

Peer としてのコンピュータの持つべき条件

Three Conditions of Human-Computer Communication

青山学院大学文学部

鈴木宏昭

Hiroaki SUZUKI

Abstract

本論文では、コンピュータを道具と見なすのではなく、コンピュータをPeer (同僚) と見なし、インタフェースを人間とコンピュータとの間のコミュニケーションプロセスとする概念化を提案する。この見解から、インタフェースは、1.) 対話性、2.) 可塑性、3.) 開自性の3つの条件を持つべきであることが主張される。最後に、この概念化と条件から、マニュアルにおいては、1.) 手続きの記述だけではなく、課題構造の記述を行うべきであること、2.) コミュニケーションを促すような文体を用いるべきであること、3.) 比喩を活用すること、が指摘される。

1 道具的コンピュータ観

近年、コンピュータを道具と見なす傾向が強まってきている。たとえば、ビジネス関係の書籍を見ても、「コンピュータを学ぶのではなく、使おう、使いこなそうという」という主張がなされている。また、学校教育においても、コンピュータは学習対象ではなく、学習を補助する道具という位置づけがなされるようになってきている。

1.1 "道具" の3条件

これらの中で用いられる"道具" という概念はいかなるものであろうか。佐伯(1997)はその著書の中で、道具としてコンピュータを見なすという観点から、道具の持つべき3つの条件を挙げている。

非規範性

道具から人間に対してあれこれと指示を出してはならない。

手段性

道具は人間の目標の達成の手段とならなければならない。

透明性

道具はそれが使われる際に、人間の意識に上らず、人間の身体の自然な延長として存在しなければならない。

これらの条件は、道具、特にコンピュータに代表される近年の情報機器の扱いにくさを考えると、説得力を持つものといえるだろう。少なからぬ情報機器が、あれやこれやと人間に指図し(非規範性条件の違反)、人間はそれらにいつまでたっても馴染めず(透明性条件の違反)、結果としてそれらを用いなくなる、あるいはきわめて限定された機能しか用いることが出来なくなる(手段性条件の違反)。

1.2 情報機器という"道具"

上記の考え方の根底には、道具を使う人間という主人が存在し、道具はその主人の召使として、それ自身は意志を持たず、忠実に仕事を手伝うという概念化が存在している。しかし、コンピュータを代表とする近年の情報機器には、佐伯の挙げた道具の3条件が当てはまる可能性は高くない。

この理由の1つは、これらの機器の多機能性に由来する。典型的な道具は、基本的に単機能である。はさみは「切る」ものであり、ホチキスは「とめる」ものであり、かなづちは「打つ」ものである。従って、これらを使う際に何をしたいかを告げる必要はない。しかし、コンピュータは基本的に万能機械であり、プログラムさえあれば何でも実行可能である。したがって、これを利用する際には、自らが何をしたいのかをまず告げなければならない。また、1つのソフトウェアに限ってみても、その機能はきわめて多様である。たとえば、コンパスは円、あるいは弧を描くためだけに使われる。しかし、ドローイングソフトでは様々な図形を描くことが出来る。

もう1つの理由は、"編集"性に由来する。いわゆる「道具」でイメージされる、単純な道具にはこうした編集性は多くの場合存在しない。一方、コンピュータでは、文章のある部分を別の場所に移動するとか、図形を回転させるなどの編集が可能である。編集を行うには、編集を行う対象、編集の種類の詳細を特定しなければならない。たとえば、「どの部分(どこからどこ)をどこへ移動する」、「どの図形をどこを中心にしてどれほど回転させる」などを告げなければならない。

佐伯の3条件が適用できない最後の理由は、応答性、対話性に由来する。多くのソフトウェアは、その多機能性や高度な編集機能によって、対話をインタフェースの基盤とせざるを得ない。たとえば、コンピュータ(及びソフトウェア)はユーザが何かを告げると、その詳細のある種の対話を通して特定しようとする。コンピュータの方から「これこれのことをしてください」というように指示が出されることもある。一方、単純な道具の使用には、対話は介在しない。単にそれを用いて結果がでてくるだけである。こうしたことから考えると、少なくとも非規範性、透明性はコンピュータのような道具には適用できないことは明白である。

