

# Web 技術の歴史

## History of Web Technologies

佐藤 和雄

**要約** Web が誕生して 20 年が経つ。Web は、はじめはサーバ上に格納されているファイル（情報コンテンツ）をクライアントが表示するだけであったが、オープン環境による技術の進化と社会的要求により、今日では単なる「情報アクセス」だけでなく「つながり」の場をも提供している。本論では、Tim Berners-Lee によって考案された基本技術－共通言語である HTML、情報位置を示す URL、情報転送のための http - を核にして発展していった Web 技術の歴史を概観し、Web を取り巻く関連技術や情報伝播の変遷について紹介する。

**Abstract** Twenty years have passed since the birth of the Web. At first, the Web service showed just files (information content) stored on the server to the client. However, today, the technological evolution in the open environment and social demand facilitate the Web service to provide just “information access” as well as a “connection” place. This paper provides an overview of the history of Web technologies based on the core and fundamental technology invented by Tim Berners-Lee, and introduces the related technology and the transition of history of propagation of information surrounding the Web.

### 1. はじめに

1991 年 8 月 6 日、スイス欧州原子核研究機構（CERN）の「NeXT」コンピュータ上で世界最初の Web サイト <http://info.cern.ch/hypertext/WWW/TheProject.html> が開設されて 20 年が経つ。もともと Web は「情報アクセス」を目的に作られた技術だが、今日インターネット上で展開されている様々なサービスは単なる「情報アクセス」だけでなく、社会との接続性を求める社会的要求に応え「つながり」（ソーシャル・メディア）をも提供している。高度な技術の積み重ねにより構築されているこれらのサービスも、Tim Berners-Lee によって考案された基本技術－共通言語である HTML (Hyper Text Markup Language)、情報位置を示す URL (Uniform Resource Locator)、情報転送のための http (Hyper Text Transfer Protocol) - を核にしている。

本論文は、Web 20 年の歴史を振り返り、現在を理解し未来を予測する一助となることを目的としている。次章以降では、まずこれまでの Web 技術やサービスの歴史を概観し、Web を取り巻く関連技術や情報伝播の変遷についてまとめる。

### 2. 黎明期

Web は典型的な分散アプリケーション・アーキテクチャであり、サーバとクライアントの二つのエンティティが情報交換のトランザクションに関わっている。初期の Web サーバの役割は、要求が発生したときにクライアントに対してファイルサービスを提供することであり、Web クライアント（＝ブラウザ）の役割は、サーバから提供されたファイル（HTML）を解

析し情報コンテンツを表示することであった。1991～1993年頃のWebは、主に研究者のコミュニティ内における情報交換のメディアとして使用されていた。

Web上で使用される情報コンテンツの共通言語であるHTMLは、1969年にIBMのCharles Goldfarbが文書の多目的利用を実現するために開発したGML (Generalized Markup Language) を起源とする。GMLは、1986年にSGML (Standard Generalized Markup Language) として標準 (ISO-8879/JIS-X4151 (1992)) へと発展した。SGMLは、DTD (Document Type Definition) との照合や省略タグの復元を行うため処理パフォーマンスが悪く、規格書の文法定義が複雑で難しいなど、Web上で使用するには問題があった。Tim Berners-Leeは、SGMLの抜本的な簡素化を行い、タグを固定化しレイアウト言語としてHTMLを開発した。HTMLは、Webページを記述するのに簡単で使いやすく、瞬く間に世界共通のWebページ用の記述言語となった。

HTMLはSGMLのDTDによって定義されたWebページ用のタグセットで、W3C (World Wide Web Consortium) によって定義されている。しかし、HTMLのDTDはあくまでも勧告であり強制力がないため、ブラウザ・メーカーが独自に拡張することもあり、DTDに厳密に従うことを求めるSGMLから見れば、HTMLは亜流であった。

SGMLの正統なサブセットとして、1998年2月にW3CはXML (eXtensible Markup Language) を勧告した。SGMLの不要な機能を削除してWebに必要な機能を追加し、SGMLの基本思想を受け継いだ正統なサブセットとして開発されたXMLは、HTMLの持つ表現情報に加えて、意味情報を持ったデータ表現形式で、ブラウザを介してコンテンツを表示するだけでなく、アプリケーション間でも正しく情報交換を行うことを可能にしている。

また、2000年1月にはHTMLをXML準拠に更新したXHTML (eXtensible HyperText Markup Language) が公開され、HTML文書もXML処理系で統一的に扱えるようになった。XHTML 1.1では文書の見栄えに関連するタグを廃止してCSS (Cascading Style Sheets) に統一し、文書構造の記述に特化した言語へと変化している。

