GUI を再発明する：ポスト・ウィンドウ・インターフェイス序論

久保田晃弘[†]　宮崎光弘[‡]　永原康史[§]　松田純一[¶]　大谷和利[‖]
杉山久仁彦[**]

2009 年 1 月 30 日

1　GUI の起源

今日の PDA(携帯情報端末) や携帯電話、デジタル AV 機器や家庭用ゲーム機を含むパーソナル・コンピュータのほとんどは、ユーザー・インターフェイス、特にグラフィカル・ユーザー・インターフェイス (GUI: Graphical User Interface) を介して操作する [1]。その起源は、古くは 1963 年にアメリカ空軍が開発した半自動式防空管制組織 (SAGE) という防空管制システムや、同じく 1960 年代にスタンフォード研究所 (SRI) Augmentation Research Center (ARC) のダグラス・エンゲルバートのチームが設計・開発した NLS(oNLine System) に遡ることができる。民間レベルにおいては、70 年代に米 Xerox 社の PARC 研究所において、アラン・ケイによる理想の個人向けコンピュータ「Dynabook」の機能の一部が、暫定ハードウエアとしての「Alto」と暫定システム・ソフトウエアとしての「Smalltalk」の組み合わせによって実装され、これが今日 Mac OS や Windows の普及によって広く知られる GUI の直接的な起源となっている [図 1]。

† 多摩美術大学 情報デザイン学科 情報芸術コース 教授
‡ 多摩美術大学 情報デザイン学科 情報デザインコース 教授
§ 多摩美術大学 情報デザイン学科 情報デザインコース 教授
¶ Mac テクノロジー研究所 主宰
‖ テクノロジー・ライター
** 多摩美術大学 造形表現学部 デザイン学科 非常勤講師

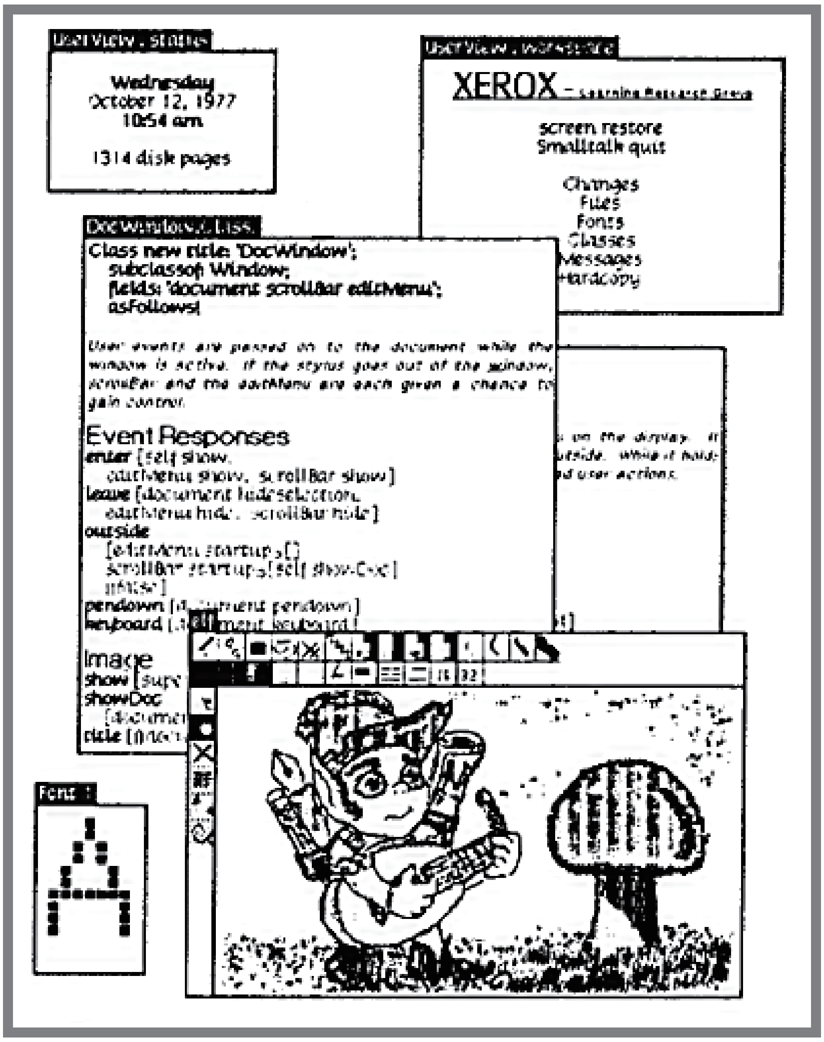

図 1　1977 年当時の暫定 Dynabook のスクリーンショット [2]

2　GUI の基本要素

今日の GUI は、オーバーラップするウィンドウとそのマウスによる操作 (移動、サイズ変更、内容のスクロール、アイコン化) や、同じくマウスによるポップアップメニューやプルダウンメニューによるアイテムの選択によって特徴付けられるが、それらを総称して WIMP インターフェイスと呼ぶ。 WIMP は Window、Icon、Menu、Pointing device の頭文字による略称で、GUI を構成する基本要素である。前述の Xerox Alto 以降制作されたさまざまな GUl が、例えば [3] や [4] にアーカイヴされているが、一見多様な様相を見せるこれらの GUI 群も、基本的にはこの 4 つの基本要素の組合せで構成されていることに変わりはない。

