

## 解説

### 新しいプログラミング環境



## 6. 究極のプログラミング環境†

有澤 誠†

### 1. はじめに

本稿では「究極の」という言葉を、時間軸で  $t \rightarrow \infty$  をさすものと解釈する。すなわち現在のプログラミング環境の影響が及ばなくなった未来の状態だとみなす。そのような十分先の世界では、業務としてのプログラミングはおそらくほとんど必要でなくなっていることであろう。プログラミングは楽しみとしてのプログラミングレクリエーションと、芸術の題材としてのアートオブプログラミングくらいになっているものと想像できる。したがってこれから考察していく究極のプログラミング環境も、これら2点を対象としたものでなくてはならない。

現在われわれがプログラミング環境あるいはプログラミングの道具と考えているものの延長上にあるものは、この意味からは究極のプログラミング環境には含まれない。たとえば、知的なエディタ、高水準プログラミング言語処理系、プログラム解析計量システムなどは、現時点で必要なプログラミング環境である。これらは技術が未成熟な時代を乗り切るための過渡的なもの、Brooks<sup>5)</sup>の言葉を借りれば、偶発的な複雑さを和らげるためのものである。究極のプログラミング環境からみれば、単なる過去の産物でしかない。

大多数のプログラム作成の仕事が自動化された状態で、なおわれわれに残されるであろうプログラミングの仕事は、高度に創造性や芸術性を必要とするものである。そこでは、人間の創造性をいかにしてコンピュータに移すかが問題となる。そうした状況で人間をたすける道具や環境がどのようなものであるべきかを議論してみたい。問題点を整理すると次のようになる。

(1) 人間の頭脳の思考の速さに追従し、思考を妨

げない形で情報をコンピュータに伝達するための環境はどうあるべきか。できれば、手指を使わずに話しかつて口述するような形式をとりたい。音声を用いずに脳波を直接コンピュータに入力する方法はあるか。

(2) 話しかつばは、人間の思考をコンピュータに伝達するために十分か。イメージ情報などは必要になるか。

(3) コンピュータ側に、人間からの話しかつばによる情報伝達を理解することができるか。人間へ情報をフィードバックさせて確認をとる必要はないか。思考の流れを妨げないためにはどのような方法がよいか。

以上3点について、具体的に考えていく。以下の議論は、1987年夏のプログラミングシンポジウムでの発表<sup>3)</sup>およびそのときの討論がもとになっており、さらにその後の MIT Media Lab. 見学の印象を含めて、考察を進めたものである。

### 2. 究極の言語としての自然言語

さきにふれた Brooks<sup>5)</sup>には、自動プログラミングとは、常にその時代に使える言語から一段階抽象度を高めた言語の使用をさしている、という意味のことが書いてある。そこで、言語の抽象度をどんどん高めていった極限が、究極の言語ということになる。それは人間が日常用いている自然言語にはかならない。

もちろん、現在の自然言語を言語階層の中間点とみなす考えかたも哲学的視点からは十分可能である。しかし、ことコンピュータと人間との共同作業という視点からは、コンピュータがわれわれが日常用いている自然言語を理解してくれて、英語や日本語や中国語でコンピュータと会話できる水準に達したなら、いちおう満足できるであろう。

また会話という点に重点を置かずに、自然言語の中でも書きことばではなく、話しかつばを中心に考える

† Ultimate Programming Environment by Makoto ARISAWA (Yamanashi University).

† 山梨大学

べきであろう。ただし人間同士の会話と異なり、コンピュータと人間との会話には、それにより適した環境が必要になる。単なる音声認識および音声合成の研究成果だけでなく、もっと新しい発想があつてよい。

もともと音声は話しことばの媒体に過ぎず、この本質ではない。話しことばを用いるとき、音声に出さなくても、頭の中だけで話したり考えたりすることが可能である。

数年前に声に出して考える (thinking aloud) プログラミングの実験をやった経験<sup>2)</sup> からすると、もしもしゃべっていることがそのままの速度でコンピュータに入力されるなら、思考をあまり妨げられずにすむように思う。もちろんこれがスムーズに行くためには、人間側にも多少の訓練が必要であろう。しかし新しいキーボードに慣れるための訓練などに比べれば、これはさほど問題にはならない。

実際、たとえば本の執筆をする際に、ペンで原稿用紙のますを一字ずつ埋めていくやりかたや、ワープロの前でキーをたたくやりかたと並んで、口述筆記というやりかたが行われている。これは thinking aloud のひとつの例であり、考える速度と記録する速度が比較的近いことで好まれている。

話すときと書くときでは、文体に差が生じることは確かであるが、編集前の初期入力としては、この差はそれほど問題にはならない。書きことばによる記述は、初期入力した文章に推敲を重ねた結果であることが多く、会話の場合のような実時間性をもっていない。したがってプログラミング環境として扱う対象は、話しことばのほうに重点を置くことになる。

### 3. 究極のプログラミング環境

プログラミング環境とは、人間の頭脳とコンピュータとのインタフェースであると考えることができる。通常は人間の視聴覚および手や指を介していることから、視聴覚や手指の補助をすることが中心的課題になり、頭脳のはたらしの補助は二次的な目標である。

