

Webデザインにおける制作プロセスの変容と創造性の行方

ソフトウェアによる支援と代行の狭間で「つくること」はどう変わるのか？

上平崇仁¹⁾ 菊池司²⁾

1) 専修大学ネットワーク情報学部

2) 拓殖大学工学部工業デザイン学科

The transformation of the production process and the creativity in Web Design

Takahito KAMIHIRA, Tsukasa KIKUCHI

E-mail:kamihira@isc.senshu-u.ac.jp,tkikuchi@id.takushoku-u.ac.jp

1) School of Network and Information, Senshu University

2) Department of Industrial Design, Takushoku University

アブストラクト

Web空間の発展に伴って、外部リソースや既製システムなどの利用が盛んになっている。Blogに例を見るような、コンテンツを管理するシステムを導入したWebサイトの制作方法は、かつての静的なHTMLを人力で繋げていた時代から大きく変化した。ソフトウェアの発達は、構築のためのパーツの再利用化や効率化を進め、同時に一点受注型のWebデザインの生産形態も変わりつつあると言える。

本稿では、普及し始めたデザイン・ソフトウェアやデザインテンプレート等の事例を紹介しながら、Webデザインの制作プロセスの変容を解説する。そして、変化したその先を考察し、コンテンツとビジュアルの分離やデザイン要素の部品化がもたらす可能性、及びソフトウェア工学の流れと融合した制作プロセスのあり方について論じる。最後に、これらの問題点として、テクノロジーによって制作を支援する際のツールと創造性の関係を考慮しながら、今後のWebデザインの制作方法の動向を展望する。

キーワード: Webデザイン, デザインプロセス, デザイン方法, 創造性

Abstract

As the World Wide Web develops, more and more external resources and ready-made systems are being introduced. The production methods of Web sites have changed greatly, and the production forms are in the process of changing.

By introducing various kinds of Design Software and Design Templates, this study explores the transformation process of the Web design. It also investigates the possible consequences of the separation of contents and visual, and the design elements being used as parts, and discusses how the production process of Web design can be combined with that of software engineering. The future developments of the Web design are touched upon at the end.

1 : はじめに

Web空間は、初期の静的なHTMLドキュメントによるハイパーテキストシステムから、テクノロジーの進化と共にさまざまな方向に多様化し拡張し続けてきた。例えば現在、テキスト主体の情報提供においては、blogに代表されるようにコンテンツを効率的に管理するためのシステム化が加速している。人力が主体であったWeb空間のリ

ンク構造は現在では多くの部分がテクノロジーによって行われるようになり、これらによって、Webページの制作側は日々の更新の煩雑さを軽減され、閲覧者は連結されていく情報を有効に活用しやすくなった。

同時に、一般の人がWebサイトを立ち上げる際の制作環境も大きく変貌した。制作を補助するソフトウェアの高機能化によって、HTMLタグ等の習熟の必要性が薄くな

り利便性が高まったことに加え、インターネット上で配布されている外部リソースやサービスを組み合わせることによって、大幅な作業の効率化を計ることが可能になった。

Web サイトは通常それ自体が作品というわけではなく、情報を伝達するメディアである。特殊な例を除いて、利用者はページのデザイン自体を見に来るわけではない。そこでは見栄えやコンテンツの視覚的完成度を高めることよりも、利用される情報自体をより伝わりやすく設計することや、更新の迅速さのほうが重要であると言える。例えば企業等においても、外部に発注するよりは担当者が手元のPCで情報を入力し、そのまま更新出来た方が本来は効率が良い。これらは書籍のような保存性の高いものとは異なり、システムと同様に「運用」と共にあるWebのメディア的特性と言える。そのため、誰もが対応できるように手数を減らすべく、コンテンツ管理システムやデザインテンプレート等の外部リソースが徐々に活用されるようになった。そして現在では、それがさらに「だれでもデザインできるようにするためのツール」、いわばデザイン・ソフトウェアが出現し始めた。これらによって制作にデザイナーが関わらなくても、傍目には見分けのつかないクオリティのアウトプットが可能になりつつある。

こういった流れの中で、スペシャリストとしてのWebデザイナーの存在と役割もまた変化せざるを得ない。今後コンピュータによる万人向けの制作支援が進み、技術的障壁も少なくなっていく状況においては、これまでのように技術労働をも含めた一点受注生産型の業務形態から大きく変わっていくであろうことが予想される。そこで、デザインを仕事としている専門家、いわゆるプロフェッショナル・デザイナーに残される仕事のエリアと発揮されるべき創造性はどのように変わっていくのだろうか。

本稿では、普及し始めたデザイン・ソフトウェアやデザインテンプレートの事例を紹介しながら、Webデザインにおける制作プロセスの変容を解説する。そしてそれらの可能性と問題点を考察しながら、今後のWebデザインの傾向を展望することが目的である。

2 : Web デザインと工程の変遷

Web制作においては、コンテンツそれ自体の情報の効果的な見せ方に加えて、そのページにたどりつくための操作性の設計も必要である。利用者がサイトを閲覧する際にそれらを双方とも成り立たせながら呈示できるように、ブラウザに表示される画面に対して適切な解を与えるよう

に設計しなければならない。そのための総合的な知の工程をここではWebデザインと呼ぶ。本章ではWebデザインに用いられてきたソフトウェア製品とそれによる工程を概観する。

2-1 : Web とデザインのためのソフトウェアの成り立ち

1989年、インターネットによる情報の分散化を重視し、文書形式の一部の整合性を省いて簡略化した巨大なハイパーテキストシステムがティム・バーナーズ・リーによって考案された。これがWorldWideWebの始まりであり、その時の文書言語がHTML (Hypertext Markup Language) である[1]。しかし、HTMLは簡略な文書構造を表示するためのものでしかなかったため、ブラウザ毎に独自に仕様が開張され始めた。ソースには色やフォントなどの記述が混合し始め、徐々に非合理的なものになっていった。そこでコンテンツとデザインの分離が叫ばれ始め、Web標準化の機関であるW3C[2]によって、HTMLは文章のコンテンツと論理構造のみを記述し、色やフォントなどのクライアント側で呈示する時のビジュアル要素は別途にCSS (CascadingStyleSheet) によって制御する、ということが取り決められた[3]。その後のセマンティックWeb[4]を目指す構造化の流れと仕様の展開についてはここでは割愛するが、このようにWebにおけるビジュアルのデザインは、本来はCSSによって記述されることになっている。しかし周知のように、当初はCSSへの対応がブラウザ毎にバラバラの極みであったために普及が遅れた。やむなくWebデザイナーたちは、どの環境でも安定した表示をさせるために、HTMLの中でtableタグを強引に解釈した「テーブルレイアウト」と呼ばれる本来からは正しくない記述[5]によってページやナビゲーションのデザインを行うという、いわば裏技を編み出した。これを巡って仕事として美しいのは表示であるべきか書き方であるべきか、デザイナーとプログラマの間でそれぞれの職業的美意識のために多くの意見の対立が見られることもあった。

このような、誰もが初めて体験する新しいメディアに対して、技術と熱気ばかりが先行する状況の中で、より直感的にデザインの作業が出来るようにするために、Web制作のためのソフトウェアが誕生したわけである。これらは当初から邪道扱いされることが多かったが、実際、初期設定の状態では奇妙な記述をするものばかりであった。なお現在でもCSSへの対応には疑問符が付くものが多い。ともあれこれらのソフトウェアによって、記述する言語が先に来るのではなく、完成形をグラフィカルに確認しながら

