〔研究ノート〕

大学における情報教育への提言
－たかがパソコン、されどコンピュータ－

伊藤 公洋*

はじめに

世界初のデジタル・コンピュータであるENIACが発明されてから60年以上が、また本格的なGUIを実現し一般に販売されたパーソナル・コンピュータ（PC）であるMacintoshがエンドユーザの手に渡るようになってからも25年以上が経過した。Windows95に始まるインターネット・ブームも既に10年以上が経過している。

現在のアーキテクチャーで設計されているコンピュータ（PCを含む）は有名なムーア博士の「ムーアの法則」により、「18か月毎に性能は２倍に、コストは半分に」を40年以上にわたり実現しているのである。

当初、軍事利用目的のために開発されたコンピュータも今では他の家電製品と同じように身近な存在となっていることは誰もが認めるところであろう。開発された当時、「21世紀までには全世界で５台は必要になる」と予測されたコンピュータも今では国内だけでも年間1千万台以上、全世界では２億台近くが出荷されているが、そのほとんどは企業や組織、そして個人が利用するために購入している、いわゆるパーソナル・コンピュータである。すなわち、現在ではPCが個人の「知的生産作業の道具」として利用されている。

この「個人のための知的生産作業の道具」という考えはヴァネヴァー・ブッシュの「memex」まで遡ることができる。彼のアイデアは素朴なものではあったが、

＊ Kimihiro ITO　四国学院大学教授、総合教育研究センター所属

その後のハイパーテキストのように関連する項目をリンクするなど、現在のインターネットやWorldWideWebの基礎となったことは否めない。彼の功績は大きいのである。しかるに、彼の当時とは異なり手軽にPCを購入できるようになった現在、我々はこの「知的生産作業の道具」を有効に活用しているのであろうか。また、この本来は便利な道具を学習者に適切に教授できているのであろうか。

この試論では上記2点を、特に後者を考察して今後の情報教育の在り方を提言してみたい。

1.「情報教育」における文部科学省の方針

2008年10月現在、文部科学省は以下のように情報教育を考えている（http://www.mext.go.jp/b menu/houdou/18/08/06082512/001.pdf）。

(a) 情報活用の実践力
●課題や目的に応じた情報手段の適切な活用
●必要な情報の主体的な収集・判断・表現・処理・創造
●受け手の状況などを踏まえた発信・伝達能力
(b) 情報の科学的な理解
●情報活用の基礎となる情報手段の特性の理解
●情報を適切に扱ったり、自らの情報活用を評価・改善するための基礎的な理論や方法の理解
(c) 情報社会に参画する態度
●社会生活の中で情報や情報技術が果たしている役割や及ぼしている影響の理解
●情報モラルの必要性や情報に対する責任
●望ましい情報社会の創造に参画しようとする態度

もちろん、これらの項目はコンピュータに限ったことではなく情報通信機器全般に対する社会生活を行う成員としてのリテラシーを概括しているものではある。しかし現在ではほとんどすべての情報通信技術がデジタル化されており、この傾向は今後益々拡大されていくことは否定できないであろう。つまり、ユビキタス

社会に代表されるように好むと好まざるとにかかわらず、情報に接する際にはその背後にコンピュータの存在があることになるわけである。

また、これらの方針は学校教育すべてにおけるものであり大学教育のみのものではない。e-learningをはじめ、ほとんどすべての教科にコンピュータなどの情報通信機器を利用することが可能であるし、活用することが望ましいことは論を待たないが、ここでは大学教育におけるコンピュータ利用および情報教育のみを取り上げることとする。

2．パソコンの三種の神器？

本学のように情報科学・工学の専門的な分野の学科がない大学では主にMicro SoftのWordやExcelに代表されるワープロと表計算の使い方を教えることで情報教育としている場合が多い。これは卒業後に就職先で求められるスキルである、という暗黙の了解の元、行われているのであろうが、実際にはその業界や企業においては独自のアプリケーションを操作することの方が多く、せいぜい報告書をワープロで作成する程度である（またそれら各書類もフォームが予め作成されており、文字入力をするだけの場合が多い）。さらに最近ではインターネットを通じてブラウザ上から操作をする、Webアプリケーションと呼ばれるシステムが普及しており、アプリケーションのヴァージョン・アップの際に利用者すべてのPCにインストールする手間が省ける、などの理由で今後も増えていくことが確実視されている。すなわち、PCの機種やオペレーティング・システム（OS）（これらをまとめてプラットフォームともいう）を問わず、インターネットに接続できる環境とブラウザがあれば通常の業務がこなせるという時代になっているわけである。

