

Title	情報視覚化を目的としたビジュアルイズフレームワークの提案
Sub Title	Design and implementation for information visualization
Author	松井, 加奈絵(Matsui, Kanae) 砂原, 秀樹(Sunahara, Hideki)
Publisher	慶應義塾大学大学院メディアデザイン研究科
Publication year	2010
Jtitle	
JaLC DOI	
Abstract	
Notes	修士学位論文. 2010年度メディアデザイン学 第61号
Genre	Thesis or Dissertation
URL	https://koara.lib.keio.ac.jp/xoonips/modules/xoonips/detail.php?koara_id=KO40001001-00002010-0061

慶應義塾大学学術情報リポジトリ(KOARA)に掲載されているコンテンツの著作権は、それぞれの著作者、学会または出版社/発行者に帰属し、その権利は著作権法によって保護されています。引用にあたっては、著作権法を遵守してご利用ください。

The copyrights of content available on the Keio Associated Repository of Academic resources (KOARA) belong to the respective authors, academic societies, or publishers/issuers, and these rights are protected by the Japanese Copyright Act. When quoting the content, please follow the Japanese copyright act.

2010 年度 修士論文

情報視覚化を目的とした
ビジュアルライズフレームワークの提案



KEIO MEDIA DESIGN

慶應義塾大学大学院
メディアデザイン研究科

松井 加奈絵

本論文は慶應義塾大学大学院メディアデザイン研究科に
修士(メディアデザイン学) 授与の要件として提出した修士論文である。

松井 加奈絵

指導教員：

砂原 秀樹 教授 (主指導教員)

稲蔭 正彦 教授 (副指導教員)

審査委員：

砂原 秀樹 教授 (主査)

稲蔭 正彦 教授 (副査)

大川 恵子 教授 (副査)

情報視覚化を目的とした ビジュアライズフレームワークの提案

内容梗概

情報爆発時代における情報過多により、従来の情報抽出、表示方法では、ウェブ利用者が必要とする情報を取得することは難しい。本論文ではこの問題の解決のために、利用者の情報取得を容易にする情報視覚化をウェブコンテンツに適応することを目標とする。

情報視覚化とは、大量の情報から必要な情報の抽出し、その情報の特性を色や大きさといった視覚的効果で表す技術である。しかし情報視覚化を行うためには、複数ソースからの情報取得、解析、フィルタリング、マイニングといった専門的知識と技術を必要とする情報分析の過程を行わなくてはならない。そのためウェブコンテンツ開発者に対し、これらの過程を行う開発負担がかかる。

そこで本研究では、ウェブコンテンツ開発者に対し、情報の配信を行っているAPIから取得した大量の情報を分析し、その後XML形式に変換した情報ソースの提供を行う、ビジュアライズフレームワークを提案する。提案のフレームワークは、情報分析の過程で時間軸や関心軸等の複数軸からの解析によって必要であると判断された情報と、その情報の性質を表す色や大きさなど視覚的効果の参照値を生成する。開発者は提供の情報ソースを使用することで、情報分析を行う開発過程を短縮することができる。その結果情報視覚化ウェブコンテンツが普及し、利用者が必要な情報を取得し易い環境構築の実現を期待した。

提案のフレームワークの有用性を確認するために、実際に開発した情報視覚化コンテンツの開発過程と、本フレームワークが提供する情報ソースを使用した場合の開発過程を比較した。その結果、本フレームワーク提供の情報ソースの使用による、開発過程の短縮が確認された。また常に変動する情報を取り扱う情報視覚化ウェブコンテンツに対し、情報ソースの使用による負荷軽減が確認された。

キーワード

情報視覚化, フレームワーク, ウェブコンテンツ, XML データ, API

慶應義塾大学大学院 メディアデザイン研究科

松井 加奈絵

Design and Implementation for Information Visualization

Abstract

In this era of information explosion, it is difficult for the Internet users to acquire valuable information by using current technologies of information extraction and presentation.

To solve this problem, a studying of information visualization can be one of the best ways because this approach can represent large scale collections of non numerical information. To create web contents which visualize meaningful information with visual effects like changing color and size by using information visualization is a purpose of this dissertation.

However, this can be difficult to utilize this study because information visualization has seven steps which are acquiring and parsing large scale data, filtering non-meaningful information, data mining, representing, refining and creating interaction. Design steps like representing, refining, creating interaction are popularized by progress of software including programming languages and web browsers. However analyzing information steps which are acquiring, parsing, filtering and data mining are still needed professional procedure for web developers.

To make them easier, this dissertation proposes a framework for information visualization. This framework provides a data source in for non-professional web developers so that they may obtain analyzed data. To utilize them, web developers do not have to hard coding when they want to create web contents with visualized information.

Efficiency of the proposed framework is evaluated by comparing information visualized web contents from scratches and from provided source that proposed framework created.

Keywords:

Information visualization, Framework, WebContents, XMLdata, API

Graduate School of Media Design, Keio University

Kanae Matsui

目 次

第1章	はじめに	1
第2章	情報視覚化の必要性	3
2.1.	背景	3
2.2.	目指す環境	4
2.3.	対象の情報	5
2.4.	情報視覚化における課題	5
2.5.	課題への取り組み、目標	6
第3章	先行研究	7
3.1.	ビジュアライジング・データ	7
3.1.1	情報視覚化を行うための手順	7
3.1.2	プログラミング言語/総合開発環境 Processing とは	10
3.2.	本研究との比較	11
第4章	ビジュアライズフレームワークの提案	13
4.1.	ビジュアライズフレームワークの役割	13
4.1.1	ビジュアライズフレームワークが行う情報分析	13
4.1.2	ビジュアライズフレームワークが行う情報ソース提供	14
4.2.	ビジュアライズフレームワークの機能	15
4.2.1	情報取得	15
4.2.2	解析	17
4.2.3	フィルタリング	17
4.2.4	マイニング	19
4.2.5	情報ソースの提供	21

4.3.	提供する情報ソースの使用方法	22
4.3.1	参照値メタ情報使用例	23
4.3.2	情報視覚化ウェブコンテンツ開発例	23
4.3.3	発展例	24
4.4.	本フレームワークが果たす役割	24
第5章	評価	26
5.1.	評価概要	26
5.2.	比較対象	26
5.2.1	ウェブコンテンツ trend-setter	27
5.2.2	ウェブコンテンツ twit-viewer	29
5.2.3	ウェブコンテンツ fu*life	33
5.3.	結果	36
5.3.1	開発過程の比較	36
5.3.2	取り扱いデータに対する負荷の比較	39
第6章	考察	42
6.1.	開発過程の変化	42
6.2.	情報処理の変化	43
6.3.	今後	43
6.3.1	取り扱いデータの拡充	43
6.3.2	情報分析の強化	44
6.3.3	利用者に対する利便性の担保	44
6.4.	まとめ	46
第7章	おわりに	47
付録 A	ウェブコンテンツ trend-setter 使用方法	54
A.1.	概要	54
A.2.	開発環境	54
A.3.	画面構成	55

A.4. 使用方法説明	55
付録 B ウェブコンテンツ twit-viewer 使用方法	57
B.1. 概要	57
B.2. 開発環境	57
B.3. 画面構成	58
B.4. 使用方法説明	58
付録 C ウェブコンテンツ fu*life 使用方法	60
C.1. 概要	60
C.2. 開発環境	60
C.3. 開発過程	61
C.4. 使用方法説明	62

目 次

3.1	情報視覚化までの7つの段階	8
3.2	情報視覚化ウェブコンテンツ開発において各々の研究が果たす役割	12
4.1	ビジュアルライズフレームワーク概念図	14
4.2	流行の情報に関する情報ソース生成の流れ	16
4.3	解析の機能	18
4.4	フィルタリングの機能	19
4.5	大きさによる視覚化例	20
4.6	色による視覚化例	20
4.7	ビジュアルライズフレームワークが提供する情報ソースの構造 . . .	21
4.8	情報ソース取得の流れ	22
4.9	提案のフレームワークによる開発過程の変化	25
5.1	trend-setter システム図	28
5.2	trend-setter 視覚化手法	30
5.3	twit-viewer システム図	31
5.4	twit-viewer 視覚化手法	33
5.5	fu*life システム図	34
5.6	fu*life 情報視覚化手法	36
5.7	一般的な情報視覚化ウェブコンテンツ開発過程	37
5.8	本フレームワークを使用した情報視覚化ウェブコンテンツ開発過程	38
5.9	trend-setter に必要なデータの取得	40
5.10	trend-setter に必要な情報ソースの取得	40
5.11	XML形式の比較	41

6.1	生成情報ソースカスタマイズ画面	45
6.2	生成情報ソース取得画面	45
A.1	初期画面	55
A.2	アイコン配置時	55
B.1	初期画面	58
B.2	アイコンステージ配置時	58
C.1	画面推移	61
C.2	fu*life トップページ	61
C.3	コメント閲覧画面	62
C.4	ブーケ作成画面	62

目 次

4.1 参照値の使用例	23
5.1 取得した情報量と必要な情報量	39
A.1 trend-setter 開発環境	54
B.1 twit-viewer 開発環境	57
C.1 fu*life 開発環境	60

第1章

はじめに

現在情報のデジタル化の進展に伴い、世界中で急激なネットワーク上の情報量の増加、いわゆる情報爆発時代の到来を迎えている。そのため利用者が真に必要な情報の取得が困難な状況に陥っている。また今後も情報量は増え続けるとの統計があり、更に深刻な状況になると予想される。そこで本研究では、膨大な情報の中から必要な情報を抽出し、情報の性質を色や大きさなどによる視覚的効果を使用して表す、情報視覚化技術に着目した。ウェブコンテンツに情報視覚化技術を適応することで、利用者が必要な情報の取得が容易となる環境の構築を目指した。

しかし、ウェブコンテンツ上で情報視覚化を行うには以下の2つの開発過程が必要となる。1つ目は情報取得、解析、フィルタリング、マイニングによって情報分析を行う過程である。2つ目は情報分析から抽出された情報を使用し、どのようにインターフェースのデザインを行うかといった情報表示の過程である。情報表示の過程に関してはソフトウェア、プログラミング言語、ブラウザなどの複合的な技術の進歩によって、専門的な知識をもたない者でも比較的容易に開発が行う環境が整いつつある。しかし情報分析の過程は、複数の提供先からの情報ソースの取得、その後の処理といった専門的な知識、技術が要求される。

そこで本研究では、情報分析の過程を経た、情報視覚化のための情報ソースをウェブコンテンツ開発者に提供する、ビジュアルライズフレームワークの提案を行う。本フレームワークではAPI(Application Program Interface)を通じて提供されるニュース情報など、ネットワークを介して取得可能な変動する大量の情報を対象としている。これらを複数のAPIから取得し、解析、フィルタリング、マイニングといった情報分析を行う。その過程から、必要であると判断された情報と、

情報の性質を視覚的効果として表す色や大きさなどの参照値をメタ情報として生成する。生成された情報は XML 形式に変換され、開発者にネットワークを通じて配信される。

本フレームワークが提供する情報ソースを使用することで、開発者は情報分析の過程を行う必要がなくなるため、開発負担が軽減されると考えられる。開発者に対する負担を削減することで、情報視覚化ウェブコンテンツ作成に対する難易度を下げ、更なる普及を期待した。

以降、本論文では、第 2 章で本研究の背景と課題について述べる。また、第 3 章にて先行研究について述べ、本研究との関連性、それぞれが果たすべき役割を明確にする。第 4 章では、ビジュアライズフレームワークの概要とシステムの詳細について述べる。第 5 章にて本フレームワークの有用性を確認するために、実際に制作した情報視覚化ウェブコンテンツの開発過程と、本フレームワークの情報ソースを使用した場合の開発過程の比較を行う。第 6 章では第 5 章で行った比較による評価を元に考察を行う。また、提案のフレームワークが今後どのような課題に取り組むべきかを明確にする。

第2章

情報視覚化の必要性

本章では本研究を行うにあたっての背景について述べる。またその背景から導き出された課題を明確にする。また本研究がその課題に対し、どのように取り組むかについて述べる。

2.1. 背景

現在情報処理を行うコンピュータ台数の増加と処理能力の向上、それらを結ぶネットワーク化とブロードバンド化が進行し、ネットワーク上にデジタル化された情報が急激に増加している [1][2]。そのため情報爆発時代と称される、利用者にとって深刻な情報過多に陥っている [3]。ICT 分野の調査会社である International Data Corporation が2007年から2008年にかけて行った調査、「International Data Corporation」[4]によると、デジタル化された情報は2010年のうちに988EB(=1.29 × 10²¹ビット)に到達すると予想されている。

この状況を緩和するために文部科学省は、「情報爆発時代に向けた新しいIT基盤技術の研究」[5]を2006年から2010年かけて試行した。この研究では以下の3つの技術に焦点を当て、情報爆発時代に向けての研究を行ったものである。

1. 爆発する大量かつ多様な情報から真に必要とする情報を取り出す技術
2. 大量情報を適切に管理し運用するためのサステナブルなシステムやソフトウェアに関する技術
3. 人間との高度な対話により容易に情報を活用できるようにする技術の発展

このように深刻な情報過多に対して、どのような技術で事態の改善に取り組むべきか、現在活発に研究が行われている。

2.2. 目指す環境

本研究では、情報過多より利用者が必要な情報を取得できない事態を問題とし、この問題の改善に向けた提案を行う必要があると考えた。先に述べた情報爆発時代に向けての研究の1点目、爆発する大量かつ多様な情報から真に必要な情報を取り出す技術として、高度な情報検索や知識抽出など、様々な研究が進められている。本研究では、それらの技術から情報視覚化に着目した。

