



研究ノート

ソフトウェアの研究について

都 倉 信 樹*

ソフトウェアという語は、計算機の物理的な部品や構成物をハードウェアとよぶことに対比してつくられ、1960年代には一般的に使われるようになった。ソフトウェアを含む合成語も多い。例えば、ソフトウェアハウス、ソフトウェア技術者、ソフトウェア工学、ソフトウェア危機、ソフトウェア製品等々。

ハードウェアは物理的な実体をもつが、ソフトウェアは物質的なものでなく無形であり、それを物理的な容器物としての、ディスクとか、テープに入れてはあるが、その内容はどのような媒体の上にあるかには依存しない。紙に書いたものでもよく、ソフトウェアは「紙と鉛筆」と、「頭」があればできるという面がある。

一方、プログラムという語があり、両者のどちらを使っても通じるような文章は多い。これは両者がほとんど同一視されうるということである。実際、ソフトウェアを作るという作業は最終的に、計算機上で動くプログラムをつくるのが目的であり、プログラムを含まないソフトウェアはまず考えられない。しかし、厳密には、両者は区別されるべきで、プログラムは計算機向きにかかれたもので、「実行」できるものでなければならないが、ソフトウェア製品の中には、プログラム以外に多数のドキュメントなどが含まれているので、ソフトウェアの方がより広い。

ソフトウェア作りの実際をみてみよう。ソフトウェアと言っても、日曜大工がホビーとしてちょっと作る棚のような程度のものから、大手建設会社が作る超高層ビルに匹敵するようなものもある。実際、コンパイラとよばれる基本ソフトを一つ作るのに、ソフトウェアハウスは、例えば、1人年という工数か、数千万円という

コストがかかるというであろう。これは個人住宅程度に匹敵する。これなどは比較的小規模のソフトウェアであるが、もっと大規模な銀行のオンラインシステムなどは数年かけて計画をねり、また、数年かけて作成し、1週に1支店ずつ旧システムと入れかえていくので、全支店が新システムになるのに、1年以上かかるというような作業で、これなど大規模なビルをこえるくらいのコストがかかることもめずらしくないようである。

このようなソフトウェアは多数の技術者の永年の努力のもとに作られるが、でき上がったものは、相当大量のドキュメントと数巻の磁気テープに入ったプログラムというようなことである。重さ当たり価格の高い工業製品の代表であるLSIチップより（無形のものであるが、適当な媒体の重さを用いて比較するとすれば、…）重さ当たり価格で高いものもあるくらいである。ソフトウェアの本質的な価値は「情報」にある。ハードウェアは同一のものを作るには、同程度の努力が必要であるが、ソフトウェアはコピーは容易にでき、ここに、ソフトウェア権利保護の問題もおこってくる。

コピーしたソフトウェアが他でも使えるなら販売することもできるが、ある特定のシステムのために作られたソフトウェア、たとえば、国鉄の「みどりの窓口」のためのソフトウェアなどは他では利用のしようがないので、販売してコストを回収するということとはできない。

「紙とエンピツ」でできるということから、小説などと似ている点もあろう。つまり、生産は人間によって行われ、自動化がむずかしく、生産性がきわめて低い。プログラムはそれを書く人の個性や、クセがある程度にじみでるが、小説のように文体を楽しんだり、内容の深さを味わうというような類いのものではない。むしろ

*都倉信樹 (Nobuki TOKURA), 大阪大学, 基礎工学部, 情報工学科, 教授, 工学博士, 情報工学

ろ、機械向けの記述という性質がつよく人間にとっては読みにくい。実は、これがソフトウェアの困難の一つの原因でもある。

さて、プログラムは計算機にやらせるべき手順の記述であるが、同じ作業をやらせるにも、そのプログラムの書き方は、原理的には無限通りの書き方がある。したがって、人によって書き方が違うのも当然で、考え方が違うとなかなか理解しにくい。ソフトウェア危機ということが言われたのは、相当前であるが、なんとかソフトウェアを技術的にしっかり作るということを目ざして、ソフトウェア工学という学問分野が生まれ研究がすすんだ。最近ではソフトウェアは、まず信頼性が高いこと、そして、読みやすいものであること、そして、そのあとに効率がよいことというように要求の順位が、昔とは変わってきた。そして、この目標をみたくべく、いろいろの研究がすすめられている。

建築物の設計コンペに対して、いろいろのデザインが提案されるであろうが、たとえば、市民ホールをつくると言えば、過去のプラン集などもあり、必要な設備や設計手順もある程度あり、先人の経験を生かして仕事をする事ができる。建築は穴居住居のころから（という逆のほりすぎかもしれないが）、人類史とともに永年の経験と知識の集積があり、それにもとづけば極端な失敗作はつくらないですむという状況である。それに比べ、ソフトウェアの方は最初の計算機が1945年頃であるから、まだ、たかだか40年の歴史にすぎず、まだまだ未経験なことも多い。しかし、基礎的な分野は相当短期間に内容も深まり、われわれの学科で講義している項目から拾っても、オートマトンと形式言語の理論、計算の理論、論理設計、情報理論、回路理論、論理学、グラフ理論や組み合わせ数学など、多岐にわたる。これらの基礎理論とともに、いくつかのプログラミング言語をつかいソフトウェア作成法についても、身につけることを目ざして教育している。

大規模な建築を日曜大工の手法でやるという無謀なことは誰も考えないが、大規模なソフトウェアの開発には、相当の素養がなければなら

ないことも理解されよう。ただまだ、その辺りの認識が低く、大学で FORTRAN でいくつかのプログラムを作ったということでもって、ソフトウェアはわかったと思う人もあるが、これは危険なことである。情報工学科は高度の専門的素養をもった技術者の養成と、そのための研究をつづけている。ただ、ソフトウェアは歴史の浅さだけでなく、非常にむずかしい要素をもっている。小説作法という本があっても、誰でもが小説が書けるわけではないように、人間の知的な作業の積み重ねであるソフトウェア作成も機械的にはいかない面が多々ある。これをなんとか克服するという事は誰しも思うことであるが、ある技術予測によると、セマンチクス（意味論）の扱いをできるようになるには、1世紀かかるであろうという。1970年代からいろいろのアプローチが一斉に花ひらいたが、理論的成果はあったものの実用化という点ではまだまだ困難が多く、1世紀という予測は別としても、これからも多くの努力が必要である。無形のつかまえにくい存在であり、評価もむずかしく、地味で世間一般の理解も乏しいという面がある。

今やマイクロコンピュータがなければ、ほとんどの工業はなりたないという状況であるが、マイクロコンピュータを使うにはソフトウェアがなくてはならず、ソフトウェアを軽視することは工業立国をめざす日本にとっては非常なマイナスになると思われる。ソフトウェアというのは、資源をもたない日本に向いており、日本人の特性にもあうと思われるが、アメリカにくらべておこなわれているといわれるのは、一つはソフトウェアの重要性の認識の低さ、したがって、力をそそがないということも一つの原因といえよう。もっとソフトウェアの研究に注力することが今後重要ではないかと思われる。紙数もつきたので、より詳しい議論はさけるが、われわれのグループとしては、これまでの研究をさらにすすめて、ソフトウェアの基礎となる種々の理論、プログラム言語、ソフトウェア・ツールの研究、マンマシン・インターフェイスの研究を継続していきたいと考えている。