

気象情報システム構築のための Web 技術の検討

A Study on Web Technologies to Develop Weather Information Systems

前 田 和 昭*

Kazuaki MAEDA

1 はじめに

通信速度の向上・ストレージの低価格化・CPU の高性能化を背景に、大量のデータを蓄積して分析できるようになってきた。例えば、気象予報のためにペタ Byte を超える膨大なデータを蓄積し、数ペタ Flops の性能を持つコンピュータを使って細部までデータ分析している。このような時代の流れを受け、データサイエンスが脚光を浴び、古くから発展してきた統計学と、新しい情報処理技術を中心に、教育研究活動が進められている。

データサイエンスの盛り上がりを受け、情報処理技術の分野では、プログラミング言語 Python が注目されている。IEEE Spectrum が毎年発表しているプログラミング言語ランキング（2019 年[1]他）の 5 年間にわたる上位 5 位を表 1 に記す。2015 年に 4 位だった Python のランキングは、2017 年に首位となり、その後首位を譲ることはない。

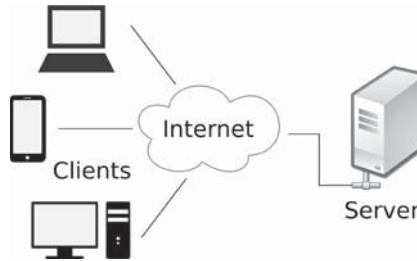
表 1 IEEE Spectrum によるプログラミング言語ランキング

	2019 年	2018 年	2017 年	2016 年	2015 年
1	<u>Python</u>	<u>Python</u>	<u>Python</u>	C	Java
2	Java	C++	C	Java	C
3	C	Java	Java	<u>Python</u>	C++
4	C++	C	C++	C++	<u>Python</u>
5	R	C#	C#	R	C#

* 経営情報学部 ― ソフトウェア工学

Department of Management Synthesis ― Software Engineering

図 1 クライアント - サーバ・モデルの構造



クライアント - サーバ・モデルでは、図 1¹⁾ に示すように、サービスを提供するサーバ (Server) と、サービスを要求するクライアント (Clients) にシステムを分割し、それらをネットワーク接続する構造でシステムを構築する。Web アプリケーションは、クライアント - サーバ・モデルの典型例であり、電子メールやネットワークプリンティングもクライアント - サーバ・モデルを土台にして動作する。Web アプリケーション開発にて Python を利用するときは、サーバ側でのみ Python を利用するのが通例である。Python を使って開発されたプログラムはサーバ側で実行され、クライアントから要求されたリクエストにしたがって、サーバによって作り出された結果がクライアントに返信され、クライアントがユーザのための表示画面を作り出す。

日常生活の多くの場面 (例えば、ショッピングやホテルの予約など) でインターネットを利用するとき、私たちは Web ブラウザを利用する。スマートフォンが普及し、インターネットをスマートフォン経由で利用することが増えるようになったものの、その裏側では Web ブラウザが動作している場合が多い。Web ブラウザが初めて公開されてから、約 30 年が経つ [2]。この 30 年間、多くの Web ブラウザが登場し、現在では、Internet Explorer (以下、IE と記す)、Google Chrome、Mozilla Firefox の 3 つで市場シェアの大部分を占めるにいたっている。

図 2 に、2009 年頃から 2015 年頃までの市場シェアの変化を示す²⁾。このグラフから、2010 年以前は太線で示される IE が過半数を占めていたものの、2012 年の中頃に破線で示される Chrome が IE を抜くことになり、2015 年には、Chrome が市場シェアの 4 割を超えたことが分かる。

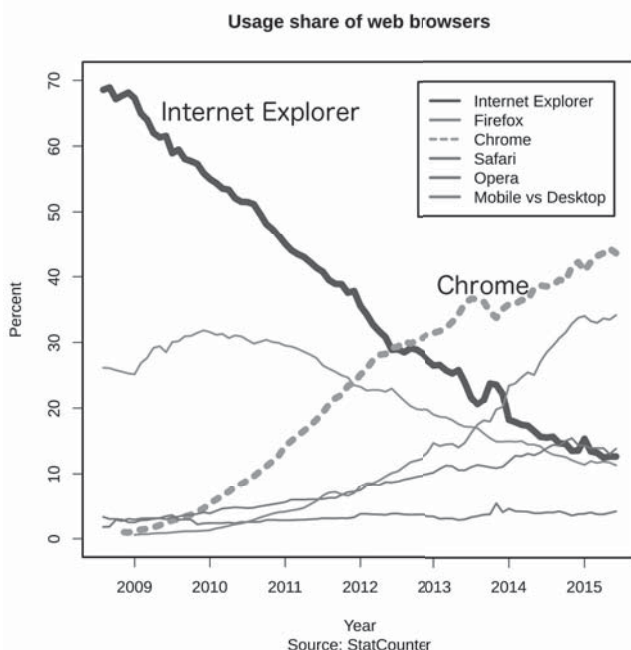
各地域における Web ブラウザの市場シェア (2020 年 1 月現在、Desktop のみ) を表 2 に示す。この表は、StatCounter 社のデータ³⁾ より、日本や各地域における上位 6 位を抜き出したもので、世界でのシェア順に上から並べてある。ただし、インドでは