情報視覚化とは、変動する大量の情報を分析することで必要な情報を抽出し、その情報の特性を色や大きさの変化といった視覚的効果で表すという手法である[6][7]。情報視覚化は当初、サイエンティフィックビジュアライゼーションや、地理情報システムなど、大規模な情報処理を行うことができる施設を必要とする分野で発展した。主な目的は、DNA など通常目視できない大量の情報を視覚化することや、気象情報など広範囲に分布しかつ高速で変化する情報を視覚化し一定の法則を発見することなど、性質を把握し難い情報の理解に対する促進であった。従来こうした大量の情報処理には高価なコンピュータ等が必要であったが、コンピュータやネットワークの技術革新によって、パーソナルコンピュータにおいても大量の情報の取得、処理が可能となった。このような変化により、個人においても大量の情報を取り扱う機会が増加したため、プログラミングのデバッグやアルゴリズムの理解など、比較的小規模な分野に対して応用されるようになっていく[8]。

本研究では、日々増加するネットワーク上のデジタル情報を、情報視覚化によって分析、その情報を使用してウェブページやアプリケーションといったウェブコンテンツ上で動的に表示することで、利用者が必要な情報を取得し易い環境を構築することを目指した。

2.3. 対象の情報

本研究で述べる情報視覚化における対象となる情報について明確にする。ネットワーク上には、以下に述べるような多種多様なデジタル情報が存在する [9]。

- デジタル文書 (ウェブ、ワード、表計算)
- 個人発信情報 (ブログ、電子メール)
- デジタルコンテンツ (画像、映像)
- トランザクションログ
- 大規模モデル計算出力

本研究では、利用者がウェブコンテンツを通じて必要とする情報の取得を行うことを前提としており、近年の動的なウェブコンテンツの特徴である、常に変動するデジタル情報の取り扱いは必須であると考えられる。そのため、上記に挙げたデジタル情報のうち、一般向けに文書化およびサポートされた API を通じて、取得が行うことができるものを対象とする。ウェブコンテンツ開発者に情報を提供している API の例として、ニュース情報、ブログ情報、画像情報、映像情報の提供を行っている、Yahoo デベロッパーズ API [10]、Google Code [11] が挙げられる。

このように現在ネットワーク上には、API を通じて様々な種類かつ形式の情報ソースの取得ができる、大変豊かな環境が構築されている。

2.4. 情報視覚化における課題

ここでは、情報視覚化をウェブコンテンツに適応させるにあたって、具体的に発生すると考えられる課題について述べる。

情報視覚化には利用者の情報取得行為を促す効果があるが、情報視覚化を行うにあたって、以下の2つの過程が必要になる。情報取得、解析、フィルタリング、マイニングを行う情報分析の過程、表現、精密化、インタラクションの開発を行う情報表示の過程である。なお本過程の具体的な方法は第3章にて詳細を述べる。

情報表示の過程に関しては、Flash などのソフトウェア、後述で詳細を明らかにする Processing などのプログラミング言語、各ブラウザの HTML5 対応などによる総合的な技術進歩により、ウェブコンテンツ開発の初心者であっても、容易な開発が可能な環境が整いつつある。しかし情報分析の過程は、専門的な知識、技術が必要となるため、開発過程が複雑化し、開発者の負担になることが予想される。そのため、情報分析における負担軽減を行うことが必要であると考えた。

2.5. 課題への取り組み、目標

上述で述べたように、ウェブコンテンツへ情報視覚化を適応させるためには、情報分析の過程がもたらす負担を軽減することが必要であると考えられる。

そこで本研究では、情報分析の過程を経た情報ソースの提供を行う、ビジュアルライズフレームワークを提案する。提案のフレームワークでは、各 API から取得した情報の分析を行い、必要な情報と、その情報の性質を視覚的表現で表す際に必要となる色や大きさなどの参照値のメタ情報を生成する。その後、これら生成した情報を XML 形式のデータに変換し、ネットワークを介して開発者に提供する。

情報ソースによる開発者の情報分析の過程による負担軽減を行うことで、情報視覚化ウェブコンテンツが増加することが予想され、利用者にとって必要な情報を取得し易い環境が構築されることを期待した。

第3章

先行研究

本章では、情報視覚化の先行研究として、Ben Fry 氏著作「ビジュアルライジング・データ:Processing による情報視覚化手法」[6] 内で語られたことを中心に述べる。その後、この先行研究と本研究の比較することで本研究の果たすべき役割を明確にする。

3.1. ビジュアルライジング・データ

書籍「ビジュアルライジング・データ:Processing による情報視覚化手法」は、作者である Ben Fry 氏が自身の博士論文として発表した、「Computational Information Design」[12] の内容を元に刊行された技術書である。

この書籍は情報視覚化ウェブコンテンツの開発をどのように作成するか詳細を述べたものであり、以下に挙げる2つの内容で構成されている。

- 情報視覚化を行う手順
- 情報視覚化ウェブコンテンツの開発を行うためのプログラミング言語、総合開発環境 Processing の使用方法

以後、この2点について詳細を述べる。

3.1.1 情報視覚化を行うための手順

情報視覚化を行う手順とは、第2章で述べた情報視覚化ウェブコンテンツ作成の2つの過程である情報分析、情報表示を行うにあたり、実際にどのような作業

すべきかをまとめたものである。Ben Fry氏は書籍内で、手順として7つの段階が必要であると述べている。

- 情報分析の過程

- 1. 情報ソースの取得
- 2. 解析
- 3. フィルタリング
- 4. マイニング

- 情報表示の過程

- 5. 表現
- 6. 精密化
- 7. インタラクション

図3.1はこの7つの段階をまとめたものである。

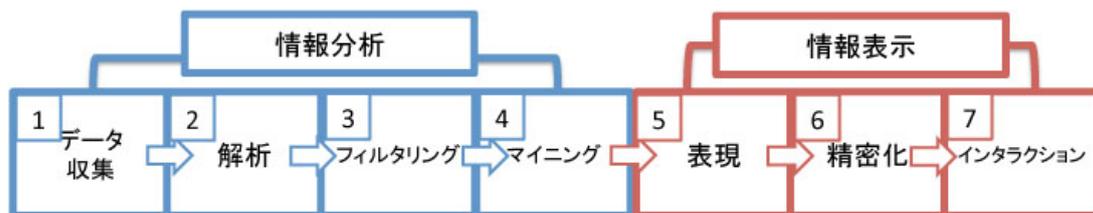


図 3.1 情報視覚化までの7つの段階

以下、各段階の詳細について述べる。

データ収集

ウェブコンテンツ開発者が、表示の対象として必要なデータを取得する段階である。この場合、データはデジタル化された情報を対象とする。ネットワークを介して取得する場合や、メディア (CD-R 等) を介して取得する方法が挙げられる。

ネットワークを介してデータを取得する際には、大規模システムから必要な情報ソースのみを選択して取得するような複雑な方法を取る場合や、利用可能なテキストファイルから取得するような簡易な方法を取る場合が存在する。

解析

収集したデータの構造を解析する段階である。デジタル化されたデータはXMLに代表されるマークアップ言語で成り立っていることが多い。各データは情報の情報を表すメタ情報群から構成されている。主なメタ情報として、生成時間や、関連情報、画像のリンク等が挙げられる。この段階では、データを構成する文字列や数字で成り立つ各メタ情報が、どのような情報を有しているのか理解することが目的である。

フィルタリング

解析した情報ソースから、ウェブコンテンツ開発上必要のないと考えられる情報を除去する段階である。開発における目的によっては、数理モデルにあてはめる、正規化を行う、といった数学処理による除去が必要となる。

マイニング

フィルタリングの段階を経たデータから、データマイニングを行う段階である。データマイニングとは、取得した情報を数理モデルや統計にあてはめ、相関関係やパターンを抽出する技術である [13][14]。情報視覚化におけるマイニングとは、この技術を使用し、必要な情報の抽出と、情報の性質を導きだし、色や大きさといった視覚的効果に付加するための値を生成することを意味する。

以上が情報分析の過程であり、以下説明を行うものが、インターフェース上における情報表示の過程である。

表現

この段階では、情報分析を行った情報ソースを表示するにあたって、基本的な形の決定を行う。円、棒グラフによる一般的なパターンや、情報視覚化で多用される木構造などのリンクを表す際に適したパターンなど、その情報の性質を表すために一番相応しいものの選択を行う。

精密化

この段階では、情報分析の過程を経て導きだされた情報の性質を表す視覚的効果の参照値を使用し、グラフィックデザインの手法を用いてインターフェースへの反映を行う。色や大きさなどの参照値から、色の濃淡や透明度の変化、アニメーションによる動きを利用して、情報の性質の表現をインターフェース上で行う。

インタラクション

この段階では利用者との対話機能を追加することで、利用者によるデータの制御や探索を行う。具体的には、利用者に対するデータのサブセットの選択機能の作成や、スクロールバーを用いた観点の切り替え機能の作成などが挙げられる。

3.1.2 プログラミング言語/総合開発環境 Processing とは

Processing とは Ben Fry 氏と Casey Reas 氏が開発したオープンソフトプロジェクトのプログラミング言語、総合開発環境である [15][16]。Processing は Java を単純化してグラフィック機能に特化したものであり、プログラミング初心者でも簡単にウェブコンテンツ開発が行えるという利点をもつ。

開発当初はアートとデザインを専門としているプログラミング初心者に対し、優れたグラフィックデザインを行うための開発環境の提供を目的としていた。だがその後、大規模なインスタレーション作品、モーショングラフィクス、複雑な情報視覚化のための機能が追加され、プログラミング上級者の要求にも耐えうる本格的なデザイン、プロトタイピングツールとして進化した。

Processing はそのシンプルなコードによる開発過程により、プログラミング初心者であっても、変動する情報ソースを使用したグラフィカルなインターフェース、高度なインタラクションの実装した情報視覚化ウェブコンテンツの制作を可能にしたことで高く評価されている。

書籍「ビジュアルライジング・データ:Processing による情報視覚化手法」では、Processing を利用して、情報視覚化の手順に沿ってウェブコンテンツの実装方法を紹介している。

3.2. 本研究との比較

以下、先行研究と本研究の果たすべき役割を明確にする。

Ben Fry氏は著書内にて、情報視覚化を行うための具体的な手順と、Processingによるウェブコンテンツへの実装方法を明らかにした。しかし、Processingはネットワーク上の情報を取得する機能が実装されているものの、インターフェース開発のためのグラフィック機能に特化しているため、大量の情報の取得やその解析など情報分析の過程を行うには適していないと述べられている。つまり、情報視覚化ウェブコンテンツ作成のための情報ソースを作成するには、別のプログラミング言語や開発環境を使用することが効率的であるとされている。

具体的には、Perl、Python、Rubyといったデータ処理を高速で行うことが可能なプログラミング言語で情報分析を行い、データベース作成のためのオープンソースであるMySQLなどを使用して情報ソースを格納する独自のデータベースの作成することである。しかしこれらの作業を行うには、別の形式のプログラミング言語や開発環境について学ぶ必要があり、開発者に対して負担を与えると考えられる。

そこで以下2つの方法を取ることで、情報視覚化ウェブコンテンツ作成に対する開発者の負担を軽減する。

- インターフェースのデザイン開発は、Processingのような情報表現の過程に優れているプログラミング言語、開発環境を利用する
- ウェブコンテンツ作成時に使用する情報ソースは、提案のフレームワークが提供する、すでに情報分析の過程が経たものを利用する

上記のように2つの研究内容を融合することで、更なる情報視覚化ウェブコンテンツの開発時に起こる作業負担の削減が期待できる。このように、情報視覚化ウェブコンテンツを開発するにあたり、情報表現にあたるインターフェースの開発においては、Processingのように簡単に表現、精密化、インタラクションの実装を行うことが可能なプログラミング言語、開発環境を使用する。また、情報分析にあたる情報の取得、解析、フィルタリング、マイニングに関しては、提案のフレームワークの提供する情報ソースのように、視覚化を行うための参照値を含んだも

のを使用することで、開発者自身が複数の開発作業を行う必要がなくなると考えられる。

情報視覚化ウェブコンテンツの開発過程における、先行研究の役割と提案のフレームワークの役割を、図 3.2 にまとめた。

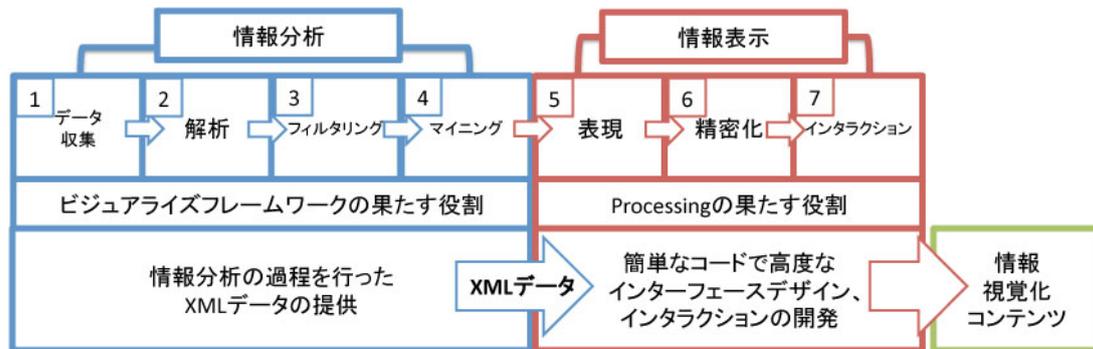


図 3.2 情報視覚化ウェブコンテンツ開発において各々の研究が果たす役割

第4章

ビジュアライズフレームワークの 提案

本章では情報視覚化ウェブコンテンツ作成のために、情報分析の過程を経た情報ソースをウェブコンテンツ開発者に提供するビジュアライズフレームワークの提案を行う。まず本フレームワークの果たす役割について述べ、情報分析を行うにあたっての機能について詳細を説明する。また、実際にどのような方法に基づいて情報ソースが生成され、その情報ソースを開発者がどのように使用するのか、例を交えて述べる。