1.3 "道具"への愛着と信頼

前節では主に情報機器を例に挙げたが、それ以外の典型的な道具に対しても佐伯の挙げた3条件が適用できないケースは数多くある。

人間と道具の間には、愛着、信頼といった関係が存在しているケースがある。たとえば、職人(たとえば、大工や料理人)は道具(かんなや包丁)に対して、特別の感情を抱きつつ、日常的にその道具をケアしている。また、プロの物書きの中には、万年筆や時には原稿用紙にまでかなりの執着を見せるケースがある。

こうした道具と人間との関係は、その道の専門家だけに留まるものではない。たとえば、初心者ドライバーですら所有する車に対して、かなりの愛着を見せるケースが少なからず存在する(ドライバーが車に語りかけたり、車の機能や乗り心地とは本質的に無関係な装飾などを行ったりする)。また、初心者プログラマーであっても、自作のプログラムをより「賢く」するために、あれこれと拡張を行うケースが見受けられる。

以上のことは、道具と人間の関係は主人と召使という関係では捉えきれないことを示している。

1.4 道具的コンピュータ観が生み出す誤解

コンピュータなどを道具と見なすことは、単に間違いというだけでなく、様々な弊害をもたらす。第1に、道具でイメージされるものが、単純で単機能な存在であるため、学習が不要であるかのような錯覚を生み出す(実際には、単純な道具でさえ、それが佐伯の3条件を満たすのは、かなりの学習期間を経た後である。)。つまり、すぐに使えると思い込み、そうでないと不平をいう、使うことを放棄してしまうという行動を生み出す。

これと関係しているが、道具的コンピュータ観は、コンピュータはユーザの言いなりになるという幻想を生み出す。道具は、そもそも召使なのであるから、不平などを言わずに、どんな時にでもユーザの言うことを聞くべきだ、という誤解である。実際にコンピュータを使用する場合には、実行不可能なことがあるのは無論であるが、通常の場合には可能な操作が、ある条件下では実行できないことはよくある。

第3に、道具的コンピュータ観では、コンピュータは意志を持たないものとされるので、ユーザに問い合わせや指示を行うものとは見なされない。したがって、コンピュータからのメッセージが無視されてしまう。実際、初心者に見られる典型的な問題行動は、コンピュータからのメッセージを読まないということである。

これと関係するが、第4に、指示の出し方、手順を覚えさえすればコンピュータは使えるという誤解を生み出す。多くの場合、ユーザが指示を出し、それに対してコンピュータの方からいくつかの問い合わせがあり、それにまたユーザが応えるという過程を経て、ある機能が実行されるケースが多い。しかし、道具的コンピュータ観を持つユーザは「まず最初にこうする、次にこうする…」という手順を覚えることにその努力の大半が注がれることになる。

2 Peer とのコミュニケーション過程としてのインタフェース

道具的コンピュータ観が間違いである、有害であるとすれば、コンピュータ及びインタフェースをどのように概念化すべきなのだろうか。本論文では、コンピュータをpeer(同僚)と見なし、インタフェースをpeer とのコミュニケーションプロセスとする概念化を提案する。つまり、我々はコンピュータというpeer といろいろなコミュニケーションをしながら、共同である仕事を達成すると考えるのである。したがって、コンピュータは意志を持たず(したがって話さず)、主人のことを黙って実行するのではなく、ユーザとコミュニケーションをしながら、ユーザを助けていくものと見なすのである。

2.1 道具の人格

コンピュータをpeer と考えた時に、問題となるのはそれがどのような人格を持っているのか、

ということである。この問題は、製作者を考慮せずに考えることは出来ない。道具はそれを考案した人の置かれた状況(問題の性質、利用可能な資源、その資源の特性)、そこでの思考方法をかなり忠実に反映している。そうした意味で、道具は製作者の分身であるともいえる。

