

オブジェクト指向を意識した応用ソフト教育

水島賢太郎

神戸女子短期大学 初等教育学科
mizusima@kobe-wu.ac.jp

大学における一般教育としての応用ソフトの教育であっても、その基礎にはコンピュータサイエンス (CS) やソフトウェア工学の成果を置かなければならない。このことを理解し、最近、CSの諸概念を取り込んだ応用ソフト教育の実践報告が見られるようになった。しかし、あいもかわらず単なる使用法の操作主義教育が蔓延している。そこで、学生にとって一見分かりやすく見えるが操作教育を教育効率の面から考察した。その結果、オブジェクト指向を意識した応用ソフト教育という方法が見出せたので実践した。この試みは、CSの諸概念を掘り出した教育実践報告でもあまり見られないもので、未だ試行錯誤の段階であるが、教育効果はある程度評価できると考えられる。さらに、この方法の問題点が、実は既成の応用ソフトの設計の問題と深く関係していることもわかった。

The application software education for the idea of 「object-oriented」

Kentaro MIZUSIMA

Department of elementary education
Kobe Woman's Collage

Even if it is education of application software as general education in a university, we must put the result of computer science (CS) and software engineering with the foundations. To understand this thing, practice reports of application software education that took in various general ideas of CS became find recently, but the operation education of the simple direction for uses is spreading now. So I considered such operation education, which seems to be considered as an easy understanding one for students, from a respect of education efficiency. As a result, I found one education method named "The application software education for the idea of 「object-oriented」", and practiced this method. To investigate the software education practice reports that took in various general ideas of CS, I could not fine such an education method introducing here. Of course this method is a stage of trial and error still, it is considered as an effective method for education. I also understood that problems of this method were related to problems of design of the application software deeply.

1. はじめに

今回のタイトルがオブジェクト指向の「ための」のではなく「意識した」とあるのは、指導内容がオブジェクト指向プログラミングとかオブジェクト指向データベースといった事柄よりは、より概括的意味での「オブジェクト指向的思考方」を目指しているからである。すなわち、「オブジェクト指向を意識した応用ソフト教育」で行いたいことは、およそ次の3項目である。

- ① 学生が将来オブジェクト指向プログラミングやオブジェクト指向データベース等を学ぶ機会があった際、その学習にスムーズに取り掛かれるため、通常操作主義的に取り扱われがちな応用ソフトの教育をオブジェクト指向の基礎概念を意識して指導する。その際、オブジェクト指向の概念を単に知識として学生に羅列するのではなく、応用ソフトの「実際の操作」を「私」がいかにか「概念として学んだか」を状況論的学習論の意味で提示する。
- ② 応用ソフトのインタフェース部分の諸問題（分かりやすさ、使い勝手のよさ）をオブジェクト指向の考え方で分析ながら授業を進め、「操作を学ぶ」から「操作を提案する」という発言力あるユーザー教育を心がける。
- ③ 応用ソフトのあり方に対して教育の側から発言できる事柄を見つけ出す。つまり「与えられたソフトをいかにか教育するか」から、「教育は応用ソフト開発に何が言えるか」を考察する。

現実の授業では、①から③は、時間的に切り分けた項目として取り扱うよりは、相互に混ざった形で行っている。これは、そのようなやり方がよいというよりは、未だ教育実践に期間が短く、教材として枯れていないというか、確立していないからである。私は、情報処理教育の実践に限らず、一般に教育実践を実践研究として考察し、発表するには最低3年かかると考えている。すなわち、着想と試行（プロトタイプ）および錯誤で1年、製品レベルの授業およびデータ集積で1年、集積データのチェックで1年というわけである。この意味では、今回の報告は最初の1年の段階に過ぎず、その意味で発表には早い気もするが、あえて発表に踏み切ることにした。それは、オブジェクト指向の定義といえば百人十色という説もある具合だから、私のようなオブジェクト指向でソフトを開発やデータベースの構築実務の経験が無い人間では、とんでもない誤解をしているかもしれないので、多くの人と議論したいからである。いま一つは、近年の応用ソフトのバージョンアップのテンポの速さのため、事例の陳腐化を恐れたためでもある。（とはいえ、現に今回の報告で主となるエクセルも、授業実践ではエクセルV5とエクセル97、自宅でのエクセル95という状態であり、ことによると「発表した時に陳腐化（誤り）」となるかもしれない。