未来を予測することは困難であるが、近い将来のコンピューティング環境へのキーワードであれば、以下のように列記することができる。

●リッチ・クライアントの進歩

現在普及しているブラウザは、HTMLというマークアップ言語に基礎をおいており、一般的なデスクトップなどGUIの快適な操作感とは程遠いものであ

るが、リッチ・クライアントとは、これを改善する技術のことである。既に数種類の技術が発表されており、どれが業界標準となるかはわからないが、リッチ・クライアント技術の普及により、快適な操作で作業できる環境が構築されるであろう。

●LAN（構内ネットワーク）、インターネット接続の速度の向上

LANにおいては、有線、無線を問わずGigaBitEthernetなど格段に通信速度が向上した技術が普及しており、インターネットへの接続もより早くなってきている。また、公衆無線LANなどのインフラも整備されてきており、より手軽にネット接続ができる環境が着々と構築されている。

●Webサービスの普及

インターネットを利用して複数のWebサイト間で情報を交換し、連携して処理を実施するための技術で、特定の機能を各自で独自開発せずに相互で利用できるようにするものである。この技術が普及すれば、個人、企業を問わず必要な時に必要な機能を検索するだけで利用可能となり劇的に作業効率が向上する可能性がある。

●データウェアハウスの普及

従来は各組織で独自に保管・保存していたデータであるが、データウェアハウスとは、インターネット接続速度の向上により、より堅牢な環境にある場所にそれらを保存するサーバ群のことを指すが、バックアップなども行われ天変地異の際にも最小限の時間とコストで通常の業務に戻ることが可能である。

●クラウド・コンピューティングの台頭

クラウドとは、インターネットのことを指し、クラウド・コンピューティングとは、これまでと異なるのはアプリケーションやデータなどすべての資源をインターネット上で管理、保存するというより積極的にインターネットを利用する方法論のことである。

つまりクラウドは、上記４つの技術の集大成を強調するために利用され始めているキーワードと言えよう。

このようにこれからのコンピューティング環境はいわゆるスタンド・アロンではなく、ネットに接続されてブラウザを通して様々な処理を行うのが一般的とな

る。つまり、パソコンの三種の神器といわれているワープロ、表計算、データベースはその座を降りようとしているのが現実なのであろうか。いや、実はそうではない。事実は、これらのスタンド・アロン環境で利用されていたアプリケーションがネット接続のもと、ブラウザ越しに利用できる環境が構築されてきている、ということなのである（オフライン＝ネット非接続時にも操作を可能とする技術も開発されてきている）。

このような時代に特定の企業が製造した特定のアプリケーションの操作方法を教えることが情報教育となるのだろうか。次はそれを考えていきたい。

3．パターンと型（スタイル）

オアシス、一太郎、松、WordStar、WordPerfect、AmiPro、Macワード。これらの商品名を聞いたことがある、使ったことがある、という方も多いであろう。すべてこれらは「ワープロ」というアプリケーションのジャンルに属する製品名である。同様にVisicalc、supercalc、multiplan、Lotus 1-2-3、Winzなどは表計算に属する製品名であり、これらのうち現在でも残っているのはLotus 1-2-3くらいである（開発は中止されているが）。

パソコン用にワープロ、表計算、データベースが開発・販売されてからすでに30年が経過するが、その間多くの企業から様々な製品が出荷され、ある時期は業界標準とも言える地位を築いていたものである。それがWindows95の発売以後、WordやExcelの独壇場となっていき、Apple社のMacintosh用のiWorksを除けば、日本語ワープロとしては、Wordや一太郎、表計算としては、Excelや三四郎、データベース（小規模向け）としては、AccessやFileMakerProくらいしか発売されていない（もちろん、OpenOfficeなどはあるが、これはオープンソースである）。その結果、情報教育の現場でも圧倒的なシェアを持つWordとExcelの操作方法を教えることが当たり前のように行われてきた。

その教授方法とは通常、以下のようなものである。

(1) アプリケーションの各部の名称を説明する。
(2) 基本的な機能の説明をする。
(3) この段階で簡単な文書（表）を作成して学習させる。

(4) より高度な機能の説明をする。
(5) これまでの機能をほぼすべて網羅した文書（表）を作成させる。

こうしたの方法自体は間違ったものとは思っていないが、数年間大学で教えてきた経験上疑問視せざるを得ないことができてきた。それは学生のレポートである。

学年を問わず、学生が作成したレポートを読むと、その書式はばらばらで、かつ非常に基本的な機能だけを用いているのである。基礎的な情報教育を受けているにもかかわらず、である。私なりに考えた結果、以下のような結論に達した。「機能の集合は全体像を表すものではない」。すなわち、各機能でどのようなことができるのか、を理解できても最終的にどのような文書（表）を作成できるか、するかということをイメージできないために基本的な機能以外のものを必要とする動機づけがないわけである。