さらに、製作者は単に個人として存在しているのではなく、その人と問題状況や思考方法を共有する人間集団＝文化の一員でもある。こうした他者や文化の存在により、製作された道具が普及するのである。

とすると、道具はユーザの言いなりになる召使ではなく、ある文化に属し、その文化の価値観や歴史や問題意識を受け継いだ、ある製作者の分身ということになる。したがって、その文化に独自のものの見方、そこで利用可能な資源の制約を受けた存在なのである。はさみを例にとってみよう。はさみの製作には、鉄やそれを加工する技術が必要なのは自明であるが、そもそも紙や布のように薄いものが存在し、それらを切ることに文化的価値が与えられる状況、そして使い手の多くが右利きであるの中で、生み出されたものである。したがって、そうした状況を共有できない人々にとっては、はさみは透明ではないし、手段性も有しない。

3 課題分割

ここで特に注目したいのは、製作者が製作に関わる時の思考方法である。一般に何かを設計、製作するという場合、もっとも基本となるのは課題分割である。つまり、目標（実現したい機能）をより単純で、厳密に定義できる課題へと分割することが求められる。たとえば、中学の技術科で行う本棚の製作を考えてみよう。本棚を作るという目標は、底板を作る、側板を作る、背板を作る、それらを接続するという課題に分割される。さらに、底板を作るという課題は、適当な大きさに板を切り出す、表面を滑らかにするなどの、さらに小さな課題へと分割される。

開発者というのは基本的にこのような思考が支配する文化の中で育ち、それを身につけた人である。こうしたことを考えれば、既存の様々な装置の操作において、課題分割という認知的操作が必要とされるのも当然のこととなる。こうしたことはコピー機、ビデオ、ワープロソフトなどのOA 機器、家電機器の操作においても同様である。予約録画という課題は、それが何日、何時から始まるのか、何日、何時に終るのか、チャンネルは何か等々を設定する課題へと分割される。ワープロにおいても同様で、移動という課題は、「何を（したがって、どこから、どこまで）」、そして「どこへ」を設定する課題へと分割される。こうしたインタフェースは、すべて彼らの思考の自然な延長なのである。一般に、理科系（特に工学系）出身者には機械音痴は少ないのは、彼らがこうした文化的背景を持っているからではないかと思われる。

こうした思考方法は製作という作業に必ずともなってくるものであるが、利用という文脈においては必ずしも自覚されているとは考えられない。実は、初心者ユーザが見せる、一見理解不可能な行動は、この課題分割という観点から説明できる。

筆者らは、コピー機を用いて、機械音痴と呼ばれる(あるいは自認する)人のパフォーマンスを分析する中で、彼らは課題分割を行わない、あるいはコピー機の製作者が仮定するのとは全く異なった形で課題分割を行っていることを明らかにした(鈴木・植田・堤, 1998)。

たとえば、「両面でソータを用いて5部コピーをとる」という場合、

1. 両面の設定を行う、

2. ソータの設定を行う、
3. 部数の設定を行う、

が必要になる。これらを設定せずに課題を遂行することが不可能なことは、半ば自明である。しかし、この課題の遂行に1時間程度かかるような重度の機械音痴には、このことは自明ではない。彼らは、すべての設定を行う前にスタートボタンを押したり、1つの設定を途中で中断して別の設定を行ったりする。

こうした人達に課題分割を教示したり、課題分割が自然に見てとれるようなディスプレイを操作させると、遂行時間が減少し、エラーも著しく少なくなる。

これらの実験は課題分割さえ獲得すれば機械音痴がなくなるということ意図したものではない。そうではなく、ユーザの文化とは異なる文化の思考方法を教える、あるいは可視化することにより、コミュニケーションを円滑にすることを目的としたものである。