このときコンピュータから人間への情報伝達が多様性に富んでいることに比べて、人間からコンピュータへの情報伝達の方法は現時点ではきわめて制限されている。これまで、穿孔カード/テープによる方法、キーボードからの打鍵、マウスやジョイスティックなどの機械的装置、と発達してきた延長では、人間の思考の速さに追従することは容易でない。コンピュータ入力のために人間の思考が中断されてしまう。たとえば木

村泉翻訳の本<sup>7)</sup>でも、いかにしてキーボード入力をスムーズに行うかを論じている。

また MIT Media Lab. の未来指向の諸研究<sup>4)</sup>をみても、高度なテレビ、電子出版、コンピュータアニメーション、ホログラフィ、グラフィックス、映像、未来の学校でのコンピュータ利用など、コンピュータから人間に働きかけるメディアを扱うものが多い。人間からコンピュータに働きかける側は、電話などによる音声入力、人間の楽器演奏をコンピュータが追従するプレイバック式学習程度しかない。それだけ、人間のもつ柔軟性がマンマシン・インタフェースに果たす役割が大きいことになる。

将来可能性がある入力機器の中では、音声入力があるであろう。このとき問題になるのは、思考自体の環境が声を出すことを許すとは限らないことである。いいアイデアを思いつき、楽しく思考を進展させることができる機会は、コンピュータ機器の完備した書斎の机の前よりも、通勤電車の中とか、他の人の講演や講義を聞いている途中とか、寝床で夢うつつときとかであることが多い。だから声に出して考えるといっても、頭の中だけでしゃべり、実際に音は出さないような環境でのコンピュータ入力を考えたい。

筒井康隆の「火田七瀬シリーズ」<sup>6)</sup>で、超能力をもつ主人公七瀬は他人の頭の中の思考を読み取ることができる。これは小説であるからどの程度まで情報伝達できるかは正確には分からない。しかし技術の問題として、普通にしゃべっているとるように頭の中で逐次的に文字列を生成していくのであれば、脳波の分析で検出することができそうな気がする。たとえば常温超電導を利用して、頭の周辺にできている磁場を検出するといった方法が可能性をもっている。

甘利俊一著バイオコンピュータの本<sup>1)</sup>を読むと、脳の情報処理機構を詳しく解明しようと研究が進んでいることが分かる。それと関連して、脳から口へ伝達するための話しことばの形に記号化された情報がどのような脳波に対応しているかという研究はどの程度進んでいるのか、大きな興味と期待をもっている。

人工知能の研究の中で、脳そのものの情報処理のメカニズムの究明と並んで、脳と感覚器や手指との情報伝達の具体的な形式を解明することが重要であろう。そのような研究から、「究極の」プログラミング環境への手がかりが得られるように思う。

ここで述べた方法は、thinking aloud を実際の声に出さずにやるだけであるから、ほんとうの思考に比

べるとかなりギャップが生じる。キーボードをたいたりマウスを転がしたりする代わりに、頭の中で聞き手を意識してしゃべるわけである。

ものを考えているときに、それをそのままコンピュータにうつすことはここでは考えていない。意図してコンピュータに情報伝達しようとする場合に、手指や声を用いずに済ませることを論じている。したがって、伝える内容はことばの形に頭の中で変換したものに限る。あいまいなイメージのままではない。

私の *thinking aloud* 実験での経験によると、考えていることをことばで言う努力にはあるエネルギーが必要で、考えていることの一部分しかことばになって出てこない。しかしことばにならない部分にはあいまいだったり不正確だったりして役にたたないものもあり、また訓練によってことばにならない部分をことばにすることもできるようになる。さしあたりは話しことばの形に表現できる情報だけに限っても、ここで論じているプログラミング環境としてはさしつかえない。

実際私は講義の予行演習などを、通勤電車の中で声を出さずに頭の中で話す形で行うことが多い。明確な話しことばになっており、火田七瀬のような超能力者が近くにいたら、その内容を正確に読み取ることができだろうとよく思う。

筒井康隆の小説には、抽象画家の頭の中にあるイメージは抽象画そのものだという話が出てくる。こちらはコンピュータによるイメージ処理のようにドットマップの形では記号化されていないと思われる。イメージの場合には、話しことばを脳波から読み取るよりはずっと困難になるであろう。

最近の軽薄短小のならいから、現在のワークステーションはいずれカード式電卓くらいの大きさになり、いつもコンピュータを携帯することが可能となる。声を出す代わりに、現在若い人がヘッドフォンを耳に付けて歩いているような形で、脳波検出入力機器を身に付けて歩く。これを介して、声にも出さないでただ頭の中で話す形式で考えていることを、そのままコンピュータに入力できる。

私の想像する究極のプログラミング環境は、ややSF的ではあるが、このようにして脳からコンピュータに直接情報伝達することから始まる。思考の補助に、カードくらいの大きさの画面に記号列や図形を表示することが必要である。それによってコンピュータから脳への方向にフィードバックをかけるわけで