実は前述のワープロ、表計算の各種製品は私が以前に使用、あるいは愛用していたアプリケーションの名称である（オアシスは違うが）。幸いにしてこれらのアプリケーションの機能の向上や変遷を経験しているため、自分のイメージする文書（表）などを作成する際には、初めて利用するものでもメニューなどから探し出すことが可能であった。つまり、ワープロならワープロ、表計算なら表計算、というもののイメージができあがっているのである。それならば、このような経験がない初心者にはどのように教えればいいのであろうか。そこにパターンやスタイルというものが登場すべき余地がある。

プログラミングの現場ではここ数年、「デザイン・パターン」という考え方が導入され実績をあげている（1995年に発表された「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」から提唱された概念である）。この考え方はこれまでに蓄積されたいわゆる「成功したプログラム設計法」をカタログ化して共通のパターンを積極的に利用することにより開発者間のコミュニケーションを円滑にし、設計を洗練させるための技術であると言える。当初は23個あったデザイン・パターンであるが、今でも様々なパターンが発見され、カタログに追加されている。デザイン・パターンは決して科学的・論理的に説明されるものではなく、いわば経験知のようなものであるが、その有効性を疑うシステム開発者はいないであろう。ちなみに失敗した開発事例を研究してカタログ化した「アンチ・

パターン」というものもある。

同様に複数の開発者でシステムを作成する場合にはスタイルが問題となる場合がある。コーディング（プログラム＝ソースコードをエディタなどで作成すること）の際に各自が独自に変数などの命名規約を使用していると、必ず行われるデバッグ作業（不具合を修正する作業）において明らかに見通しが悪くなる。これはコメント（プログラムの動作には直接関係ないが、その内容をメモしておくためのもの）の書き方やインデントのサイズなどにも当てはまる。スタイルの統一、というのは効率や利便性を向上させるポイントとなっているのである。

これら情報システム製作の現場で用いられている方法を情報教育に取り入れられないのだろうか。私は取り入れられると思う。そのための方法として、

(1) 大学において独自の文書フォーマットの作成方法を教える。これは統一的なスタイルを教えることを意味する。つまり、(ⅰ)レポート、(ⅱ)小論文、(ⅲ)卒論などの論文に応じて、どのようにすれば簡単に「体裁」を作成できるか、を教えるわけである。

(2) 表計算の場合には、(ⅰ)関数の種類（どのようなものがあるか)、(ⅱ)すでに作成されているシートを整形、(ⅲ)どの種類のグラフがそのシート（データ）に適しているかを中心にパターンを教える。

(3) OLEによる複数のアプリケーションの連携

道具は選んで利用することを教えるためであると考えている。

要するに、「スタイルとは仕事をてっとり早くこなすために必要」、「パターンとは利用するアプリケーションの概念を習得するため」、ということである。

このような方法で初心者に学習させることにより、場数を踏めばWord以外のワープロやExcel以外の表計算も比較的短時間に利用可能となり、そのスキルは一生使えるものになると思われる。

重要なのは作成したいドキュメントに必要なアプリケーションは何であるかを判断でき、どうすればストレスを感じずにそれを利用できるか、なのであり特定のアプリケーションの操作方法ではない。そのためにもパターンとスタイルが必要なのである。

4. The dark side of Black Box

システム開発の現場では、昔から大規模で複雑なシステムを構築する際に採用してきた手法がある。“Divide and Conquer”＝「分割して統治せよ」である。規模が大きく複雑に見える問題でも、小さな部分に分解してみると手をつけやすくなる、というごく当たり前のことであるが、意外と見落とされる戦略である。またこれはシステム開発に限ったことではないが、「複雑さ」に対処するための方策として「抽象度をあげる」というものがある。特にオブジェクト指向では、継承、カプセル化などの概念を用いて抽象度をあげることにより堅牢で変更に耐えうるシステムの構築を目指している。そして、現在日常的に利用されている家電製品や自動車から情報機器まではいわゆる部品を調達し、それを組み合わせる（アセンブル）することにより製造されるようになった。それ自体はより素早く製品を製造するための手段なのであるが、結果的にわれわれの周りにある製品はブラックボックス化してきている。