2. 「オブジェクト指向を意識した応用ソフト教育」に至る背景

2-1 情報リテラシー教育の観点から

真の意味で情報化社会に突入しつつある今、情報に関する基礎概念の理解は、もはや一部専門家のものではなく、広く一般人の基礎教養として教育する必要性が出てきた。これをリテラシーとしての情報教育（以下、情報リテラシー教育）という。

情報リテラシー教育の幾つかは小中高等学校での教育段階で行われるようになってきているが、現状ではまだ大学教養教育として行なわざるを得ない状況があり、その際、圧倒的多数で取り扱われるのが（ホームページ作成やインターネットのブラウジング、電子メールといった一寸応用ソフト的でないものも増えているが）ワープロや表計算ソフトや画像処理ソフトなどの応用ソフトである。

本学会の情報教育のあり方に対する各種報告書や本研究会の報告、文部省主催の情報処理研

究集会等を踏まえた情報教育者にとって、「特定ソフトの取り扱いを覚える教育＝単なる応用ソフトの利用操作教育」は価値ある情報リテラシー教育ではないというコンセンサスが確立して来ている。その結果、情報リテラシー教育のあり方について大きく次の3つの内容が重要視されるようになってきたと思われる。

- ①たとえ応用ソフト教育でもその基礎にコンピュータサイエンス（CS）を置く
- ②情報倫理やネットワークモラルといった情報学的内容を置く
- ③特に文系で重視されているものとして、専門教育に積極的にコンピュータを取り入れる

この場合、②にせよ③にせよ、もっとも基礎となる①の教育が不十分であると、その教育目標は達成できない。しかし、例えば文部省研究会の多数の報告を見ても分かるように、情報処理学会が文部省委託研究として報告した「一般情報処理教育に関する実態調査」等の一連の報告は参考文献としての引用頻度は多数見られるものの、その意味を実際の教育実践としていかに取り扱ったか（扱えるのか）が明確に主張として読み取れる報告はきわめて少ない[1]。たとえば、CS頻出概念などがシラパス的書き方で取り扱っていると書かれても、それらが単なる知識として教えられているのか否かは判断できない場合が多い。

CSの頻出概念に限らず、一般的に「概念」とは個別具体的な世界を抽象化して得られるものである。いわば帰納的に抽象化された概念は、それが生な形の知識として教えられたのでは、単なる「暗記知識」となり、学生の身につかない（図1）。したがって概念に関わることをきちんと教育しようと思えば、図2に示すように、教員は概念を表現できる具体物を通して教育する必要がある。このような抽象概念を具体物に適應するということの意義は、単に学生指導上に留まらず、教員自身が当初分かっていたと思っていた概念の意味や価値を価値が、史的意義をも含めてより深く「なるほど」という「納得のレベル」で分かる契機となることである。

図1

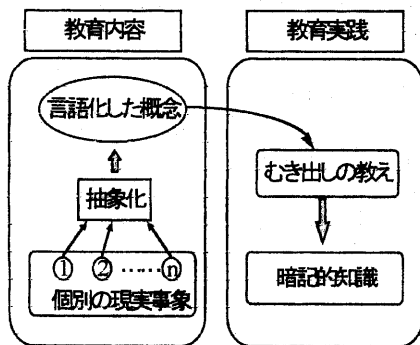
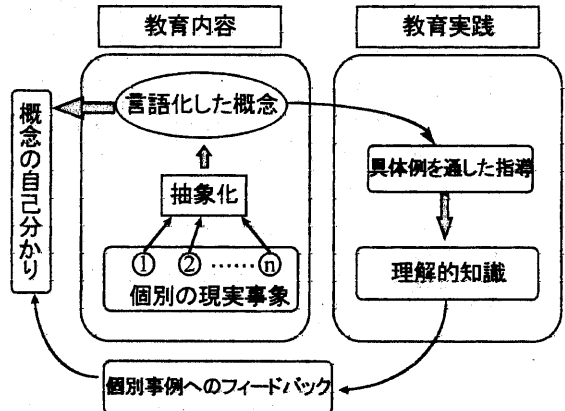