3 GUI のデザイン原則

1984 年に発表された Apple 社の Macintosh(Mac OS) のデスクトップが今なお GUI の典型 (exemplar) として参照されるのは、その完成度の高さや普及によるだけでなく、同社の Human Interface Group と Technical Publications Group が 1987 年に「Human Interface Guidelines: The Apple Desktop Interface」[5] を出版したことが大きい。Apple 社はコンピュータが持つ多彩な能力と機能を、コンピュータの非専門家が容易に利用できるようにするための布石として「Apple Desktop Interface」を位置づけ、Macintosh はこの Apple Desktop Interface が最も高い完成度でインプリメンテーションされた機種だと捉えた。Macintosh のインターフェイスとして Apple Desktop Interface があるのではなく、 Apple Desktop Interface を実現するために Macintosh というハードウェアがあることが肝要である。この時点で Apple 社はすでに、ハードウェアではなく、ソフトウェア中心主義を宣言していた。

この本で最も有名なのは、「Apple Desktop Interface の基本的な 10 原則」としてあげられたインターフェイスの一般的なデザイン原則 (General design principles) である。

比喩の使用 Metaphors from the real world
操作の直截性 Direct Manipulation
見たものを指示する (憶えてキー操作するかわりに) See-and-Point (instead of remember-and-type)
一貫性 Consistency
スクリーンで見たままをプリントする WYSIWYG (what you see is what you get)
ユーザによるコントロール User Control
フィードバックとダイアログ Feedback and Dialog
寛容性 Forgiveness
安定度 Perceived Stability
美的完成度 Aesthetic Integrity

このように、Apple 社は単に製品を開発・販売するだけでなく、その背景にある思想や原則と具体的な実現方法を記述することで、それまでの GUI デザインの試行錯誤を総括し、同時にその後の GUI デザインのスタイルを方向付けた。同時に、自動車や楽器のインターフェイス同様に、サードパーティーを含むさまざまなアプリケーション間で、操作の一貫性を保つことができるようになった。その後の OS のアップデートにもかかわらずインターフェイスの基本的な構造がほとんど変っていないことが、このデザインの完成度の高さを実証している [図 2]。

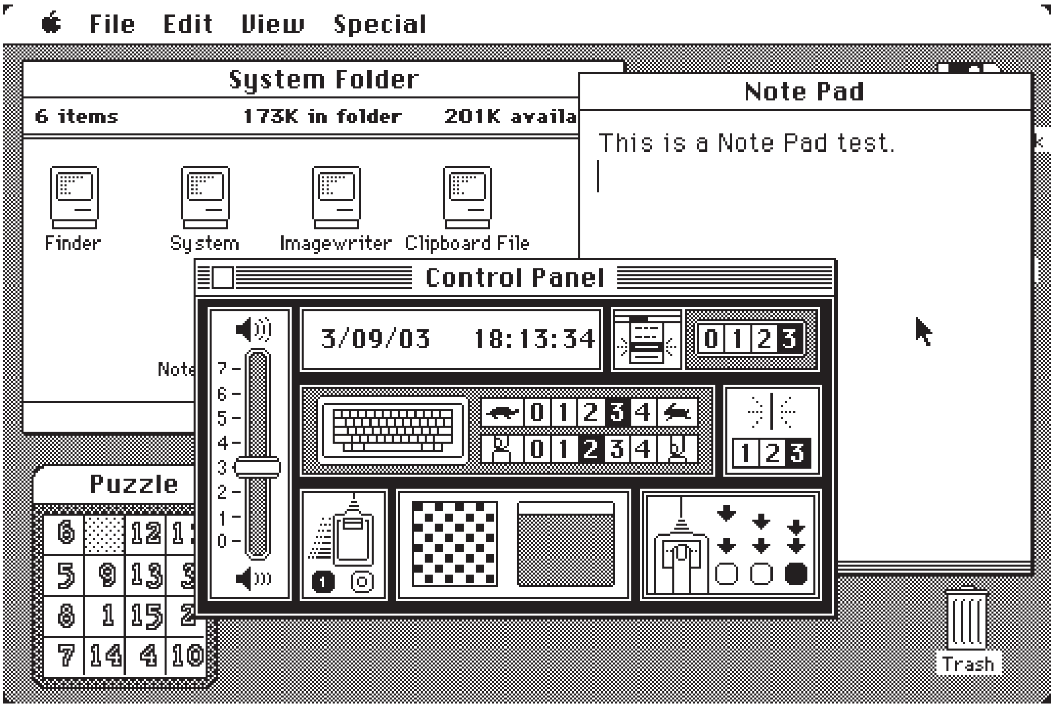

図 2　Macintosh System1.1 (1984) の Apple Desktop Interface[4]

4　デザイン原則の展開

87 年に発表された GUI のデザイン原則 (ガイドライン) は、その後も 93 年の「Macintosh Human Interface Guidelines」[6] や今日の OS X における「Apple Human Interface Guidelines」[7] へと改訂されつつ、Macintosh 以外のコンピュータや OS にも展開していった。代表的なものとしては、Apple 社に解雇されたスティーブ・ジョブズが創立した NEXT 社により開発された NEXTSTEP のガイドライン [8]、Microsoft 社の Windows 95 [9] や Windows Vista [10] のガイドライン、X Window System 上のデスクトップ環境である OSF/Motif や GNOME のガイドライン [11][12] などがある。同様に、PDA や携帯電話である Apple 社の Newton [13] や iPhone [14]、Palm 社の Palm OS [15]、Microsoft 社の Pocket PC [16] などにもそれぞれ、インターフェイスのガイドラインが設けられている。

これらのうちいくつかを一覧表にまとめると、表 1 のようになる。各々のデザイン原則に細かな項目や表現の違いはあるが、「Metaphors/Naturalness」「Direct Manipulation」「Consistency」「User Control」「Feedback」「Aesthetics/Simplicity」といったキーワードは、ほとんどのガイドラインに共通していると同時に、こうした共通のキーワードが今日の GUI を象徴している。