### 3. Webアプリケーションの登場

1991年以降、Tim Berners-LeeはWebをオープンな状態で周知していった。これにより、Webの基本技術はインターネットで共有される重要な知的資産として広く利用され、クライアント、サーバ共にオリジナルな実装が出現することになった。この流れの中で、1993年にNCSA (National Center for Supercomputing Applications) Mosaicブラウザが登場し、1994年にはNCSAからスピンオフしたMarc AndreessenらによってNetscape社が設立され、Netscape Navigatorが公開された。

1993年、NCSAによりNCSA HTTPdがリリースされた。NCSA HTTPdは、単に存在するファイルをクライアントへ送るだけでなく、動的なコンテンツを生成する機能を2種類実装していた。一つは外部のプログラムを実行して出力結果をクライアントへ送るCGI (Common Gateway Interface)、もう一つはHTML上に決められたフォーマットで記述したコメントを基にサーバが出力時に外部コマンドを実行し、その結果を動的に埋め込むSSI (Server Side Include) である。本章では、CGIを始めとする初期のWebアプリケーション構築に使用された技術について解説する。

### 3.1 Common Gateway Interface (CGI)

初期の Web サーバは、クライアントからの要求に応じてサーバ上に保存されている HTML ファイルを送信する単純なものであった。1993 年の CGI の登場により、Web の用途は、研究者のためのコミュニティのドキュメント交換メディアから、大規模なユーザのコミュニティ向けの、より動的なメディアへと変化した。CGI を使用することにより、以前のようにあらかじめ作成されている静的な情報を送信するだけでなく、プログラムの処理結果に基づいて動的に HTML 文書を生成・送出することができるようになった。CGI プログラムからのファイルの読み取り、他のコンピュータとの通信、およびデータベースへのアクセスが実現し、Web をベースとしたアプリケーションへの道が開かれた。

一見便利そうな CGI プログラムではあるが、プラットフォームごとに発生する CGI プログラムの互換性の問題の他に大きな問題が二つあった。一つは、CGI プログラムは Web サーバとは別の独立したプロセスとして起動されるため、CGI に複数のアクセスがあると、複数の CGI プログラムが動くことになり、サーバコンピュータに負担がかかるという点である。もう一つの問題は、CGI プログラムが、基本的には HTML 文書を出力するものであることに起因している。CGI プログラムが最終的に生成する HTML 文書はプログラム内に埋め込まれるため、まず HTML で画面設計をし、それをプログラム中の標準出力メソッドにいちいち置き換えていくという作業が必要で、直感的に見通しづらかった。さらに、CGI プログラムの場合には、論理的な処理と、表示に必要な HTML を生成する部分とが分離されずに、同じプログラム中に混在することになり、それに起因する CGI プログラムのソース管理の難しさは、大規模な Web アプリケーションを作る上での障害となった。また、画面の変更にはプログラムの変更が必要であることから、保守性の面でも問題があった。

### 3.2 クライアント・サイド・スクリプティング

1994 ~ 1995 年頃、Netscape 社の Brendan Eich は、まだ静的だった Web ページへのインタラクティブな機能の追加とブラウザの機能向上を目的に「LiveScript」というスクリプト言語を開発した。Sun Microsystems (現 Oracle) 社と Netscape 社は LiveScript を基に「JavaScript」を共同開発し、1995 年に発表した。翌 1996 年 3 月にリリースされた Netscape Navigator 2.0 が JavaScript のサポートを追加し、広く認知されるようになった。

1996 年 5 月には Sun Microsystems 社により、Microsoft 社を含む数社に JavaScript のライセンスが供与されたが、1997 年 Netscape 社の Netscape Navigator と Microsoft 社の Internet Explorer のブラウザシェアの取り合い (第一次ブラウザ戦争) で、両社が競い合うように様々な拡張機能を盛り込み、結果として言語としての互換性が保たれなくなった。

その後、これらをまとめる為に 1997 年 8 月 ECMA (ヨーロッパ電子計算機工業会) により ECMA-262 規格という JavaScript における標準規格 (仕様) が公開された (この仕様に準拠する JavaScript を通称 ECMAScript という)。また、JavaScript から HTML を参照・制御する方式も統一仕様がなく、開発者はブラウザの種別やバージョンを強く意識して実装する必要があったため、W3C は 1998 年 10 月に DOM (Document Object Model) と呼ばれる標準 API を勧告した。静的な HTML の内容を CSS と JavaScript 等のクライアント・サイド・スクリプト言語を用いて動的に変更する Web 技術を総称して DHTML (Dynamic HTML) と呼ぶ。また、JavaScript の非同期通信と DHTML を使ってページの必要な部分だけを書き換