ブラックボックスはその内部の詳細に立ち入らなくていい、という点では優れた長所を有しているが一旦、その内部の詳細を知らなければならなくなると悲惨な状況となる。今まで気にもしていなかった事項を理解しなければならなくなるからである。それでもブラックボックス化することの優位性は変わらず、そのブラックボックスに不具合が見つかれば多くの場合は交換するだけで終わってしまう。そのため、これからもますますブラックボックス化した製品が登場してくることはまず間違いがない。

もちろん、パソコンもかなりブラックボックス化された製品である。触れる部分はキーボードやマウスなどの入力装置、変化がわかるのはディスプレイやプリンターなどの出力装置だけ、というのが普通であって、パソコン自体に不具合が生じても初心者はどこがその原因なのかを調べることもできない。結局、専門家かメーカーのサポートセンターへの問い合わせということになるのであるが、これがまた問題の始まりとなる。あまりに高度にブラックボックス化されているため、その問題点（症状）を説明することができない場合が多いのである。友人の小児科医から聞いた話であるが、子供はボキャブラリーが少ないため、調子が悪くなると「ポンポが痛い」というそうである。頭が痛いにもかかわらず、にであ

る。それは保護者が自分の子供に不調を感じると「ポンポが痛いの？ポンポなの？」と尋ねるからであろう。

コンピュータをハードウェア的に見た場合、機能を分割すると、(ⅰ)入力装置、(ⅱ)記憶装置、(ⅲ)演算装置、(ⅳ)出力装置、(ⅴ)制御装置の五大装置に大別できる。これらもまた、ブラックボックス化されたものであるが、より単機能なものであるため理解はしやすいといえよう。

同様にソフトウェアもブラックボックスである。特にOS（オペレーティング・システム＝基本ソフト）はその最たるものであろう。OSは単独では何も行っていないように見えるが、実際にはハードウェアとそのOS上で稼働するアプリケーションなどとの間を取り持っており、携帯電話などにも利用されている技術である。もちろん、アプリケーションの各機能を実現している処理内容もブラックボックス化されているのではあるが、その複雑さはOSの比ではない。

ではここで問題を一つ。

問「USBメモリを利用するには単純にUSBポートに装置を挿入するだけであるが、取り外すには『装置の安全な取り外し』という手段を経なければならない。しかるに、CD－ROMやDVD－Rなどやメモリカードスロットに挿入されるSDカードやメモリ・スティック（おもにデジタル・カメラやデジタル・ビデオのデータを保存するために用いられる記憶媒体の一種で、商品名である）はそのような手段を経ずに取り外すことができるがどうしてだろうか。」

答「CDやDVDドライブ、メモリカードスロットはあくまでも記憶媒体を装着するための装置であるから、装置そのものはパソコン本体から切り離さることはない。それに対してUSBポートはプリンターを始め、様々な装置が装着される可能性があり、その一つがUSBメモリという外部記憶装置（記憶媒体ではない！）からである。」

つまりパソコン本体に当初から接続されている装置が記憶媒体を利用するものであれば、それらはファイル・システムにマウントされており、CD－ROMやSDカードが挿入されていない場合にはその容量が「ゼロ」なだけなのであるのに対して、USBメモリは装置そのものをマウントさせるため、アンマウントというファイル・システムから切り離す作業が必要となるためである。

ここにブラックボックス化の弊害が潜んでいる。CDやDVDより小さなUSBメ

モリが「外部記憶装置」だと把握できるであろうか。また、さらに小さなマイクロSDカードがCDなどと同じ記憶媒体であると理解できるのであろうか。

重要なのは、ソフトウェア的にはOSに「ファイル・システム」が、ハードウェア的には「バス」いうものがあり、それによってデータの読み書きが可能になっている、という点である（USBはUniversal Serial Busの略）。この例を用いたのは、ソフトウェアとハードウェアのブラックボックス化の弊害の端的なものだと思ったからであるが、他にもこのような例は枚挙に暇がない。多分、コンピュータの初心者でかつ特定のアプリケーションしか利用したことがないのであれば、先の問の解答を読んでも全く理解できないであろう。それが一番の弊害なのである。

近年、高等学校の履修漏れでスポットライトを浴びた「情報A（B、C）」という教科であるが、その教科書には「コンピュータはデジタルです。０と１の２進数で動いています」とだけ記述され、なぜ2進数なのか、ソフトウェアはどのようにしてハードウェアを駆動しているのかが説明されていることはまれである。確かにコンピュータは「ものすごい速度で計算を実行するだけの機械」なのであるが、その機械がどうしてワープロや表計算、インターネットに接続したり、印刷ができるのかは、上記のような説明だけでは理解することは不可能なのではないだろうか。