表 1　代表的なインターフェイス・デザイン原則の比較

Apple (1987) [5]	Apple (1992) [6]	Apple (2008) [7]
Metaphors	Metaphors	Metaphors
Direct Manipulation	Direct Manipulation	Direct Manipulation
See-and-Point	See-and-Point	Reflect the User's Mental Model
		Explicit and Implied Actions
Consistency	Consistency	Consistency
WYSIWYG	WYSIWYG	WYSIWYG
User Control	User Control	User Control
	Modelessness	Modelessness
Feedback and Dialog	Feedback and Dialog	Feedback and Communication
Forgiveness	Forgiveness	Forgiveness
Perceived Stability	Perceived Stability	Perceived Stability
Aesthetic Integrity	Aesthetic Integrity	Aesthetic Integrity
		Managing Complexity in Your Software

Next[8]	Microsoft[9]	GNOME[12]
		Design for People
		Don't Limit Your User Base (Accessibility/Internationalization and Localization)
Naturalness		Create a Match Between Your Application and the Real World
Direct Manipulation	Directness	Provide Direct Manipulation
Targeted Action		
Modal Tool		
Consistency	Consistency	Make Your Application Consistent
User Control (Modes/Action for the User)	User in Control	Put the User in Control
Using Mouse		
	Feedback	Keep the User Informed
	Forgiveness	Forgive the User
	Aesthetics	
	Simplicity	Keep It Simple and Pretty

5 The Anti-Mac Interface

Apple 社が Human Interface Guidelines を発表してからおおよそ 10 年後の 1996 年、当時サン・マイクロシステムズ社に所属していた Don Gentner と Jakob Nielsen が「The Anti-Mac Interface」と題した論文を発表した [17]。この論文で Gentner らは、Macintosh の GUI がデザインされた際には、

- コンピュータを使用したことがない初心者を対象としていた。
- (映像や音響制作ではなく) オフィスワークを対象としていた。
- ハードウェアのスペックがまだ貧弱であった[*1]。
- スタンドアロンでの使用を前提としていた。

といった制約 (環境) があったが、ユーザの熟練、ハードウェアの高性能化やインターネットの普及などにより、インターフェイスのデザイン原則が表 2 のように変化していくべきだと主張し、それを Anti-Mac インターフェイスと呼んだ。

表 2　Anti-Mac デザイン原則

Mac	Anti-Mac
Metaphors	Reality
Direct Manipulation	Delegation
See and Point	Describe and Command
Consistency	Diversity
WYSIWYG	Represent Meaning
User Control	Shared Control
Feedback and Dialog	System Handles Details
Forgiveness	Model User Actions
Aesthetic Integrity	Graphic Variety
Modelessness	Richer Cues

[*1] 初代 Macintosh のハードウェアのスペックは、128K のメモリ、クロック 7.8336MHz のモトローラ 68000CPU、512 × 342 のモノクロ 2 階調ディスプレイ、400KB フロッピーディスクドライブ等であった。

Gentner らの主張は Apple 社のデザイン原則を批判することが目的なのではなく、環境や時代の変化によって、インターフェイスのデザインがどのように変化していくかを予測しようとしたものである。確かに、人間のスキルや感覚が環境に影響されながら、時代と共に変化していくものであるとすれば、インターフェイスのデザイン原則もそれに伴なって、常にアップデートし続けていくのが自然である。

この論文が発表されてから 10 年後の現在、この提案を再検討してみると、その内のいくつかは、すでに身近なものとして実現していることがわかる。例えば、GUI という新しいメディアが登場した際に、それを何らかの既存のメディアの比喩 (Metaphor) を介して受容することは、クリエイターとユーザーの双方において必要不可欠であるが、そうした既存のメディアのメタファーの枠を越えた、新たなメディア固有の特質や表現が、メディアの成熟と共に社会の中で共有されていく過程を経ることで、今日ではコンピュータ上のデスクトップを、現実世界のデスクトップと対応させて理解したり使用している人は、むしろ少数となった。コンピュータのデスクトップはコンピュータのデスクトップであり、何か他のものの比喩として存在しているのではなく、今や固有のメディア/実体 (Reality) として、広く社会の中で認知、使用されているというべきである。

同様に、膨大なインターネット上のデータを操作するのに、See and Point による直接操作はとうてい不可能であり、サーチエンジンに代表されるように、言語によるタスクの委譲 (Delegation) が今では日常茶飯事のように行われている。同時にブログサイトや画像・動画共有サイト、ソーシャル・ブックマーク・サイトを含む SNS(ソーシャル・ネットワーキング・サービス) は、ここでいう Shared Control そのものであり、その背後ではシステムによる検索や分類といったさまざまな操作や、ユーザーのモデル化が行われている。今日では、インターネット上で検索や購買など何かを行えば、それを誰かが必ず管理、監視していることの利便性ではなく弊害の方が、むしろ問題となっている。

他にも、Anti-Mac interface の基本原則としてあげられている 5 項目：

- The central role of language (**言語中心**)
- A richer internal representation of objects (**内部表現の活用**)
- A more expressive interface (**表現的なインターフェイス**)
- Expert users (**熟練ユーザー**)
- Shared control (**共有コントロール**)

のうち、2 番目の「より豊かなオブジェクトの内部表現」は、今日のデータベースや (セマンティック・) ウェブにおける XML によるメタデータやタグの記述であるし、ディスプレイのカラー化、大画面化や高解像度化は、インターフェイスの表現を向上させた。今日では、こうした Anti-Mac interface のパラダイムやガイドラインは、伝統的な Mac interface の中に多数取り入れられている

といって良い[*2]。

6 Post-WIMP Interface

次世代のインターフェイスの探求は、その後「Post-WIMP User Interfaces [18]」あるいは「Perceputual User Interfaces [19]」などのパラダイムで、より多くの研究者によって考察されるようになった[*3]。