える Ajax (Asynchronous JavaScript And XML) を、Google が 2004 年頃から使用し注目される技術となった。

1996 年 8 月、Microsoft 社は Netscape Navigator 2.0 に実装された JavaScript に対抗するものとして、JavaScript 互換のスクリプト言語 JScript と共に VBScript を Internet Explorer 3.0 に実装した。しかし、VBScript が Windows 版 Internet Explorer でしか動作しなかったことなどから、クライアント・サイド・スクリプト言語としては JavaScript 程には浸透しなかった。また、1996 年、Netscape 社は Web サーバ Enterprise Server 2.0 にサーバ・サイド JavaScript 実行環境 (LiveWire) をリリースしたが、これも当時はあまり注目されなかった技術ではある。しかし、2009 年に発足した CommonJS<sup>\*1</sup> により共通仕様が策定され、node.js<sup>\*2</sup> や Rhino<sup>\*3</sup> などの実行環境も整備されたため、注目を集めつつある。

### 3.3 Java applet

1996 年 3 月、Netscape Navigator 2.0 で JavaScript と共にリリースされた Java applet は、インタラクティブ性を高められる技術の一つとして注目を浴びた。しかし、当時はまだ Web 関連の技術や環境が十分に発達しておらず、Netscape Navigator に搭載されていた JavaVM や Sun Microsystems 社から提供されていた JRE の JavaVM の起動や動作が遅かったり、Java applet をダウンロードするのに必要な高速回線を利用できるユーザが少なかったなどの理由により、Java applet の注目度は下がっていった。加えて、Microsoft 社製の JavaVM が Sun Microsystems 社の著作権を侵害して独自仕様の実装を進めたり、ベンダーごとの JavaVM の実装が仕様に合わず、互換性の面でも問題があった。Java applet は、その後の Shockwave や Flash の台頭もあり、Web 上でインタラクティブ性を実現する用途には広く使われているとは言えない状況である。現在では、利用シェアが大きいとは言えないものの、チャットや CG アニメーション、ゲーム、教育機関による学習システムなどでの利用を見ることが出来る。

### 3.4 ポスト CGI

Microsoft 社は、CGI の性能問題を解決するため 1996 年にリリースされた Internet Information Server (IIS) 2.0 に Internet Server Application Program Interface (ISAPI) を用意した。ISAPI は、サーバサイドに配置するプログラムを DLL (Dynamic Link Library) として作成し、マルチスレッドで実行できるようにした技術である。ISAPI を利用すれば、IIS が提供する API を利用して、Web サーバの機能を細かく制御できると同時に、実行時のリソース負荷も軽減できるというメリットがあったが、マルチスレッドアプリケーションのデバッグのしにくさなどからあまり普及しなかった。

1997 年、Sun Microsystems 社は、Java Servlet の仕様を公開し、同社が提供する Java Web Server と同時に発表した。Java Servlet は、ISAPI と同様にスレッドとして起動され、メモリー消費量を低減しサーバリソースを効率的に利用できるメリットがあった。Servlet は、いわば、Java Applet をサーバ側で走らせて、サーバの機能を拡張しようとするもので、HTML を生成するという点では CGI と同じであった。しかし、Java を利用することによって、プラットフォーム毎のプログラムの互換性の問題もクリアできた。

## 4. Web アプリケーションの普及

1990年代後半になると、HTMLを出力するスクリプトを記述するというCGIのアプローチと異なり、SSIのようにHTML中の動的な部分にのみスクリプトを埋め込むページ・インライン・モデルが出現し、より本格的なWebアプリケーションの時代となる。本章では、Webアプリケーションの普及に貢献した三つのページ・インライン・モデルを紹介する。

### 4.1 PHP: Hypertext Preprocessor (PHP)

1995年にRasmus Lerdorfが、インターネット上に公開している経歴書のアクセス履歴を解析するために作成したスクリプト群「Personal Home Page Tools」がPHPの原型である。この時点では、PerlスクリプトをC言語のCGIとして利用するために開発したものであった。その後、Rasmus Lerdorfが持っていたフォーム処理のシステムと統合、データベース接続機能を追加し、Webアプリケーションを作成できるような汎用性を持たせて1997年4月「PHP/FI (Personal Home Page/Forms Interpreter)」としてオープンソース化した。