1) 図 1 は、<https://en.wikipedia.org/wiki/Client-servermodel#/media/File:Client-server-model.svg> より LGPL のもとで使用する。

2) 図 2 は、<https://en.wikipedia.org/wiki/Browserwars#/media/File:Usageshareofwebbrowsers> (SourceStatCounter).svg より CC BY 3.0 のもとで使用する。

3) <https://gs.statcounter.com/>

図2 Web ブラウザ市場シェアの変化



IE が 6 位未満となるため空欄とした。ここでの Edge は、2020 年 1 月に新版がリリースされる前の版でのデータを示している。現在の世界における Web ブラウザ市場シェアは、表 2 が示すように Chrome が約 7 割まで達し、IE は 1 割を下回るようになった。

インターネットの爆発的な普及を背景に、研究から日常生活まで Web ブラウザは不可欠な存在となっている。本稿では、ソフトウェアと Web が十分に統合されていることが重要であるという主張にもとづき、Web コンテンツの利用について PowerPoint に関する考察、Web ブラウザを活用するための技術と応用について述べる⁴⁾。

表 2 各地域における Web ブラウザ市場シェア (Desktop, 2020 年 1 月)

	世界	日本	アジア	北米	欧州	インド
Chrome	68.78%	57.41%	76.23%	63.04%	61.41%	80.29%
Firefox	9.87%	8.65%	8.01%	8.60%	14.09%	10.47%
Safari	8.64%	10.25%	5.34%	13.63%	9.42%	2.50%
Edge	4.71%	9.52%	2.61%	7.68%	5.59%	1.64%
IE	3.70%	12.38%	3.07%	5.09%	4.20%	
Opera	2.49%	0.69%	1.82%	1.01%	3.45%	1.47%

4) 本稿は、発表済みの論文 [3] の内容に解説を追加して作り直した研究ノートである。

筆者が進める研究では、これまで独自に取り組んできた研究成果を土台とし、Web ブラウザを先端情報技術プラットフォームとして利用する方法を追求している。とくに、

- ・ソフトウェアは Web フレンドリであるべきか
- ・Web フレンドリであるために必要なものの何か

について興味を持ち研究を進めている。ここで言う Web フレンドリとは、筆者が提案している言葉で、「ソフトウェアが Web と十分に統合されていること」を意味する。

次節では、先端情報技術プラットフォームを検討するために開発を続けている気象情報システムの概要について述べる。3 節では、PowerPoint を例にして Web フレンドリについて解説する。さらに 4 節では、Web ブラウザの応用例と性能比較について述べ、5 節で WebAssembly について触れる。

