは、単にプレゼンテーションの最中に用いられるだけではなく、プレゼンテーション終了後のハンドアウトとしても使われることが多い。しかしながらこの二つの機能を一組のスライドで賄うのには問題がある。本来プレゼンテーションは、プレゼンターが聴衆に語りかけるものである。画面に映し出すのはその語りかけを支援するための最小限の情報であるべき。逆にハンドアウトとして渡す際には、それだけで受取手がほとんどの情報を得られることが期待される。前者の目的にフォーカスすれば画面上の情報は少なくなり、ハンドアウトとして渡されても意味が取れないことになる。後者の目的にフォーカス

すれば、表示される文字は必要以上に多くなり、プレゼンテーション時において聴衆は退屈な「画面上の文字の読み上げ」に付き合わされることになる。

こうした問題はプレゼンテーションの補助を行うアプリケーションはどのような機能を持つべきか、という根本的な点に立ち戻ることなく従来のスライドという枠組みの中で問題を解決しようとしていることから生じるものではないだろうか。

こうした問題意識に基づき、プレゼンテーションを支援するビジュアルエイドの表示制御はどのように行うべきかという考察に基づいたプレゼンテーション用ビジュアルエイドシステムを開発した。

## 2 プレゼンテーション用ビジュアルエイドシステムの開発

### 2.1 本システム開発前の実運用

著者は以前より、プレゼンテーションのビジュアルエイドは、画面上の詳細なレイアウトの定義より画面に表示されるタイミングのコントロールを主眼とするべきであり、スライドではなくプログラムとして記述すべきである、という主張を持っていた。この構想にしたがって開発したのが Gozen[6] である。このシステムは独自のプレゼンテーション記述用言語を用いてビジュアルエイドの表示仕様を記述するものであった。筆者はこのシステムを持ちいて合計4回社外向けプレゼンテーションを実施した。その結果当初の目論見通りビジュアルエイドの記述が直感的になり、かつプレゼンテーションに合わせて表示タイミングを細かく制御できるというメリットを確認できた。それと同時に以下のような問題点も明ら

かとなった。

1. 独自プログラミング言語を使用することによる、信頼性の低さ。
2. エディタのコード補完機能がサポートされないことによる開発効率の低さ。
3. プログラムのモジュール化が十分でなく、長いプレゼンテーションになると全体の把握/編集が困難になる。
4. プレゼンテーションのストーリーをプログラム中のコメントとして残すことは可能だが、プログラムの編集・更新とともにコメントとして残したストーリーと表示内容が乖離する。

本研究ではこれらの問題点を解消することを目的とした。

## 2.2 システム設計方針

システム開発にあたり以下に示す設計方針をたてた。

- ビジュアルエイドをモデルとビューに分離する。モデルとはプレゼンテーションのストーリーであり、自然言語で記述する。その中から画面に表示すべき部分を指定し、表示仕様を示すビューを設定することによりビジュアルエイドを構成する。
- ビューの記述は、編集が容易なビジュアルプログラミング言語により行う。
- 上記の特性を生かし、従来のスライド型プレゼンテーションソフトウェアでは達成が難しかったプレゼンテーション上のニーズに対応する。

以下項目ごとに説明する。本研究ではプログラムの設計パターンとして広く用いられている Model-View-Controller パターンから着想を得て、プレゼンテーションのストーリーをモデルとして扱い、それをどう画面に表示すべきかのビューと分離することを試みた。これにより、モデルであるストーリーを表示方法の記述と混同することなく検討することが可能となる。また「ストーリーのうち、画面に何を表示すべきか」という点を明確に意識して定義するため、従来のスライド型プレゼンテーションで起こりがちだった「ストーリーの検討をスライドの作成と混同する」「本来ストーリーとして記述すべき“話す内容”をスライドに記述する」といった間違いを防ぐことが期待できる。

2項目目に関して、ビューの記述は Scratch[3] に類似したビジュアルプログラミング言語を用いた。これにより、ビューの記述が直感的かつ容易になる。この方法を

とることの欠点として記述の柔軟性が失われる点が挙げられる。しかし前述したようにプログラミング言語を用いてプレゼンテーションの記述を行った経験から通常のプレゼンテーションにおいて必要なプログラミングのパターンが抽出できていたため、それらをサポートすることでビジュアルエイドの記述が可能であると判断した。

3項目目に関し、従来のスライドを利用したプレゼンテーションソフトウェアではスライド単位でしか情報の表示が制御できないため実現が難しい機能があった。例えばスライド内でユーザ操作に応じ逐次的に表示を変更しているのに、プレゼンテーションソフトウェアからの Twitter への投稿がスライド単位でしか行えないなどである。またプレゼンテーション時に「次に何を話すか」を参照する際もノートもしくは次スライドしか参照できず柔軟性に欠けた。そうした問題を解決することを狙った。