原理を理解できていないからトラブルに遭遇すると、そのトラブル（症状）の内容をも説明することができないのだと思われる。昔から言われるように、問題に対処するにはその問題点を切り分けて特定する必要があるわけである。私見ではあるが、人間には程度の差こそあれ、生まれながら応用力があると思っている。理由は、応用力がなければ成長するまで生存していけないからである。応用力がなく、すべて記憶力だけで生きていけるというのであれば、様々な危険をすべて体験しなければ、危険回避能力など発達するわけがない、というのが理由である。少なくとも大学生まで成長している個人には応用力が備わっていると考える非合理的な根拠はない。

それにもかかわらず、現状の情報教育では応用力を無視して操作方法などの記憶力に依存している傾向がある。そうではなく、コンピュータの基本原理を教えて身につけてもらう方が時間はかかるかもしれないが、根本的な解決策なのであ

る。幸い、コンピュータが発明されて以来60年間、処理速度やサイズは劇的に変化したがその基本原理は変わっていない。

またブラックボックス化の弊害として、「無機質、非人格性」というのも考えられる。つまり、単なる機械としてのコンピュータだと捉え、そこにある人間ドラマを無視することである。MicroSoft社のBill Gates氏やApple社のSteve Jobs氏は有名であるが、コンピュータがここまで発達してきた裏には様々な歴史があるのも事実である。数理論理学を確立したブール代数のブール（プログラム言語の論理型のことをブーリアンと呼ぶ)、コンピュータの基礎を考案した19世紀のエイダやチューリング・マシンのチューリング、ストアード・プログラム（現在のコンピュータはすべてこの方式を採用している）のフォン・ノイマン、情報科学の基礎を構築したクロード・シャノン、オブジェクト指向を考えたアラン・ケイなどのソフトウェアの先駆者や初のデジタル・コンピュータであるENIACを製作したモークリーとエッカート、マウスを考案したダグラス・エンゲルバートやICを考案したショックレーや前述のムーア、イーサネットのメトカーフなどのハードウェアの先駆者が大勢いるわけである。彼ら以外にも前述のブッシュやハイパーテキストにより人類の知識を結びつけることを夢見たネグロポンテなどの業績も忘れてはならないだろう。多くの科学者、技術者や企業家の努力の賜物として現在のPCが存在するわけである。「歴史に学ばないものは未来に対処できない」の諺通り、コンピュータの歴史も重要であると考えている。

以上のことにより、アプリケーションの学習の前にコンピュータの基本原理を学ぶことの重要性を具体的に指摘したい。

(1)デジタル・コンピュータの歴史

アナログ・コンピュータの歴史も重要であるが、現在主流のデジタルのものの歴史を中心に教える。特にENIACなどは真空管（最近の学生は見たこともない！）で構成されているが、原理は同じであることや、当初からOSが利用されていたわけではなかったこと、様々なPCが登場したことなどを教える。

(2)デジタル＝２進法ということ

電圧によってオンかオフであることを表現できるため、２進法を用いていることの説明。

(3)ブール代数によって計算が可能であること

AND、OR、NOTなどを使用して演算ができることを説明する。

(4)２進数の限界＝浮動小数点の導入

実数はそのままでは２進数では表現できないことを学習してもらう。

(5)ハードウェアの基礎

前述の五大装置の意味や役割を説明する。

(6)ソフトウェアの基礎

ソフトウェアの基本構造である「順次」、「条件分岐」、「繰り返し」を説明し、演算や処理の概念を理解してもらう。

(7)OSの役割

最もブラックボックス的であるOSの担当している役割を説明する。

(8)OSの操作方法

クリック、ダブル・クリック、ファイルのコピーや移動と削除、ドラッグ＆ドロップ、マウスの右クリックによるコンテキスト・メニューなど基本的な操作を教える。可能であれば、複数のOSを例示するのが好ましい。

原理を理解する、ということは応用力をさらに発展させられる素地を身につけると言えるのではないのであろうか。

５.「早い、安い、うまい、のうち２つを選べ」（作者不詳）

「早い、安い、うまい」はファースト・フードのキャッチ・フレーズである。問題はこのうち「二つ」を選ばなくてはならない場合、二律背反ではなく、条件のトレード・オフをどのようにするか、という問題である。