図2



なお、図2は、オブジェクト指向アプローチと対照するものでもある。

2-2 教育効果から見た個別応用ソフト操作教育の問題

情報リテラシー教育で個別応用ソフトの使用方法を操作主義的に扱う問題点は図3にまとめられる。この図から分かるように、個別応用ソフトの操作主義教育は、図1に見られた抽象化さえ抜けているのだからより問題だといえる。これの解決として最近盛んに主張されるのが、情報学や別の教育内容とのドッキングである（図4）。

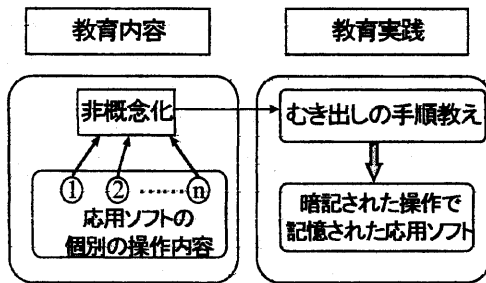
図3の操作主義派の問題点は次のように要約できる。

- ① 現実的タスクが存在しない学生の場合、単なる操作指導すら難しく、かつ、すぐ忘れられる。職場の現実的タスク下での学習効果に比べれば、学校という状況下での教育は能率がわる

い。これは、詳細なマニュアル的テキストを与えようと、レジメ風の資料でも、単に参考書の指示に留まっても、ヘルプの使い方だけでも、こと「操作部分の学習効果」の面ではさほど差はない。これはセンター試験の得点が大学入学後に著しく低下するのと同じである。格言風に書けば「学生時代に半年かかる操作教育は社会現場での具体的タスク下では3日でマスターできる。貴重な青春時代を操作学習で無駄にすべきではない」となる。

② 操作の暗記的学習の効果が冪関数的で、概念理解を伴う学習の効果は指数関数である。冪関数はかならず指数関数に追い抜かれる。

図3



に使うことは、裸でジャングルを歩くようなものである。

図4

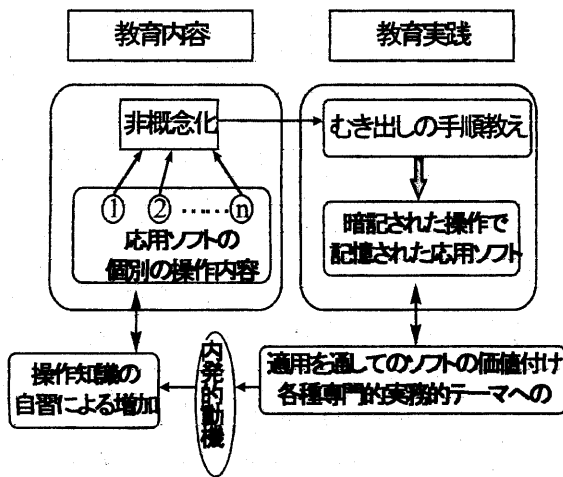


図4に示した問題点は、現実的タスクによって操作主義を越えようとするものである。問題は、現実的タスクをコンピュータ等の情報機器を使う効果で動機付けたものであり、操作主義派の問題点を一定程度解消している。しかし、情報化社会においてコンピュータをヘビーに使用すれば使うほど「情報化社会の影」にさらされるリスクも大きくなる。システムクラッシュ、ウイルス、クラッキング、パスワード管理などの諸問題がいかなる原理に関わっているかを理解せずに情報化機器を仕事