## 2.3 システム設計

開発したアプリケーションのスクリーンショットを図 1 に示す。

本システムのインターフェースはモデルエディタ、ビューエディタ、プレゼンテーションコントローラの三つの部分に別れている。モデルエディタは、基本的にテキストエディタである。プレゼンテーションのストーリーをこの上で編集する。このエディタには、画像や動画などをドラッグすることで、そのパスを含めることができるが、基本的には自然言語でストーリーを記述する。

ここからプレゼンテーションのビジュアルエイドを作成する際には、モデルエディタ上で画面に表示したい部分を選択し、引用ボタンを押す。すると引用したテキストの前後に括弧が表示され、括弧内のテキスト末尾を拡張子として解釈し、そのタイプ（画像、動画、文章）に対応するプログラムコンポーネントがビューエディタに追加される。画面上での表示位置、順番、フォントサイズなどは全てこのビューエディタ上で設定する。次にビューエディタ上でストーリーから引用されたデータの「見栄え」をビジュアルプログラミングを使用して設定していく。設定された結果は、プレゼンテーションコントローラ部分にリアルタイムに表示されるので、編集結果をただちに確認することができる。

以下実装詳細について述べる。

## 2.4 実装詳細

### 2.4.1 ビジュアルプログラミング言語

本システムでは、大別して以下に示す 6 種類のコンポーネントを用いてビューを記述する。

1. 表示：テキスト/画像/動画の表示
2. Wait: 停止してユーザの操作を待つ

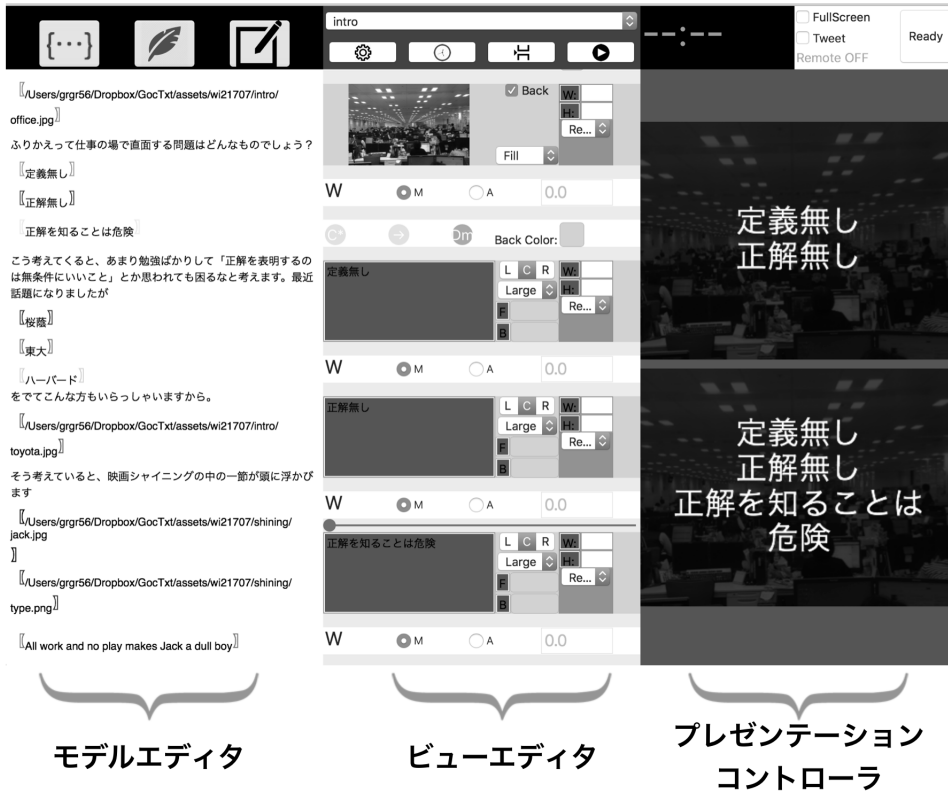


図1 開発したアプリケーションのスクリーンショット

3. 設定：背景色，表示要素を配置する方向，表示要素の削除
4. 再生：動画の再生
5. Tweet:Twitter に投稿する文章
6. ノート：リモコン上に表示する注釈

動作の基本は，画面上に複数の要素を並べて表示しユーザの入力を待つ，というものである．ユーザの入力がなされると，次のコンポーネントで指定された動作を行う．それぞれのコンポーネントでは，表示する際のフォントサイズ，背景色，サイズ，画面上での位置などを GUI を用いて設定することができる．画面上の位置指定は，絶対位置での指定も可能である．しかしこれまでに行ったプレゼンテーションの経験から，多くの場合表示要素を並べる方向を指定した上で，単純に追加し，レイアウトはシステム側で自動的に行うことで十分な場合が多い．一例としての記述を図2に示す．