4.1. ビジュアライズフレームワークの役割

本フレームワークの役割は、容易に情報視覚化ウェブコンテンツに作成するための情報分析の過程を経た情報ソースを、ウェブコンテンツ開発者に提供することである。この役割を果たすための一連の流れを、図 4.1 にまとめた。

以後、本フレームワークがどのように取得したデータを分析、情報ソースを生成、それらをウェブコンテンツ開発者に提供するかについて、具体的に述べる。

4.1.1 ビジュアライズフレームワークが行う情報分析

本フレームワークが行う情報分析において、以下の2つの目的をもつ。

- 情報を提供する API から取得したデータから必要な情報のみを抽出すること

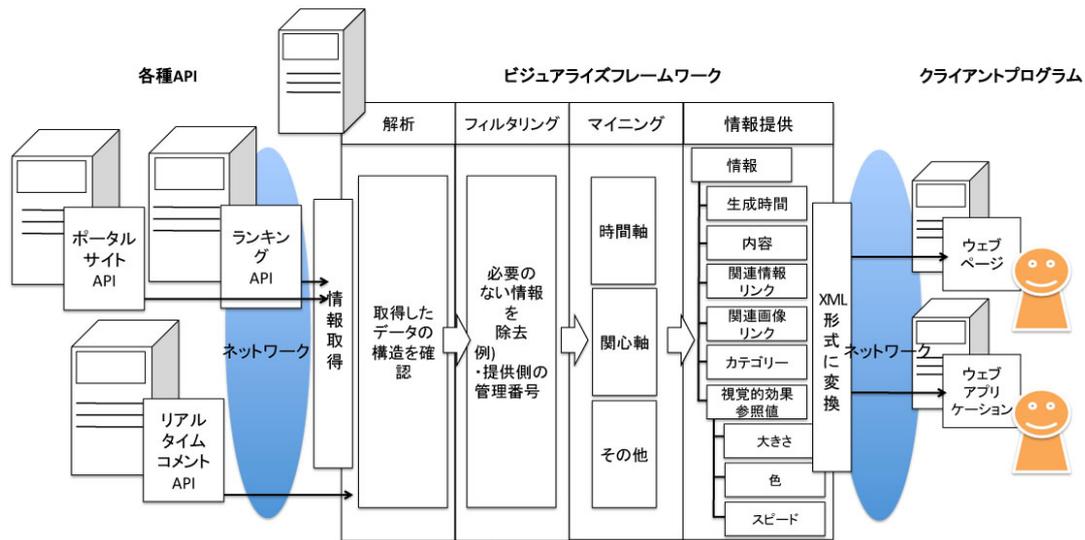


図 4.1 ビジュアルフレームワーク概念図

- 取得した情報の特性を視覚的効果で表す色や大きさなどの参照値をメタ情報として生成すること

この目的を果たすために本フレームは、前章の図 3.2 で表した情報分析の過程である、データの取得、解析、フィルタリング、マイニングの4つの段階を行う。

4.1.2 ビジュアルフレームワークが行う情報ソース提供

ビジュアルフレームワークでは、情報分析を経た情報ソースを XML 形式のデータに変換し、ネットワークを介して、ウェブコンテンツ開発者に提供する。XML(Extensive Markup Language)とは、文書やデータの意味や構造を記述するためのマークアップ言語である [17]。XMLは統一的な記述方式を用いながらも、開発者自身が独自の意味や構造をもたせることができるため [18]、本フレームワークの提供する情報ソースの内容に合わせた構造を作成することが可能である。

4.2. ビジュアライズフレームワークの機能

上記で明らかにした提案のフレームワークの役割を、実際の機能ではどのように行うのかについて述べる。今回機能を説明するにあたり、流行の情報を表すための情報ソースの生成を目的とした。

前述の Ben Fry 氏は著作「ビジュアライジング・データ:Processing による情報視覚化手法」、第 1 章「問いは何か?」において、大量の情報から何の情報を視覚化したいかを明確にすることの重要性について述べている。つまり必要とする内容によって分析方法は異なり、情報を表す性質においてもその目的に適したものをを用いるため、何を目的とした視覚化を行うのか、最初に開発者が明確に定義すべきであると説いている。

今回この定義に基づいて、ウェブ上の情報の特性の 1 つである即時性に着目し、流行のニュース情報を視覚化するための情報ソースの生成を目的とした。流行のニュース情報を生成するためには、どの時間帯の情報だと判断するための時間軸と、人々の関心の度合いである関心軸を用い、取得した情報を分析する必要があると考えた。提案のフレームワークを使用して流行のニュース情報を生成することで、本フレームワークのもつ機能を明確にする。

図 4.2 は、今後説明する機能の一連の流れをまとめたものである。以下、各機能の詳細について述べる。

4.2.1 情報取得

フレームワーク内の情報取得の段階における機能について述べる。本段階では情報取得として、目的に対してどのような情報ソースを使用すべきか、情報提供を行っている各 API の中からどの API を使用するのか、選定を行う必要がある。今回は時間と関心が含まれたニュース情報に関連するデータが必要になるため、以下 2 つの API を使用した。

- 新着のニュース情報を提供する API
- 新着のニュース情報におけるランキングを提供する API

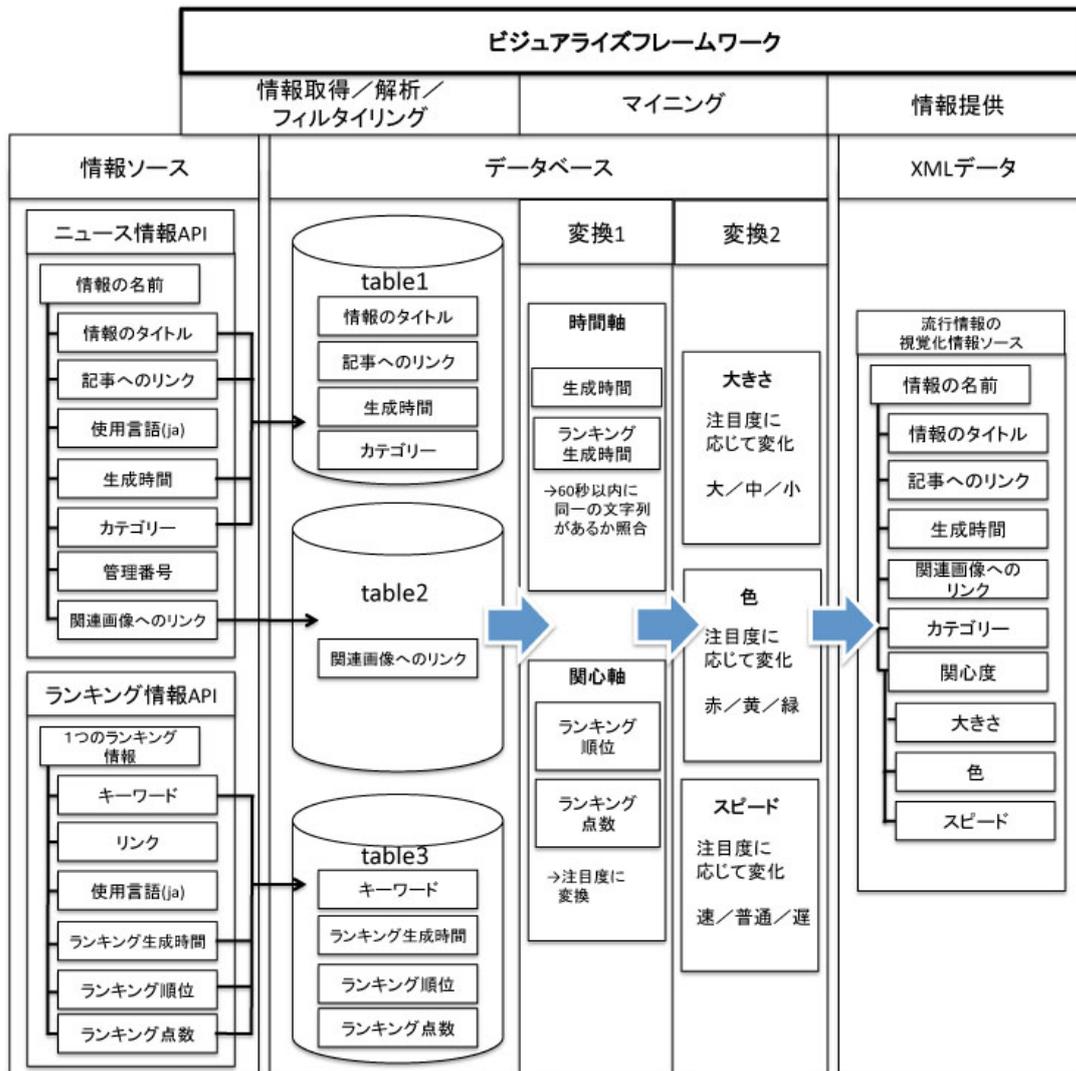


図 4.2 流行の情報に関する情報ソース生成の流れ

共に XML 形式のデジタル情報を配布している API であり、ネットワークを介して情報取得が可能である。それぞれの API が情報提供を行うために指定したエンドポイント URL に、フレームワークが HTTP リクエストを送り、HTTP レスポンスとして各データの取得を行う。

4.2.2 解析

フレームワーク内の解析の段階における機能について述べる。本段階では、情報取得の段階で各種 API から取得したデータの構造の解析を行う必要がある。今回は多くの API が採用している XML 形式のデータを取得したものとした。

一般的な API が提供するデータは、1つの情報を構成するメタ情報が含まれている。XML データでは開発者がそれらメタ情報の構造を決定することができるため、各 API によってメタ情報の構造が異なる。

そのため解析の機能として、これら異なるメタ情報は何の情報を意味しているのか理解する必要がある。今回は図 4.2 における、ニュース情報 API、ランキング情報 API が提供している XML データのメタ情報の構造を理解することが、解析における機能である。図 4.3 に、1つの XML データがどのように解析されるのかをまとめた。

4.2.3 フィルタリング

フレームワーク内のフィルタリングの段階における機能について述べる。本段階では、解析の段階で理解したデータのメタ情報の中から、流行のニュース情報を生成するにあたって必要ないと思われるメタ情報の除去を行う。今回は例として使用言語のメタ情報や、API 開発者が付与した管理番号のメタ情報を除去した。この過程を表したものが図 4.4 である。

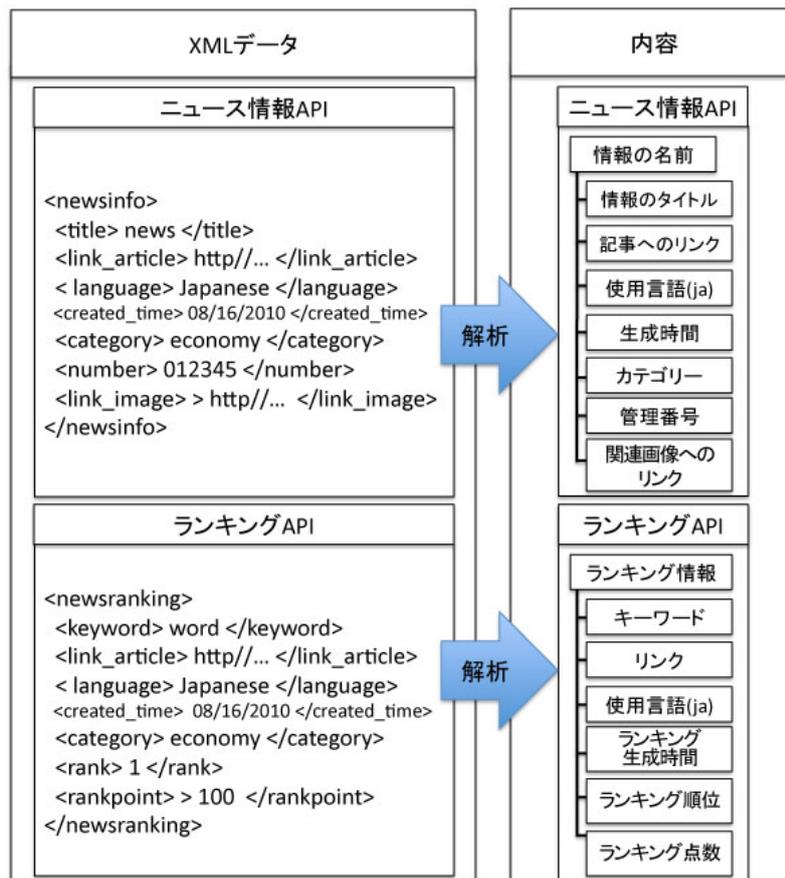


図 4.3 解析の機能

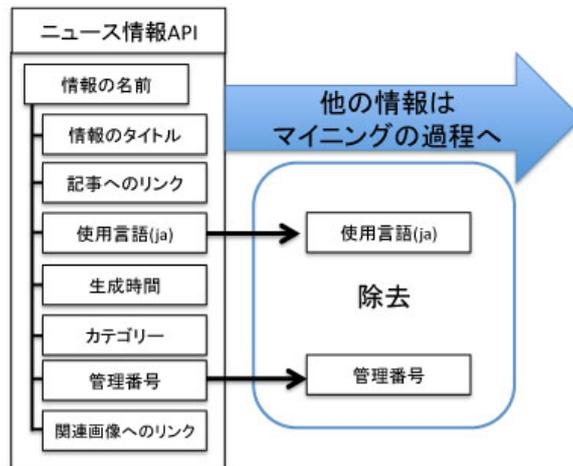


図 4.4 フィルタリングの機能

4.2.4 マイニング

フレームワーク内のマイニングの段階における機能について述べる。本段階では流行の情報として必要な情報の抽出、それらを大きさや色、スピードといった視覚的効果で表現するために必要な参照値を生成するために、これまで分析を行った情報を時間軸と関心軸によって変換する。