デザインを検討することができるようになった。ワープロ感覚でコンテンツを編集出来るようになり、ソースを読んだり JavaScript を記述したりするのが苦手なデザイナーを強気に支援した。現在、販売されている主な製品として Macromedia の DreamWeaver[6], Adobe の Golive[7], IBM の Homepagebuilder[8], Microsoft の FrontPage[9] がよく知られている。

その中で DreamWeaver は勝手にソースを書き換えなかったためにデザイナーに好まれ、またこれらの中では高性能であったことが理由でデザイン業界に普及した。また同じ Macromedia が販売する Flash[10]によって、映像や音や動きなどのマルチメディア要素やマウスイベントのインタラクション等の要素も飛躍的にひろがり、現在では非常に高いクオリティのインタラクティブな表現も可能になっている。そして要求されるスキルが高度化するに従って、プロの現場では職種毎に制作の分業化も定着しつつある[11]。現在でも多種多様なコンテンツが日々生産され、利用されているが、これらのソフトウェアの普及が多くの人たちの Web サイト制作を支援し、敷居を下げた大きな要因であることは間違いないだろう。

2-2 : Web テクノロジーの周辺から

Web 開発における技術的側面に注目すると、BBS で用いられていた CGI 等の Web プログラミングの発達により、ブラウザ上で動的に処理が行えるようになった。それによって運用に FTP を必要としないで Web サイトが構築できるようになっている。例えば Wiki[12]は、ブラウザ上でだれでもページを書き換えたり生成したりしてサイトを構築していくことができる Web ツールである。編集ボタンをクリックすれば巨大なフォームが現れ、書き換えればその場でページが動的に更新されるというユニークな機能をもつ。Wikipedia[13]は、この技術を用いて世界中の人々と協同で作っていくオンライン百科事典作りのプロジェクトである。

そして もう一つ、ページのコンテンツをデザインと分離してデータベースとして管理し、Web ページを動的に生成するツールとして CMS (コンテンツマネジメントシステム) [14]ツールが普及し始めた。これらを使用することによって手軽にブラウザの管理画面から情報を更新できるようになり、遠隔協同作業による Web コンテンツ構築も可能になった。CMS は運用フローが重視される大規模サイトでは比較的早くから用いられていたが、今では一般ユーザーにまで裾野が急速に広がっている。

MovableType[15]などで作成される Blog もこの一種である。よく知られている CMS ツールにフリーウェアの Xoops[16]や Plone[17]、販売されている製品に NOREN[18]などがある。今後はよほど小規模のサイトでない限り、CMS を使わずにサイトを立ち上げることは少なくなっていくだろうという予想もある[19]。なお、これらではページや情報の流動性ゆえ、いつでも変更可能なようにメニューやトピック等を完全に固定することはなく、テンプレートやスキンによって柔軟に変更可能なように制御されている。ここでは活発に情報を更新する人が「使う」ことを前提にサイトが運用されており、デザイナーや管理者のような特権的な人が細かく見栄えを調整しアップロードしていたような、かつての静的な運用形態とは異なっている。[図 1]。

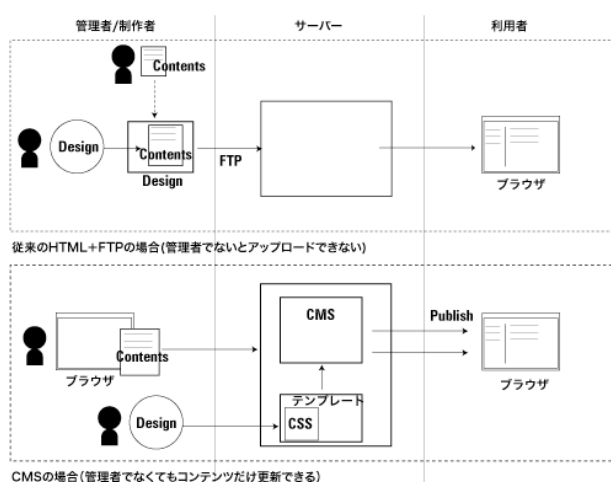


図1 HTML+FTPの場合とCMSの場合の比較

これはメディアとしての様態が変化するに従って、そこで求められるデザインも変化している例であると言える。このようなツールによるサイト構築ではビジュアルのデザインが独立しており、CSS が効果的に機能するようになっている。現在は未だ CSS はソースの直接の編集が中心であり習得がやや難しいが、じきに HTML と同じようにより直感的に編集する手法が確立されると予測される。

また、従来はメンテナンスが極めて大変であった Flash サイトにおいても CMS 化がみられるようになった。ベルギーの gourp94[18]は、Flash を用いた高いクオリティのウェブ制作で知られるが、独自に開発した"Flash94"という、Flash サイトの更新やメンテナンスに特化した専用 CMS を開発し、クライアント先に提供している。彼らが制作したドイツの D' Art Design Gruppe のサイト[19]は、会社の業務実績のポートフォリオとして公開されている

が、通常の企業内システムと同等なほどに運用重視の構築が行われており、優雅な動きによる視覚的演出とともにクライアント自身による管理の簡単さが両立されている[図2]。このようなクライアントによる運用のためのシステム化は、日本でも高い技術力を持ったウェブ制作会社を中心に組み込まれ始めている。

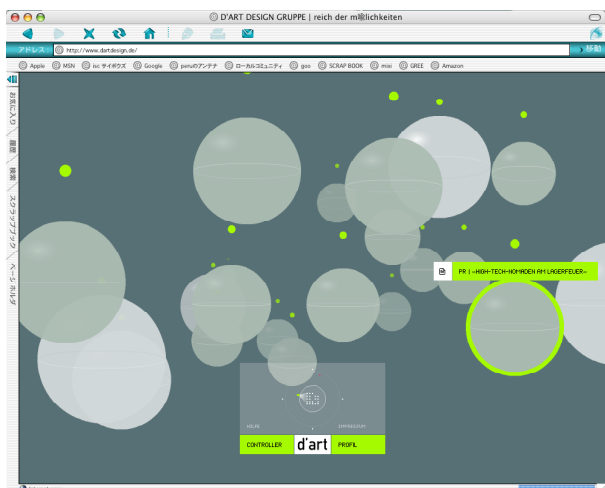


図2 D' Art Design Gruppe

3：デザイン支援ツールからデザイン代行ツールへ

3-1 機能特化の次に

ソフトウェアは機能が追加されて多機能になればなるほど、価格や操作感から初心者には敷居が高くなっていく傾向がある。例えばFlashは、初期の頃は直感的な操作によってアニメーションが非常に簡単に制作できるということが大きな特長であったが、バージョンアップの度に入門者にはわかりにくいソフトになっていった。そこでFlash Maker[22]のように、ごく簡単にアニメーションが作れるということだけに特化したソフトウェアが生まれたりもしている。年賀状用ソフトウェアと同様に、用途を絞りこむことによる使い勝手のよさを目指したあり方と言える。こういったWebデザイン用ソフトウェアの例としてFreeway[23]がある。これはテンプレートを書き換えながら作業を進める形であるが、一切ソースコードを出さないという徹底ぶりです。2Dソフトウェアのようにページを制作することが可能であり、ソースアレルギーを持つ初心者を支援する。

しかし、ここで重要なこととして、これらの提供するインターフェイスの「簡単に」が、いつのまにか作業の結

果まで「簡単に」へと、すなわち自動的にデザインする行為の主体の転換が行われがちな点を指摘しておきたい。実のところ、ユーザー側の体験としては、これらは明確に区別しにくいところがある。特にコンピュータ利用の経験が浅い人ほど、少ない手数で派手なことが出来ればその威力に騙されてしまいがちである。ソフトウェアは、作業を手助けする「デザイン支援」ツールから、作業を自分の代わりに自動的に進める「デザイン代行」ツールの意味を持ち始めたといえる。それらを以下の3つの代表的な事例から解説する。