これらの主張や提案の多くは、そのディテールには違いがあるものの、基本的には以下のような内容となっている。

- コンピュータの性能が向上することで、インターフェイスも進化するべきである。インターフェイスは 2D より 3D の方が優れており、大画面高解像度のマルチ・ディスプレイや立体視 (3D) ディスプレイ、複数のポインティングデバイス (マルチタッチ) などが特に有用である。
- 視覚だけでなく、聴覚や触覚など、人間のさまざまな感覚を活用すべきであり、キーボードやマウスだけでなく、音声 (聴覚) や触覚など、さまざまな情報をセンスするインターフェイス・デバイスを活用するべきである。
- インターフェイスをより直接的、直感的に操作できるようにすることで、コンピュータをより人間に近づけるべきであり、そのためには、自然言語やジェスチャー (身振り)、音声認識技術などを利用するべきである。
- コンピュータ・ビジョンや人工知能 (AI) の技術を用いることで自律させ、コンピュータに人間のような知覚を持たせたり、コンピュータを人間のようにコミュニケートさせるべきである。

こうした、いわば「マルチメディア」「マルチモーダル」あるいは「バーチャルリアリティ (VR)」

[*2] Anti-Mac interface のパラダイムを実際に実装していったのが Unix/Linux や Xwindow をベースにしたフリーソフトウェアやウェブであり、それが Macintosh や Windows の文化に流入していった、という指摘もある (`http://www.advogato.org/article/130.html`)。そうした意味では、Gentner らの主張は Anti-Mac というよりもむしろ Post-Mac というべきだろう。

[*3] 90 年代の Apple 社においても、未完に終わった OS の Copland や Gershwin のデザインでは「(Point and Click に代わるものとしての) Speak and Delegate」などの次世代インターフェイスのパラダイムが盛んに議論、検討されていた。

「オーギュメンティドリアリティー」といったキーワードに関連した提案は、既存のコンピュータ科学や工学の分野における技術的パラダイムの転用であり、時代のムードや流行も含めて、当時は一見もっともな主張に聞こえていた。しかしながら現実には、当時盛んに提案された 3D デスクトップ/ブラウザや、VR 技術や AI 技術を用いたインターフェイスは、いずれも実用化はされず、結局はキーワードの博覧会で終了してしまった。

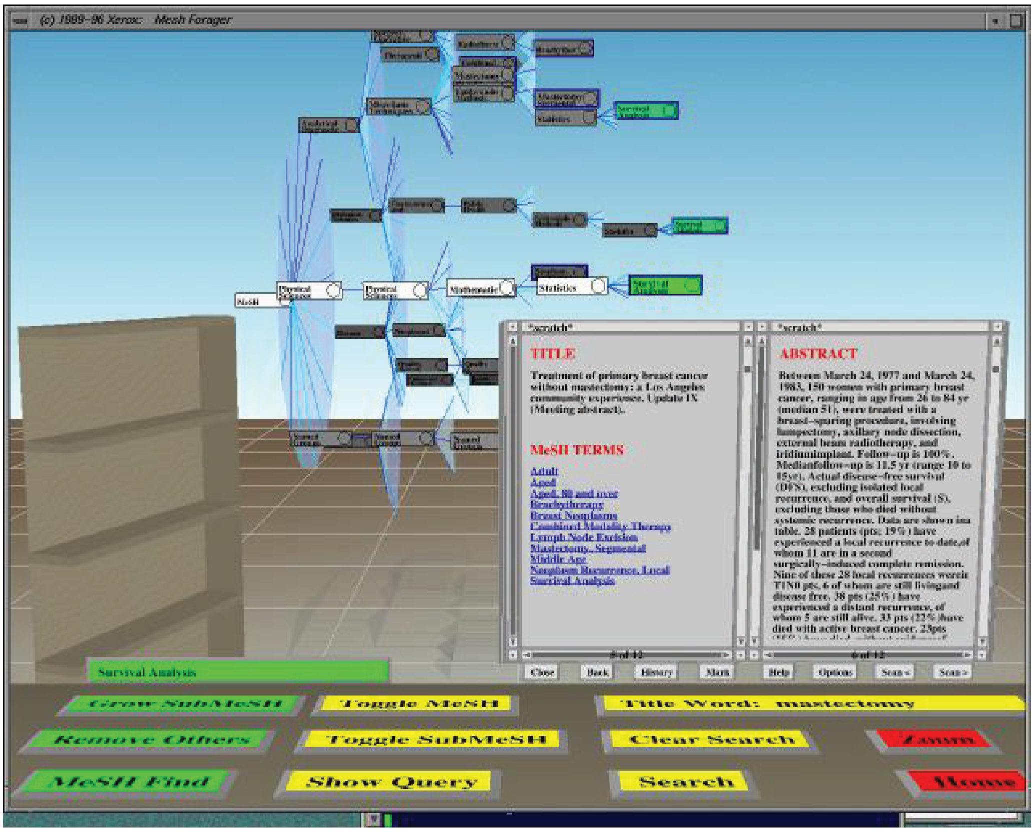

図 3　90 年代に提案された 3D インターフェイスの一例 [20]

ディスプレイやマウスはどんなに大型化、高解像度化しても、所詮は 2D のデバイスであり、それを介して行う仕事も 2D の枠内にとどまらざるを得ない。GUI におけるメタファーとは、表象のメタファーというよりはむしろ構造のメタファーというべきであり、インターフェイスの見栄えだけを 3 次元にしても、本質的に何かが解決されるわけではない。データのブラウジングやマイニングの場合も、コンピュータで取り扱う大量の情報は、3 次元どころかそれより遥かに高い次元を含んでいるので[*4]、それを 3 次元にマッピングしようが、2 次元にマッピングしようが、たかが 1 次元の違いでは、やはり本質的な違いはない。