結局のところ、CS等の基本概念を身につけることが重要となる。ポイントは、「概念」ということである。概念は抽象化の産物である。とすれば、応用ソフトを操作主義に陥らず学習するとは、応用ソフトの操作を抽象化し、概念としていかに理解するかに帰結するのである。極論すれば、その抽象化がいかなるものであってもよいのだが、ニュートンではないが「ずっと遠くが見えるのは、巨人の肩に乗っているからだ」である。そして、CSの頻出概念にせよ、今回扱うオブジェクト指向にせよ、それは確かに「肩」となる。

3. オブジェクト指向を意識して応用ソフトを指導するとは

3-1 概念チャンクとオブジェクト指向

私がオブジェクト指向を応用ソフト教育との関連で取り組み始めたのは、私がオブジェクト指向のプログラミング言語でソフトを開発したりオブジェクト指向データベースの構築に関わったりした関係ではない。私の原点は常に教育にあるので、「オブジェクト指向」との関わりも教育実践の中から生まれてきた。

前述したように、操作主義的応用ソフト指導は、結局学生に暗記的に受け取られ、長期的に見れば学習効率が悪い。したがって個別操作を概念として抽象化し、概念化の意義を感じさせることが重要である。この抽象化による概念的「分かり」を「概念チャンク」という。すなわち、『あることを「分かった」と実感できるのは、次の2つの状態になったときである。①(略)②複数の

観察事例が、統一された概念として認識できた場合であり、「概念による知識のチャンク化」[2]による理解なのである。これのワープロ指導への実践は次のようになる[同報告]。

具体的には、文字列や表、図形(オブジェクト)と操作(オペレーション)で括る。そうすると、操作は次のような概念にチャンク化できる。

① 『何(オブジェクト)を → どうする(操作)』型(中略)

② 『何を(オブジェクト)を→どこに、なにと→どうする』型(中略)

当然、これらのチャンク化は、プルダウンメニューでのチャンク化とは異なったカテゴリーでの分類となる。(下線今回)

さて、オブジェクト指向はソフトウェア開発の方法論として発達してきたが、その発端は1967年のSimula67というシミュレーション言語にあるといわれている。フィルサリによれば『オブジェクトで作成したモデルは、実世界の模型であり、実世界のシミュレーションである。(中略)オブジェクト・モデルが実世界に近ければ近いほど、それらの振る舞いもより現実に近づく』、また『オブジェクト指向という見方は(中略)認知科学や知識表現の分野では(中略)よく利用されている』となる[3]。また、春木はオブジェクト指向を解説した本で、『哲学者ヴィットゲンシュタインの言葉に見られるように(中略)言語により表現することは実世界を自分の思考の中に写像することです。写像された世界に対しては自分の思考を及ぼすことが可能になる』とも述べている[4]。つまり、オブジェクト指向アプローチは、現実の抽象化による概念的理解に深く関わった方法なのである。

では、私が応用ソフト教育としてまとめた「概念チャンク」の見方に、オブジェクト指向の基礎概念、つまり抽象化されたオブジェクト、クラス、メッセージ、メソッド、関係あるいはカプセル化(情報隠蔽)、継承性など取り込みながら実践すれば何が見えるのか、どこまでやれるのか。これを、ワープロよりは難しいとされる表計算ソフトの指導で試みることにしたのである。

3-2 取り扱った応用ソフトと内容

実際の指導は、表計算ソフト「エクセル」を使った。対象学生は原則としてワープロ既習者であるが、ワープロ→表計算の順序は本質的でない。

私が担当している情報処理実習関係でエクセルに配当できる時間は(複数の大学の関係で)3コマから6、7コマ程度で幅がある。取り扱う内容は表計算ソフトに共通な基本部分である。指導項目を普通のシラバスや参考書やマニュアル目次風に書けば次のようなものである。

1. エクセル起動から、終了まで
2. ファイル(ブック)の保存と呼び出し。
3. セルへのデータ入力
4. 計算式と関数
5. 相対座標と絶対座標
6. セル操作(複写、貼り込み、形式を選択して貼り込み)
7. 印刷
8. 罫線
9. グラフ
10. セルや文字の修飾(複写、削除、セルの色等)
11. リスト
12. 行や列の操作
13. データベース機能(ソート、フィルター)
14. 図形処理