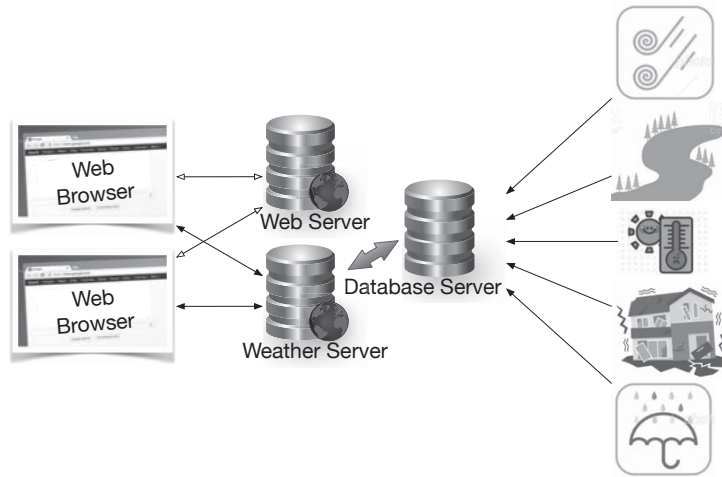
2 気象情報システムの概要

ここ最近、台風や地震などによる自然災害が増えてきたことで、災害時の対策を効率化することが重要な課題になってきている。その課題解決のために、筆者は企業と共同で、気象情報の可視化と人員配備のための気象情報システムの開発を進めている。気象データの対象となるのは、風、河川、温度、地震、雨の 5 種類である。これらの気象データを閲覧・分析・管理するために、機能豊富な画面を 10 種類以上作成した。例えば、複数箇所の地震データを同時に閲覧できるように、複数のグラフを 1 画面で同時に表示する。また、多種類の気象データを 1 つの画面で概覧するために、地図を反映したダイアグラムの上に簡易データを表示し、そこから詳細データ画面に遷移できるようになっている。

この気象情報システムでは、(現在のところ) 約 50 台の測定機器から即時データを獲得し、図 3 に示すように、獲得したデータをデータセンタ内のデータベースに保管する。クライアントには、標準の Windows が稼働する省電力タイプの小型 PC を使う。電源投入時に Web ブラウザが自動的に立ち上がり全画面を占有し、決められた Web サーバにアクセスするように設定しているため、ユーザは OS の諸機能を使うことができず、気象情報システム専用機として利用する。このような構成をとることで、クライアント側の保守を簡略化でき、クライアント PC に問題が起きたときは、単純に PC を交換するだけで良い。

このシステムの Web サーバには、広く使われている Apache httpd[4]を採用し (図 3 の Web Server)、画面とそれに付随するプログラム一式をクライアントに提供する。時間とともに変化する気象データは、独自に開発した気象サーバが提供し (図 3 の

図 3 気象情報システムの概略図



Weather Server), 各クライアントが決められたタイミング (1 秒に 1 回, 1 分に 1 回など) で気象サーバにアクセスする。プロジェクト開始時には, Windows 標準搭載の IE を使っていたものの, 気象サーバに毎秒アクセスするプログラムを試作中, IE の性能が貧弱なことが問題 (4 節で述べる) となり, Chrome に置き換えることになった。

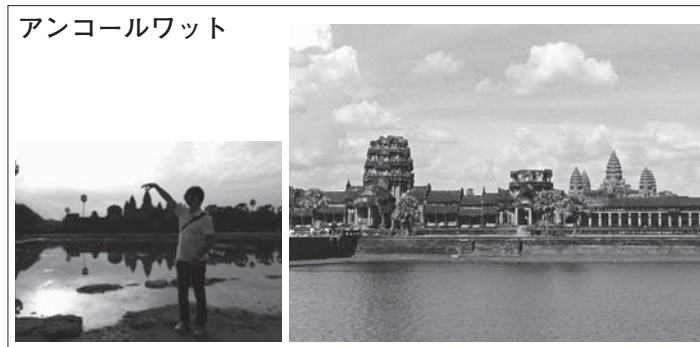
3 PowerPoint と Web コンテンツの利用

PowerPoint がプレゼンテーションソフトウェアとしてリリースされたのは 1987 年頃で, WWW が登場する以前のことである [5]。したがって, PowerPoint は, Web コンテンツを利用することを前提に作られたわけではない。例えば, 筆者がアンコールワットを訪問したときに撮影した画像と, Wikipedia に掲載されている画像⁵⁾ をダウンロードした後グレースケールに変換した画像を貼って作成した PowerPoint スライドを図 4 に示す。このスライドには自由に利用できる画像を使っているため, スライドの配布に対して制限はない。しかし, 利用する画像の規約にて, Web ブラウザで表示する以外のダウンロードが禁止されているとしたら, 図 4 のように画像をコピーしてスライドを作成することはできない。

アンコールワットの場所を示すために, PowerPoint スライドに Google マップ画面

5) 図 4 右側の画像は, Wikipedia の Web サイト (<https://ja.wikipedia.org/wiki/アンコール・ワット>) にて, パブリック・ドメインに公開されている。

図 4 アンコールワットの写真



のコピーを貼り付けることを考えてみる。この画面をコピーして作成したスライドを再配布する場合は、利用規約の禁止行為になるので注意が必要である。もし Google マップ画面のコピーを貼り付けることが許可されたとしても、スライドに貼り付けるのは画像でしかない。その位置に関することを伝えるために、Google マップが提供する機能を使って世界全体を俯瞰した後に、ズームインしながら詳細を表示するスライドショーを行うことができない。Web ブラウザに切り替えて必要な Web サイトを表示すれば、そこで提供されるサービスを利用できるものの、PowerPoint スライドから Web 上で提供されているサービスを直接利用することができない。