3-2 ID for WEBLIFE

IDforWeblife[24]は、2004年7月にデジタルステージ社[25]が販売を開始した初心者でも極めて簡単にフルFlashサイトを制作できるオーサリングツールである。「ホームページ作りは、ここまでカンタンになります」のキャッチフレーズ通り、サイト制作の煩わしい工程を減らし極限まで易くしつつも楽しく演出しようとする製品コンセプトが、パッケージからソフトのインターフェイスに至る隅々にまで強く表れている [図3]。



図3 IDforWeblife

予め用意されている多数の「カートリッジ」(デザインテンプレート)に対して画像やテキストを差し替えたりしながら編集し、次々にページを追加してだけでサイトが完成されていく。事前にFTP設定さえしておけば、ボタン一つでアップロードも簡単である。直感的かつ派手な演出のユーザーインターフェイスも含め、まさにサイト制作のための環境をデザインしている事例といえるだろう。しかし、これによって作られたサイトは、発売当初の初期バージョンにおいては、パワーポイントのテーマや画面エフェクト等に通じる動きの均一さやナビゲーションのデ

フォルト感から、やや金太郎飴のような画一化された印象をもつ感は否めない。しかし、Flash 制作経験のある上級者であれば、無料で配布されている開発キットを使用してオリジナルな自作カートリッジを開発し、配布することもできるという点で拡張性も所有しており、一見は初心者向けを装いながら、上級者のユーザーを効果的に取り込みコミュニティとして普及することを狙っており、ビジネスモデルとしても計算されていることが伺える。デジタルステージ社は、VJ 人口を飛躍的に増やした MotionDive[26]や、デジカメ画像からムービーを簡単に制作する LifewithPhotoCinema[27]で知られるが、敷居をさげつつも若者らの表現欲求を強くくすぐるソフトウェア制作で異彩を放っている。技術的な障壁を極限まで減らし作ること自体を楽しませようというコンセプトは極めて画期的であり、いずれもソフトウェアとしてグッドデザイン賞を受賞している。代表の平野が言うように[28]、ページ自体を作る作業の時間を少しでも減らし、コンテンツの方に力を注げることは本来望ましいことであり、デジタルステージ社のみならずこのような切り口の展開は今後も期待されてよい。このソフトウェアは「デザイナーが隣にいたような感覚」と謳っているように、デザインを代行するような性格が色濃い。

3-3 theWorkFrame

洗練された Flash のインタラクションに特化した市販パッケージとして、デザイン会社 FICC inc[29]が販売している the WorkFrame[30]がある。これはキャッチコピーの”レディメイドウェブサイト”および、商品名である”WorkFrame”の名の通り、変数とスキンファイルで構成や見た目を自由にカスタマイズし、クオリティの高いフラッシュサイトを簡単に構築することができる。この製品は単独のソフトウェアではなく編集は Flash を使って行う。要するに一種のテンプレートであり、ベースになるレイアウトと動きがセットになった雛形を販売するものである。これによって制作されたサイト例がいくつか公式ウェブサイトで紹介されているが、傍目には全く既製品と感ぜさせない。例えば 2004 年秋に開催されたセントラルイースト東京(CET'04)の公式ウェブサイト[31]は、短期的なイベントであったが限られた制作時間の中で使用され、告知や参加アーティストの作品紹介に高い効果を上げている。またこのサイトの質の高さによってイベントのブランド性を上げているとも言えるだろう。特に Flash サイトで最も

質を左右するマウスイベントのインタラクションに関しては、心地よい動きが追求されており、強いインパクトを持つ[図 4]。



図 4 CET'04

このようなテンプレートとしての販売はまだストックフォトや画像素材集ほど根付いているとは言えないが徐々に広がっている模様である。例えばデザインエクステンジ社が運営する DEX WEB[32]では、同様にデザイナーが制作した Web デザインテンプレートや素材を豊富にダウンロード販売しており、他にも ThemeGarely[33]などが知られている。ここでは「型」を買っているだけであり、むしろこれまで各自の制作者が水面下でやっていたこと、例えば書籍を見ながらテクニックを流用したり、良くてきた Web サイトの効果を模倣したりして手に入れていたことが、需要と供給のニーズによって市販パッケージというかたちで顕在化し、売買されたようになっただけであるという見方もできる。

3-4 N-GEN

N-GEN[34]は、サンフランシスコのデザイン会社 MoveDesign[35]が制作した実験的ソフトウェアである。しばらくウェブサイトからデモバージョンを配布していたが、製品化のために配布は中止されている模様である(2005 年 2 月現在)。このソフトは、「デザインマシン」の名を冠している通り、適当なタイトル文字やサブタイトル文字をフォームに入力し「n-Generate」のボタンを押すと、クリック毎にその文字を素材にしたグラフィックスをどんどん自動生成していく。またメニューから違うモジュールを選択すると、まるでデザイナーが変わったかのように違う芸風の画像群を生成する。このモジュールが生成するデザインのそれぞれの「テイスト」のアルゴリズムを制御している。初めて触れた人は到底人力ではかなわないよう

な多様なテイストのバリエーションを、一つのモジュールが次々に生成し続けるのに大変驚かされる。またメニューバーから「view」を選択すると、Web ページだけではなくポスターやジャケットなどに対応したフォーマットに変更することも出来る。そして最後に気に入ったものが表れた際にはエクスポートすれば、素材用の画像として書き出すことが出来る [図 5]。

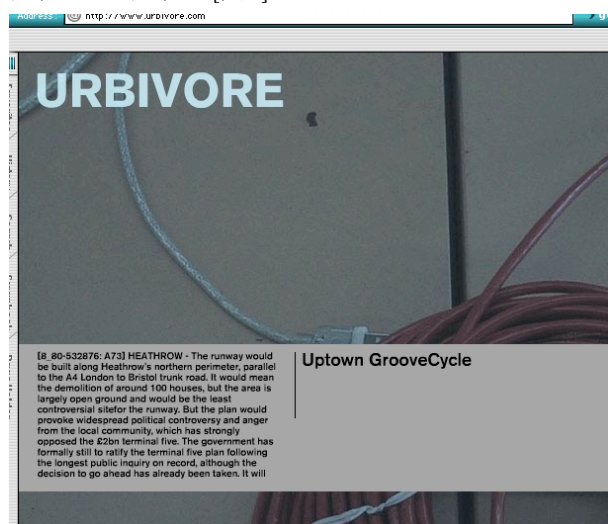


図 5 N-Gen 制作例

これは一種の自動生成ソフトウェアと言え、名刺デザインソフト[36]などにその類型が見られる。必要情報を入力すれば、いくつか予め準備されたアルゴリズムにそってアウトプットを生成していくという機能は似ている。それらのビジュアルを重視した発展形と位置づけられるだろう。また肝心のアウトプットの質であるが、プロの目として見れば細かい要素ごとのバランスに自動生成らしい緩さが目立つものの、普通の人が見ればそれほどの違和感はないと思われる。様々な人にソフトの動作画面を見せてその反応を観察したが、鑑賞に堪えるレベルのものが制作できると言えそうである。

なお、筆者が試用したデモバージョンでは画像のインポート等の機能は制限されていたが、これらの機能制限がない製品版であれば、より手軽に好みの素材を使った組み合わせのアウトプットを生成することが出来る。ここで人間側がすることは、クリックすることと生成された物に反応して気に入ったものを選ぶことだけである。現在はソフトウェアの存在自体に驚かされるような段階だが、このような実験的なソフトウェアは、やがて身近になるかもしれない。