4 道具と人間の対話のための条件

コミュニケーション的に道具、インタフェースを考えると、別の条件が導かれる。以下の条件を考える際に重要なことは、これらはコミュニケーションという「場」の条件であり、道具側、ユーザ側が持つべき条件と解釈されてはならない。コミュニケーションがうまくいかない責任は双方にあるのであり、どちらか一方が全責任を負うというのは正しくない。したがって、インタフェースが適切でないという場合、そのインタフェースを単に非難したり、またユーザの無能を嘆いても、事態は改善されない。コミュニケーションの「場」への双方の関わり方が問題にされねばならないのである。

4.1 対話性

この条件は、コミュニケーションそのものための条件である。この条件を道具が満たすには、道具側に

1. ユーザが行為の目的を伝える手段があること、（目的レベルでの対話ができるインタフェースは少ない（カット、ペーストで移動というのは目的レベルの対話ではなく、手段レベルである）
2. 伝えられたことが了解されたのか否かを伝えること、
3. 自分の現在の状態を伝えられること（自分の現在の状態を表示しないものがある（バッテリーの入っていない8mmカメラ等）、してはいるが意味不明のもの（古いMacの爆弾マークととともに出てくるエラー番号）、

が必要であり、ユーザ側には、

1. 「道具は何かを伝えようとしている」という信念、
2. 対話が成り立たないことがあるという信念、
3. 道具は状態を持つという信念、

が必要である。

4.2 可塑性

おそらく道具-人間とのコミュニケーションが、人間-人間のそれともっとも異なるのは、可塑性である。佐伯(1997)が指摘しているように、特定の、そして初心者には恣意的と思えるようなやり方でしかコミュニケーションができないという現状は改善されねばならない。

人間の場合、コミュニケーションがうまくいかない時には、別の言い方をしてみる、例を出して説明するなどのいろいろな会話のモードがあり、聞き手の方でも相手の特性を推測し、様々な解釈を試みる。

よって、道具側では、別の伝え方、別の解釈の仕方を実装しているかということ、つまり冗長性が保証されていることが重要である（たとえば、macro言語を持っているワープロや、X-windowのように自分の会話しやすいWindow Managerを選択できるものなど。）。

一方、ユーザ側のポイントは自らが伝えようとしている目的-行為がどのような構造を持っているのかを考え直してみるということ重要である。これには課題分割という考え方が、少なくとも現在の道具に対しては、大変に有効である。

4.3 開自性

これは、はじめて道具に触れる時（あるいは問題状況が発生した時）、その道具の製作者、あるいはその人が属する文化が、道具を含む状況をどのように概念化しているのかが、了解できるようになっていなければならないという条件を指している。また、ユーザはそれによって自らが行ってきた類似の行為がどのように構成されているのかを製作者の文脈の中で問い直さねばならない。

簡単にいえば道具側の要件としては構造、組織化がはっきりと見えることが必要になる。自らがどのような意図の下に、そしてどのような制約の下に、インタフェースを設計したのかをはっきりと提示する。また、ユーザは、その新しい構造に接することにより、自らを問い直す中で、自らの文化を相対化していくことが必要となる。

今まで佐伯(1997)が述べた道具の持つ3条件を批判的に取り上げ、人間とコンピュータのコミュニケーションの促進のための条件を提出した。ここで明確にしておきたいのは、佐伯のアプローチと本論文のアプローチの関係である。佐伯は確かに道具について不適當な概念化を行い、道具の条件を導き出している。しかしながら、佐伯は1992年の論文で、コンピュータと人間との関係を異文化間コミュニケーションとする主張を行っている(佐伯, 1992)。さらに、道具の3条件を明示した1997年の著書においても、製作者文化の問題や手順主義の問題点などを取り上げており、そもそも本論文は佐伯のそれらの主張を出発点としている。したがって、本論文は佐伯の理論の批判的拡張と見なされるべきものである。

5 コミュニケーションを促すマニュアル

以上のような考察から、ユーザのマニュアルの理解を促すためのいくつかの示唆が得られる。ここでは、1). 手続き主義の克服、2). マニュアルの文体、3). 比喩の活用、について簡単に述べる。