このように、利用規約のために Web コンテンツを利用できなかったり、Web ブラウザ経由で使えるはずのサービスが PowerPoint スライドで使えない理由は、PowerPoint が Web ブラウザの機能を提供していないからである。世界中の人達がインターネットを利用し、Web 上でコンテンツやサービスが豊富に提供されているからこそ、PowerPoint などの代表的なアプリケーションは Web フレンドリ、すなわち、Web と十分に統合されるべきと考える。

4 Web ブラウザの活用例と性能評価

4.1 Web ブラウザの活用例

データサイエンスの盛り上がりを受け、情報処理技術の分野ではプログラミング言語 Python が注目されている。Python 利用者には、Web 時代を反映し、Web ブラウザ上で会話的に Python プログラムを試しながら文書を作成・閲覧する Jupyter が普及している。Jupyter は、2017 年にコンピュータ関連学会の ACM から Software System Award を受賞した[6]ことから分かるように、多くのユーザを獲得してい

る。Jupyter を使って CSV データを表示している画面を図 5 に示す。Jupyter は、Web ブラウザ上で Mathematica ノートブックに似た使い勝手を提供する。Python 利用者は Web ブラウザを使い、図 6 に示すように、Web サーバを経由して Python プログラムによるデータ分析を行う。

Web ブラウザからの利用を前提としている Jupyter から、Web が提供する様々なコンテンツとサービスが利用可能であるため、Jupyter は Web フレンドリを実現しているといえる。しかし、著者が Jupyter を評価する途上で、全て標準の Web を土台としていることによる制限を知ることになった。もし、更新が頻繁に必要な可視化プログラムを開発する場合（例えば、マウスを使ってグラフをズーム・回転させる機能など）、Web サーバを介した頻繁なデータの送受信が必要となり、スムーズな使い勝手を利用者に提供できない。

気象情報システムの開発を進めるにあたり、標準の Web で提供する部分と独自に気象データを提供する気象サーバに分割した（図 3 の Web Server と Weather Server）。今後、Web ブラウザを活用した先端情報技術プラットフォームを構築するならば、Jupyter の基本システムとは異なり、気象情報システムを開発した知見にもとづき、構築を進める必要があるだろう。

図 5 Jupyter にて CSV データを表示している画面

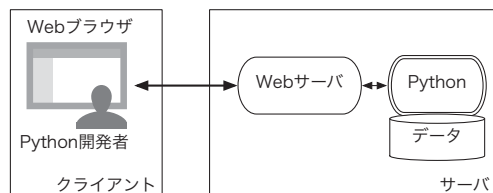
	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Embarked
0	1	0	3	Braund, Mr. Owen Harris	male	22.0	1	0	A/5 21171	7.2500	NaN	S
1	2	1	1	Curtis, Mrs. John Bradley (Florence Briggs Th...)	female	38.0	1	0	PC 17599	71.2833	C85	C
2	3	1	3	Hekkinen, Miss. Laina	female	26.0	0	0	STON/O2. 3101282	7.9250	NaN	S
3	4	1	1	Futrelle, Mrs. Jacques Heath (Lily May Peel)	female	35.0	1	0	113803	53.1000	C123	S
4	5	0	3	Allen, Mr. William Henry	male	35.0	0	0	373450	8.0500	NaN	S

Titanic: Machine Learning from Disaster

Overview

This data is a training set to build your machine learning models.

図 6 Jupyter の基本システム構成



4.2 Web ブラウザの性能比較

気象データの閲覧・分析のためのシステム開発の途上で、大量のデータを迅速に Web ブラウザで送受信しながら、閲覧・分析する機能を実装する必要があり、Web ブラウザの実行性能が追いつかない問題に何度も遭遇した。その問題を解決するために、毎回アドホックに解決案を探ることになった。その結果、Web ブラウザでのプログラム実行性能を向上させることが急務となり、調査のために性能評価実験を行うことになった。

以下では、2018 年の情報システム学会全国大会・研究発表大会にて発表した論文 [7] から、性能評価実験の結果の一部を抜粋して概略する。その詳細は、論文 [7] を参照していただきたい。