ある。

さきほど、コンピュータから人間への情報伝達の方法は多様性に富むと述べたが、こんどはおもしろいことに、コンピュータから脳へ直接情報伝達するほうが容易でない。人間のもつ多様性はほとんど感覚器にたよっている。コンピュータの記号情報を脳に直接入力する方法は、その逆に比べてより困難であるような気がする。

こうしてみると、コンピュータへプログラミング情報を伝達する場合に、話しことばだけでは不十分かもしれない。なぜなら、話しことばは、会話によるフィードバックを前提としたものであり、一方通行の情報伝達手段としては書きことばほど信頼性がないからである。

コンピュータから脳へフィードバックがかけられない場合には、視聴覚の助けを借りてそれを行うしかない。ひとつはコンピュータからも音声によってヘッドフォンから情報を返す方法、もうひとつはカード状のワークステーションのディスプレイに表示した情報を読み取る方法がある。最近いろいろな研究が進んでいる。マルチメディアあるいはハイパメディアは、こうしたコンピュータ側から人間への情報のフィードバックに、特に強力な手段となるのではないかと思う。

最後に、話しことばによる情報をコンピュータに理解させる点については、私は比較的楽観している。音声によるコンピュータ入力が *aloud* か *in silence* かという相違はあるが、話しことばの理解について、現在研究されている音声認識の手法がそのまま利用できる。入力となる波形が音声波形か脳波であるかという部分を除けば、音素、語、文、文章などに階層的に識別し、文脈から意味論モデルを構成して自然言語解析をするという方策で、コンピュータによる話しことばの理解が可能であろう。

むしろさきほど述べたように、話しことばとして自然言語だけを用いるのではなく、*Speak Lisp* あるいは *Speak Prolog* のように、頭の中に明確に記号化した形で想い浮かべるだけであれば、自分の好きな人工言語も合わせて使いたくなる。プログラミングに関する情報伝達だから、少なくとも部分的にプログラミング言語を用いることは自然であろう。

ただはじめから全部プログラミング言語で思考できるなら、プログラミング環境などを論じるまでもない。思考過程にはプログラミング言語と自然言語がか

らみ合った形で関与してくるため、プログラミング環境を介してその境界部分をスムーズにしているであろう。究極の環境では高水準プログラミング言語処理系は不要か書いたが、高水準プログラミング/思考言語は必要かもしれない。プログラミング支援のための環境や道具を論じていると、いつの間にか言語へ戻ってしまうようである。

#### 4. おわりに

コンピュータの世界では、「こうあってほしい」という技術的諸問題は、遅かれ早かれ必ず実現されてきた、というのがおおかたの歴史である。そうすると、本稿で述べてきた  $t \rightarrow \infty$  でのプログラミング環境も、意外に早い時期に実現するかもしれない。そのときに、また誰か夢多き研究者が、「究極の環境」としてどんなものを思い描くか、想像すると楽しい気持ちになる。

マルチメディアの情報処理が、Brooks のいう偶発的な複雑さを解消するためだけでなく、本質的な複雑さにかかわってくるなら、究極のプログラミング環境へ一歩ずつ近づいていける。人間の頭脳とコンピュータとの直接情報伝達の媒体こそが、究極のメディアということになるわけである。

MIT Media Lab. の諸研究の評価は人によってまちまちのようであるが、私個人としてはこうした未来指向、究極指向の夢のある研究を忘れたくない。小さな技術の進歩を蓄積することの重要性は十分承知しているが、目先の経済性の効用だけを目的とした研究だけで満足したくない。コンピュータを人間が安楽に

暮らしていくための手段として利用するだけでなく、人間により広い想像力の可能性を与えるような媒体として用いていきたい。現在はまだ手が届かない未来に向けての研究も、ぜひ並行してやっていきたいものである。

謝辞 夏のプログラミングシンポジウムで、建設的な意見を述べていただいた諸氏、MIT Media Lab. 見学の機会を与えてくださった慶應義塾大学相磯秀夫教授および斎藤信男教授、本稿をまとめる機会を与えてくださった編集担当委員日本アイ・ビー・エム(株)大場充氏に感謝いたします。

#### 参考文献

- 1) 甘利俊一: バイオコンピュータ, 岩波書店(1986).
- 2) 有澤 誠: プログラム作成過程分析の心理実験とプログラミングツールについて, 第21回プログラミングシンポジウム報告集 pp. 10-20 (1980).
- 3) 有澤 誠: 究極のプログラミング環境への私見, 1987年度夏のプログラミングシンポジウム報告集 pp. 135-138 (1988).
- 4) Stewart Brand: Media Lab-Inventing the Future at MIT, Viking (1987) (和訳あり).
- 5) Frederick P. Brooks, Jr.: No Silver Bullet-Essence and Accidents of Software Engineering, Computer 20-4, pp. 10-19 (1987).
- 6) 筒井康隆: 家族百景, 七瀬ふたたび, エディプスの恋人, 新潮文庫.
- 7) Tom Gilb, Gerald M. Weinberg: Humanized Input-Techniques for Reliable Keyed Input, Winthrop, 1976 (木村 泉, 米澤明憲共訳: 計算機入力の人間学, 共立出版, 1986).

(昭和63年11月16日受付)