なお、13.の図形はワードやパワーポイントで既習のことが多いが今回の説明上入れた。

さて、オブジェクト指向を意識して指導すると、このような項目の立て方がそもそも問題となる。つまり『コンピュータのモデル化の対象は、基本的には「もの」である。従来の手法であると、コンピュータ上での実現手段が機能中心であったため、「もの」および「もの」どうしの関係をわざわざ機能の観点からまとめなおしていた。これには、発想の転換が要求され、最適なアプローチではなかった。』[5]。その結果、オブジェクト指向アプローチが生まれてくる。こうしてみると、先の項目の立て方は「機能中心」と「オブジェクト指向」が混在しており、各項目のどれを教えるのかといった発想では、オブジェクト指向を意識した指導は出来ないことが分かるだろう。

オブジェクト指向を意識しての指導では、各種操作は機能ではなくオブジェクトとの関連、つまりオブジェクトの属性（プロパティ）や処理（メソッド）関係にしたがって再分類しなければならない。同様、①マウスと特殊キー（ShiftやCtrlキー）の使い方 ②各種ボタン ③左クリックと右クリックの操作も、知っていれば便利という指導ではなく、オブジェクト指向の文脈で取り扱わなければならない。ツールボタンも同様である。さらに、ソフトを操作する人間もオブジェクト（スーパー・オブジェクト）と見て、具体的操作を各種オブジェクトとメッセージと関係として意識しなければならない。

3-3 指導内容の各論

以下の事柄は、オブジェクト指向として指導したい事柄であるが、実際行ってみるとうまく扱えないケースが多々あった（以下、斜体明朝体部分）。私の理解不十分もあろうが、多くはエクセル等の応用ソフトのインタフェイスが機能中心に設計されており、それにオブジェクト指向が混ざっているため、すっきりオブジェクト指向で扱えなかったからである。

3-3-1 マウス

- ①左ボタンは、オブジェクトの選択機能に割り与えられている。
- ②右ボタンは、操作機能を便利にまとめたものではなく、そのオブジェクトが受け入れることの出来るインスタンスをスーパーオブジェクト（＝人間）に可視化する機能を持つ。この意味で、オブジェクト指向を意識した指導の際、きわめて重要となる。*問題は、エクセルの設計がオブジェクト指向に徹していないことにあり、これが指導を難しくしている。すなわち、右ボタンでポップアップするメニューが、オブジェクトのテキストフレーム的仕様になっていない。*
- ③マウスによるセルのドラッグは、オブジェクトの指定であり、かつ上位クラスの作成でもある。特殊キー（ShiftやCtrlキー）を絡めた複数のオブジェクトの指定は、より上位のクラスの形成をより明確にする。*しかし、複数の応用ソフト間では統一が取れていない。*

3-3-2 オブジェクトとしてのセル

- ①セルは、表計算ではもっとも下位に属するクラスであり、データとして文字や数値、色、パターン、罫線をプロパティと、数式や複写や削除といった処理を持つ。指導では、これらは同時に一気に扱う。
- ②セルのプロパティは、インターフェイスにおいて可視化しているが、実態はカプセル化されて、またモジュールとして独立している。
- ③セルの数値データを変更すると、それを数式処理しているセルに反映するが、これはオブジェクト間がメッセージを介して関係していることをよく示している。
- ④セルは、列や行、リスト構造といったセルの上位クラスでまとめられ、さらにワークシート、ブックといったより上位のクラスをもつ。
- ⑤上位クラスでまとめられたセル群のプロパティは、下位クラスのセルに継承される。下位クラスのプロパティは上位クラスには影響しない。なお、セルのコピーの場合、「形式を選択して貼り付け」の指導が「貼り付け」より優先度が高まる。この際、複写元と複写先の関係に、クラスの継承をどう判断して良いか分からない問題が生じる。
- ⑥リスト処理をオブジェクト指向のクラスとして取り扱おうと、うまくいく場合と行かない場合がある。