次バージョンのPHP/FI 2.0は、利用者の増加とともにさらなる機能改善が望まれたため、イスラエル工科大学のプロジェクトとしてZend社の創業者であるZeev SuraskiとAndi Gutmansらが改良し、1998年6月にPHP3.0 (PHP: Php is Hypertext Preprocessor)として公開した。

さらに、PHP3.0に安定性、拡張性などのパフォーマンス向上を行うため、PHPのコアになる部分を再構築し、その名称をZend Engineとして搭載したPHP4.0が2000年に登場し、爆発的に普及した。

### 4.2 Active Server Pages (ASP)

Microsoft社がCGIに代わる仕組みとして開発し、1996年12月にリリースされたIIS 3.0に実装されたのがASPである。ASPは、デフォルトのサーバ・サイド・スクリプト言語としてVBScriptを採用し、習熟の容易さから成功を収めた。

また、ASPは、クライアント・サイド・スクリプトと共に使用することにより、より高度なWebアプリケーションを可能とした。クライアント・サイド・スクリプト言語として一旦はJavaScriptに敗れたVBScriptは、ASPのデフォルト言語として地位を確固たるものとした。

ASPは、クライアント・サイド・スクリプトとappletをサーバに拡張し、サーバ・サイドでのVBScriptだけでなくJavaScriptおよびActiveXまでも有効にした。クライアント・サイドで馴染みのあるスクリプト言語の使用と、開発ツールとして定評のあるMicrosoft Visual Studioの使用により、1996年～2000年にかけてWebアプリケーション開発の主流となった。

### 4.3 Java Server Pages (JSP)

JSPは、Sun Microsystems社が開発した、Java Servlet技術をベースとしたスクリプト言語で、1999年6月にJSP 1.0の仕様が公開された。ASPと同様な技術であるが、ASPはスクリプト言語としてJavaScriptまたはVBScriptを使用するのに対して、JSPはServlet形式のJavaコードに変換され、コンパイル・実行することにより、スクリプト言語よりも実行速度が上がる利点があった。また、ASPはMicrosoft社のIIS専用なのに対し、JSPはJavaテクノロジーを使用しているため、様々なWebサーバ用の実行環境が提供されていることも利点

である。

## 5. Web アプリケーションから Web システムへ

ページ・インライン・モデルは、複雑なコードが HTML という「レイアウト構造」の中に直接埋めこまれるため、コードの可読性や保守性の低下を招くこととなった。また、インターネットの普及や業務アプリケーションでの使用が進むにつれ、Web ページに対するユーザ・ニーズが高度化し、デザイン、コンテンツ、ロジックがそれぞれに複雑化したため、レイアウトとロジックが渾然一体となったページ・インライン・モデルでは対応できないようになっていった。

そこで、プログラムを Model, View, Controller の役割に分割して並行開発を可能とする MVC アーキテクチャや、トランザクション管理やデータベース・アクセスといった機能を大規模/分散環境に提供する基盤が整備され、Web ページは Web アプリケーションから Web システムへと、スケーラビリティの上でも複雑度といった観点からも飛躍的な進化を遂げた。本章では、MVC アーキテクチャを採用した J2EE と ASP.NET について説明する。

### 5.1 Java 2 Platform, Enterprise Edition (J2EE)

1999 年、Sun Microsystems 社は J2EE 標準 1.2 をリリースした。これは、サーバ・サイド Java の最も基本的な機能である Servlet と JSP、JDBC (Java Database Connectivity) や JTA (Java Transaction API) といった Java テクノロジーを一本化したもので、これにより Java テクノロジーによる企業システム向け Web アプリケーション開発のための最低限の基盤が整備された。その後、Craig McClanahan が、企業規模の Web アプリケーションのコードの複雑さを軽減するために、アプリケーションに一定の構造的な枠組みを与え、汎用的な機能を部品化することで設計とコーディングを簡単化する開発フレームワーク「Struts」を 2001 年に公開した。Struts は MVC パターンを実装したもので、シンプルな構造やその取り組みやすさ、さらに無償であることなどから爆発的に普及、MVC フレームワークのデファクトスタンダードとなった。