時間軸による変換

ニュース情報 API は、主に各通信社が配信する最新のデータを使用している。これらのデータ内にはニュースに対する重要度等、尺度に関するメタ情報は含まれていない。今回は流行のニュースの情報ソースを生成するため、ランキング API のデータを使用する。取得したニュース情報とランキング情報のデータに含まれている生成時間のメタ情報の照合し、一定時間内に同一の文字列があるか判別を行う。合致したものは流行の情報として抽出する。

関心軸による変換

時間軸によって抽出した流行の情報に対して、関心度による重み付けを行う。取得したランキング情報データ内に、ランキング順位に関するメタ情報と、ランキング点数に関するメタ情報が含まれているため、このメタ情報を使用して情報の関心度を数値化する。ランキング順位が高いものはその順位が高いほど、ニュー

スとして関心度の高いものと判断した。またそれに伴い、ランキング点数は数値が高いほど、ニュースとして関心度の高いものであると判断した。この関心度の度合いにより、流行のニュース情報を表す視覚的効果の参照値の具体的な数値を決定する。

視覚的参照値のメタ情報生成

今回流行のニュース情報を視覚的効果で表す参照値として、大きさ、色、スピードの変化を使用する。関心軸によって数値付けられた度合いを使い、各々の参照値に具体的な数値を付ける。またこれらの参照値は、人間の情報認知を参考にして付けられたものである。[19][20]。以下、それぞれの視覚的効果の参照値の詳細を述べる。

- 人間は大きいものから先に視覚的に認識するため、大きさの参照値では、関心度の高いものから大、中、小といった値を与える
- 人間は暖色から寒色にかけて誘目性が高いと認識するため、色の参照値では関心度の高いものから赤、黄、緑といった値を与える
- 人間は動きのあるものに対しスピードの早いものから認識するため、スピードの参照値では、関心度の高いものから速度が早い、普通、遅いといった値を与える

誘目性とは、色がもつ人の関心を惹きつける力を指す。図 4.5、4.6 は、関心度の変化を大きさで表した場合の例である。



図 4.5 大きさによる視覚化例

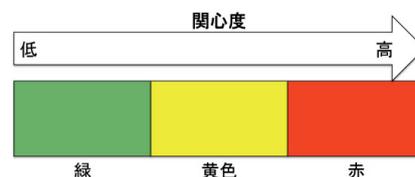


図 4.6 色による視覚化例

以上マイニングの段階により、流行のニュース情報として必要な情報と、その情報の性質を視覚的効果で表すための参照値を生成する。

これまでが本フレームワークにおける情報ソース生成までの一連の流れである。またこの流れは 60 秒ごとに繰り返され、取得データの内容に変更があった場合、情報ソースもまた最新の内容に変更される。

4.2.5 情報ソースの提供

これまでに生成したデータを流行の情報ソースとして配布する、フレームワーク内の提供の段階における機能について述べる。

流行の情報として抽出された情報、またその情報の特性を視覚的効果として表すための参照値を、情報ソースとして XML 形式のデータに変換する。図 4.7 は情報ソースの内容を、XML 形式に変換した一例である。

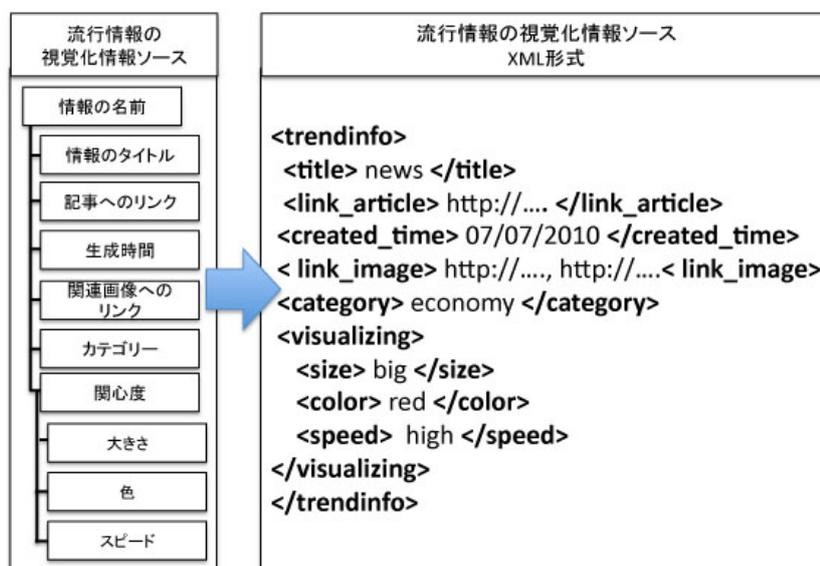


図 4.7 ビジュアライズフレームワークが提供する情報ソースの構造

XML 形式の情報ソースは、開発者にネットワークを通じて提供される。

具体的な配布の方法は REST[21] に準拠するものとする。ウェブコンテンツ開発者は、本フレームワークの指定したエンドポイント URL に対し HTTP リクエストを発行し、本フレームワークから返された HTTP レスポンスによりフレー

ムワークが提供する情報ソースを取得することが可能である。これら一連の流れを図4.8にまとめた。

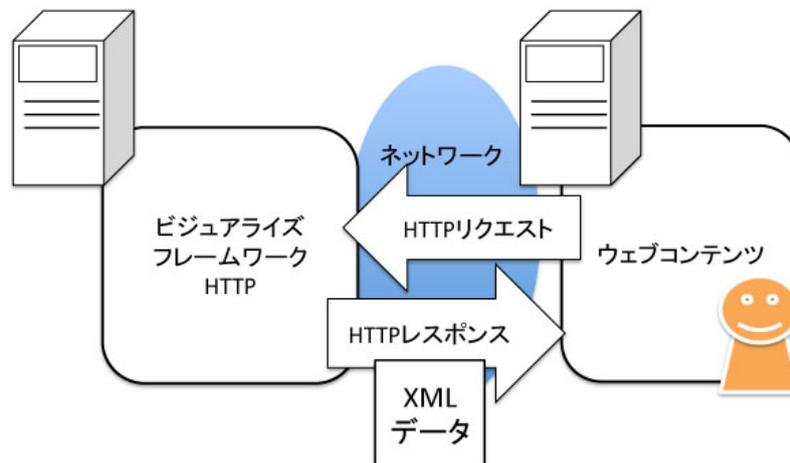


図 4.8 情報ソース取得の流れ

なお提供される情報ソースは 60 秒ごとに最新のものに切り替わるため、開発者はフレームワークに HTTP リクエストを送る度に、最新の流行のニュースの情報ソースを取得することが可能である。

4.3. 提供する情報ソースの使用方法

提案するフレームワークが提供する情報ソースの使用例について述べる。提供の情報ソースは以下 2 点の内容を含んでいる。

1. 対象とする情報を構成するメタ情報
2. 対象とする情報の性質を視覚的効果によって表すための参照値メタ情報

ウェブコンテンツ開発者がこの情報ソースに対し、どのような使用方法が可能か述べる。

4.3.1 参照値メタ情報使用例

今回流行のニュース情報の特性を表す視覚的効果の参照値として、大きさ、色、スピードの値を生成した。これらの値は、ウェブコンテンツのインターフェース上のデザインに使用されることを目的としている。具体的には、フォントサイズの変化や、インターフェース上に配置されたアイコン等のオブジェクトの変化に使用することが可能である。ウェブコンテンツ開発者は、開発時にこれらの参照値を使用することで、抽出された流行のニュース情報の表示を行うだけでなく、視覚的効果を使用して流行の情報の性質をインターフェース上で表現することができる。参照値のインターフェースへの活用例を、表 4.1 にまとめた。

表 4.1 参照値の使用例

参照値	インターフェースデザインの活用例
大きさ	フォントサイズ、オブジェクトの大きさの変化
色	フォント、背景、オブジェクトの色の变化
スピード	オブジェクトへのアニメーションの追加

4.3.2 情報視覚化ウェブコンテンツ開発例

提案のフレームワークが提供する情報ソースを使用し、どのような情報視覚化ウェブコンテンツが作成可能か例を述べる。

開発者は情報ソースを使用し、以下のようなウェブコンテンツが作成可能である。

- 文字情報を視覚化したウェブコンテンツ
- 参照値による変化を表したアニメーションをメインとしたウェブコンテンツ
- ゲーム要素を含んだウェブアプリケーション

このように、開発者はフレームワークの提供する情報ソースを使用して、情報を伝えるために文字を中心としたウェブコンテンツや、アニメーションを中心としたウェブコンテンツなど、目的に応じて多彩なウェブコンテンツの開発が可能になると考えられる。

4.3.3 発展例

また3章で述べた Processing のように、プログラミング初心者であっても大量の情報を扱った電子アート作品が作成可能な開発環境が登場したことで、Information Aesthetics[22][23] のように、情報伝達に芸術的な要素を兼ね備える研究分野が発達した。この分野は、情報視覚化技術と視覚化を目的とした電子アートの技術、各々の確立された研究内容から導き出された、データの精密さ、美しさ、インタラクション性の高さを基軸に情報表示を行うことを目的としている [24]。

これらの研究領域では開発時に、情報を視覚的に表すために必要な、本フレームワークが提供する視覚的効果を表す参照値を含むデータを作成する必要がある。また同時にそのデータを使用して、デザインの美しさと、機能性の高いインタラクションの開発を行わなくてはならない。しかしこれらの開発を行うには、デザイン技術とプログラミング技術の双方が必要となり、研究を行うにあたり障壁が高い。

しかし、本フレームワーク提供の情報ソースを活用することで、インターフェースのデザインや、インタラクションに対する開発過程に注力することが可能になると考えられる。

4.4. 本フレームワークが果たす役割

以上、提案するフレームワークが提供する情報ソースの生成の例、生成のための機能、配信のための機能について説明を行った。また、ウェブコンテンツ開発者が提供する情報ソースをどのように使用するか例を交えて述べた。

本フレームワークが提供する情報ソースを使用することで、開発者は情報分析の過程を行うことなく、情報視覚化ウェブコンテンツを作成することができる環

境を目指した。提案のフレームワーク、また提供する情報ソースが、情報視覚化ウェブコンテンツの開発にもたらすと期待する変化を図 4.9 にまとめた。

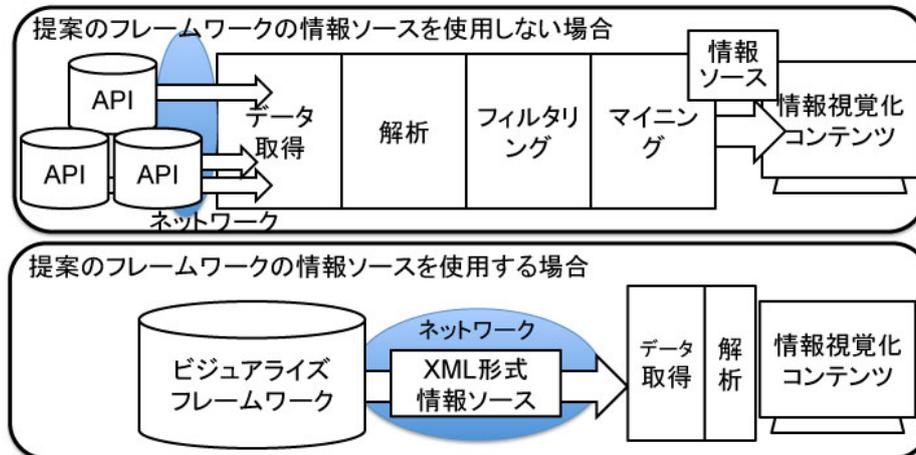


図 4.9 提案のフレームワークによる開発過程の変化

情報視覚化ウェブコンテンツを作成するにあたり、開発者は通常、複数の API からのデータ取得、解析、フィルタリング、マイニングを行う必要がある。しかし、提案のフレームワークを使用することで、フレームワーク提供の情報ソースを取得、解析し、必要なデータを使用してウェブコンテンツの開発を行うだけで良い。よって、提案のフレームワークを使用することで、簡潔な開発過程になると考えられる。

次に記す第 5 章によって、本フレームワークが提供する情報ソースが、本研究の目的である開発者の開発負担を軽減するか、また情報ソースがウェブコンテンツに対する負担を削減するののかについて評価を行う。

第5章

評 価

本章では、第4章にて提案したビジュアルライズフレームワークに対する評価について述べる。今回、実際に作成した3つの情報視覚化ウェブコンテンツを評価対象として使用し、それらの開発過程と提案のフレームワークが提供する情報ソースを使用した場合の開発過程の比較を行った。また各ウェブコンテンツが必要とするデータの処理と、本フレームワークが提供する情報ソースの使用の処理、各々がウェブコンテンツに与える負荷を比較した。

5.1. 評価概要

本評価において、慶應義塾大学メディアデザイン研究科で行われたリアルプロジェクトにて開発した、3つの情報視覚化ウェブコンテンツを使用する。

また定性評価については以下の2点の項目について行う。

1. 本フレームワークが提供する情報ソースが実現する、情報視覚化ウェブコンテンツの開発過程の変化の検証
2. 本フレームワークが提供する情報ソースが実現する、情報ソースが情報視覚化ウェブコンテンツに与える負荷軽減の検証

5.2. 比較対象

ここでは、3つの情報視覚化ウェブコンテンツの作成の流れ、必要なデータの分析方法、またデータを使用してインターフェース上にどのような視覚化を行っ

たかを説明する。

5.2.1 ウェブコンテンツ trend-setter

概要

本ウェブコンテンツはニュースサイトの利用者に対し、最新かつ関心度の高い流行のニュース情報を視覚化して提供することを目的に制作されたものである。

ニュースサイトでは、情報は生成時間の早いものからウェブページの上部に表示される場合が多い。多くの利用者はウェブページの上部から記事を閲覧する傾向があるため [25][26]、利用者の閲覧時間帯によって取得する情報の質が左右されてしまう。