3-3-2 オブジェクトとしてのグラフと図

- ①グラフも図も最下位の個々のグラフのインスタンスから、最上位の「グラフ」クラスまで構造を持っており、上位のプロパティは下位に継承される。
- ②複数のグラフや図を選択しグループ化すれば、より上位のクラスとなり、継承がある。
- ③グラフと図は上位クラスとしてグループ化できるが、どちらもセルとはグループ化できない。この解釈は難しい。すなわち、グラフとセル範囲は、別種のクラスのオブジェクトであり、グラフは

セルに従属している。オブジェクト指向でこのような非入れ子的関係性をいかなる概念として取り扱うべきかの問題がある。

4. まとめと考察

4-1 オブジェクト指向から見た、マイクロソフト応用ソフトの問題ー 抽象化の面からー

「MSオフィス」という複数の商品群をオブジェクト指向の面から統一的に扱う際、同一の操作（例えばオブジェクトの指定）などが、応用ソフトによらず共通のもの、異なったものがあるため、学習成果が転移出来ないことがある。たとえば、マウスによるオブジェクトの指定における Shift キーや Ctrl キーの機能が、ワードとエクセルでは異なっている。その原因は、人間（スーパーオブジェクト）に対する過保護というでしゃばり問題で、これが概念チャンクとしての学習を妨げることがある。同様、インタフェイスの設計思想が不明解で、機能主義とオブジェクト指向が混在しており、学習を妨げるところがある。たとえば、一太郎同様ワードでも罫線が紙面とも密着度の強さは、その典型といえる。

4-2 教育効果とトレードオフ問題

- ①オブジェクト指向を意識して指導しているのだから、当然、オブジェクト指向アプローチの概念指導の効果がある。
- ②人間をスーパーオブジェクトとして一体的に扱うことにより、エクセルのVBAというオブジェクト指向言語でのプログラミング指導時に（まだ実践していないが）効果があると思われる。
- ③表計算ソフトとしての概念指導には不要な内容（セルの色、形式を選択して貼り込み、グループ化など）がオブジェクト指向の良い例になることが多く、指導項目が増える。したがって、決まった指導時間では両者間のトレードオフが起こる。しかし、両概念を扱うのだから時間がかかるのは当然ともいえるので、時間を増やせばトレードオフは解消する。
- ④なにより、操作主義に見られる暗記主義が克服でき、概念的学びが可能になる。

4-3 オブジェクト指向アプローチの「ソフトウェア工学」パラダイムからの解放

オブジェクト指向を意識した実践教育の実例もみられるように、エクセルのインタフェイスは、オブジェクト指向ではすっきり理解出来ない。つまり、オブジェクト指向的なものと機能主義がごちゃ混ぜになっている観がある。これは経験した応用ソフトが、ワープロ（松、一太郎、ワード）であれ表計算（ロータス、エクセル）であれ、パワーポイントであれ同様である。さらに、どのソフトのマニュアルもまた参考書も、すべて非オブジェクト的な機能中心主義的に書かれている。これが、応用ソフトの学びを難しくしている原因の一つと考えられる。

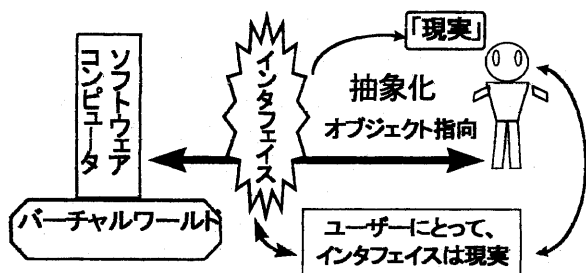
フィル・サリーは先に紹介した著書の第5章に「インタフェイス設計こそオブジェクト指向で」とあるぐらいだから、各種応用ソフト開発者は、十分このことを認識していると思われるのだが、なぜ使いやすいインタフェイスはなかなか生まれないのだろう。