5.1 手順主義を越えて

一般にマニュアルは手順を事細かに述べていくという形式をとる。「まずxを行う。次に、yを

行う。次にzを行う.....」という形式が典型的である。その一方で、個々の操作がタスクあるいはシステム全体の中でどのような位置を占めるのかについて記述してあるマニュアルは極めて少ない。これらは専ら操作の手順を逐一伝えているという意味で、手順主義と呼ぶことができる。

手順主義では、ソフトウェアの構造、インタフェースの構造の理解は、専らユーザの帰納的推論に頼らざるを得なくなる。純粹に理論的に考えると、こうした帰納によって正しい構造を推論することは極めて困難である。実際、誤った一般化を行ったり、一般化がそもそもできずに個々の手順をいちいちマニュアルで確かめながら行うユーザは少なくない。

課題分割の実験からも同様のことが示唆されている。つまり、初心者が戸惑う原因の1つは、課題がどのように分割されているか、どのような構造になっているかがマニュアルを読んでも理解できないことである。

我々は、コピー機の課題の分割構造が直観的に理解できる支援画面を付け加えたインタフェースにおける初心者ユーザの行動を分析した。その結果、支援画面を用いたユーザはそうでないユーザに比べて、エラー回数が有意に少なかった。また、多くのユーザが支援画面は大変に有効であったと報告している。

これらのことから、マニュアルにおいても個々の操作手順に加えて、それらが課題全体の中でどのような位置付けになっているのか、課題およびインタフェースの構造がどのようなものを明示すべきであることが導かれる。

5.2 マニュアルの文体 マニュアルの文体が相手を意識させるようになっていない点もまた問題である。典型的なマニュアルは次のような文体で書かれている。

1. 編集メニューから、設定を選択します。
2. 表示をクリックします。
3. フォントまたは配色をクリックします。
4. 必要に応じて設定を変更します。
5. OK をクリックします。

ユーザは実際にはその操作を行うことにより、ソフトウェアとコミュニケーションをしているにもかかわらず、ここにはユーザの行うべきことしか書かれていない。つまり、ここでは相手が全く存在していないかのような文体が用いられている。これでは、ユーザはコミュニケーションを行っているという意識を持つことは出来ない。たとえば、

1. 編集メニューから、設定を選んで、「設定をしたい」と伝えて下さい。
2. すると、表示を設定したいのか、...を設定したいのかを伝えるボックスが現れますので、「表示」を選んで、「表示を変えたい」と伝えて下さい。
3. フォントを変えたいのならば、「フォント」を選んであげて下さい。
4. 自分が変えたいフォントを伝えたならば、「OK」をクリックして、「これでいいよ」と伝えて下さい。

というように書き直すことで、ユーザがソフトとコミュニケーションをしているという意識を持つことが可能になると思われる。

5.3 比喩の活用

開発者は熟達ユーザはさまざまな比喻やジャーゴンを用いている。たとえば、

1. 「ピーエス吐かせちゃってよ」
2. 「エディタぬけてさー...」
3. 「すいませーん、僕のやつ殺してもらえます?」

などはよく使われる。これらは単にエキセントリックな表現と見なしてはならない。1では、プログラムが入力を受け取り、ある出力を出す（吐く）という概念化に基づくものであるし、2はOSとアプリケーションの関係をある種の空間的な枠組を用いて概念化したものであるし、3はプロセスがある種の生き物のように（したがって自律的に）動くという擬人化に基づいている。

熟達者のこのような比喻は、彼らが豊かな、そして適切なイメージを持ちながら、コンピュータに接していることを示している。マニュアルにおいても、これらの比喻を積極的に用いながら、熟達者が持つイメージを伝え、豊かな概念化を促すことは可能であると思われる。

参考文献

- [1] 佐伯胖(1992) ヒューマン・インタフェースは異文化交流の場である。認知科学の発展5, 5 - 28.
- [2] 佐伯胖(1997) 新・コンピュータと教育。岩波書店。
- [3] 鈴木宏昭・植田一博・堤江美子(1998) 日常的な機器の操作の理解と学習における課題分割プラン。認知科学5, 14 - 25.