4：デザインプロセスの変容についての考察

前章にて示した代行と支援の境界や、またテクノロジーによって自動化されたものがどこまで知を伴う「デザイン」なのか、については非常に見解が分かれるところであるが、このような傾向を持ついくつかのソフトウェアの事例を紹介した。これらはまだ一部の利用者に知られているのみであり、実用性という点では未成熟な点も多いかもしれないが、省労力という点で考えるとまもなく CMS のような自動化の流れとも合流するのではないかと予想される。したがってこれらのソフトウェアは現時点でのシェアは小さいとしても、今後に与える影響は以外に大きいのではないだろうか。

本章では、これらのソフトウェアによる支援と代行、およびそのプロセスを巡って、いくつかの立場による視点から考察する。

4-1 目的と手段

まず最初に、ポジティブな立場として、サイトをつくる技術（手段）と中身（目的）は別の話であり、手段の自動化によって、もっとも大事な目的に焦点を当て生産性を高めることができるようになるのは好ましいことである、という意見がある。IDforWeblife の宣伝[37]にもあるように、忙しい日々の中でもなるべく更新しやすいように、煩わしい手段をソフトが肩代わりするのであり、それが簡単になったからこそ、その分個人が出来ることの目的の選択肢が増えると言うことは間違いのないところであろう。第 2 章に述べたことと関連するが、Web デザインとは単にビジュアルの設計だけではない。Jeese James Garrett[38]は、Web デザイン時に考慮すべきユーザーエクスペリエンス[39]についてレイヤー構造が有ることを指摘し、それを階層図で表現しているが、[40] これによってもっとも具体的な VisualDesign からもっとも抽象的な UserNeeds の階層まで、Surface(表層),Skelton(骨格),Structure(構造),Scope(要件),Strategy(戦略)の 5 つの段階に分かれる [図 6]。

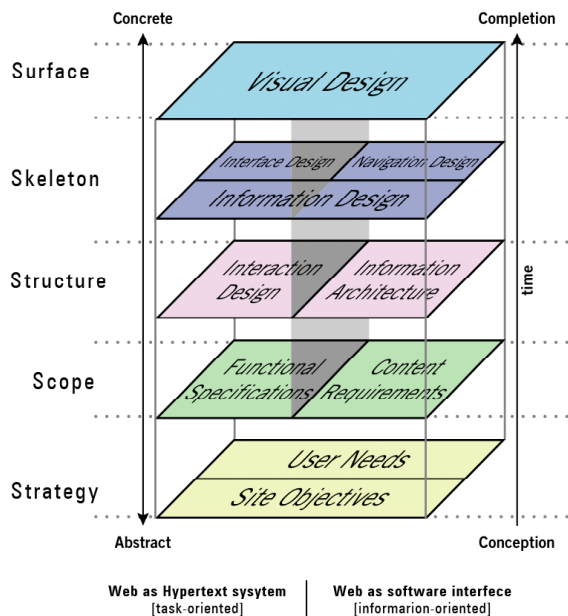


図 6 J.J.Garrett によるユーザーエクスペリエンスの階層図 (一部改変)

これに照らし合わせて考えてみると、これらのソフトウェアが代行するのはビジュアルデザインやナビゲーションであり、最上位の Surface やせいぜい次の Sklton を照射するにすぎない。実際のところ、具体的に目に見えやすい Surface の他にも計画しなければならないことは数多くある。例えば、図で3階層目に位置する情報アーキテクチャ[41][42]的な部分は、ユーザーが情報にアクセスする際に、使い手側の視点にたった文脈やサイト移動のわかりやすさなどを設計する大変重要なところであるが、こういった工程は大規模な商用サイトを除き、あまり重視されていないのが現状である。ビジュアルデザインの周辺の手段を簡略化することによって、これまで優先順位が低く焦点が当たりにくかったこれらの部分にも力を割けるようになることは意義深いことである、と言える。

やや本題から逸れるが、ページ構成を主としたこの情報アーキテクチャの部分を中心に支援するツールも開発されている。Ridual[43] は、野村総研が開発した XML ベースのサイトマップ構築ツールである。このソフトウェアでは、元になるテキストデータからサイトマップを自動的に数パターンに可視化する[図 7]。それによって、各ページのビジュアルの検討を行う前に、サイトの全体像や構造をグラフィカルに把握しながらページ構成やナビゲーション等の設計を行うことができるようにする。Web のビジュアル部分を扱うソフトウェアではなく、これまで手動でおこなっていたサイトマップ等の構成作業を支援するという重要なコンセプトを持ったツールである。

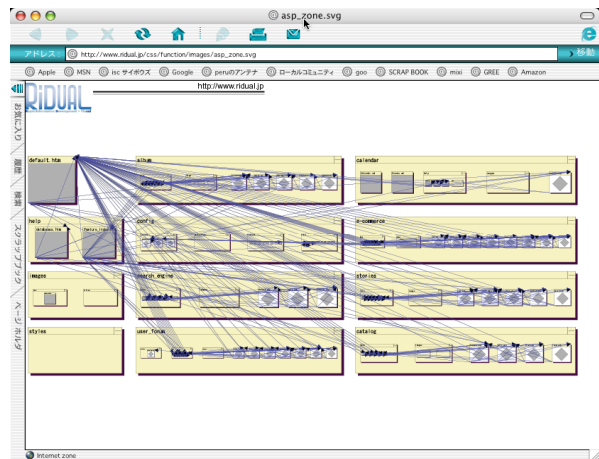


図 7 リンク構造を可視化した、Ridual が描くサイトマップの一例

4-2 ファッション化：つくることから選ぶ事へ

次に紹介する意見は、選択の多様化によって個人の価値観を体現しやすくなる、というものである。どこまで自動化されてもかつ選択肢が増えても、最終的には価値判断から答えを決定し情報を取り纏める個人の「センス」が逆に問われるわけであり、次第にそれは文化としての「ファッション」と近い状況になっていく。個人の体型などの制限がない分、よりその選択の自由度は高い。そして、洋服を作ることと洋服をチョイスするセンスにはほとんど関係がないように、Web のセンスも、ビジュアルデザインが作れる必要とは関係がなく、むしろ検討の機会が増えればそれだけ選択のセンスの質も上がる可能性は高い。このような Web の着衣に近いようなあり方は、一部の Web コミュニティではすでに普及している。例えば、日記サービス最大手のはてなダイアリー[44]でも利用されている tDiary.org[45]のテーマギャラリーを例にとってみる。これはフリーウェアとして多くの参加者によって支えられている CSS のテンプレート集であるが、すでに 200 種類程度まで増えている。



図 8 tDiary のテーマギャラリーを Zphoto[46]で一覧するツール[47]

様々なテイストの例が一同に集められており、ダウンロードして入れ替えれば手軽にビジュアルを切り替える事が出来る。作者は自分だけのものではなく人に使ってもらうためにファイルを提供する。ここでは、CSS ファイルがビジュアルデザインのための「部品」として扱われていると言える。このような例は、他では、MovableType における CSS 配布として Dr.Blog[48]が知られている。これらは、ユーザーのポリシーや好み、さらには気分を反映して、しばしば着せ替えられつつも本人の価値観にあわせて最適化される。このように、見栄えを一括で変更できる手軽なスキンチェンジは多くのブログサービス[49][50]や一部の Wiki クローン[51]などでもすでに広く提供されている。また大規模サイトでも、慶応 SFC の Web サイト[52] (制作: IMJ[51]) のように、ビジュアルデザインと中身の情報を分離することで実現した「デザインチェンジボタン」を持つサイトも現れている [53]。