むしろ人間の認知にとって重要なのは、速度やスケールであるから、そうした観点からは複雑なア

[*4] 例えば n 個のデータのすべての関係を表現するためには、n 次元空間 (n 行 n 列の行列) が必要になる。

ルゴリズムや解析を用いた巨艦大砲主義的でモノシリックな技術よりも、小さな技術をブリコラージュ的に多数組みあわせていくのが効果的だ。インターフェイスにおいては、複雑なことをゆっくりやるよりは、単純なことを高速に行う方がはるかに有効である。

聴覚や触覚、ジェスチャーを用いたデバイスも、肝心なのはその精度である。人間がさまざまな感覚を複合的に組みあわせたり、意識的・無意識的なフィードバックによるコーディネーションから生まれる総合的な作業の緻密さに比較して、データグローブや音声認識などの精度は、そのまま用いたのでは今なお貧弱である。指先が行う精密な動きや、声のニュアンスがもつさまざまな意味を活用し、ユーザーにフラストレーションを抱かせないためにも、より高度なデバイスの組み合わせや活用のしかたを検討する必要がある。少なくとも現時点においては、高精度な解像度を持つマウスやタブレット、熟練によって高速タイピングが可能になるキーボードの方が、日常のプロフェッショナルな仕事を行うためのインターフェイスとして、遥かに優れているといわざるを得ない。

7　PDA や携帯電話のインターフェイス

結局、Post-WIMP インターフェイスが最初に実用化されたのは、高速な処理能力や広大なメモリーをもったパーソナル・コンピュータではなく、ハードウェア的にはむしろ初代 Macintosh と比較できる程度の性能しか持たない小型の PDA であった。今日の PDA のルーツになっているのは、Macintosh と同じ Apple 社によって、1993 年から 98 年にかけて販売された、世界初の PDA である「Newton」であった。Newton は価格が高価であったことや、サイズが大きすぎたこと、処理速度が遅かったことなどから、決して商業的に成功したとはいえないが、Post-WIMP インターフェイスとしての、ペン入力やジェスチャー技術を最初に実用化した功績は非常に大きい。

Newton の手書き認識は、単に手書きのアルファベット文字を認識するだけでなく、ポインティングによる文章の入力位置の自由な指定や、文字を囲むことで選択したり、引っ掛くことで消したりという、校正記号に近いシンボルを使った編集操作によるジェスチャー機能を実現していた。さらにユーザーが書いた文字を整形してくれるだけでなく学習し、さらにはその学習データを用いて、ユーザーが次に何を書こうとしているかを推測してくれた。これらは Anti-Mac インターフェイスにおける「Delegation」「System Handles Details」「Model User Actions」の具体例でもある。

Newton が開拓した PDA という、ポータブルなコンピュータの可能性は、その後ジェフ・ホーキンスによって開発され、1996 年から販売されるようになった「Palm」によって、広く社会の中で認知、使用されるようになった。Palm のデザイン方法論は、前述の Post-WIMP 的な高度化、複雑化とは逆に、PDA という限られたリソースを効率良く活用するために、機能や外見をミニマルにすることで、使いやすい実践的なインターフェイスを構築しようとすることであった。Newton では、自由にどこでも通常のアルファベットを書くことができた手書き認識を、Palm(の Graffiti)

ではプログラムが認識しやすい独特の記法と入力場所の限定によって、その速度や認識率を向上させたのが良い例である。

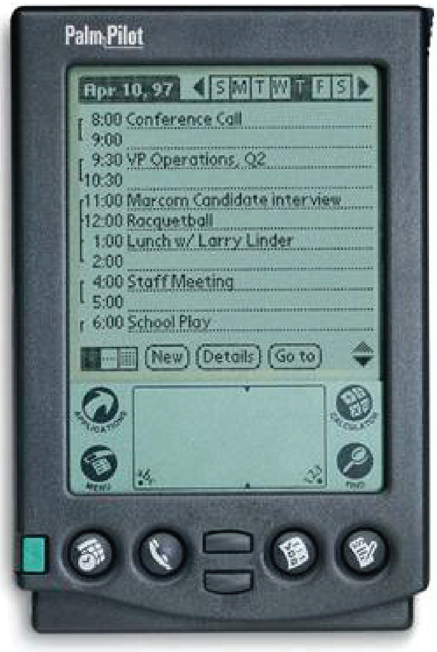

図 4　1996 年に発売された初代 Palm

こうした方向性は、さまざまな技術によってインターフェイスを人間に近づけていく Post-WIMP 的な方向性ではなく、むしろインターフェイスをシンプルかつミニマルにすることで、人間のが本来有している認知の並列性や高速性、巧緻な操作を可能にする熟練性を引き出していこうとするものだ。それは、GUI におけるキーボード・ショートカットの活用や、テンキーと親指 1 本だけで漢字かな (絵文字) 混じり文を高速に入力できる程に成熟した、今日の携帯電話のインターフェイスの有り様に通じる考え方であるといえる。

Newton や Palm にも表 3 のようなユーザー・インターフェイス・ガイドラインが設けられていた [6] [15]。Newton のデザイン原則は、「Modeless」が「Stability」に置きかわっている以外は、Macintosh のガイドラインとほぼ同様の内容である。ここで Stability というのは、PDA の入出力デバイスの小ささから生じる複雑さを緩和させるための、安定した参照点の必要を述べたものである。インターフェイス要素をなるべく少なく共通化したり、Macintosh のディスプレイに固定されたメニューバーや、今日の iPhone におけるホームボタンのように、どういう状態にあっても常に変化しなかったり一定の状態に復帰できるインターフェイスがその一例である。