本コンテンツでは、このような情報の質の偏りを防ぐため複数の API から取得したデータを使用する。最新のニュース情報のデータと、検索キーワードランキングのデータを使用し、生成時間による時間軸と、検索された回数による関心軸によって情報分析を行う。その結果必要な情報の抽出と、また流行の度合いとして大きさとスピードによる視覚化的効果の参照値を生成した。本コンテンツのシステム概要を図 5.1 に表した。また、本コンテンツの使用方法を付録 A にまとめた。

以下、本コンテンツの使用するデータの生成方法、またそのデータを使用し、インターフェース上でどのような視覚的表現を行ったかについて述べる。

使用データ生成

流行のニュース情報を提供するウェブコンテンツの作成のため、以下2つの API を使用した。

- Yahoo!トピック API[27]
- Yahoo!検索ランキング API[28]

これらの API から取得したデータに対し、情報分析の過程である解析、フィルタリング、マイニングの過程で抽出、変換を行う。この分析により、本コンテンツ

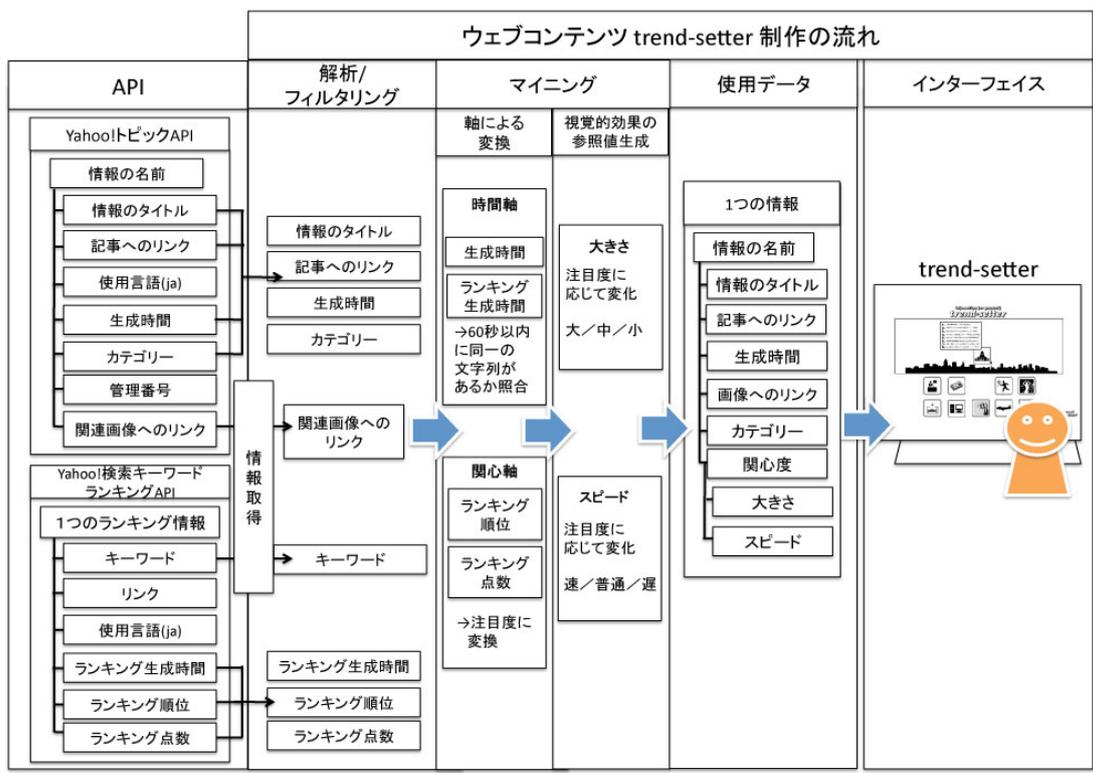


図 5.1 trend-setter システム図

に必要な流行のニュース情報と、その情報の性質を視覚的に表す大きさ、スピードの参照値を生成する。

また、このデータを生成するために API から取得した XML データのオブジェクト数は 110 つであった。また本コンテンツは 60 秒ごとに最新のデータを各 API から読み込むため、更新ごとにこのオブジェクト数を読み込み、情報分析を行わなくてはならない。

視覚化手法

本コンテンツにおいて、生成したデータをどのようにインターフェース上に視覚化したのかについて述べる。

今回情報の視覚的効果を行う参照値は、大きさとスピードである。これらの参照値を使用する対象物は、情報のカテゴリーを視覚化しインターフェース上には配置されたアイコンである。大きさの参照値はアイコンの大きさの変化に比例し、スピードの参照値はアイコンのジャンプアニメーションの飛距離に比例する。アイコンの大きさやジャンプの飛距離の変化が大きければ大きい程、そのニュース情報の関心度が高いことを意味する。

これらの変化によって関心度の度合いが視覚的に表現されるため、利用者は情報のもつ性質を直感的に理解することが可能である。図 5.2 に、視覚的効果の参照値がインターフェースに変換された際の表現をまとめた。

5.2.2 ウェブコンテンツ twit-viewer

概要

本ウェブコンテンツは、マイクロブログサービス Twitter[29] 利用者に対して、利用者が閲覧登録を行っている他利用者の更新情報を視覚化することを目的に制作されたものである。

利用者がリアルタイムコメントを発信することができる Twitter は、更新頻度が非常に高い。時間軸の早いコメントから順にインターフェースの上部に表示されていくため、利用者が必要なコメントの取得を逃してしまう可能性がある。

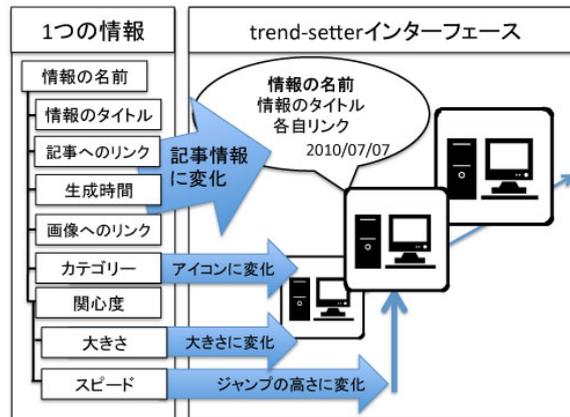


図 5.2 trend-setter 視覚化手法

本コンテンツでは、大量のコメントから利用者にとって必要なコメントを取得できるインターフェースの開発を目指した。本コンテンツのインターフェース上では、利用者が閲覧登録を行っている他の利用者はアイコン化される。利用者はコメントを取得したいと思う他利用者のアイコンのみを選択することで、その他利用者の最新コメントを表示することができる。本コンテンツでは更新頻度から生成された視覚的効果の参照値を含むデータを使用し、その参照値はアイコンに反映される。利用者はアイコンの変化によって、他利用者の更新状況を視覚的に確認することができる。

この機能によって、利用者が Twitter 上の大量のコメントから真に必要とするものだけの取得を可能になることを目指した。本コンテンツのシステム概要を図 5.3 に表した。また、本コンテンツの使用方法を付録 B にまとめた。

以下、本コンテンツに使用するデータの生成方法、またそのデータを使用して、インターフェース上でどのような視覚的効果を行うかについて述べる。

使用データ生成

まず本ウェブコンテンツの作成のため、Twitter 社がウェブコンテンツ開発者に提供している API に、「ユーザー情報」と「ステータス情報」を取得するための 2 つの HTTP リクエストを送る。API からの HTTP レスポンスにて各データ

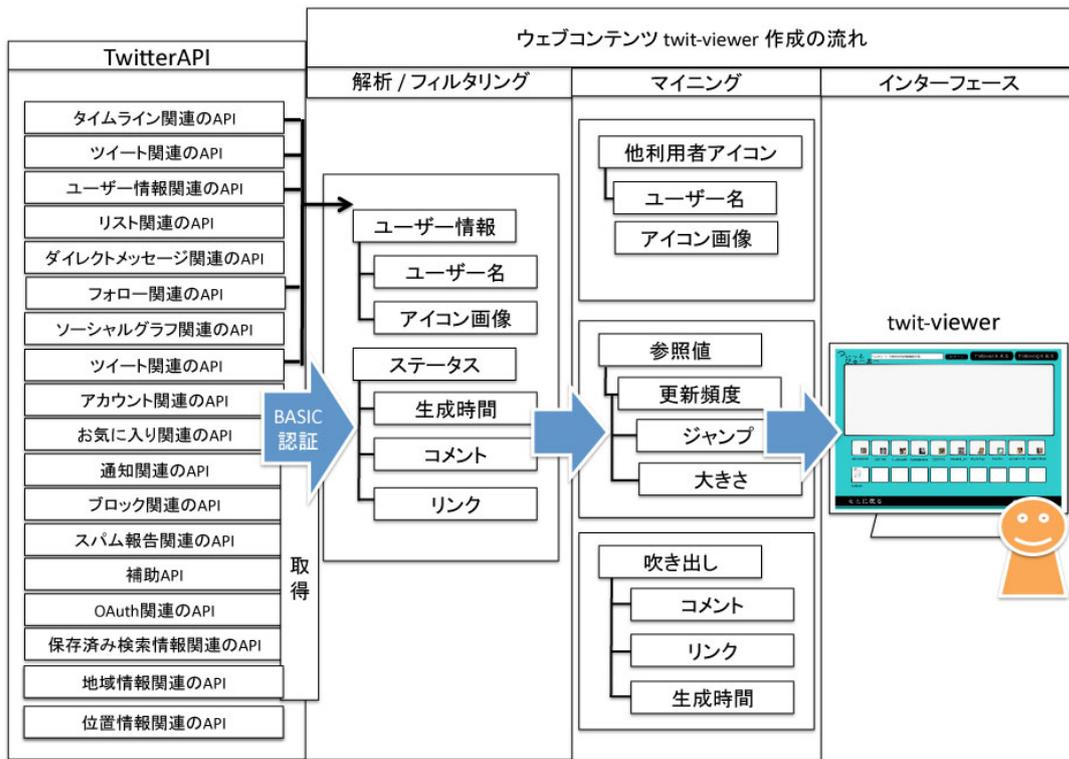


図 5.3 twit-viewer システム図

を取得する [30]。

取得した情報ソースを情報分析の過程である、解析、フィルタリング、マイニングによって、必要な情報の抽出、変換する。この分析によって、本コンテンツに必要な、利用者が閲覧登録を行っている他利用者のユーザー名、アイコン画像、コメント、コメントの生成時間を取得する。また、更新状況を視覚化するための大きさとスピードの参照値を、コメントの生成時間による更新頻度によって作成する。

またこの情報ソースを取得するために必要な XML データのオブジェクト数は、2つであった。しかし、TwitterAPI の提供している XML データは、その構造が複雑であり、1つのコメントごとに 25 個のメタ情報から成り立った XML データを取得する必要がある。また本コンテンツは最新の更新状況を確認するために、10 秒ごとに最新の情報ソースを TwitterAPI から読み込むため、更新ごとにこの構造から成り立つオブジェクト数を読み込み、情報分析を行わなくてはならない。

視覚化手法

本コンテンツにおいて、生成したデータをインターフェース上でどのように視覚化したか述べる。

今回更新状況の視覚的効果を表す参照値は、大きさとスピードである。これらの参照値を使用する対象物は、本コンテンツのインターフェース上に設置されたアイコンである。このアイコンはデータから取得した、利用者が閲覧登録した他利用者の、ユーザー名、アイコン画像から作成されたものである。利用者はこのアイコンの動きによって、視覚的に他の利用者のコメントの更新情報を理解することができる。大きさとスピードの参照値は、他利用者の更新頻度によって変化し、更新頻度が高い場合に大きさとスピードの参照値は高くなり、頻度が低い場合に参照値は低くなる。参照値が高い場合、大きさの参照値によってアイコンのサイズは大きくなり、またスピードの参照値はジャンプアニメーションの飛距離となり、アイコンは高くジャンプする。更新頻度が低い場合は大きさとスピードの参照値が低くなるため、アイコンのサイズは小さくなり、またジャンプアニメーションは起こらない。

利用者はこれらアイコンの変化という視覚的効果によって、他の利用者のコメントの更新状況を理解することができる。図5.4に、生成したデータがインターフェース上でどのように視覚化されるかまとめた。

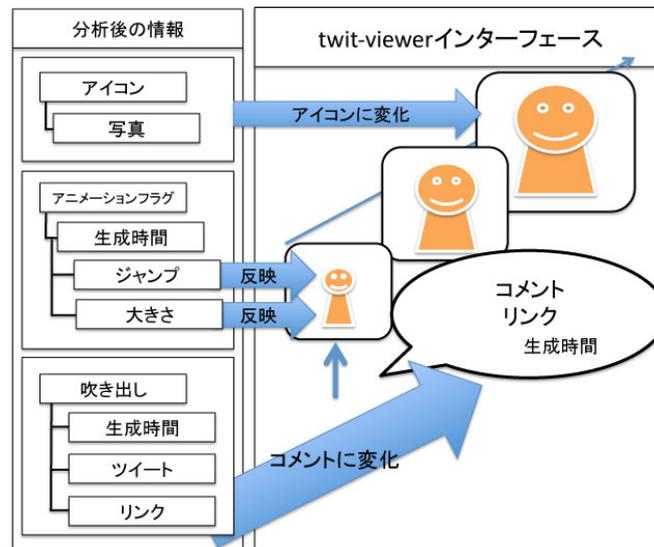


図 5.4 twit-viewer 視覚化手法

5.2.3 ウェブコンテンツ fu*life

概要

本ウェブコンテンツは、マイクロブログサービス Twitter の利用者、主に小学校低学年の女兒に対して、花を基軸としたデザインがなされたインターフェースを提供する Twitter クライアントである。本論文におけるクライアントとは、サービスの管理をより機能的に使用するためのソフトウェアを指す。インターフェース上では、利用者と利用者が閲覧登録を行った他利用者のコメントは花として視覚化される。