コンピュータというシステム（＝体系）を使用している場合には、この種の問題は必ず存在するものである。が、情報教育の現場においては、このような問題に対処する方法を教えることはまれである。つまり、一種のクリーン・ルームのような状態が「通常」であるかのように講義で説明がなされ、同一条件でのトレード・オフと異なる条件間でのトレード・オフを指摘することはないように思われる。

上記の例で言えば、条件が３つでそのうちから２つを選ぶため組み合わせは３種類しかないが、例えば実際のシステム開発の現場などでは、そもそも条件がい

くつありそのうちどれを選択が可能であるかは、要件を分析しなければ、いや、分析をしても不明である場合が多い。

よしんば、条件をすべて洗い出せることができたとしても上記のように「悩ましい状況」に遭遇する可能性の方が高いのである。「ウォータフォール型」というのは、名前の通り、開発の工程を滝に見立てて、(ⅰ)要件分析、(ⅱ)設計、(ⅲ)コーディング、(ⅳ)テスト、(ⅴ)デバッグという段階をそれぞれ順序立てて行う方式であり、決してこの各段階を遡ることがないように各段階を完璧にこなしてから次に進む、というある意味、完全主義的な方法である。

それに対して「アジャイル型」というのは、完璧な要件分析ができるのを待たずにわかっている範囲内で分析を行い、それを元に設計をしてコーディング以降の段階に進み、ともかくもシステムを作成して、システムを利用するユーザにその成果物を提示してフィードバックを受けながら、徐々にシステムを構築していく方法である（アジャイルとは「俊敏な」、「身軽な」という意味である)。そのため、ウォータフォール型とは異なり、上記５段階を何度も繰り返しながらシステムの構築をすることとなる。この方法は、「完璧な分析ができる」とか「完璧な分析ができたら完璧な設計ができる」という従来の誤謬を反省して考えられたものである。つまり、積極的な試行錯誤を推奨しているわけである。

ここで留意してもらいたい点は「行き当たりばったりのやり方」と「方向性を持っている試行錯誤」は異なる、ということである。確かにほとんど要件分析もせずに設計をして、そのままシステムの構築に向かうのであれば、完成品を見ることはできないであろう。そもそも、設計すら不可能である。そうではなくて、様々なトレード・オフがあることを見越した上で、いわゆる「見切り発車」をして、ユーザに試しに使ってもらえる、ともかくも稼働できる、システムを作成するわけである。このアジャイル型の方法論の優れている点は、前述の５段階を繰り返す（一回の繰り返しをイテレーションという）ことにより、条件間のトレード・オフをさらけ出すことができる、ということにあると私は考えている。つまり、当初は「悩ましい」状態であったトレード・オフがイテレーションが進むにつれて、関係各位のコンセンサスを得るようになり、落ち着くところに落ち着くわけである。

同様のことが情報教育にも必要ではないのだろうか。各種のアプリケーション

の利用方法を教え、そのようにすれば一糸乱れず完璧な成果物（＝ドキュメント）が作成できる、ということも必要であるが、次の段階（脱・初心者）には、上記のようなアジャイルなプロセスを経ることによって、条件間のトレード・オフを理解して、自分の成果物をブラッシュ・アップさせることが大事であると理解させる必要がある。

PCは知的生産の道具である、ということは、適切な使い方がある、ということである。PCの一番の特徴は「いろんな道具のフリ（＝真似＝代用）ができる」ということであり、今後も様々なものがデジタルで表現できるようになるため、その応用範囲は拡がるであろう。しかし、その道具には根源的にトレード・オフが存在するのである。早くて、安くて、うまいものを実現するための努力は必要であるが、いつでもそうできる保証はないのも事実だという覚悟もまた必要なのである。

6．プログラム言語学習は必要か

結論から言うと、大学における情報教育においてプログラム言語学習は必要ではない。もちろん、情報科学や情報工学の専門学科・学部や理数系の大学では必要であろうが、それ以外の大学では多分必要ではないであろう。それは音楽専攻の大学以外では音楽理論や実技、体育専攻の大学以外では体育理論や実技が必要でないのと同じ意味においてである。

しかし、音楽や体育専攻の大学に進学しなくても小学・中等・高等学校では音楽や体育の授業はあるし、それらを習っている。にもかかわらず、どうしてプログラム言語の学習をしないのであろうか。それは次のような理由があるからと思われる。(ⅰ)プログラム言語が英語ベースになっているから。そのため、ある程度の英語力がなければ記述することや理解することができないため。(ⅱ)プログラム言語に詳しい教員を確保できない。(ⅲ)本格的なコンパイラなどは特定企業の有償ソフトウェアである。これら以外にも、学習時間が取れない、なども理由であろうが大きくは上記の３点であろう。