一方で、Palm のガイドラインは、それまでのものとは、ちょっと違った構成のものになっている。表 3 であげたものは、インターフェイスというよりも、サイズやバッテリーなど PDA の環境をリ

表 3　PDA のデザイン原則

Newton	Palm OS
Metaphors	Pocket Size
Direct Manipulation	Fast and Simple
Feedback	Low Cost, Long Battery Life, and High Value
See and Point	Seamless Connection with Desktops
Consistency	
User Control	
Forgiveness	
Stability	
Aesthetic Integrity	

ストアップしたもので、それぞれの項目の下にさらに以下のような、具体的小項目があげられている。

- Pocket Size
  - Applications Must Limit Data Entry (**データ入力の制約**)
  - Menus Are Hidden (**メニューバーを隠す**)
  - No Button Toolbars (**ツールバーボタンを用いない**)
  - Less Is More (**より少きはより多き**)

- Fast and Simple
  - Perceived Speed Is Important (**知覚の速度が重要である**)
  - Minimize Required Steps (**必要な手順を最小化する**)
  - Minimize Taps (**タップは最小化する**)
  - Be Consistent (**一貫的であれ**)
  - Optimize Frequent Tasks (**よく行う作業を最適化する**)

- Low Cost, Long Battery Life, and High Value
  - Keep User Costs Down (**使用者のコストを少くする**)

- Seamless Connection with Desktops

Palm のガイドラインの重要な点は、メニューバーやツールバーなどのインターフェイス要素や、タップなどのデバイスに具体的に言及することで「PDA の GUI のデザインは PC のデザインとどこが違うのか」ということを明確にしたことである。加えて、このガイドラインは人間とコンピュータの情報のやりとりを (増やすのではなく) なるべく少なくすることや、インタラクションの量よりも速度の方が重要であることを明言した。前述のように、インターフェイスにとって「速度」は非常に重要である。同じインターフェイスであったとしても、その応答時間や表示速度が異なれば、ユーザーの使用感は大きく異なる。Apple の Safari や Google の Chrome といった昨今のウェブ・ブラウザは「高速」「シンプル」をモットーとし[*5]、そこから生れる新たなユーザー体験を広くアピールしている。

Apple 社のデザイン原則や、それに対する Anti-Mac インターフェイス原則のいずれにも、Aesthetics や Graphics に対する項目があるにもかかわらず、「Speed(速度)」のような時間に関する項目が明示されていないのは、今考えれば不思議なことだといえる[*6]。

8 今日の GUI の特徴

こうした GUI の歴史を踏まえて、もう一度今日のパーソナル・コンピュータや PDA/携帯電話のインターフェイスを検討してみよう。 今日の GUI の特徴といっても多岐に渡るが、ユーザーの視点からみた特徴のひとつとして、WIMP インターフェイスの象徴的存在であった「(オーバーラッピング・) ウィンドウ」を用いなくなってきたことがあげられる。具体的には、今日のウェブ・ブラウザ、メイル・アプリケーション、音楽ソフトウェアなどにおいて、複数の重なり合うウィンドウではなく、単一のウィンドウの中を (サイドバーなどにより) いくつかの区画 (ペイン) に区切って用いる傾向が多い。その背景にあるのは、ディスプレイの大画面化や高解像度化により、以前より大量の情報が一度に表示できるようになったことだが、なにより情報が部分的に隠蔽されるウィンドウの重なりは、それだけでインターフェイスを複雑にする。ウィンドウの起源は、ディスプレイの狭小さによる一種の妥協の産物であり、今日のディスプレイはそうした妥協を行わずに情報を表示できるようになりつつある[*7]。Post-WIMP 論者が主張したように、GUI は 3D 化しなかった。その第一の理由は、ウィンドウを含む GUI を 3D 化しても、このオーバーラッピング化は解消されるわけでなく、むしろ遠近法の本来の特徴として、3D 化によってオーバーラッピングされ

[*5] 例えば Google Chrome のインターフェイスにはフレームもツールバーもない。

[*6] とはいえ、腕時計アイコンやプログレスバーを表示して、ユーザーの心理的な待ち時間を軽減することなど、デザイン原則の「Feedback and Dialog 」の項には、時間の要素が暗に含まれているといえる。なお、今日最も普及しているコンピュータの GUI である Windows を開発した Microsoft 社は、自社でハードウェアを開発しないこともあり、当初から常にソフトウェアの速度を重視していた。それに対して Apple 社は速度の問題は (自分たちが作る) 将来のハードウェアで解決されると考え、常にやや過負荷気味の UI をデザインしてきた。