本コンテンツは、インターネット使用率が年々高まっており、SNS やチャットなどのウェブコンテンツに触れる機会が高い小学生に向けて [31]、花というデザインを取り入れることで Twitter の使用を促すことを目的に制作された。

Twitter のインターフェースは、9割が文字情報で構成されているため、対象者にとって親しみ易いものではない。そこで、以下の機能を実装した。

- インターフェースに表示されるコメントをアニメーションが付加された花で表示する
- 閲覧登録を行った他利用者と交わした返信コメント (Twitter におけるリツイート機能) のやり取りが花のブーケに変化する

これらの機能によって、対象者が Twitter に対して親しみをもって使用してもらうことを期待した。本コンテンツのシステム概要を図 5.5 にまとめた。また、本コンテンツの使用方法を付録 C にまとめた。

以下、本コンテンツに必要なデータの生成方法、またそのデータを使用し、インターフェース上でどのような視覚的効果を行うかについて述べる。

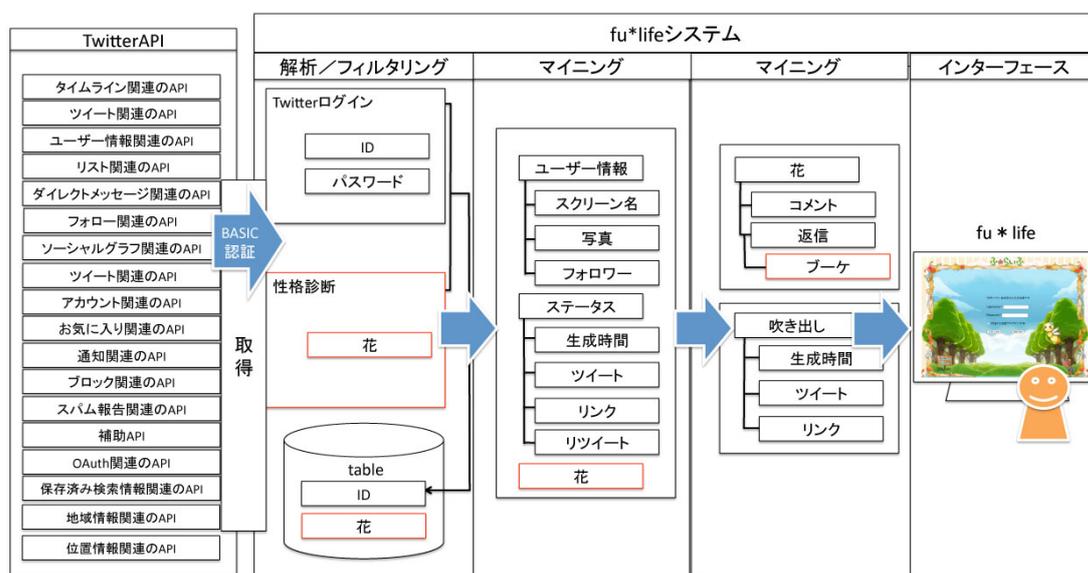


図 5.5 fu*life システム図

使用データ生成

本コンテンツの作成のため、Twitter 社が開発者に提供している API に、「ユーザー情報」と「ステータス情報」を取得するための 2 つの HTTP リクエストを送

る。その後 API から送られる HTTP レスポンスにて各データを取得する。

取得した情報を情報分析の過程である、解析、フィルタリング、マイニングの過程で抽出、変換を行う。

この情報分析から、利用者と利用者が閲覧登録をしている他利用者の、ユーザー名、アイコン画像、コメント情報、返信コメント情報を必要なデータとして抽出した。また返信コメント情報を視覚化するため、返信コメントが行われたことがデータから読み取れた場合、それをブーケ作成のための参照値として生成する。

またこのデータを生成するために API から取得した XML データのオブジェクト数は4つであった。しかしこれらを取得するためには、オブジェクト数1つあたり25個のメタ情報から成り立った HTTP レスポンスを受け取らなくてはならない。また本コンテンツは15秒ごとに最新の情報ソースを API から読み込むため、更新ごとにこのメタ情報から成り立つオブジェクトを読み込み、情報分析を行わなくてはならない。

視覚化手法

本コンテンツにおいて、生成したデータを使用してインターフェース上でどのような視覚化が行われたのか述べる。

今回情報を視覚化するために使用する参照値は、コメント情報、利用者と他利用者による返信コメント情報の2つである。コメント情報は、本コンテンツによって各利用者に割り振られた花に変化する。また Twitter のインターフェース上には、他の利用者のコメントに対して返信の機能がある。この機能が使われたコメントのやり取りは、花が集まったブーケとして表示する。返信コメント情報の参照値が1以上の場合は、インターフェース上にブーケ作成ボタンが出現するものとした。このボタンを押すとコメントの花はブーケに変化する。

図 5.6 に、生成したデータがインターフェース上でどのように視覚化されるかまとめた。

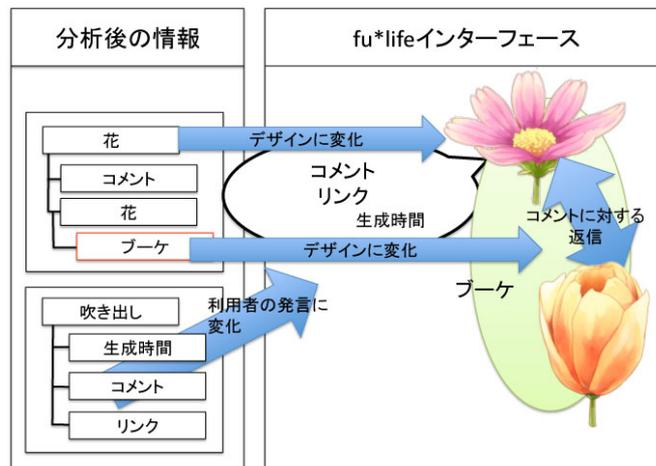


図 5.6 fu*life 情報視覚化手法

5.3. 結果

これまでに述べた3つの情報視覚化ウェブコンテンツの開発過程と、提案するフレームワークが提供する情報ソースを使用した場合の開発過程の比較を行う。提案のフレームワークを使用して、3つのウェブコンテンツに必要なデータを情報ソースとして生成した。また、提案のフレームワークが提供する情報ソースが各ウェブコンテンツに与える負荷軽減について述べる。

5.3.1 開発過程の比較

比較を行うために、上述の3つのウェブコンテンツの開発過程の抽象化を行った。抽象化された過程は、以下5つの段階から成り立つ。

1. 情報収集
2. 情報解析
3. 情報のフィルタリング
4. 情報のマイニング

5. 表現、精密化、インタラクションによるインターフェースのデザイン

ウェブコンテンツ作成において、提案のフレームワークの提供する情報ソースを使用した場合は、以下3段階の過程に抽象化できると考えられる。

1. 情報収集
2. 情報解析
3. 表現、精密化、インタラクションによるインターフェースのデザイン

比較の結果、提案のフレームワークを使用すると、2段階の過程を短縮することができるが確認された。つまり情報分析における、フィルタリングとマイニングの段階の短縮を確認することができた。また情報表示の過程である表現・精密化にて使用する参照値が情報ソースに含まれていることから、表現、精密化における開発段階を短縮することが可能であると考えられる。図5.7、5.8に3つの情報視覚化ウェブコンテンツ開発過程、また提案のフレームの提供する情報ソースを使用した場合のそれぞれの過程を抽象化したものをまとめた。

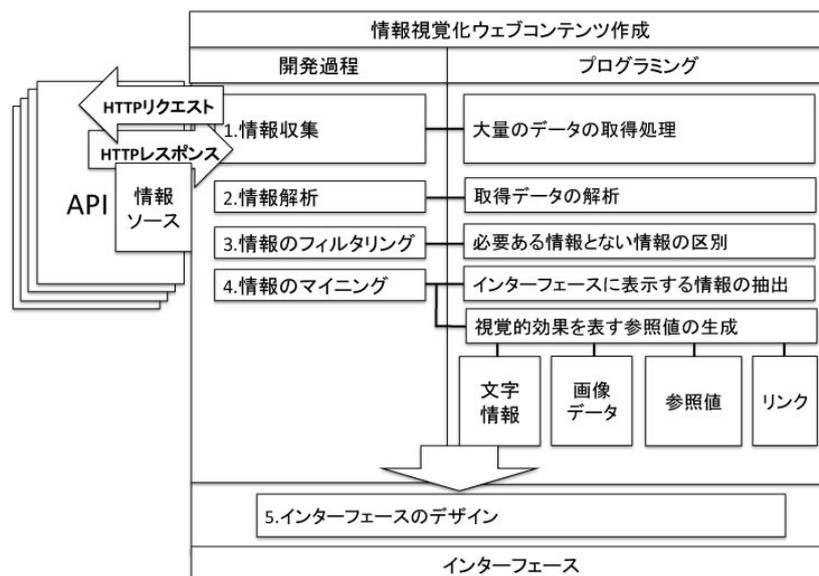


図 5.7 一般的な情報視覚化ウェブコンテンツ開発過程

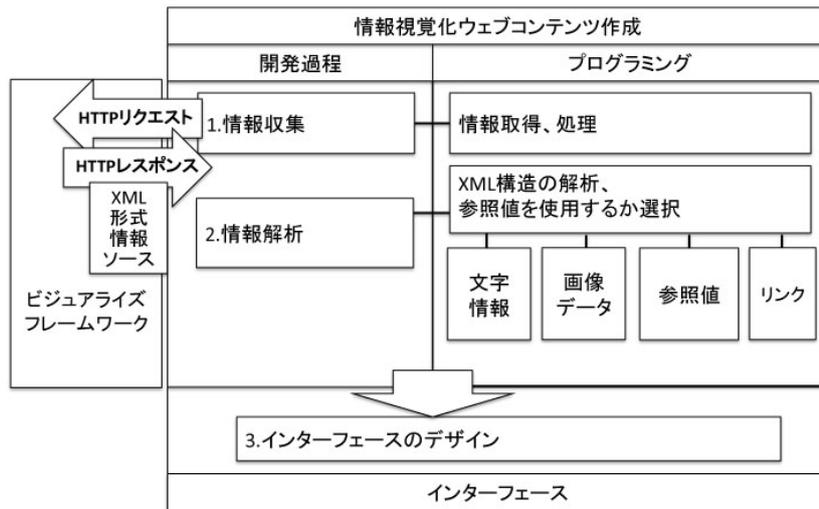


図 5.8 本フレームワークを使用した情報視覚化ウェブコンテンツ開発過程

情報収集の過程に関しては、一般的な情報視覚化ウェブコンテンツ開発過程、本フレームワークを使用した場合の開発過程の双方において行わなくてはならないが、複数の API を使用する場合に利点が生じる。第 4 章で述べたように、XML 形式のデータは制作者自身がその構造を制作することができるため、各 API は独自の XML フォーマットを使用していることが多い。そのためウェブコンテンツ開発者は、使用する API が増えるごとにその API の構造を理解する必要がある。しかし提供の情報ソースを使用すれば、1 つの XML 構造の理解のみで良い。また、各 API で起こり得る仕様や規格の変更に対応する必要がなくなる。このように過程が短縮されるだけでなく、必要な過程の内容においても利点が生じると考えられる。

フィルタリングの過程に関しては、一般的な情報視覚化ウェブコンテンツ開発過程においては、取得したデータ全てのメタ情報から必要なもの、必要でないものを選別する必要があるが、本フレームワークを使用した場合は目的のデータのみで構成されているためフィルタリングを行う必要はない。

マイニングの過程に関しては、フレームワークの提供する情報ソースには必要な情報とその性質を表す視覚的効果の参照値が含まれているため、情報視覚化ウエ

ブコンテンツ開発過程においては同様に行う必要がない。

5.3.2 取り扱いデータに対する負荷の比較

情報ソースの量は、情報視覚化ウェブコンテンツの開発にて行われる情報取得、その後の情報分析の過程において多大な影響を与える。その取得量が多いほど、複雑な情報分析を行わなくてはならないからである。また情報視覚化ウェブコンテンツは変動する情報に対応しなくてはならないため、他ウェブコンテンツと比べ、情報取得は比較的短い間隔で繰り返し行われる。上記の理由から情報視覚化ウェブコンテンツにおいて、取得するデータの量はコンテンツの動作の安定化をはかるために重要であると考えられる。

以下の表に、開発した3つの情報視覚化ウェブコンテンツ開発において取得する必要がある情報量と、情報分析の過程を経た情報視覚化のために必要な情報量をまとめた。なお数値は、取得したXMLデータを構成しているメタ情報の数値である。

表 5.1 取得した情報量と必要な情報量

	取得したメタ情報数 (個)	必要なメタ情報数 (個)
trend-setter	1210	990
twit-viewer	50	10
fu*life	100	40

提案のフレームワークを使用することで、trend-setter は 220 個、twit-viewer は 40 個、fu*life は 60 個のメタ情報を削減されることが確認できた。trend-setter は 60 秒ごとに、また twit-viewer は 10 秒ごとに、fu*life は 15 秒ごとに情報取得を行うため、1 度の取得行為において上記の表に記した取得したメタ情報数だけ処理を行わなくてはならない。