それならば、大学だったら可能であろうか。無償のソフトウェアを利用してプログラム言語に詳しい教員さえいれば多分可能であろう。実際、理数系の大学で

は有償・無償のソフトウェアを利用して教育を行っている。しかしそれは飽くまでも専門としての科目・講義であり、週２回以上の講義を数年に渡り受講し、かつ多くの課題やレポートをこなしていくから可能なのである。逆に言えば、それくらいの学習時間を取らなければ本格的なプログラミング言語の習得は難しい、ということを意味している。事実、使用に耐えうるようなシステムをプログラム言語を使って構築できるようなるためにはプログラム言語の学習以上にハードウェアやアルゴリズム、数学や開発の方法論など学ばなければならないことがたくさんあるのである。

一般教養としてのプログラム言語の学習ということには、かなり高いハードルがあるが、それでも私は数年に渡って四国学院大学でオブジェクト指向言語であるJAVAを教えてきた。JAVAは無料で利用できる開発環境があり、それにフリーソフトであるエディタ（プログラム開発に特化されている文書作成ソフトウェア）を利用しての授業であるが、やはり半期14回の講義では基本的な概念を教えるのが精いっぱいである。当然、受講している学生も表面を撫でる程度のことしか学べないため、フラストレーションがたまると思われる。それは受講後に自分が作成したいと思うプログラムを独力では作成できない、ということがわかるからである。そのため昨年度からは初心者＋αの講義を開講しているが、それによって独力でシステムを構築できるかどうかは現在研究中である。

最近ではphyton=⟩やphythonといった無償のオブジェクト指向型スクリプト言語（コンパイル作業を必要とせず、手軽に動かすことができる開発言語）があるが、一般教養としてのプログラム言語の学習には今後活用されると予想される。問題点は、GUI環境で動作可能なVisual Studio（MicroSoft社の統合開発環境システム）のようなものがまだない、ということであるがこれもそのうちにJAVAにおけるEclipse（無償で提供されている統合開発環境システム）に匹敵するツールが登場してくるであろう。このような取っつきやすいツールの存在価値は大きく、そのツールを利用することで煩わしい設定作業やGUI構築のための作業をビジュアルに行うことが可能であるばかりでなく、単純な入力ミスの低減、データベースへの接続や既に開発されているコンポーネント（同じような機能をまとめた道具類）の利用ができるようになり、一貫したシステム構築をサポートできるようになるわけである。

プログラム言語の学習ほど学習曲線が極端に変化するものはないであろう。また、自然言語である通常の語学学習のように一つの言語を習得すると他の言語の習得がかなり楽なものになる。これは自然言語の比ではない。それは人口言語であるプログラム言語の原理は、現状ではほとんど同じであるからである。そのため、スクリプト言語を学習することでより本格的なプログラム言語の学習への入り口になることが期待される。

7．真の「知的生産技術のための道具」になるために

先に述べたようにコンピュータとは「いろんな道具のふりができるもの」である。OSが発明される以前はコンピュータとは全て専用機であった。つまり、科学技術計算専用、給与計算専用などである。OSが登場した以降、アプリケーションを読み込ませて様々な方面に利用できる能力が備わったわけである。

この「汎用性」というものは、単機能の集合を意味するわけではない。10数年前に販売されていた「ラテカセ」（ラジオ、TV、カセットテープが利用できる製品）とは根本的に異なり、アプリケーションやドライバ（ハードウェアを制御する特別のソフトウェア）などをインストールすることにより、様々な道具に「変身」することができるのである。しかし、その「変身」の度合いは従来からの概念を上回るものでもある。例えば、ワープロは「紙と鉛筆」というメタファー（隠喩）から発想されたであろうが、それだけのものであろうか。同様に表計算ソフトは「集計用紙と電卓」だけのふりをしているものではない。ブラウザは様々な情報を表示できる、いわば「電子の紙」というものであるが動画や音声・音楽も再生することが可能である、という一例を見ればわかるように以前に存在していた道具の機能を発展させる方向でアプリケーションは進化しているのである。

進化がもたらすものが全て良いことばかりではない。主題からは外れるが、イノベーション（もちろんPCや各種のアプリケーションもイノベーションの一種である）を開発したイノベーターはその技術なりアイデアの利用方法全てを把握できることはない。電信、電話、ラジオ、TVといったイノベーションは全て「教育目的のため」（ロバート・V・ブルース著、唐津一訳『孤独の克服　グラハム・ベルの生涯』NTT出版、1991年）に考案されたものなのである。いわく、