以下上記プログラムでの動作を説明する．1でそれまでの画面表示要素をクリアする．次に2で指定された画像を背景に表示する．ここまで動作したところで3でユーザの入力を待つ．ユーザがプレゼンテーションを進める入力をするると，4で次に表示する文字を見やすくするため背景画像を半透明の面でマスクする．5で文字を表示．ここでフォントのサイズと文字列の左右



図2 ビジュアルプログラムの例

方向の配置を設定している。6 でユーザの入力を待つ。ユーザが入力を行えば、7 で別の文字列を表示し、8 でユーザの入力を待つ。このように本システムでは直感的に画面上の表示を設定することができる。仮に最初から全ての要素を表示したければ3と6のWaitコンポーネントを削除するだけであり、極めて簡単に一括表示、あるいはユーザ操作に合わせた逐次表示を選択することができる。

またこれらの複数のコンポーネントをモジュールという単位にまとめることができる。この機能により例えば“Introduction”、“Demo”、“Algorithm”などの意味のある単位ごとにプレゼンテーションの記述、管理が可能となる。

Tweet機能を用いることにより、モデル上の任意の記述をハッシュタグをつけた上でTwitter上にタイミングを指定して投稿可能である。またこの機能はプレゼンテーションを一旦終了した後は、プレゼンテーション全体を再読み込みするまで機能しない。これによって、質疑応答の際に情報をブラウズしている際に再度情報がTweetされることを防ぐことができる。

### 3 システムの使用例

実装したシステムを用いて社外向けのプレゼンテーション(ARG 第10回Webインテリジェンスとインタラクション研究会)を行った。そこで得られた知見について以下に列挙する。

- ストーリーと、表示仕様を分離するコンセプトは期待通り機能した。
- ビジュアルプログラミングによるビューの記述も可能であり、リアルタイムで画面のプレビューが見られるのはプログラムを行う上で大変便利なのが確認できた。
- プレゼンテーションの本番中何度か「次に何をしゃべるか」とつまった場面があったが、次画面ノート表示機能により支障なくプレゼンテーションを進めることができた。
- 質疑応答の際に、途中の画面を見せようとした場合現在の方式では簡単に指定することができない、という問題がある。

信頼性において市販ソフトウェアより劣る自作システムを外部向けのプレゼンテーションに用いるといった心理的な重圧は別として、上述のように概ね当初狙った機能は実現できていると考えられる。

### 4 関連研究

プレゼンテーション用のプログラムに関する研究は多いが、それらのほとんどは従来型のスライドベースのプレゼンテーションソフトウェアを前提としている。その一方でPowerPoint,Keynoteなどのプレゼンテーションソフトウェアを用いず、テキストベースでプレゼンテーションを記述するシステムも使われるようになってきている。代表的なものとして、Deckset[1]およびreveal.js[2]を挙げる。Markdownでプレゼンテーションを記述することが可能である。この際プレーンのテキストで表示内容を記述し、表示の仕様(フォントサイズなど)は最小限の記号で指定する。これはある程度プレゼンテーションの画面に表示する「内容」と「見え方」を分離する試みと考えることができコンセプトにおいて本システムと共通する点がある。宮本らによるLaTeXで記述された論文からプレゼンテーション用のスライドを自動生成する試み[5]は、論文をモデル、プレゼンテーション用スライドをビューと捉えれば本研究とコンセプトにおいて共通する点があると考えられる。

### 5 むすび

プレゼンテーションのストーリーをモデルとして捉え、それをどのように表示するか「ビュー」と分離したプレゼンテーションのビジュアルエイドシステムを開発した。今後本システムを用いた社外向けのプレゼンテーションをさらに行い、知見を蓄積、システムに反映させることでより柔軟、かつ強力なプレゼンテーション支援システムの開発を行っていききたい。

#### 参考文献

- [1] Deckset. <https://www.decksetapp.com>.
- [2] reveal.js. <http://lab.hakim.se/reveal-js/>.
- [3] J. Maloney, M. Resnick, N. Rusk, B. Silverman, and E. Eastmond. The Scratch Programming Language and Environment. *Trans. Comput. Educ.*, 10(4):16:1-16:15, Nov. 2010.
- [4] カーマイン・ガロ. スティーブ・ジョブス 驚異のプレゼン. 日経BP社, 2010.
- [5] 宮本雅人, 酒井浩之, 増山繁. 論文LaTeX原稿からのプレゼンテーションスライド自動生成. *知能と情報: 日本知能情報ファジィ学会誌*, 18(5):752-760, Oct. 2006.
- [6] 大坪 五郎. Gozen:プレゼン用「ビジュアル・エイド」のあるべき姿. 第20回インタラクティブシステムとソフトウェアに関するワークショップ, 2012.