また図 5.9 に、trend-setter におけるデータ取得に必要なプログラムのスケルトンコードを記す。本コンテンツでは 10 種類の各カテゴリーの最新ニュースデー

```

/* trend-setter に必要なデータの取得 */
load http://dailynews.yahoo.co.jp/fc/economy/rss.xml(経済ニュース提供 URL)
load http://headlines.yahoo.co.jp/rss/sh_mon_bus.xml(金融ニュース提供 URL)
load http://seiji.yahoo.co.jp/column/article/rss.xml(政治ニュース提供 URL)
load http://headlines.yahoo.co.jp/rss/spnavi.xml(スポーツニュース提供 URL)
load http://dailynews.yahoo.co.jp/fc/entertainment/rss.xml
(エンタテインメントニュース提供 URL)
load http://headlines.yahoo.co.jp/rss/cbn_dom.xml(医療ニュース提供 URL)
load http://dailynews.yahoo.co.jp/fc/computer/rss.xml
(テクノロジーニュース提供 URL)
load http://dailynews.yahoo.co.jp/fc/science/rss.xml(環境ニュース提供 URL)
load http://dailynews.yahoo.co.jp/fc/local/rss.xml(交通情報ニュース提供 URL)
load http://weather.yahoo.co.jp/weather/promo/rss.html
(天気ニュース提供 URL)
load http://searchranking.yahoo.co.jp/rss/total_ranking-general-rss.xml
(検索ランキング提供 URL)

```

図 5.9 trend-setter に必要なデータの取得

タ、また検索ランキングのデータを取得する必要がある。データを取得するためには、上記のように各データ配布を行っている 11 つの各エンドポイント URL に HTTP リクエストを送る必要がある。

また図 5.10 に、提案のフレームワークから提供される情報ソース取得時のプログラムのスケルトンコードを記す。

```

/* 提案のフレームワークからのデータ取得 */
load http://infomationvisualization.com/trend.xml

```

図 5.10 trend-setter に必要な情報ソースの取得

提案のフレームワークの提供する情報ソースには、各カテゴリーの情報はすでに統一された XML データとして配布されるため、開発者は HTTP リクエストを 1 回のみ発行するだけで、データ取得を行うことができる。

また、上記で行った HTTP リクエストにより取得した、通常のコンテンツ開発

時に必要なデータとフレームワークが提供する情報ソースの XML 構造のサンプルを図 5.11 にまとめた。

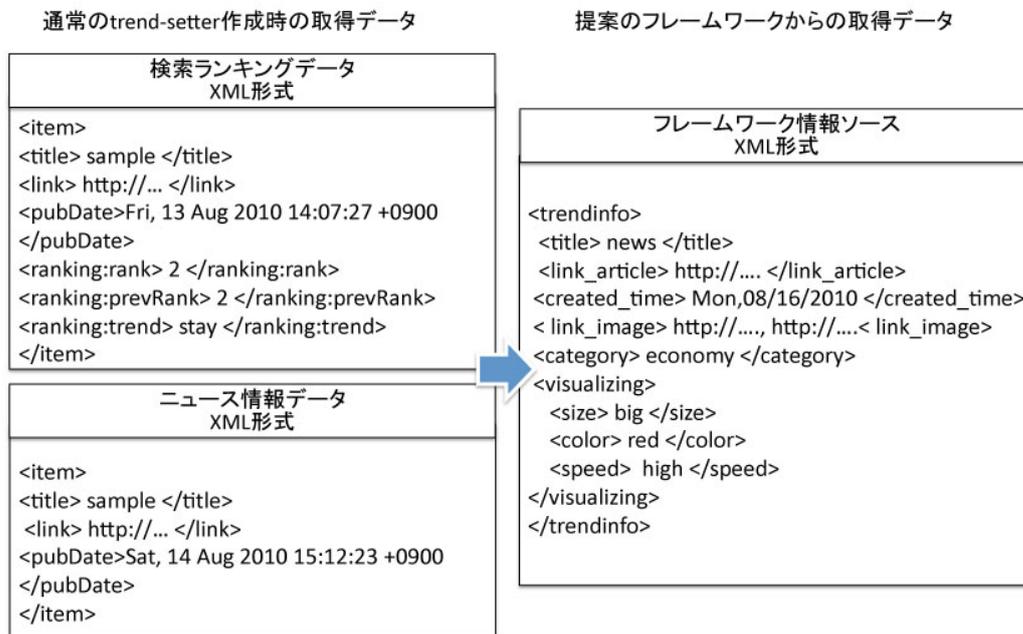


図 5.11 XML 形式の比較

今回は1つの提供先からデータの取得を行っているため大きな XML 構造の違いはないが、他の API を使用した場合、生成時間やタイトルといった共通の内容であってもタグや表記方法が異なる場合が多い。本フレームワークを使用することにより、これら各 API がもつタグや表記方法は統一され、XML 構造の違いに対応する必要はなくなるという利点が挙げられる。

上記の観点からも同様に、提供の情報ソースを使用することによるデータ取得時の負荷軽減が可能になると考えられる。

第6章 考 察

本章では第5章の結果に基づいて、提案するフレームワークが提供する情報ソースが情報視覚化ウェブコンテンツの開発に与える影響について考察を行う。

以下、提案のフレームワークの提供する情報ソースを使用することで、ウェブコンテンツ開発者に対する開発過程の短縮に関する有用性、またコンテンツに対する負荷の軽減がどのように行われるかについて述べる。

6.1. 開発過程の変化

第5章において、提案のフレームワークが提供する情報ソースによって、情報視覚化ウェブコンテンツの開発過程の削減が可能であることが確認された。今回検証によって2過程の削減が確認されたが、この2過程を行うにあたり、ウェブコンテンツ開発者が行わなくてはならない作業を以下にまとめた。

- 2過程の具体的な設計
- 設計後のプロトタイピングの作成、評価
- 設計確定後の実装

これら作業を行う必要性がなくなるため、提案のフレームワークが提供する情報ソースは、開発者にとって開発過程の負担の軽減をもたらすと考えられる。

6.2. 情報処理の変化

同じように第5章において、提案のフレームワークが提供する情報ソースによる情報視覚化ウェブコンテンツに対する負荷削減が確認された。

情報視覚化ウェブコンテンツは変動する情報への対応が求められるため、開発過程において、大量の情報処理に耐え得る構造の設計を行う必要がある。

しかし、提案のフレームワークの提供する情報ソースは、必要であると判断された情報と視覚的効果を行うための参照値で構成されているため、ウェブコンテンツ内で全ての情報を処理する必要がなくなる。複数のAPIからの情報取得を行わなければならない情報視覚化ウェブコンテンツにおいて、提案のフレームワークが提供する情報ソースの使用は有用性が高いと考えられる。

この結果複数のAPIから情報取得と比べ、ウェブコンテンツ作動に対する安定化への担保が期待できる。

6.3. 今後

提案のフレームワークの更なる有用性を追求するために、今後取り組むべき課題について述べる。

6.3.1 取り扱いデータの拡充

今回提案のフレームワークが情報取得するために使用したAPIは、ニュース情報API、検索ランキング情報API、TwitterAPIであった。今後対応のAPIを増やすことにより、情報ソースの内容としての充実、また種類の拡充を目指す。

第4章では流行の情報ソースの生成方法について述べたが、Google Map API[32][33]や、foursquare API[34]のような土地情報や位置情報を提供するAPIを組み合わせることによって、更に使用用途の広がりをもった流行の情報ソース生成の可能性が[35]。

また今後、気象情報や人間の行動履歴など、様々な種類のセンサ情報のネットワーク上での増加が考えられる。このようなAPIのように体系は確立されておら

ず、しかし有益な情報ソースに成り得るデータに関しての取り組む必要性が考えられる。フレームワークのセンサ情報の取り扱いを可能にすることにより、Social Visualization[36]のような、人間の生活情報の視覚化に対する情報ソースの提供が可能になると考えられる。

6.3.2 情報分析の強化

上述のように API の取り扱いの増加やセンサ情報の対応により、形式の異なるデータを取り扱う必要がある。

この結果、第4章で述べたフレームワーク内のマイニングの段階で使用した時間軸や関心軸のような、情報分析を行うための軸の拡充が必要になると考えられる。

今後、新たな軸の作成に取り組むため、ニューラルネットや決定木、自己組織化マップや関連規則、クラスター分析やサポートベクターマシン、ベイジアンネットワークや潜在意味解析 [37] などデータマイニングの手法として使用されている解析方法の使用を検討する。

6.3.3 利用者に対する利便性の担保

今回本フレームワークの提供する情報ソースの利用者が筆者のみであったため、情報ソースの取得に対する利便性の担保に対して課題が残る。以下、具体的な課題についてまとめた。

1. 情報ソース取得時のフレームワークの安定性
2. GUI(Graphical User Interface) への移行の検討

1点目において、現在提案のフレームワークの情報ソースの配布は HTTP リクエスト、レスポンスを介して行われているが、多数のウェブコンテンツ開発者からリクエストを受けた場合のフレームワークの安定性に関する評価は行われてい

ない。今後、実際に使用される場合の想定環境に基づいた検証が必要だと考えられる。

2点目において、ウェブコンテンツ開発者の本フレームワークの情報ソースを取得にあたり、GUIの導入を検討している。現在必要な情報ソースの取得に対して、それぞれ異なるエンドポイント URL にアクセスする必要があるが、GUIの導入にあたって、ウェブコンテンツ開発者は自身で必要な情報ソースの選択を視覚的に行うことが可能になる。また今後提供する情報ソースの種類増加や、ウェブコンテンツ開発者による情報ソースのカスタマイズが実現された際、GUIによって開発者に対する更なる利便性が担保が期待できる。以下、図 6.1、図 6.2 に本フレームワークにおける GUI の画面イメージをまとめた。

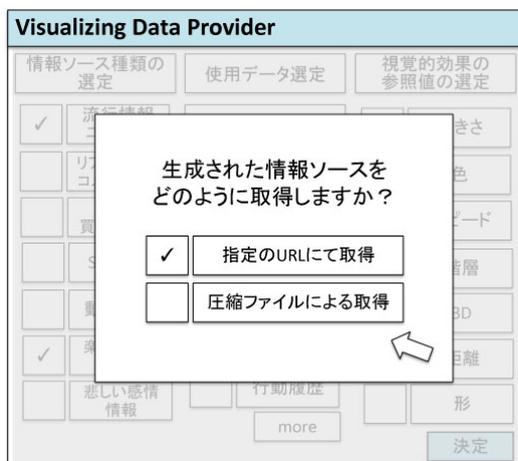
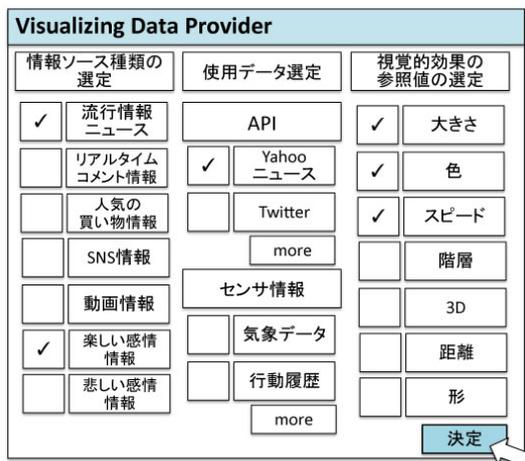


図 6.1 生成情報ソースカスタマイズ画面 図 6.2 生成情報ソース取得画面

このように、開発者は取得したいと思われる情報ソースの選択、そのソースの使用データの選択、また使用したい視覚的効果の参照値のカスタマイズを行うことができる GUI の作成を目指す。この機能より、本フレームワークの提供する情報ソースの拡張性の追加を行うことが可能になると考えられる。

また、その後 HTTP リクエスト/レスポンスによる取得、圧縮データによる取得など、データ取得に関するユーザビリティへの対応を目指す。

6.4. まとめ

本研究では、情報爆発時代において、利用者にとって必要な情報取得が困難な状況にあることを問題とし、大量の情報から必要な情報の抽出と視覚的な効果を行う情報視覚化技術をウェブコンテンツに適応させることで状況の改善を目指した。

しかし、情報視覚化ウェブコンテンツ作成のためには、複数の API から取得した大量の情報の処理、その情報から必要な情報の抽出を行うといった、情報分析という複雑な開発過程を必要とする。これらの過程はウェブコンテンツ開発者への負担になると考えられる。

そこで本研究では情報視覚化を行うための情報ソースを提供するためのビジュアライズフレームワークの提案を行った。第5章にて行った評価から、本フレームワークの有効性が確認できた。また、更なる有効性の確立のため、本フレームワークが取り組むべき課題についてまとめた。

第7章

おわりに

本研究では、情報過多によって利用者が必要な情報の取得が困難な現状を改善すべく、情報視覚化技術のウェブコンテンツへの適応を目指した。

情報視覚化は変動する大量の情報を取得、必要な情報を抽出し、その情報の性質を視覚的効果で表すことで、情報取得を容易にする技術である。現在情報視覚化を行う環境は以下2点で進化しつつある。情報視覚化に必要な情報ソースの取得は、様々なAPIの公開によって容易に行うことが可能となった。情報を視覚的にインターフェースの表現に関する技術は、ソフトウェア、プログラミング言語、ブラウザの進化によって、総合的に容易になっている。しかし、取得した情報を変換し、視覚化のための参照値を生成する過程である、解析、フィルタリング、マイニングといった情報分析を行うには、未だ専門的な知識を必要とする。そのため、情報視覚化ウェブコンテンツの開発はウェブコンテンツ開発者にとって時に負担になる。

そこで、本研究では視覚化のための情報分析が行われた情報ソースを開発者に提供する、ビジュアライズフレームワークの提案を行った。提案のフレームワークでは、利用者にとって必要な情報を取得した情報を提供するために、複数の情報ソースから大量の情報を取得する。それらを時間軸や関心軸などの複数軸によって抽出、変換し、更に色や大きさなどの視覚的効果を行うための参照値を生成する。最終的にこれらをXML形式に変換し、インターネットを介して開発者に提供する。この情報ソースを使用することによって、開発者は開発の過程を増やすことなく情報視覚化ウェブコンテンツの作成を行うこと可能になる。