4-3 テンプレートの先に

そして、これらを考える上で避けられないのが、ビジュアルデザインにおける一種のフォーマット性である。視覚言語[55]の組み合わせは理論上無限かもしれないが、人間が判別できる心地よさや分かりやすさを感じる組み合わせは、ある意味では有限であるという一面もある。そして優れたデザインには様々な原則が見え隠れする[56]。日々消費されるビジュアルのデザインは、実際のところ様々な類型に源流をもつバリエーションが多い。優れた一流デザイナーでも作品を続けて観察してみれば、作風に一種の規則性のようなものがあるように、創造のプロセスは神秘的なようにみえても過去にストックしている資産を効率的に流用したり交配したりしながら制作しているものである。また頻繁に盗作騒ぎが起こるのは、単純な真似だけではなく人間の潜在的な無意識の共通性に起因するという面もある[57]。こういった点を考慮すると、デザイン作業初期のプロトタイプ作成の段階で似たようなアイデアに悩まされる場合などには、発想支援としてそこから逃れるためにも多様な展開や組み替えを目指して要素のシャッフルを行うことは必要である。そのような検討材料の一つとしては、大いに利用すべきであろう。

なお、コンピュータを利用した自動生成によってアウトプットの幅を広げる実験的な試みは、音楽や文学などでもよく行われている[58][59][60][61]。

さらに言うと、これらのフォーマットやアルゴリズムを取り出して普遍的なものとして洗練させる事ができれば、

ソフトウェア開発における「デザインパターン」[62]のようなシステムへと展開される可能性は高いだろう[63]。デザインパターンとは、過去のソフトウェア設計者が発見し編み出した設計のノウハウを蓄積し、名前をつけ、再利用しやすいように特定の規約に従ってカタログ化したものである。Bob Klaassen は、用途別のウェブサイト設計において要素の類型をデザインパターンを用いて評価を行い、その資料を公開している[64]。

デザインは、単純に機能だけで判断される問題ではないので評価が難しいところもあるが、このように web デザインにパターンを導入することで、デザイナーの直感よりも設計のアプローチや検討材料を増やすことが可能になることは間違いのないところである。デザインパーツの部品化は、例えば古いところと言えば「フォント」などはそれぞれの作者のテイストを持ったものがパッケージ化され再利用されるように作られたものであったし、tDiary のテーマに見られるように、個人から離れて誰でも使えるような共有のものになったりする点で、すでに部分的には進みつつあったが、個人のアイデアに頼るほかはなかった非効率なデザイン作業が、徐々に外部知識化され、デジタルの特性を活かしたかたちとしての再利用可能な「部品化」のようなあり方が本格的に始まりかけていると言えるだろう。このような状況での Web デザインを行う人は、たとえて言うならばインテリアコーディネータのように、ニーズや目的にあわせて全体を把握し、的確に既製の部品を選択しながら問題解決を図るような立場になる。そしてこの選択ルーチンの手順を合理的にシステム化し、誰でも使えるようにしたものがソフトウェアということになる。

5. ソフトウェア工学におけるオブジェクト指向

このようにデザイン・ソフトウェアを考察していくと、ソフトウェア工学の影響と融合していることに気づく。ソフトウェアとは、開発工程をサイクリックに繰り返しながら進化していくものであるとする基本的な考え方に基いて、ソフトウェアの設計方法も大きく変革を遂げてきた。本章では、まずその変革の歴史を顧みることからはじめ、デザイン、特に技術革新やデザイン開発工程の早い Web デザインにおける流れと比較しながら将来を展望する。

5. 1 ソフトウェア設計法の歴史と基本的な考え方

1970年代の前半は、主にプログラミングのスタイルや方法論を通じてプログラムの構造が研究された時代である。主な設計法には以下のようなものがある[5-1]。

- ・ モジュール化法 (Pamas)
- ・ 段階的詳細化法 (Wirth)
- ・ ワーニエ法 (Warnier)
- ・ トップダウンプログラミング (Mills)
- ・ 構造的プログラミング (Dijkstra)
- ・ 複合設計 (Myers)
- ・ 構造化設計 (Constantine)

ここでは、これらの設計法に関しての詳細は割愛するが、これらのソフトウェア設計法では「構造」という語がよく登場する。すなわち、ソフトウェア技術者にとってプログラム構造と読み易さというものが、いかに関心が高いことであったかを示している [65]。1970年代の後半になると、構造化プログラミングの枠を越えて各種の設計法が提唱されるようになった。

- ・ ジャクソン法 (Jackson)
- ・ システム階層分割 (Ross)
- ・ トップダウン設計法 (Yourdon)

さらに1980年代になると、抽象化の概念から端を発したオブジェクト指向設計法 (Booch, Buhr) が脚光を浴びるようになった[66][67][68]。

一般に、小規模なシステムでは全体の見通しがよいので特別な設計法を使わなくても問題を解決することは可能である。したがって、特別な設計法が必要になるのは大規模なシステムを設計する場合である。大規模なシステムの設計では、全体の見通しをよくするため、通常次のような手順を踏んで設計作業が行われる。

第一段階：モジュール分割の設計

問題を複数の小さくて単純な構成要素に分解

第二段階：モジュール内の設計

分割して得られた構成要素の内部処理を設計

最初のモジュール分割の設計段階では、システムを階層化することによって全体を分割していく方法がとられる。大規模なシステムでは、一気に適度な大きさのモジュールに分割するのは無理なので、階層を作って段階的にサブシステムへと分割していく。このとき重要な点は、このサブシステムがそれぞれ独立した機能単位になるようにすることである。次いで、このサブシステムの機能を更に細かいサブシステムの機能へと分割を繰り返す[図9]。

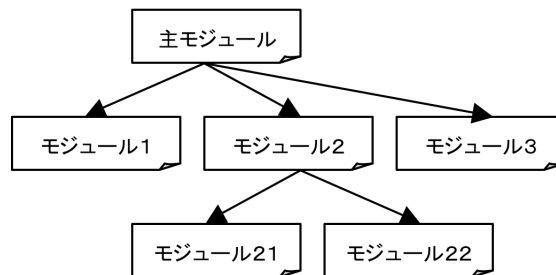


図9 サブシステム化によるモジュール分割

各モジュールの詳細設計の段階では、主にトップダウン設計法が使われる。これは、モジュールの主手続き部分を最上位の主プログラムとし、そこに含まれるべき内部処理の部分を複数の副プログラムに分割して呼び出す形式にする。ここで重要なのは、トップダウン設計という考え方は、必ずしも副プログラムに分割しない場合でも有効に使える点である。全体の処理を概念上で分解して考えることにより、複雑な内容をうまく詳細化して具体化することができる。

トップダウン設計に対して、ボトムアップというアプローチも存在する。ボトムアップの設計では、ある機能を実現するのに既存のプログラムを組み合わせることによって行う。大雑把に言えば、トップダウンは「分解」であり、ボトムアップは「統合」である。オブジェクト指向では、このボトムアップ的な考え方をとることが多い。

5-2 オブジェクト指向とオブジェクト思考

ソフトウェア工学においてオブジェクト指向という考え方が注目されるようになったのは、従来のソフトウェア開発方法では解決できない問題点が、次第に明らかになってきたからである。大規模なソフトウェアを開発する場合、従来の構造化技法を中心とするソフトウェア設計法では、いくら人手をかけても開発のスピードが上がらないという事態になってきた。ソフトウェア開発における生産性という面で行き詰まりを生じていたのである。そのような状況下で登場してきたのが、オブジェクト指向という設計方法である。

オブジェクト指向プログラミングを正しく理解するには、コンピュータ内部でのデータとその取り扱い方(動作)に関して、従来の考え方を捨てて発想の転換をはかる必要がある。