[*7] 今日の GUI でほぼ標準装備となっている、複数のスクリーンを瞬時に移動できる仮想デスクトップも、同様の機能を持つ。

る領域が増大してしまうことである。

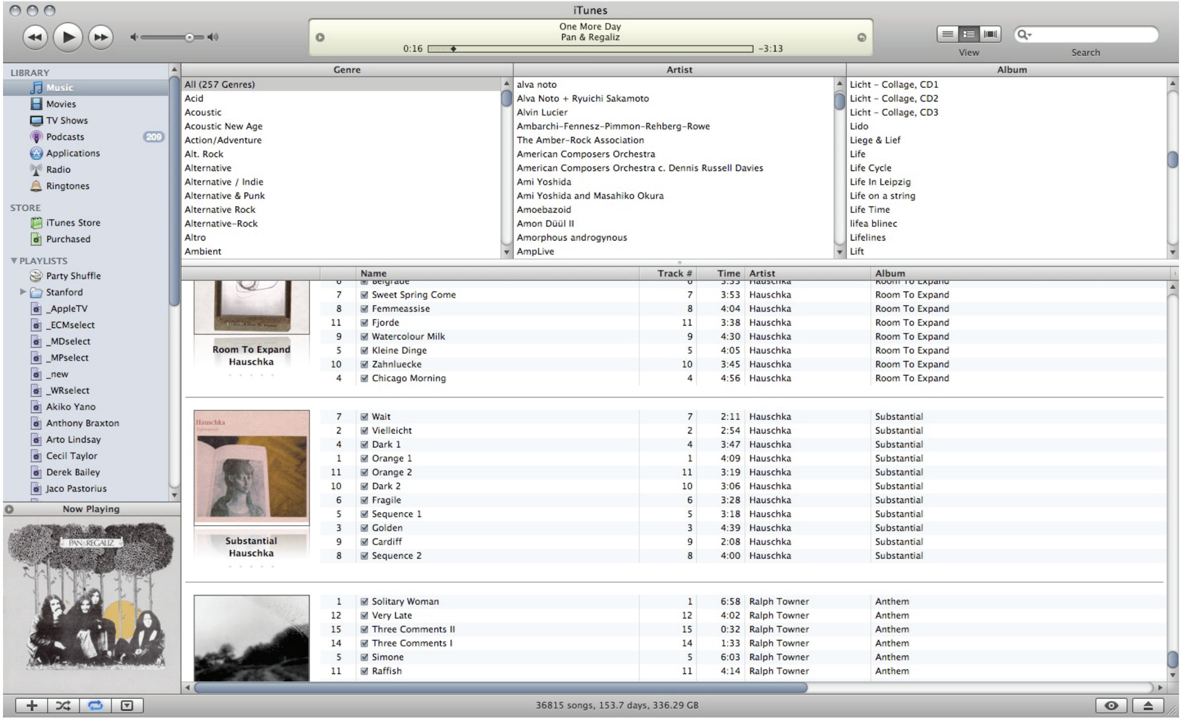

図 5　複数のペインによって構成される iTunes のインターフェイス

ウィンドウレスの GUI のきっかけとなったのは、パーソナル・コンピュータではなく、PDA や携帯電話、あるいは携帯音楽プレイヤーのインターフェイスであった。これらの携帯機器のインターフェイスは、前述の Palm のデザイン原則にあるように、人間と機械の情報のやりとりを、できるかぎり少なくすることが求められている。その結果、こうした携帯機器のインターフェイスにおいては、ダイアログ等の表示を除いて、オーバーラッピング・ウィンドウは一切用いられず、メニューバーやツールバーによらない少数のボタンと、仮想デスクトップ同様のフルスクリーン画面の切り替えを中心にしたインターフェイスが採用されてきた。これらは、携帯機器のみならず、DVD やブルーレイディスクなどの再生機器や TV のセットトップボックスのインターフェイスにおいても同様であり、今日のパーソナル・コンピュータの GUI はこうした機器のインターフェイスからの影響を大きく受けているように見える。

インターフェイスにおけるインタラクションの最小化は、インターネットの普及以降、コンピュータを介して扱う必要がある情報が飛躍的に増大したこととも深く関係している。ウェブだけでもすでに 100 億ページを越えるといわれている膨大なインターネット空間とそこでやりとりされる情報を WYSIWYG ならぬ WYSIATI(What You See Is All There Is、あなたが見ているものがそこにあるもののすべてである)[17] で表示したり、直接操作 (Direct Manipulation) することは不可能であり、少数の単語で膨大なウェブ空間にアクセスする検索エンジンのように、「委譲

(Delegation)」や「System Handles Details」といった間接操作技術がインターフェイスの中心にくるようになった。こうした間接操作技術にとって必要なことは、人間の入力の手順や量の最小化であり、それはいいかえればインタラクションの量よりも速度が重要ということである。インターフェイスを人間に近づけるというのは、インターフェイスを人間のように振る舞わせることではなく、インタラクションをシンプルにすることで、むしろ人間の知覚や思考の速度に対応できるような速度 (と速度から得られる量) を獲得することである。

9 GUI の将来

GUI における Window の役割が終焉を迎えた今日、それに代ってフルスクリーンベースのインターフェイスで重要になってくるのは何だろうか。そのためにまず必要なのは、インターフェイスにおける空間と時間の操作法の再考である。オーバーラッピングするウィンドウというのは、部分的な重ね合わせによる空間の操作方法であり、今日のデバイスにおいては、空間の効率化というそのメリットよりも、重ね合わせによる煩雑化というデメリットの方が大きくなってきた。

今日の GUI において、ウィンドウに代わるものとしてクローズアップされているのが、ズームイン/アウトという空間操作法である。Google Map や Google Earth のようなジオグラフィカルなデータを扱うアプリケーションや、 Adobe Photoshop や Illustrator のようなグラフィック・アプリケーションでは、スムーズなスクロールと同時に、情報の表示領域の拡大縮小を行うズームイン/アウトが必要不可欠である。2008 年 6 月にリリースされたウェブブラウザの Firefox 3 ではテキストの拡大/縮小だけでなくページ全体のズームが可能になり、iPhone のインターフェイスにおいても、フルスクリーン表示と、2 本指で行うズームイン/アウトが有効に使用されている*8。ウィンドウというのは、いわば視点を固定したまま、操作対象を並行移動することであるが、それに対してズームイン/アウト + スクロールというのは操作対象を固定したまま前後左右に視点を自由に移動することに相当する。今日のグラフィック処理の高速化が可能にしたスムーズ (高速) なズームイン/アウトは、対象を 2 次元のまま固定することで、大量の情報に対して Stable かつ Flexible なアクセスを可能にし、単なるアプリケーションの 1 機能としてだけでなく、今後の GUI の基本機能のひとつとして、これまで以上に活用されることが予想される。