2001 年の J2EE 1.3 と、2003 年の J2EE 1.4 では、EJB (Enterprise JavaBeans) によるトランザクション管理やデータベース・アクセスといった機能を大規模/分散環境に提供する基盤が整えられた。また、EJB のような重厚な機能群を必要としないアプリケーションを開発するために、「Spring」や「Seasar2」に代表される「DI (Dependency Injection) コンテナ」と「Hibernate」や「TopLink」といった「O/R (Object/Relational) マッピング・ツール」がオープンソース・プロダクトとして登場した。2003 年～2006 年頃には、Struts + Spring + Hibernate のように、MVC、DI × AOP (Aspect Oriented Programming)、O/R マッピングの各機能について、プロダクトを組み合わせて利用することが一般的になり、EJB コンテナを持たないサーブレット・コンテナ上の軽量フレームワーク開発が広まった。

2006 年にリリースされた Java EE 5 では、主要なデファクト標準フレームワークが標準搭載され、Java EE 5 に含まれる EJB 3.0 は、EJB コンテナ自体に DI × AOP 機能を備え、O/R マッピング仕様である Java Persistence API (JPA) も策定され、Java EE の主要機能が軽量フレームワークと同じ容易さで利用できるようになった。また、MVC フレームワークの標準規格として JavaServer Faces (JSF) が策定された。

2009年、Java EE 6 がリリースされ、すべてのフレームワーク機能を統一的にサポートするフルスタック・フレームワークを実現するために、JBoss Seam の仕様を基に、Java EE の標準規格となったフレームワーク群全体にわたる統一 DI 仕様が定められた。

## 5.2 ASP.NET

2002年1月にリリースされた ASP.NET は、Microsoft 社が「.NET 戦略」に基づいて開発した「.NET Framework」の主要機能の一つである。ASP はサーバ・サイドの単なるスクリプト・エンジンであったが、ASP.NET は Web アプリケーションや Web サービスの開発・実行環境を提供した。また、ASP で使用されていたインタプリタ言語としての VBScript や JScript が廃止され、ASP.NET ではコンパイル言語である VB.NET や C# が使用できるようになった。

ASP.NET では、HTML 部分とコード部分を分離する「コード・ビハインド・モデル」を採用し、ポストバックと ViewState によりイベント駆動型モデルを実現している。これらの仕組みは WebForm と呼ばれ、HTML や HTTP の仕組み、ブラウザとサーバ間のデータのやりとりなどを抽象化し、既存のクライアント・アプリケーションを開発するように Web アプリケーションを開発できる環境を提供する。

2007年1月、ASP.NET Ajax がリリースされ、同年11月、ASP.NET 3.5 に統合された。ASP.NET Ajax は、JavaScript のコーディングをほとんどせずに Ajax を使ったページ作成を可能としている。2009年3月には、ViewState のパフォーマンスの問題やクライアント・サイド・スクリプトとの親和性等の問題、そしてより高度な Web アプリケーションを開発するために、MVC アーキテクチャを採用した ASP.NET MVC がリリースされた。

## 6. 関連技術

本章では、Web を取り巻く関連技術として、必要な情報を引き出すための検索エンジン、情報を間違いなく提供するためのコンテンツ管理システム、そして Web の「使いやすさ」を向上させるユーザビリティについて紹介する。

### 6.1 検索エンジン

検索エンジンは、インターネットの普及初期にはすでに存在していたが、その歴史は、Google 以前と以降に大別できる。

Google 以前の 1993～1998 年は、文書内容そのものから価値判定を行っていたが、情報提供者が自サイトへのアクセスを呼び込むために SEO (Search Engine Optimization) 対策を施し、検索結果が情報提供者の意図を多分に含むことになり精度が低下していった。

1998年に登場した Google は、「リンクされるホームページには価値がある」という法則 - 「人気の高い情報」≡「満足度の高い情報」- を発見し、それを検索結果に反映した。人気の高さ（第三者の評価）は、文書内容と比べて情報提供者の意図を含みにくく、検索精度が向上した。

しかし、「知りたかった情報」は必ずしも「人気の高い情報」とは限らない。見ず知らずの「第三者の評価」ではなく、志向のあった「第三者の評価」として、Google 以降、最近ではキュレーション<sup>\*4</sup>が人気を集めている。また、各種目的別に多様化した検索エンジンが現れ、ブログの

情報に特化した検索、商品情報の検索に特化した商品検索サイト、検索エンジンでは判断できない抽象的な条件などでの検索を人手に求めた「人力検索」「ナレッジコミュニティ」など次々と新しい検索エンジンやサービスが生まれている。