私たちが生活する現実社会とは矛盾と混乱に満ちている。つまり、決して分かりやすすくない（社会学という学問分野すらある）。実は人間の現実との関わりは、現実と格闘し、現実を抽象化したモデルを作り、そのモデル化した「現実」によって現実を理解しているのである。つまり、人間にとって現実とは頭脳に写像されたモデルとしての「現実」なのである。この意味で「分かりやすい現実」は存在しない。現実とは、学ばれて、理解されて「現実」となる。

さて、メーカーにとってインタフェイスはコンピュータ上のバーチャルワールドを可視化したものだが、ユーザーにとっては現実である（図5）。とすれば、そもそも「分かりやすい現実」は存在しないのだから、「分かりやすいインタフェイス」を作ることはそもそもできない。出来るとすれば、それはメーカーの思い上がりである（最近の応用ソフトには、分かりやすいでしょう、という押し付けのなんと多いことか）。大切なことは、インタフェイスは人工的現実であり、とすれば「分かりやすい現実」は出来なくとも「学びやすい現実」は作れるはずで

ある。たとえばユーザーがオブジェクト指向の方法で抽象化して頭脳に「現実」としてモデル化する際、その「学び」を混乱させない首尾一貫性を持つことである。

図5 ユーザーにとってのオブジェクト指向



次の例が分かりやすいだろう。あなたが本屋の店主とする。もし、あなたがお客に「分かりやすい」本の置き方を考えたとして。ジャンル別？、著者名順？、発行日順？。どれでもない。ミステリーを読みたい客ならジャンル別が良いし、お好みの著者の本なら著者別。つまり、店主は「分かりやすさ」を決められないのである。しかし、どの配置にしておいても、それが一貫していればお客は容易にその店の現実を学べるのである。

最悪は、ジャンルと著者名と発行順がごちゃ混ぜになっていることである。

次に注意しなければならないことは、「学びやすさ」と「便利さ」は別概念ということである。生存的合理性と論理的合理性の概念[5]でも示したように、処理する情報量に対する記憶要領によって「便利さ」も変わる。もし人間の記憶能力が100万倍大きければ、プルダウンにせよ階層型ファイル管理にせよ、面倒なだけだ。

ソフト開発のオブジェクト指向がなぜインタフェース設計にすっきりと学びやすく反映してこないのか。この問題を教育との関わりで考えると、とりもなおさず「情報処理やコンピュータを教える」という受け身の教育から「教えから情報処理やコンピュータに提言する」という立場をも取り込むことである。これがなければ、応用ソフト教育はいつまでも新しいバージョンを追っかけ、その内容を操作主義で「いかにうまく教えようか」というサイクルから逃れられないであろう。

参考文献

1. CSの概念を指導に入れた実践教育とは、例えば、
水島賢太郎：操作主義でないアプリケーションソフト教育、情報処理研究会論文集、P355-358、文部省(平成6)、および、日常メタファーとCSメタファーの両面による情報科学の指導実践、P183-191、文部省(平成9)
神村伸一：計算機科学の頻出概念の理解を目指した情報リテラシー教育、「コンピュータと教育」Vol.97 No.125、P65-72、情報処理学会、(1997)
などが参考になる。
2. 水島賢太郎：「出来る」かあら「分かる感動」への教養情報処理教育—人間、社会、日常性とコンピュータサイエンス—、「コンピュータと教育」Vol.97 No.94、P1-8、情報処理学会、(1997)
3. フィルサリー著、本位田真一 監訳：オブジェクト指向モデリング、P6-8、日経 BP 出版センター、(1955)
4. 春木良且：オブジェクト指向への招待、P20、啓学出版、(1989)
5. 江村潤朗 監修：図解コンピュータの大百科、P954、オーム社、(平成7)
6. 水島賢太郎：生存的合理性と論理的合理性の概念—合理性の認知科学—、神戸女子短期大学学会論攷 35 巻、(平成2)