

ソフトウェア危機*

高橋 秀俊**

電子計算機が世に出てから 20 年、その間にそれは予想をはるかに上回る進歩を遂げ、とくに、いわゆる情報革命の形で、その力を社会のあらゆる層に浸透させつつあることは、誰もが知るとおりである。この趨勢に対し、わが国は出発点で約 10 年の遅れをとったとはいえ、その後の官民一致の努力の結果、ようやく差は縮められ、世界第 2 の電子計算機生産国として、アメリカに 2~3 年の遅れで追隨できる程度まで、こぎつけたようである。

しかしながら、以上は電子計算機のハードウェアについてであり、一方、電子計算機を中心とする、いわゆる情報産業は、ハードウェアの能力を十二分に発揮させるに必要な各種のシステムプログラム、あるいはさらに個々の応用分野の要求にこたえる応用プログラムなど、いわゆるソフトウェアの裏づけが必要であり、さらには、人間と計算機を含んだ大きいシステムの設計を行なうシステムエンジニアリングなど、広いスペクトルをもった総合産業であることに、注意しなければならない。そうして情報産業の重心はすでにソフトウェアの方に移っており、アメリカなどでは、さらに、システムエンジニアリングの方に重点が置かれてきているようである。

さて、わが国の現状は、遺憾ながら、ソフトウェアにおいては、アメリカはもちろん、ヨーロッパの主要国にくらべても、まだ 2 流の水準であることを認めざるを得ない。このことは、本会の会員がようやく 3,000 名あまりということにも、反映されていると思われる。この立ち遅れの原因はいろいろあり、一部の人に責任を転嫁するのは妥当でないが、結局は、ソフトウェアの重要性に対する認識が、指導者の間に欠けていたこと、ひいては、生産されたハードウェアがソフトウェア人口を再生産することに投入される率の、低かったことに帰着されよう。最近、ようやくこの点の認識もあらたまりつつあり、次第に改善に向かうことを信

ずるものであるが、もし、今日のままで推移するならば、5 年ならずして、ハードウェアの生産における今日の地歩も、他にゆずることになるであろう。

このように、ソフトウェアに関してわが国は、とくに悲観的な状態にあるのであるが、これを世界的にみても、電子計算機工業はソフトウェアの面で、一つの危機に直面していると断ぜざるを得ないのである。アメリカにおいてさえ、IBM のシステム 360 のソフトウェアは、予期しなかった困難に逢着しているようであり、また、一時絶大な期待をもって迎えられた Time Sharing System も、小規模のものは各所で実用されているが、最初最もその効果が期待された、大規模な TSS は、最も注目を浴びている MIT の Multics が、まだ、本格的にはたらいっていないということである。このようなソフトウェアのつまづきは、今後ますます大型化して行こうとする、電子計算機システム自体の将来に、暗いかげを投げかけるものである。

このような状況は何を意味するものだろうか。それは、単なる一時的・偶然的なできごとだろうか。残念ながら、私には、そうでないように思われる。むしろ、これはソフトウェアのもつ本質的な問題につながるように思えるのである。

それは、システムの“複雑”さに関係している。システムの複雑さを定量的に定義することの必要性は、von Neumann なども指摘しているが、適当な方法は見つかっていない。しかし、たとえば、システムを記述するに必要な長さなどが、その一つであろう。そういう意味の複雑さについて考えるとき、電子計算機の中央処理装置は、いろいろな機械の中では、ずばぬけて複雑な方に属する。それに反して記憶装置は、素子数はもっと多いにもかかわらず複雑ではない。同じようなものの規則的な配列が、主体を占めているからである。

さて、このように考えるとき、今日の大規模計算機のソフトウェアは、ハードウェアをさらに上回る想像を絶した複雑さをもっているのである。そうして、そういう複雑なものをつくることは、二つの面で、困難に

* The Software Crisis, by Hidetosi Takahasi (University of Tokyo)

** 東京大学・理学部

ぶつかるのである。

まず、ソフトウェアそれ自身が複雑で、それは、たくさん部分の複雑なからみあいであるから、部分部分の情報を受け渡しのための余分な負担がかかる。 n 個の部分から成るシステムでは、その部分の間の情報交換に、 n^2 に比例する余分な命令を必要とすると考えると、各部分の語数が決まっているとすると、有効に使える語数は、 $an - bn^2$ に比例することになり、 n が増すと、有効な語数は極大を過ぎると、かえって減少する勘定になる。これは極端な考え方だとしても、非常に複雑なシステムは、有効な仕事をする部分の割合が、非常に低くなることは確かである。

もう一つの困難は、そのようなソフトウェアシステムをつくる過程にある。そのような大きいソフトウェアは、多数の人がチームをつくってやらなければならないが、そのときは、それらの人の間の情報交換に同じ問題が生じて、いくら人を増しても、生産力は上が

らないという結果になるのである。

以上は、複雑なシステムをつくることに対する一般的な困難を述べたのであるが、とくに、その困難がソフトウェアにかかってくる理由は、電子計算機の急激な大型化にある。大型化は中央処理装置の複雑化をももたらすが、それはむしろ緩慢である。それに対し、記憶容量の増大のピッチは、はるかに急速であり、そうして、それは必然的に、より多能の、より複雑なソフトウェアへの誘惑を呼びおこす。そうして、その実現に、前述のような難関が待ち受けているのである。

もちろん、今日問題になっている程度のソフトウェアの複雑さが、最終的な限界になるとは思われない。危機はいずれは打開され、さらに複雑なものへと発展するであろう。しかし、今後の道はますますけわしさを加えて行くものと、覚悟することは必要であろう。

(昭和44年9月13日受付)