## 6.2 コンテンツ管理システム (CMS)

コンテンツ管理システム (Contents Management System : CMS) とは、Web サイト管理者でなくても、デザイン化されたテンプレートのもと、テキストや画像などが含まれたドキュメント、あるいはクライアント・サイド・スクリプトなどの様々な形式のコンテンツの作成、編集、管理ができる機能を提供するものである。これらは一定のルール、プロセスおよびワークフローによって制約、検証される。また、ほとんどの CMS は、データベース管理システムを使用してコンテンツを一元管理する。

CMS の歴史は意外に古く、1995 年には商用製品が開発されていた。CMS は、商用やオープンソース、高機能のものや機能がほとんどないものが、欧米だけでなく日本、韓国など様々な国や地域で開発され、全世界で 3000 以上のプロダクトがあると言われる。このように多種多様なプロダクトが生まれた主な理由は、企業の Web サイトの構築を情報システム部門ではなく広報や営業部門が担っていたため、Web 制作会社、印刷会社、広告代理店、翻訳会社といったソフトウェア産業とは異なるプレーヤーがサービスを供給してきたことにある。そして、それらのプレーヤーにより、Web を簡易的に更新するような CMS の原型ツールが個別にスクラッチ開発された。その機能も、コンテンツ制作を重視したもの、Web サイト管理を重視したもの、コンテンツ配信 (デプロイ) 方法を重視したものなど、サービスを供給する会社により様々である。

一方、大手ソフトウェア・メーカーは、CMS に関してはポータル製品やドキュメント管理から ECM (Enterprise Content Management) へのアプローチを進めてきたが、企業 Web サイトが情報システム部門の管轄外だったため、Web コンテンツの CMS は供給してこなかった。このように多種多様な CMS は、単体で使われることは少なく、制作過程ではコンテンツ制作に強い CMS を、デプロイ過程ではワークフローが充実している CMS を、運用過程ではサイト管理に強い CMS を使用するなど、組み合わせられて使われることが一般的である。

また、Web サイトが企業の事業基盤として大きな地位を獲得してきた近年、CMS も基幹システムの一つとして位置づけられるようになり、Oracle が 2007 年 10 月に Oracle Universal Content Management (2006 年 11 月 ECM 製品ベンダーの米 Stellent 社を買収) を製品ラインナップに加えるなど、大手ソフトウェア・メーカーが参入するようになってきた。

## 6.3 ユーザビリティ

ユーザビリティとは、ユーザ・インターフェイスにおいて「使いやすさ」を示す品質特性の一つであり、コンピュータおよびコンピュータ搭載機器の高機能・多機能化、そしてそれらをより簡単に使いたいというニーズに応えるために、古くから人間工学、HCI (Human-Computer Interaction)、ユーザビリティ工学、認知工学等で研究されていた。

Web においては、Jakob Nielsen が 1994 年に Web デザインの最初の調査をし、1996 年に「Top Ten Mistakes in Web Design」<sup>[6]</sup>を著すなど初期の頃から様々な取り組みがなされていた。そして、1999 年に人間中心設計のプロセスを定めた「ISO 13407:1999 Human-centred

design processes for interactive systems」<sup>\*5</sup>が発表されるのと、Web サイトの使用目的の多様化の時期が重なり、広く注目を集めることとなった。

現在では顧客が初めて企業と接する場所が Web サイトになることが増え、B2C 型の Web サイトでは、単なる「使いやすさ」だけではなく、リアル店舗と同等の顧客経験価値（カスタマ・エクスペリエンス）を提供することが重要になってきた。特に EC サイトや広告収益モデルを展開するサービスなどでは、企業自身や企業の製品・サービスに対する顧客の信頼性、愛着度といった顧客ロイヤリティを高め、新規顧客の獲得や顧客の維持に繋げる必要がある。そのためには、カスタマ・エクスペリエンスを効果的に提供することが重要で、顧客を特定できればできるほど、質の高いエクスペリエンスを提供できるようになる。それには、トレーサビリティが重要であり、この分野でのユーザビリティでは、顧客の動きを捕捉・分析し、フィードバックし、ユーザビリティをさらに向上させるというサイクルが必要となる。

一方、企業の基幹系業務アプリケーションでは、情報システム部門からの視点や機能の充実の優先、業務単位のシステム化やマルチベンダー化により、操作性の統一感が損なわれた。最近では「使いやすさ」を重視し、業務効率の向上を図るため、ユーザビリティガイドラインを策定する企業も増えている。特に官公庁系においては、2009年7月1日に開催された各府省情報化統括責任者（CIO）連絡会議第37回会合において、「電子政府ユーザビリティガイドライン」<sup>[7]</sup>が決定されるなどユーザビリティの重要性が認知されるようになってきた。