従来のプログラムでは、その動作を記述する場合、次の

ような気持ちでプログラムを作っていたはずである。つまり記憶装置の番地という「引き出し」に入っているデータを自分で取り出し、それを処理する手続きが置かれた別の番地へ運ぶ、という具合にして自分が記憶装置内部を動き回ってプログラムを実行していた。

ところが、オブジェクト指向ではプログラマは絶対的な権力を持つ管理者となる。自分はもはやメッセンジャーボーイではなく、有能な部下をたくさん持った管理者であると思う必要がある。管理者はメモリ内の引き出しを調べて右往左往する存在であってはならず、どっかりと座って命令を発してさえすればよいのである。

このオブジェクト指向という仕組みで重要な点は、データとそれを処理する手段であるメソッドはすべて受け手を持っていて、その詳細については管理者である自分は一切関知しないでよいという点である。ただ振る舞いとして出てくる結果のみを見ていればよい。これがオブジェクト思考である。

オブジェクト指向の技術を活用すると、操作のためのコードと、データ形式に関する情報を扱うコードとを分離することが可能となる。その結果、その部分をソフトウェア部品として取り扱い、他のプログラム中で、あるいは他のプロジェクトで再利用することが可能となる。しかも、このようにして作られた部品のプロトタイプが1つあれば、その複製をいくらかでも作ることが可能となる。この点が従来のライブラリサブルーチンの利用と著しく異なる。

ただし、ここで注意すべきは、オブジェクト指向の導入によりソフトウェア部品を作る手段が用意されたということであり、良い部品を作るにはそれ相応の努力が必要であることは言うまでもない。

5-3 デザインプロセスにおけるオブジェクト指向

ここまではソフトウェアにおける設計法に関して概観した。次にデザイン、特にグラフィックデザインにおけるプロセスを概観する。

ある著名なデザイナーが仕事を請けおひ、下絵の段階からひとり（もちろん、アシスタントがいる場合もあるであろう）で作品を仕上げた時代は、請け負った仕事のテーマからコンセプトを決定し、作品の全体を大まかにイメージするところから始め、徐々に詳細を決めて制作していくというプロセスが、一番単純な、しかしながら一番基本的なプロセスであった。これはソフトウェアのトップダウン設計法であることが出来る。すなわち、全体の概略から詳細に具体化していく「分解」というプロセスであ

る。ところが、デザイン要素（例えば写真であるとか、キャッチコピーであるとか）の専門性が高まってくるにしたがい、また、デザイン作業にパソコンが導入され生産性が求められるようになると、デザインプロセスにも変化が見られるようになった。現在のアートディレクターと呼ばれる人々は、作品の方向性を決定しラフなイメージをデザイナーやカメラマン、コピーライターなどに伝える。ラフなイメージを伝えられたそれぞれの部署では、そのイメージに合致するようにアウトプットを制作し、最終的にそれらのアウトプットを統合して作品とする。これは、ソフトウェア設計法でいうところのボトムアップ的なプロセスである。アートディレクターという職能は、ただ出てくる結果を最終作品に統合する役割を担うのであり、オブジェクト指向的立場であると言える。また、デザイン作業がひとりの作家性に頼っていた時代から、徐々に複数の人間がプロジェクトチームを組むようなグループワークに変化していった点もソフトウェア開発と似ている点であろう。

しかしながら、Web デザインのような「普遍性」よりも「生産性・新規性」が求められる分野では、人的・時間的にさらに制限が強くなってきている。そこで第4章までに述べたデザイン・ソフトウェアが誕生した、と見ることが出来る。これらは、グラフィックデザインの分野が長年培ったアピランスに関する法則[69]に基づいて、デザイン要素を部品化しユーザーに部品を統合させるというオブジェクト指向的なデザインプロセスを提供している。

またグラフィックデザインに対して、近年非常に注目されているデザイン分野に情報デザインがある[70]。情報デザインでは、グラフィックデザインがアピランス、つまりビジュアルを重視するのに対して、コミュニケーションを重視する[71]。Web サイトは制作者（情報提供者）と利用者とのインタラクションによるコミュニケーションの場と考えることができる。そこでは新しい情報を即時に提供する生産性や新規性、そしてユーザビリティなど、グラフィックデザインにはない視点が重視される。デザイン・ソフトウェアではそのためのコンテンツ内容に関するデザインや、サイト全体でのコンテキストデザインにおいては支援することはない。それは制作者に委ねられ、逆に言えばこの部分に制作者が注力できるように支援するのである。

6：ツールと創造性の関係

本章では、これらのソフトウェアやシステムの利用や普及が進む先に、Web デザインにおける作り手側の創造性の領域について、その問題点を挙げながら考察する。

6-1 作ることの価値

否定的な意見を見てみると、もっとも多い反応が同一のアウトプットによる「没個性化」ということである。だれもが同じ OS で同じソフトを使い、同じ処理をすれば、当然出来あがってくるものもまったく同じになる。人間の感覚は不思議なもので、既製のものに対してその製品を知っていれば逆にマイナスの方向に印象づけられてしまうところがある。例えばウェブ探索中に偶然たどり着いた IDforWeblife 製のサイトから流れるデフォルトのアニメーションは、初めて見る人と、ソフトを使ったことのある人とは、相当印象が違っただろう。PowerPoint を見慣れている人にとって PowerPoint の画面エフェクトに好感を抱く人は少ないのと同じように、こういった既製の効果に対して、初めて見たときは目を引かれたものの、すぐに否定的な反応に変わる人は多い。それらの意見をピックアップしてみると「表面的に整っているとしても、それが人と同じになってしまうのは嫌だ」「既製品を使っていると思われたくない」「多少拙くても個人サイトは自分なりにハンドメイドで作りたい」などと、自分で作ることに對しての付加価値を見出そうとする立場の意見が目立つ。個人の価値観を体現する個人 Web サイトの場合と、他人や組織から Web 制作を受注する場合とは、切り分けて考えなければ話が混同されてしまいがちであるが、この付加価値の原因を推測すると「料理」と共通するものがあるように思える。例えば、本当は味が落ちるとしても、外注するよりも手作りの方が美味しく感じることはある。逆にアウトソーシングの仕方によっては印象が落ちることもある。また、作ること自体を楽しんでいる人にはそもそも自動化は必要のない話となる。デザインの場合は論理以外の部分で評価される部分もあり、単純に合理性だけでは語れない点に関しては、こういったソフトウェアによるアウトプットの一つの限界が見える。

6-2 ツールと主体

次に重要なこととして、これらのテンプレートやツールの機能が逆に人間が行うことを決め始めるといふ、創造性がツールによって規定される危険性を挙げておきたい。これはデジタルの危険として以前から言われ続けている[72]

が、これは要素をコントロールするだけの感覚が発達していない人間が道具を使う際の歪みというべきものかもしれない。通常、物作りを学びながら上達していく過程には、同時に試行錯誤による経験の蓄積がベースとして存在する。例えばタイトルロゴ一つを画面に置くにしても、多くの下積みを支えられた繊細なバランス感覚に因って最適な位置は自然に導かれるのであり、そういった繊細な感覚に基づく身体知がプロとしての技術的特権性へと繋がっている。ここで適切な調整を行うだけの判断力が育っていない人間が、一足先に便利さに頼る以上、ツールに「使われて」しまうことになるのはある意味では仕方がないとも言える。また、失敗から傑作まで起こりうるアナログ制作の多様性に比べ、N-Gen のようなソフトウェアを使用した場合、生産性は保証される代わりに、アウトプットの幅は非常に小さくなる。いくら生成ボタンをクリックし続けても、平均的なクオリティから向上していくことは、おそらくない。