「この技術を利用すれば、遠隔地にいる学習者のためになる」、「目が不自由な学習者のためになる」などである。これらの技術・装置が普及していった歴史はそれとは異なっているのが現実であるにもかかわらずである。

PCにも同じような状況が生まれている。端的な例がコンピュータ・ウィルスに代表される「愉快犯」的なものである。残念なことにコンピュータ・ウィルスも一種のソフトウェアであり、その起源はインターネットでどのようにこのソフトウェアが増殖するか、ということに興味を持った学生が行ったことに端を発している（この学生の父親はコンピュータ科学者であった）。その後、営利目的のあるなしを問わずにこの種のソフトウェア（現在では悪意のあるソフトウェアの総称としてマルウェアと呼ばれている）は膨大な量になっており、日常的にその脅威に晒されることになってしまっているが、その理由は「コンピュータ・ウィルス作成キット」なるものがネット上で出回っているからである。これは各種のパラメータ（変数）をいじるだけで、新種のウィルスを作成できるものであり、そのための専門知識はほとんど必要がない。小学生や中学生でも興味があれば簡単にウィルスを作成できる道具が存在するわけである。これは言ってみれば、小学生や中学生が結果がどうなるかもわからずに高性能なマシンガンを所持しているのと変わりがないことなのである。幸いなことにこのウィルス作成キットはせいぜいマシンガン程度の被害しか与えられないのであるが、悪意のあるプログラムを作成する人間のスキルが向上すれば、核爆弾のような被害をもたらすツールを作成して配布する危険性がないわけではない。人間の知的興味と欲望はどこまでいくのかはわからないが、少なくともまともなシステムを構築できる技能と哲学を持っている科学者・技術者はこのようなもの（ウィルスなど）を作成しないとは思うが（彼らはまっとうなシステムを作るために忙しいし、マルウェアを作っても結局誰が作成したかを特定されることを理解している）、一般の利用者には大変な問題であることに変わりはない。

このような危険があっても多分、我々はPCをネットに接続して利用し続けるであろう。それは「費用対効果」で説明される、よりメリットを追求するからであるが、メリットを追い求めるための態度が必要である。コンピュータ・ウィルスに感染しないようにするためにはネットに接続しなければいい。訳のわからない状況に陥らなくするためにはPCを利用しなければいい。確かにそうであるが、

それではせっかく人類が獲得した「知的生産技術のための道具」を有効活用できないこととなる。

システム開発において言われている「狼人間を倒せる銀の銃弾はない」という言葉がある。この含意は「何にでも有効な方法論は存在しない」ということであるが、情報教育でも同様である。しかし、(ⅰ)原理を理解し、(ⅱ)適切な道具を選択して、(ⅲ)目的を明確にして、(ⅳ)戦略を立てて、(ⅴ)実行に移すことができるならば、この人類始まって以来の「何にでも変身できる道具」を使いこなせることができると確信している。

8．最後に

情報教育における方法論は未だ発展途上であると思われるが、以下のことを指摘してこの研究ノートを終わりたい。

(1)情報科学や情報工学の成果を教えることのみが情報教育ではない。

確かに情報科学や情報工学は人類の生活を劇的に変化させてきたが、その成果を教えるだけが使命ではない。科学に対する工学や実験に対する理論のように両者はその成果をフィードバックすることで進展を期待することができるわけである。

(2)成果物（＝最終的なアウトプット）をイメージさせることの重要性。

ドキュメント、システムを問わずどのような成果物を自分が得たいか、がイメージできなければ制作の端緒も訪れないであろう。最終的なアウトプットがイメージできれば、どのようなステップを踏むべきか、自分ができること、できないことを理解することが可能となるはずである。

(3)経験（場数）に勝る情報教育はない。

「習うよりも慣れろ」の言葉通り、積極的にPCを利用する環境を提供することが重要である。最近ではネットブックに代表される低価格のノート・ブック・パソコンが販売されており、これらを個人で所有することにより日常的にPCを利用させる環境を提供すべきである。

前述のアラン・ケイの言葉であるが、「未来を予測する一番の方法はそれを発明することである」というのがある。

情報教育も教育工学の一種であり、これまで述べてきた困難や誤謬や誤解も存するであろうが、「必要は発明の母」をもじれば「必要とパターンやスタイルは学習意欲への両親である」だとは言えないだろうか。今後の情報教育の研究はまだまだ続く。たかがパソコンという道具なれど、コンピュータという未知の道具なのである。