各 Web ブラウザ毎の実行時間を計測するための JavaScript プログラムを作成し、代表的な Web ブラウザを使って性能評価実験を実施した。表 3 は、代表的な Web ブラウザにて SVG 画像を描画する JavaScript プログラム 2 つを、3 台の PC にて実行したときの実行時間を一覧表示したものである。この結果から、IE と Edge のときの実行時間は、他 Web ブラウザの実行時間の数倍かかり、IE と Edge の実行性能が悪いことが分かる。また、Chrome, Vivaldi, Opera の実行時間が似た値を示していることは、Web ブラウザでの表示機能を実装しているモジュールが同じであるという事実から納得できる。

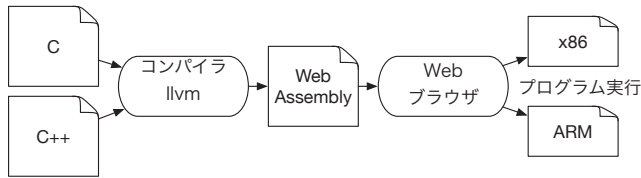
表 3 各 Web ブラウザ毎の JavaScript プログラム実行時間（単位：ミリ秒）

		Chrome	Firefox	Vivaldi	Opera	IE	Edge
PC 2018	時計	22.03	31.13	21.88	29.76	108.05	147.66
	棒グラフ	108.86	233.64	110.56	97.88	433.12	360.06
PC 2016	時計	15.20	16.07	15.29	15.85	84.74	100.49
	棒グラフ	108.08	197.7	111.30	103.08	447.88	517.08
PC 2014	時計	28.58	41.46	35.94	40.60	100.12	132.06
	棒グラフ	171.42	315.64	163.10	164.80	510.70	586.94

5 Web ブラウザが提供する新しい技術

Web ブラウザでプログラムを高速に実行させるための技術が注目されている。例えば、Native Client（以下、NaCl と記す）が Google によって開発されていた。NaCl を使うと、C 言語で作成したプログラムが、バイナリーコードのまま Web ブラウザで実行可能になる。しかし、NaCl は 2019 年の第 4 四半期に Chrome から削除される

図 7 WebAssembly を利用するときの手順



と発表された[8]。

その代替案として注目をあびているのが WebAssembly である[9]。WebAssembly は、Web ブラウザ内で実行可能な低水準言語を提供する。C や C++ などの高水準言語で記述されたプログラムを、図 7 に示すようにコンパイラを介して WebAssembly に変換した後、Web ブラウザ内に実装された処理系で実行する。最新の Web ブラウザ（Chrome, Firefox, Safari, Edge など）で利用可能になっている[10]。

6 おわりに

本稿では、Web フレンドリが重要であるという主張にもとづき、PowerPoint には問題があり、Jupyter は Web フレンドリを実現していることを示した。また、Web ブラウザを気象情報システムに適用する事例を示した。現在 WebAssembly の活用が複数の分野で試行されている。今後、先端技術情報プラットフォーム実現のため Web に関連する技術を調査しながら、新たなチャンスを探っていきたい。

参考文献

- [1] The Top Programming Languages 2019, <https://spectrum.ieee.org/computing/software/the-top-programming-languages-2019> (2019 年 10 月 24 日参照)。
- [2] CERN 2019 WorldWideWeb Rebuild, <https://worldwideweb.cern.ch/> (2020 年 2 月 29 日参照)。
- [3] 前田和昭, Web ブラウザに関する最新技術の検討, 情報システム学会第 15 回全国大会・研究発表大会 (2019)。
- [4] Welcome! -The Apache HTTP Server Project, <https://httpd.apache.org/> (2020 年 2 月 29 日参照)。
- [5] Microsoft PowerPoint -Wikipedia, https://en.wikipedia.org/wiki/Microsoft_PowerPoint, (2020 年 2 月 29 日参照)。
- [6] Jupyter receives the ACM Software System Award -Jupyter Blog, <https://blog.jupyter.org/jupyter-receives-the-acm-software-system-award-d433b0dfe3a2> (2019 年 11 月 12 日参照)。
- [7] 前田和昭, SVG によるユーザインタフェース構築の検討, 情報システム学会第 14 回全国大

会・研究発表大会（2018）.

[8]WebAssembly Migration Guide, <https://developer.chrome.com/native-client/migration>（2019 年 11 月 12 日参照）.

[9]WebAssembly, <https://webassembly.org/>（2020 年 2 月 29 日参照）.

[10]Can I use... Support tables for HTML5, CSS3, etc, <https://caniuse.com/#feat=wasm>（2020 年 2 月 29 日参照）.

本論文は，クリエイティブ・コモンズ表示-継承ライセンスの下で利用可能である。