こういった観点からみれば、身体活動を伴わず作れてしまう点や、試行錯誤の経験が完全に欠落している点で、細かく見る感覚が弱まっていくという傾向は否めないとこころである。

しかし上述したように、作業が自動化されたからこそ代わりに可能になった事も多いため、今更テクノロジーを忌避するのは現実的ではない、というのが一般的な見方だろう。ツールによる利便性と創造性の関係は、単純な善悪二元論で判断せず、慎重になる必要がある。このバランスについては今後も多くの検討を行わなくてはならない。これらの問題はデザインに限らず、現代の社会一般に通底している点が多い。

6-3 作る側と使う側

20 世紀終盤頃には、技術的特権性を基にしたかつてのヒエラルキーは IT 革命によって破壊される、と盛んに言われた。実際、Web デザインの草創期には混沌とした状況の中で、誰もが同じ環境にあり、プロとアマチュアがほとんど同じような環境で試行錯誤しながら制作に励んでいた。個人が先進企業を凌ぐ質のものを作ることも多く、敷居は低くなりクリエイターの裾野は格段に広げられた。しかし、現在、ノウハウが蓄積されツールが高度化していくにつれ、もうひとつのヒエラルキー、ツールを作る側と使う側の階層が再び引き起こされつつあるとも言える[図 10]。



図10 階層の変化イメージ図（数居の先に再び生まれつつある階層）

このようなデザイン・ソフトウェアが台頭することで、なおそのような傾向が強まることは大いに予想される。例えば3章で挙げた the WorkFrame のような市販パッケージ製品を相手に、10 倍以上のコスト差を跳ね返すほどのクオリティを提示できるデザイナーは相当絞られてしまう。したがってプロフェッショナルとして戦えるのはやはり数少ない精鋭だけであるという当然の事実が導かれる。おそらくその比率は近代以前とそれほど変わらないだろうと予測される。現在のメディアアートを牽引する者がそうである[73]ように、プログラミング能力とデザイン能力を併せ持った者だけがデジタル環境における真の創造性として、一種の「神の言語」を手に入れるということも言えるかもしれない。

これらの階層はどこまで強力なのだろうか。大抵の人はこの壁を意識化すると自らの創造性の限界に考え込み立ち止まってしまう。しかし、Web デザイン界の第一人者である中村勇吾[74]は言う。

「昔、ジョン前田[75]が言っていたんですけど、Photoshop や illustrator みたいなアプリケーション使ったものを作っている人は、そのアプリケーションを作った人の夢の中でたださまよっているだけである、という話があって、夢をさまよう人と夢のもとを作る人っていう二元論があるんですけど、僕はそう思ってないんです。もちろん、ある人が敷いたプラットフォームってのがあって、その上でやるんですけど、でもそれは別にその人の夢の中のような閉じられた感じではなくて、彼に敷いてもらったところからどこまで行けるかっていう、それがその人との勝負だって感じがあるんです。環境を作る側か、その環境を盲目的に追従する側か、どっちか極端に行こうとする人が多い。極端に真面目な人は『やっぱり俺はアプリケーションを作る側にならなくてはならないそれこそが真のデザイン』とか思ったり。みんなその両極を向いてて中間で面白いところが実はあんまり目立たない」[76]

このような、ツールを出発点にした先の創造性こそが大事なのだという意見は傾聴に値する。ヴィレム・フルッサーは、プログラムが決められているチェスゲームで遊ぶブ

レイヤーを引き合いに出しながら、写真装置と自由に戯れる写真家の立場に、システムとそこに従属する人間が一体になった新しい創造の可能性を示した[77]が、中村の発言はフルッサーとほぼ同じ視点であることがわかる。中村の制作手法とその作品は、マクロメディアの開発者にも Flash の可能性について衝撃を与え、その後の Flash の方向性にも強い影響を与えたという。ソフトウェアの進む方向性を、それを作る人ではなく創造的に使う側が指し示すこともあるという事例である。ここには、ソフトウェアが生み出した物に満足しているのとは 180 度逆の姿勢がある。マクロメディアは、このような点を早くから考えており、使いやすさを求めるだけではなく、開発の初期段階から世界各地のヘビーユーザーの声を聞き、ツールとユーザーのコミュニティ的な広がりを意識した戦略をとっている[78]が、そういったツール開発側と向かい合うためにもユーザー側の持つ創造性は重要になっていくだろう。

7：むすび

以上、Web デザインにおける制作プロセスの変容を見てきた。本稿で取り上げたような問題は、技術が成熟していく段階での過渡期の出来事なのかもしれない。しかしこうして見てみると、すでに 90 年代のそれとは大きく変化してしまっていることは明らかである。アナログと同様に一点受注の形式で成り立っていた Web デザインも、デジタルの特性により近い形に移行し始めたと言え、このような流れは省労力と低コストもあり今後もより高速に発展していくと考えられる。これらに対して多様な意見があることは論を待たないが、一定の質が普及することによって一般化し、文化として底上げされるような効果も有るのではないか。その地平の先で本当に才能ある者は、必ずそこから一段階上の創造を行うだろう。また、Web の世界は、表現も技術も可能性はまだ未知数であるが、どれほど技術が進化しようとも見るのは人間であり、そのサイトの目的を考えるのも人間である。その意味は今後も変わることはない [79] 。

そして、これまでの事例と考察を元にして判断すると、Web デザインの今後の傾向としては、これまで混在していた、組み立てる (Build) と創造する (Create) が、インテリアやファッションの分野と同じように、ある程度棲み分けしはじめるのではないかと予測できる。サイトのニーズに合わせるように既製の部品を組み合わせるスピーディかつ効率的に作る方向と、それで満足できないような

場合、より品質や機能を重視したオーダーメイドの方向と分担していくことで、きめ細かなそれぞれの目的や用途にあわせたサイト構築が可能になるのではないだろうか。

このような流れの兆候に対して、Web 業界からも示唆的なコメントを見ることも増えてきた。前述した Ridual を開発した三井英樹は、高度な仕事の割に報われない Web 業界の疲弊した状況を憂いながら、「これだけ単なる情報共有型サイトが手軽に構築できる時代になったからには、それ以外のサイトには「機能」が要求される。＜中略＞もはや趣味的な延長線や個人プレーでは作り得ない領域に入っている。きちんと分業体制の組める知識集約型の組織が必須だ」とし、「Web 開発も扱う金額に対して比例する保証サービスのなものに変わっていくべきなのかもしれない」[80]と今後の展望を述べており、業界として生き残るために、より顧客サービスを高めて専門化する方向へ向かう考えを表明している。

こういった点をふまえると、教育側としてもこれまでのような目的と手段が混在したような草創期のままだに近い Web 制作の演習なども根本から検討し直す必要があるだろう。残念ながら筆者らが属する情報メディア教育の世界では未だ技術主体が多く、学びの手法も模索中である。そしてまた重要な「目的」に対しても、明確な指導や評価も確立していないまま共に変化への対応も求められ、検討すべきことは山積みとなっている。

なお、本稿のきっかけとなったのは、筆者らが CMS ツールに触れはじめた体験にあわせ、情報学部で出会った一部の大学生らの Web 制作のスタイルであった。インターネットから既製のパーツをダウンロードして抵抗無く器用に組みあわせて作っていく姿は、従来のデザイン教育で行われてきた制作手法とはまるで異なっており、そのあり方について考えさせられた経験から始まった。それに対して明確な答えを出せるほどの言葉は未だ持ち合わせていないが、現在の Web 制作プロセスの概観を掴むために、おぼろげながら整理を試みた次第である。同様の問題はあちらこちらで起こっていると思われる。本稿が微力ながら手助けになれば幸いである。