同様に GUI における時間の操作法として、アニメーションの活用がより重要になっていくだろう。Apple 社の最新の MacOS 10.5 においては、iTunes アプリケーションにおけるアニメーションによるブラウジング手法であった Cover Flow が基本ファイル管理/シェル・プログラムである Finder のプレビュー/検索インターフェイスに取り入れられた。同様に、同社の写真管理ソフトウェアである iPhoto においては、アルバムに含まれる写真のブラウジングに、マウスオーバーに

*8 OS X のユニバーサル・アクセスのズーム機能をプレゼンで活用する人も多い。

図 6　自在なズームやスクロールが可能な Google Earth のインターフェイス

よる高速なアニメーションが取り入れられるようになった。限られた空間の中で、一瞬の映像でも知覚できる人間の高度な認知能力を活用したアニメーションによるインターフェイスは、ズームイン/アウト同様に、今日の高速なグラフィック処理技術をフルに生かしたものであり、そこから生まれる新たなインタラクションの探求は、今後の GUI にとってより大きな役割を果すことになるだろう。

以上の項目を、 将来の GUI の特徴としてまとめると、以下のようになる。

- シンプルな**フルスクリーン**表示とその切り替えによる 2D 空間の活用
- コンピュータに処理を委譲する**間接操作**によるインタラクションの最小化
- 人間の知覚能力をフルに活用する**速度**に対するクリティカルな要求
- **ズーム・イン/アウト**による視点と解像度の柔軟な操作
- 時間を操作する**アニメーション**による認知の効率化

インターネットという大規模なデータベースへのアクセスや、CPU や GPU の高速化によるハードウェアの進化の活用により、 Macintosh の登場以来 20 年以上の歴史を持つ GUI は今日、ポス

図 7　アニメーションを活用した Cover Flow のインターフェイス

ト・ウィンドウとも呼ぶべき次のステージに変革しつつある。そこから生まれるものは、かつて「Post-WIMP User Interfaces」や「Perceptual User Interface」として多くの人に想像されたものとは、かなり異なる姿になっていくだろう。今日のインターフェイス・デザインにおいて必要なのは、こうした歴史を参照しながら GUI を再発明することであり、ウィンドウの消滅はその第一歩というべき象徴的事項であるといえる。

謝辞

本研究は、2007 年度多摩美術大学共同研究費「インターフェイスの歴史とインタラクションの未来」の支援を受けて行われた。関係各位に厚く御礼を申し上げる。

参考文献

[1] 大谷和利「GUI の起源と進化の概略」情報の科学と技術，Vol.49，No.12，情報科学技術協会，`http://ci.nii.ac.jp/naid/110002829603/`，1999.

[2] Daniel H. H. Ingalls「The Smalltalk-76 Programming System Design and Implemen-

tation」Fifth Annual ACM Symposium on Principles of Programming Languages, http://users.ipa.net/~dwighth/smalltalk/St76
/Smalltalk76ProgrammingSystem.html, 1978.

[3] Graphical User Interface Gallery, http://toastytech.com/guis/index.html

[4] GUIdebook: Graphical User Interface gallery,
http://www.guidebookgallery.org/

[5] Apple Computer Inc.「Apple Human Interface Guidelines: The Apple Desktop Interface」Addison-Wesley, 1987.

[6] Apple Computer Inc.「Macintosh Human Interface Guidelines」Addison-Wesley, 1992.

[7] Apple Computer Inc.「Apple Human Interface Guidelines」
http://developer.apple.com/documentation/UserExperience/Conceptual
/AppleHIGuidelines/, 2008.

[8] Next Computer Inc.「Nextstep User Interface Guidelines: Release 3」Addison-Wesley, 1992.

[9] Microsoft Corporation「The Windows Interface Guidelines for Software Design: An Application Design Guide」Microsoft Press, 1995.

[10] Microsoft Corporation「Windows Vista User Experience Guidelines」
http://msdn.microsoft.com/en-us/library/aa511258.aspx, 2007.

[11] Open Software Foundation, Inc.「OSF/Motif Style Guide」
http://wwweic.eri.u-tokyo.ac.jp/computer/manual/lx/SGI_Developer/books
/Motif_SG/sgi_html/index.html, 1992.

[12] The GNOME Usability Project「GNOME Human Interface Guidelines 2.0」
http://library.gnome.org/devel/hig-book/stable/, 2004.

[13] Apple Computer Inc.「Newton 2.0 User Interface Guidelines」Addison-Wesley, 1996.

[14] Apple Computer Inc.「iPhone Human Interface Guidelines」
http://developer.apple.com/iphone/library/documentation
/UserExperience/Conceptual/MobileHIG/, 2008.

[15] Palmsource Inc.「Palm OS User Interface Guidelines」Iuniverse Inc,
http://dogbert.mse.cs.cmu.edu/charlatans/References/Tech_Doc
/Palm%20OS%205.0%20Docs/Palm%20OS%20UI%20Guidelines.pdf, 2002.

[16] Microsoft Corporation「Pocket PC User Interface Guidelines」
http://msdn.microsoft.com/en-us/library/ms854763.aspx, 2005.

[17] D. Gentner & J. Nielsen「The Anti-Mac interface」Communications of the ACM, Vol.39, No.8
http://www.useit.com/papers/anti-mac.html, 1996.

[18] Andries van Dam「Post-WIMP User Interfaces」Communications of the ACM, Vol.40, No.2, 1997.

[19] Matthew Turk & George Robertson「Perceptual User Interfaces」Communications of the ACM, Vol.43, No.3, 2000.

[20] Marti A. Hearst & Chandu Karadi「Cat-a-Cone: An Interactive Interface for Specifying Searches and Viewing Retrieval Results using a Large Category Hierarchy」Proceedings of 20th Annual International ACM/SIGIR Conference, http://people.ischool.berkeley.edu/~hearst/papers/cac-sigir97 /sigir97.html, 1997.