## 7. 「情報伝播」の変遷

1990年代前半、Webの主な利用者は研究者やパソコン通信を利用していた「限られた人達」であった。この頃は、それまでとは違い、特定のプロバイダーの領域を越えて情報入手できることに価値があり、情報提供者と情報受容者とが明確に別れ、Tim O'Reillyが言うように「一方通行」の時代（Web 1.0）であった。そして、1995年のWindows 95の発売、ブロードバンドやインターネットに接続可能な携帯電話の普及等を経て、「限られた人達」だけではなく、「普通の人達」がインターネットを利用するようになった。「Web 1.0」時代は、「情報アクセス」が中心で、検索エンジンが情報伝播の役割を負っていた。

そして、2004年「Web 2.0」の時代になると、それまで「一方通行」だった情報の流れは「双方向」となり、誰もが情報発信者となりえ、情報のフラット化が進み、インターネットはグーテンベルクが活版印刷を発明して以来の「情報爆発」といわれる媒体となった。様々なソーシャル・メディアが登場し、他者との「つながり」と「情報アクセス」が密に結びつくようになってきた。

Web1.0の時代には情報を個人が探し出すだけであったが、個人によって検索・収集された情報が、ソーシャル・メディアによって信用できる個人（インフルエンサー）の信頼関係＝「つながり」に基づいて伝播する。この時代を代表するものの一つであるブログ、SNS（Social Network Service）、口コミサイト、COI（Community Of Interest）サイトなどはCGM（Consumer Generated Media）と呼ばれ、消費者の購買行動にまで影響を与えるようになった。

そして、現在「Beyond Web 2.0」や「Web 3.0」と言われる次の時代を迎えつつある。Web 3.0について、Robert O'Brien<sup>\*6</sup>は、

Web 1.0：集中化した「彼ら」Centralised Them

Web 2.0：分散化した「私たち」Distributed Us

Web 3.0：非集中的な「私」Decentralized Me  
と定義した。Web 3.0は、フラット化されて「つながり」あつたさまざまな「私たち」の情報を、Web 1.0のように特定少数の情報発信者（「彼ら」）によって集権的にコントロールして体系化するのではなく、「私」自身のために「私」自身によって体系化できるような技術やサービスで、キーワードはパーソナライゼーションとレコメンデーションである。パーソナライゼーションとレコメンデーションはすでにEC分野では実用化されているが、問題は誰が自分に代わって必要な情報をパーソナライズしてレコメンデーションしてくれるのかということであり、まだ議論の途中である。

## 8. 日本ユニシスグループの取り組み

インターネットの利用チャネルが、PC中心からスマートフォンやタブレット端末の普及と共に時間と場所を問わないモバイル機器へと拡大し、インターネット利用者の裾野が拡大している。このような流れの中、企業も既存のビジネスの枠組みを越えて、異業種領域へ新規参入を図るなど、インターネットを通じて産業の構造的変化も進んでいる。

そして企業のWebサイトは常に競合他社との競争に置かれ、他社にはないサービスをいち早くリリースし競争優位を勝ち取る必要があり、我々ベンダーにはスピード感を持った開発が求められている。また、開発中も市場や競合他社の動向による仕様変更や納期の短縮が求められることもある。これらの顧客からの要求に応えるために、日本ユニシスグループでは短期軽量開発への取り組みを始めている。仕様変更や短納期を実現するための開発プロセスの策定やECサイトで使用される再利用を前提とした機能部品群の開発、コード自動生成ツールの開発に取り組み、実案件への適用を始めている。さらにユーザビリティ&デザインセンターを設立し、「使いやすさ」を重視した操作性の向上や顧客経験価値の向上を目指し、設計プロセスや設計・構築技法の確立に取り組んでいる。また、近年では実店舗を持つ小売業がネットショッピングに力を入れ始めている。これらのWebサイトは、インターネット流通の拡大により以前のような副次的な位置付けから実店舗と同等の位置付けとなり、リアルとバーチャルのマルチチャネルによる販売拡大を目指している。このようなWebサイトでは、実店舗と同等の顧客経験価値の提供や、実店舗では得ることが難しかった顧客の購買行動をWebアクセスログの分析等により把握することが重要である。日本ユニシスグループではWebマーケティングの取り組みをSNSなどのCGMへと拡大し、様々なメディアから顧客の志向を収集・分析し企業活動のPDCAを支援する体制を強化しつつある。