参考文献、参考 Website

- [1] Information Management:A Proposal. Tim Berners-Lee,
<http://www.w3.org/History/1989/>
 [2] World Wide Web Consortium, <http://w3c.org>
 [3] Cascading Style Sheets, level 1,

- <http://www.w3.org/TR/REC-CSS1>
 [4]-進化する Web- セマンティック Web 入門
 齋藤信男、萩野達也監訳 ジャストシステム 2004
 [5] Throwing Tables Out the Window, Douglas Bowman
http://www.stopdesign.com/articles/throwing_tables/
 [6]Macromedia DreamWeaver
<http://www.macromedia.com/jp/software/dreamweaver/>
 [7]Adobe Golive
<http://www.adobe.co.jp/products/golive/main.html>
 [8]IBM HomepageBuilder
<http://www-6.ibm.com/jp/software/internet/hpb/>
 [9]Microsoft Frontpage
<http://www.microsoft.com/japan/office/frontpage/prodinfo/default.msp>
 [10] Macromedia FlashMX2004
<http://www.macromedia.com/jp/software/flash/>
 [11] 「Web デザインワークフロー」 生田昌弘他 ソフトバンクパブリッシング 2003
 [12]Wiki <http://c2.com/cgi/wiki?WikiWikiWeb>
 [13]Wikipedia <http://ja.wikipedia.org/>
 [14] 「Web コンテンツマネジメントシステム入門--共同作業としての Web サイトの構築と運営」 ラッセルナカノ (著)、インフォデザイン+ソシオメディア (訳) ピアソン・エデュケーション 2002
 [15]Movable Type <http://www.sixapart.com/movabletype/>
 [16]Xoops <http://jp.xoops.org/>
 [17]Plone <http://plone.org/>
 [18]Noren <http://www.ashisuto.co.jp/prod/noren/>
 [19]WebDesigner2.0 進歩し続ける Web デザイナーの考え方 長谷川恭久 ソシム 2005
 [20]group94 <http://www.group94.com/>
 [21]D' Art Design Gruppe <http://www.dartdesign.de/>
 [22]FlashMaker <http://www.ifour.co.jp/product/flashmaker2/>
 [23]FreeWay <http://www.act2.co.jp/freeway/>
 [24]ID for WebLIFE
<http://www.digitalstage.net/jp/product/id/index.html>
 [25]DigitalStage <http://www.digitalstage.net/>
 [26]MotionDive
<http://www.digitalstage.net/product/motiondive/index.htm>
 [27]LifewithPhotoCinema
<http://www.digitalstage.net/jp/product/life/index.html>
 [28]ID 公式サイト内「これからのなし」
<http://www.digitalstage.net/jp/product/id/>

- [29]FICC Inc. <http://www.ficc.jp/>
- [30]TheWorkFrame <http://www.theworkframe.com/jp/>
- [31]CET' 04 <http://www.centraleasttokyo.com/>
- [32]DEX WEB <http://www.dex.ne.jp/>
- [33]ThemeGarely <http://www.themegallery.com/>
- [34]n-Gen <http://www.n-generate.com/>
- [35]Move Design <http://www.movedesign.com/>
- [36]アジェンダ 名刺制作ソフト一覧
<http://www.agenda.co.jp/products/>
- [37] <http://www.digitalstage.net/jp/product/id/>
- [38] <http://blog.jig.net/>
- [39]デザイン領域の新たな広がる広がりとしての Experience Design
- 「モノ」から「コト」、そして「Experience」へ広がるデザイン領域 - 菊池他 芸術科学会論文誌 3巻第1号 pp. 35-44
- [40] “The Elements of User Experience
Jeese James Garrett, New Riders Publishing
- [41]「Web 情報アーキテクチャ 最適なサイト構築のための論理的アプローチ」 Louis Rosenfeld, Peter Morville 篠原稔和 監訳
オライリージャパン 2003
- [42]「それは情報ではない」リチャード・ソール・ワーマン MdN
コーポレーション 2001
- [43]Ridual <http://www.ridual.jp/>
- [44]はてなダイアリー <http://d.hatena.ne.jp/>
- [45]tDairy <http://www.tdiary.org/>
- [46] Zphoto <http://namazu.org/~satoru/zphoto/>
- [47] <http://tdiary2.tdiary.net/theme/zphoto.swf>
- [48] Dr.Blog <http://drblog.jp/>
- [49]LivedoorBlog <http://portal.blog.livedoor.com/>
- [50]Ameba blog <http://ameblo.jp/>
- [51]PukiWiki <http://pukiwiki.org/>
- [52] 慶応義塾大学 SFC キャンパス <http://www.sfc.keio.ac.jp/>
- [53] IMJ <http://www.imjp.co.jp/>
- [54]WebDesigning 2004/8月号 p048
- [55]視覚言語 Langage of Vision G・ケベッシュ グラフィック社 1973
- [56]「ハッカーと画家 コンピュータ時代の創造者たち」
PaulGraham 著 川合史郎監訳 オーム社 2005 p139
- [57]「色彩と感性のポリフォニー」小町谷朝生 勁草書房 1991
- [58]音楽研究所: 自動作曲研究所
<http://www.asahi-net.or.jp/~HB9T-KTD/music/Japan/Research/AlgoComp/index.html>
- [59]Cut me Up!
<http://ux01.so-net.ne.jp/~ev-net/cutup/cutup.html>
- [60]DadaDodo <http://www.jwz.org/dadadodo/>
- [61] Webcollage <http://www.jwz.org/webcollage/>
- [62]「オブジェクト指向における再利用のためのデザインパターン」エリック・ガンマ他(著) 本位田 真一 (訳)
ソフトバンクパブリッシング 1999
- [63]「interCommunication No.47 CODE/コモンズ/ソフトウェア」
NTT 出版 2004
- [64] Welie.com <http://www.welie.com/patterns/index.html>
- [65]「C++プログラマのためのオブジェクト指向入門」 木下 恂
サイエンス社 1993
- [66]「オブジェクト指向プログラミング入門」 Timothy A. Budd
トッパン 1992
- [67]「C++ Programming Style」 Tom Cargill Addison Wesley
1992
- [68]「TURBO C++ for Windows Inside & Out」 Herbert Schildt
Osborne McGraw-Hill 1992
- [69] Kinross R., Conversation with Richard Hollis on graphic design history, Journal of Design History 5 (1), pp.73-90, 1992.
- [70] Web ナビゲーション技術にみる情報デザイン・情報視覚化の最近の動向・菊池他 芸術科学会論文誌 Vo.4, No.1, pp.1-12
- [71] Sless D., What is information design?, Designing Information for People, pp.1-16, Canberra: Communication Research Press., 1994.
- [72]Out of Design 江並直美 美術出版社 1998
- [73]「アートとコンピュータ」藤幡正樹 慶応義塾大学出版会 1999
- [74]<http://www.yugop.com/>
- [75]<http://maedastudio.com/>
- [76] 季刊 des マクロメディア特集 翔泳社 2004 p82
- [77]「写真の哲学のために」 ヴィレム・フルッサー(著) 深川雅文 (訳) 勁草書房 1999
- [78]季刊 des マクロメディア特集 翔泳社 2004
- [79]「Web ワールドのすべて---制作・管理・運営・インターネット決済サービスまで---」 佐藤嗣、上平崇仁他 クレオ 2003
- [80]Web 開発の今後
<http://www.ridual.jp/cgi-bin/Column/column.cgi?no=77&reno=no&oya=77&mode=message&page=0>