提案のフレームワークの有用性を検証するために以下の評価を行った。実際に開発した3つの情報視覚化ウェブコンテンツの開発過程と、提案のフレームワー

クを使用した場合のコンテンツの開発過程の比較を行った。また、フレームワークが提供する情報ソースを使用することによる、ウェブコンテンツに与える負担軽減をはかった。その結果、提案のフレームワークを使用することで、2つの開発過程の短縮が確認された。また、提供の情報ソースはコンテンツ作成において必要なメタ情報で構成されているため、情報ソースを使用した場合と、各データをコンテンツ内で処理した場合と比べ、負荷が軽減されたことがわかった。

今後の課題として、取得データの種類の拡充、それに伴う情報分析の確立が必要であると考えた。また、開発者が情報ソースのカスタマイズを行うことができる GUI の対応に取り組む。

謝 辞

本研究の指導教員であり、幅広い知見からの確な指導とご指摘や暖かい励ましをしていただきました主査である慶應義塾大学大学院メディアデザイン研究科教授 砂原秀樹氏に心から感謝いたします。また、副査である慶應義塾大学大学院メディアデザイン研究科教授 稲蔭正彦氏に感謝いたします。研究指導や論文執筆など数多くの助言を賜りました慶應義塾大学大学院メディアデザイン研究科教授 加藤朗氏に心から感謝いたします。

本研究を進めていく上で様々な励ましと助言、お手伝いをいただきました、慶應義塾大学大学院メディアデザイン研究科後期博士課程 山内正人氏、廣海緑里氏、修士課程 廣井慧氏に感謝いたします。また研究活動を支えていただいた、慶應義塾大学大学院メディアデザイン研究科 NetworkMediaProject の皆様に感謝いたします。また時に苦楽を共にした Nancy Lan-lan Ma、Pianpian Zhou、慶應義塾大学大学院メディアデザイン研究科 2008 年秋入学の全ての同窓生に感謝いたします。

最後に、病気にも関わらず経済面、生活面で常に応援し続けてくれた父親、精神面で常に支えてくれた母親、常に励まし続けてくれた兄に心から感謝いたします。常に癒しを与えてくれた愛猫マーシュと愛犬ミミィに感謝します。また、大学院で学ぶことに深い理解と多大な支援をしてくれた亡き祖父に感謝いたします。

全ての方の温かいご支援がなければ、本論文を完成させることは不可能でした。心から感謝いたします。

以上をもって謝辞といたします。

参 考 文 献

- [1] 情報流通インデックス研究会報告書. http://www.soumu.go.jp/main_content/000030652.pdf.
- [2] 秋山龍平. 情報大爆発—コミュニケーション・デザインはどう変わるか. 宣伝会議, 2007.
- [3] K. Ellison. Too much information! *Frontiers in Ecology and the Environment*, Vol. 7, No. 2, pp. 116–116, 2009.
- [4] J.F. Gantz, C. Chute, A. Manfrediz, S. Minton, D. Reinsel, W. Schlichting, and A. Toncheva. The diverse and exploding digital universe: An updated forecast of worldwide information growth through 2011. *IDC white paper, sponsored by EMC*, 2008.
- [5] 喜連川優, 松岡聡, 松山隆司, 須藤修, 安達淳. 情報爆発時代に向けた新しい IT 基盤技術の研究. *人工知能学会誌*, Vol. 22, No. 2, pp. 209–214, 2007.
- [6] Ben Fry. ビジュアルライジング・データ:Processing による情報視覚化手法. O'Reilly Japan, 2008.
- [7] 増井俊之. 情報視覚化の最近の研究動向. 電子情報通信学会第 9 回データ工学ワークショップ, 1998.
- [8] 増井俊之. インターフェイスの街角本当に使いやすいユーザー・インターフェイスの極意. *インターフェイスの街角*, 2005.
- [9] 情報大航海プロジェクト. http://www.meti.go.jp/policy/it_policy/daikoukai/index.html.

- [10] Yahoo デベロッパーズ API. <http://developer.yahoo.co.jp/>.
- [11] Google Code. <http://code.google.com/intl/ja/>.
- [12] Benjamin Jotham Fry. *Computational information design*. PhD thesis, 2004. Supervisor-Maeda, John.
- [13] 元田浩, 津本周作, 山口高平, 沼尾正行. データマイニングの基礎. オーム社, 2006.
- [14] J. Han and M. Kamber. *Data mining: concepts and techniques*. Morgan Kaufmann, 2006.
- [15] C. Reas and B. Fry. *Processing: a programming handbook for visual designers and artists*. The MIT Press, 2007.
- [16] Processing.org. <http://processing.org/>.
- [17] F. Bry and S. Schaffert. The XML query language Xcerpt: Design principles, examples, and semantics. *Web, Web-Services, and Database Systems*, pp. 295–310, 2009.
- [18] エリック・T・レイ. 入門XML. O'Reilly Japan, 2004.
- [19] C. Ware. *Information visualization: perception for design*. Morgan Kaufmann, 2004.
- [20] C. Ware. *Visual thinking for design*. Morgan Kaufmann Pub, 2008.
- [21] 山本陽平. Webを支える技術 -HTTP、URI、HTML、そしてREST. 技術評論社, 2010.
- [22] Robert Klanten, Nicolas Bourquin, and Thibaud Tissot. *Data Flow: Visualising Information in Graphic Design*. Die Gestalten Verlag, 2008.

- [23] Robert Klanten, Sven Ehmann, Nicolas Bourquin, Thibaud Tissot, and Johannes Schardt. *Data Flow 2: Visualizing Information in Graphic Design*. Die Gestalten Verlag, 2010.
- [24] A. Lau and A. Vande Moere. Towards a Model of Information Aesthetics in Information Visualization. *Proceedings of the 11th International Conference Information Visualization*, pp. 87–92, 2007.
- [25] Jenifer Tidwell. デザイニング・インターフェース—パターンによる実践的インタラクションデザイン. O’Reilly Japan, 2007.
- [26] Bill Scott. デザイニング・ウェブインターフェース—リッチなウェブアプリケーションを実現する原則とパターン. O’Reilly Japan, 2009.
- [27] Yahoo!ニュース -RSS. <http://public.news.yahoo.co.jp/rss/>.
- [28] Yahoo!ニュース -検索ランキング RSS. <http://searchranking.yahoo.co.jp/rss/>.
- [29] Twitter. <http://twitter.com/>.
- [30] 辻村浩. Twitter API プログラミング. ワークスコーポレーション, 2010.
- [31] インプレス RD, 財団法人インターネット協会. インターネット白書 2010. インプレスジャパン, 2010.
- [32] Google Maps API. <http://code.google.com/intl/ja/apis/maps/>.
- [33] Rich Gibson. Google Maps Hacks 第2版—地図検索サービスをもっと活用するテクニック. O’Reilly Japan, 2001.
- [34] foursquare API. <http://foursquare.com/apps/>.
- [35] 水野忠則, 峰野博史. 3. 情報共有空間のためのセンサコンピューティング. 情報処理, Vol. 48, No. 2, pp. 135–141, 2007.

- [36] L. Sun and J. Vassileva. Social visualization encouraging participation in online communities. *Groupware: Design, Implementation, and Use*, pp. 349–363, 2006.
- [37] 豊田秀樹. データマイニング入門. 東京図書, 2008.

付録 A

ウェブコンテンツ trend-setter 使用方法

A.1. 概要

本付録は、ウェブコンテンツ trend-setter の利用者へ使用方法を説明するものである。

本コンテンツは、コンピュータ言語である ActionScript3.0 で作成され Flash-Player9 以上のバージョンをインストール済みの、全てのコンピュータで使用可能である。

A.2. 開発環境

本コンテンツの開発環境を表 A.1 に記す。

表 A.1 trend-setter 開発環境

OS	使用ソフトウェア	プログラミング言語
Mac OS X 10.6.3	Flash	ActionScript3.0

A.3. 画面構成

図 A.1 は本コンテンツの初期画面であり、情報のカテゴリーを表すアイコンが10つ配置されている。アイコンのカテゴリーは経済、金融、政治、スポーツ、エンタテインメント、医療、テクノロジー、環境、交通、天気情報である。

利用者は自分が必要とする情報のカテゴリーを表すアイコンを選択し、画面上部のステージ上にアイコンを配置する。配置後アイコンからふきだしが表示され、そのカテゴリーにおける流行の情報が表示される。図 A.2 は、アイコンをステージ上に配置した時の画面である。



図 A.1 初期画面



図 A.2 アイコン配置時

A.4. 使用方法説明

本コンテンツ上では以下の機能を使用することが可能である。

- 利用者が取得したい情報のカテゴリーを表すアイコンを選択し、ステージ上に配置する
- アイコンがステージ上に配置されるとアイコンからふきだしが表示され、各カテゴリーにおける流行の情報8件が表示される
- 1個から10個アイコン全てをステージに配置することが可能である

- 必要ないと思われるアイコンはステージ外に配置すると自動的に初期位置に戻る
- ステージ上のアイコンには流行の度合いによる大きさの変化、ジャンプのアニメーションが起こる

付録 B

ウェブコンテンツ `twit-viewer` 使用方法

B.1. 概要

本付録は、ウェブコンテンツ `twit-viewer` の利用者による使用方法を説明するものである。

本コンテンツはコンピュータ言語である `ActionScript3.0` で作成され、`Flash-Player9` 以上のバージョンをインストール済みの全てのコンピュータで使用可能である。

また本コンテンツは `TwitterAPI` を使用しているため、利用者は自身で `Twitter` のアカウントの取得を行う必要がある。

B.2. 開発環境

本コンテンツを開発環境を表 B.1 に記す。

表 B.1 `twit-viewer` 開発環境

OS	使用ソフトウェア	プログラミング言語
Mac OS X 10.6.3	Flash	<code>ActionScript3.0</code>

B.3. 画面構成

本コンテンツでは利用者が取得した Twitter アカウントでログインを行う必要がある。このアカウントでログイン後、図 B.1 の画面に推移する。本コンテンツでは、利用者が閲覧登録を行っている他利用者の情報はアイコンとして表示される。

利用者は他利用者のアイコンをステージ上に置くことで、他利用者の最新のコメントを閲覧することができる。アイコンをステージ上に配置すると、図 B.2 のようにふきだしが現れ、他利用者の最新コメントが表示される。



図 B.1 初期画面



図 B.2 アイコンステージ配置時

B.4. 使用方法説明

利用者の Twitter アカウントによるログインを行った後、利用者は以下の機能を使用することが可能である。

- 利用者が閲覧登録を行った他利用者はアイコンとして表示され、利用者は閲覧を行いたい他利用者のアイコンを選択する
- アイコンがページ上部のステージ上に配置されるとふきだしが表示され、ふきだし上に他利用者の最新のコメントが表示される

- ステージ上に配置されたアイコンは、コメントの更新があった場合、大きさの変化とジャンプによるアニメーションが起こる
- 1画面につきアイコンは20個のみ表示されているが、利用者が20名以上閲覧登録を行っている場合は、ボタンを押すことで新しい20名の情報を取得する

付録C

ウェブコンテンツ fu*life 使用方法

C.1. 概要

本付録は、Twitter クライアントである fu*life の利用者による使用方法について説明するものである。

本コンテンツはコンピュータ言語である ActionScript3.0 で作成され、Flash-Player9 以上のバージョンをインストール済みの全てのコンピュータで使用可能である。

また本コンテンツは TwitterAPI を使用しているため、利用者は自身で Twitter のアカウント取得を行う必要がある。

なお本コンテンツの開発環境、使用方法は 2010 年 2 月時点のものである。

C.2. 開発環境

本コンテンツを開発環境を表 C.1 に記す。

表 C.1 fu*life 開発環境

OS	使用ソフトウェア	プログラミング言語
Mac OS X 10.6.3	Flash	ActionScript3.0

C.3. 開発過程

利用者は以下の流れに従い、本コンテンツを使用する。

1. コンピュータから本コンテンツのトップページの URL にアクセスする
2. トップページから利用者の Twitter のアカウントを使用してログインを行う (参照: 図 C.2)
3. ログインが成功した場合 4 番の推移に移動、失敗した場合は 5 番の推移に移動
4. ログインが初めての場合は 6 番の推移へ、2 回目の場合は 7 番の推移に
5. 利用者にアカウントの取得を促すため Twitter 公式ページへ自動的に移動する
6. 利用者は性格診断に答え、自身のコメントを視覚的に表す花の種類を決定する
7. メインページへ (参照: 図 C.3、図 C.4)

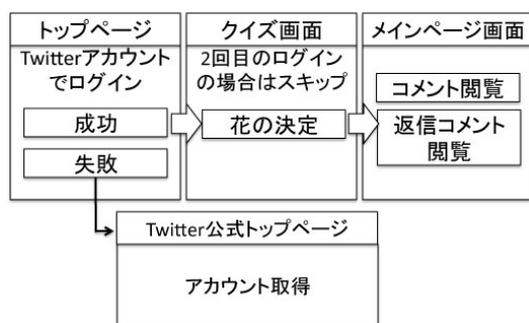


図 C.1 画面推移



図 C.2 fu*life トップページ



図 C.3 コメント閲覧画面



図 C.4 ブーケ作成画面

C.4. 使用方法説明

利用者は本コンテンツにおいて、以下の機能を使用することが可能である。

- Twitter アカウントによるログイン/ログアウト
- 利用者自身の全コメントの閲覧
- 利用者が閲覧登録を行っている他利用者の全コメントの閲覧
- 他利用者と返信コメントのやり取り行った場合、そのやり取りを花のブーケに変化させ、保存する