日本ユニシスグループは、ECサイトやチケット販売サイト、楽曲配信サイト等いくつかの企業Webサイトの構築に携わり、多くの経験と技術を蓄積している。しかし、Webは常に進化し、変化している。今後もWeb技術そのものだけでなく、その技術が現実社会においてどのような場面に適用できるのか、ということを常に考えながらWebに取り組んでいかなければならない。

## 9. おわりに

Web 20年の開発技術の歴史を振り返った。要素技術のほとんどは、2000年代前半にほぼ出揃い、以降、技術進歩のスピードは鈍化している。

Tim O'Reillyは、Web 1.0時代の旗手としてNetscape社を挙げ、Web 2.0時代の旗手とし

て Google を挙げている。前者はソフトウェア産業による技術開発の時代を表し、後者はサービス産業による技術適用の時代を表している。これは、今ある IT 技術をどう使うか、現実社会の場にどのように適用するか、という適用先の開拓と応用力が益々重要になることを示唆している。Web2.0 以降、重要となってきた「つながり」は、人間が中心であり、社会・文化・地域特性を無視しては考えられず、従来のように海外の売れているサービスを単純に持ち込んでも解決できないであろう。そのためには、Web 発展の原動力である、「デジタル化」、「オープン化」、「パーソナル化」の三つの社会要求の変化を常に追いつける必要がある。

本特集号により、日本ユニシスグループの取り組みと Web 技術のトレンドをご理解いただき、事業拡大に向けた課題解決の一助にいただければ幸いです。

- 
- \* 1 CommonJS は、ブラウザ以外で JavaScript を利用する際に問題となる領域の解決を目的とし、2009 年 1 月に Kevin Dangoor によって立ち上げられたプロジェクト。当初、ServerJS だったが、2009 年 8 月に CommonJS となる。
  - \* 2 Google が開発した JavaScript エンジン V8 を使ったアプリケーション・サーバー。
  - \* 3 Mozilla プロジェクトによってメンテナンスが行われている、Java で書かれた JavaScript インタプリタ。
  - \* 4 語源 Curator。博物館や美術館などで展覧会を企画し、展示物を整理し見やすくする専門職を指す。転じて、情報のあるテーマに基づいて収集し、それ自体にコンテンツとしての価値を持たせて共有することを指す。
  - \* 5 JIS Z 8530:2000 人間工学—インタラクティブシステムの人間中心設計プロセス。
  - \* 6 ネットメディアである「ReadWriteWeb」が 2007 年 4 月に行った Web 3.0 の定義に関するコンテストの優勝者。

- 参考文献**
- [1] 浜屋 敏, 「Web 3.0 とインフォメディアリ」, Economic Review Vol.12 No.4, 富士通総研, 2008 年 10 月, p94 ~ p95
  - [2] Mike Lehmann, 「J2EE and Microsoft .NET」, Oracle Corporation, 2002 年 4 月
  - [3] M. Keith Mortensen, Rob McGovern, Charles Liptak, 「ASP.NET と Struts : Web アプリケーションのアーキテクチャ」, Microsoft Corporation, 2004 年 2 月
  - [4] Tim O'Reilly, 「What is Web 2.0?」, 2005.09,  
<http://radar.oreilly.com/archives/2005/09/what-is-web-20.html>
  - [5] Tim O'Reilly, 「What Is Web 2.0」, 2005.09,  
<http://oreilly.com/web2/archive/what-is-web-20.html>
  - [6] Jakob Nielsen, “Original Top Ten Mistakes in Web Design”, Jakob Nielsen’s Alertbox for May 1996, <http://www.useit.com/alertbox/9605a.html>
  - [7] 「電子政府ユーザビリティガイドライン」, 電子政府ガイドライン作成検討会, 首相官邸, 2009 年 7 月  
[http://www.kantei.go.jp/jp/singi/it2/guide/security/kaisai\\_h21/dai37/h210701gl.pdf](http://www.kantei.go.jp/jp/singi/it2/guide/security/kaisai_h21/dai37/h210701gl.pdf)

※上記参考文献に含まれる URL は、11 月 18 日時点での存在を確認。

**執筆者紹介** 佐藤 和雄 (Kazu Sato)

1987 年ユニバック・ソフト・エンジニアリング(株)入社。汎用機用通信制御装置 (DCP/TELCON シリーズ) 通信ネットワーク関連プログラムの開発を担当後、携帯事業者向けメール GW システム開発、衛星放送事業者向け番組編成システム、音楽配信システム開発等に従事。2007 年日本ユニシス(株)に転籍し、現在、システム統括部長。

