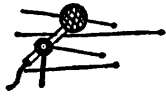


講演



研究・開発雑感†

—ソフトウェア工学によせて—

大 野 豊††

講演をはじめの前に、急逝された長友福田伸一氏の霊に慎しんで哀悼の意を表します。

1. はじめに

本総会での講演は本学会の会長、副会長の経験者が主にやるという余りよからぬしきたりがあるようで、私のところに順番がまわってきましたが、さし当りまとまったお話がありませんので、私の経験を通じた感想のようなことでその責をふさがせて頂きます。

私は長いこと国鉄の研究所にいて、国鉄内の情報システムの研究・開発にあれこれ関与してきたが、故福田伸一氏と同じ研究室で勉強し議論し、教えて頂いたことも多々あり氏の急逝は残念でならない。私は昭和47年に大学の教師としてそれまでとは全く異なった道を歩くことになり、以来、いささかの当惑と感想を持ちながら今日に至っている。情報学あるいは情報科学全般については、とてもお話する力がないので、数年前前から関心をもっているソフトウェア工学などにかこつけた若干の感想を述べることにする。

2. 情報学における分極について

昭和49年の本学会総会で清野武教授が“雑談の情報科学論”と題し講演されており[†]、当時私も大変興味深く拝聴させて頂いたが、その中で情報学におけるいくつかの面の分極化のお話があった。たとえば、情報の学問における“厳密性”ないしは“純粋性”を志向する動きと、“技術的”ないしは“実利的”方向への推進が見られ、“純粋数学的情報学”と“工学的情報学”との分極が存在し、その橋渡しは一般に極めて乏しいといわれた。このお話は清野先生がもともとは電磁気学の大家であられたので、その辺の連想から考えつかれたかどうか、は余分な推量であるが、この問題はいろいろな学問分野で常に新しい問題であるにちがいない。以前計測自動制御学会が徳島で大会を開催したと

き、最近の制御理論の実用性について討論会があり、私も呼ばれて一言申した記憶があるが、いって見ればこれは制御工学の方の分極の議論でもあったといえる。制御の方ではかつてのフィードバック理論が理論的にも実用的にも調和した完成の域に達し、新しく行われている理論が実用から離れて宙に浮いているという批判が多かったわけである。私が京大に移る決心をする前に清野先生にお会いしたとき、情報学では数学的情報学に偏る危険があるので、少なくともそうはしたくない、といわれたので、私でも大学でやっていけるかもしれない、と感じた次第である。

情報学ではどんな高度な理論を論じても、最終的には計算機というハードウェアに乗るプログラムとして有効に動かないと話にならない、という歯止めのような所があって、この点では、大規模システムの制御理論が大規模システムによって簡単には試みられない、という事情とは余程ちがっているように思える。

もっとも、情報学というのは基本的には計算機とは関わりないので、たまたま便利な道具として計算機を利用するにすぎない、という考え方があり、あるいはその通りかもしれないが、現実には情報学の基本的な核の1つに計算機科学があって、その対象はもともとはハードウェアの計算機である。そこには基本的原理も基礎理論もあり、学問としては比較的たしかな足場があるように見える。

しかしながら、ソフトウェアとなると急にその足場が宙を行く感があり、そのことから情報学全般が何となく頼りないものになってくる。上記の基本的核の外に核と称すべきものは何か、ということになると誰も明確には答えてくれない。

よくいわれることであるが、情報学の基本原理は何か、その基礎理論と称すべきものがあるか、これに答えることは大変むずかしい。そして、この点では、制御工学がフィードバック制御理論を結局は基本としていることにくらべると、極めて容易ならざる問題を含んでいるといえる。また一方、情報学がこれから体系

† 情報処理学会第20回通常総会特別講演(昭和54年5月17日)

†† 京都大学工学部

化されるべき若い学問であるしるしともいえよう。

3. 情報システム開発での経験

先述の通り、国鉄在職時代に主に専用オンラインシステムの研究・開発に関わっていたが、今から考えると冒険的な技術でやらざるを得なかったことがしばしばで、たとえば、穂坂先生と一しょにやらせて頂いた座席予約システムでは、そのための計算機的设计も同時に行ったという、今日では余りすすめられない開発活動が必要でもあったが、大勢の関係者の大変な努力で大成功であったので、結果としてはよかったということになる。しかし、このように巧くいった例ばかりでなく、私が関係したシステムでも日の目を見なかったものがあり、むしろ当然のことながら大きな教訓であったと考えている。

その巧く行かなかったシステムの私にとって最も印象的なのは、貨物情報処理システムである。これは最初の座席予約システムが成功したのに勢をえて、以前から発想されていた貨物輸送に関する情報を扱うシステムを開発しようと、国鉄本社の電気局や貨物営業局と検討をすすめたわけである。当時、丁度米国でも同様の動きがあり、米国の諸鉄道で貨物輸送の情報処理を行うオンラインシステムを開発し、運用をはじめている所もあった。一方、わが国鉄内では、本社の事務管理統計部で貨物の諸統計のバッチ処理を以前から行っており、貨物情報処理はその方向からの機能拡張で進めて行くべきである、という提案もなされていた。そこでオンライン派とバッチ派とのかなり長い間の論争があり、本社の各方面の調整もあったが結論は出ずに日時は流れた。一般にバッチ処理を中心にするてきた人たちは、座席予約システムや米国諸鉄道のシステムの成功を目にしているにも関わらず、オンラインシステムを困難視、危険視しており、一方、私などオンライン派は今から思えばかなり冒険的な技術もとに主張していた所もあった。たとえば、単に大容量ファイルをもったオンラインシステムではなく、今日でいえば分散処理、計算機ネットワークさらには貨車番号の自動読み取りなどの技術を必要とするものであり、長い研究の積み重ねはあったものの、実用上未経験のものが多くあった。これらの論争を通じて最も重要なことは、このシステムが出来上ったとして果してどれ程の利益あるいは増収が見込まれるか、という当時各所のシステムで問題であったことに満足のいくような予測が十分でできなかったことである。私共は米国

の諸鉄道のシステムの調査を現地において行い、これらについての解答を引き出そうとしたのであるが、米
国鉄道とわが国鉄の環境事情の相違は抜き難く、思
うような解答は出せなかった。結局、私共の主張する方
向での開発は延期され、特殊な貨車（地域間急行貨
車）の予約という極めて限られた機能の小さい規模の
オンラインシステムを開発し、さらにバッチ処理も強
化して行くという方向が決定された。このような決定
が本社で行われたことは、技術的に見れば保守的で安
全側のものであり、研究者にとっては物足りないもの
であったが、本社の現実組織を運営し責任を負う立
場からは当然の決定であったと、私自身納得したが、
国鉄が貨物輸送サービスをして行く限り、私共の考
えたようなシステムが必要であろうという気持は今
でも変りはない。

もう一つの例は研究成果が巧く実用システムに組み
込まれた新幹線の運転管理システム COMTRAC に
ついてである。このシステムの研究には私自身鉄研で
の最後の数年間に研究プロジェクトを担当したので
あるが²⁾、研究から開発実施への道程はやはりかなり
けわしいものであった。このシステムでも研究と開発
との間のギャップを埋めるための多くの論争と、その裏
打ちをするための研究とがあった。このシステムは主
機能として列車の動きを追跡し、信号および進路を自
動制御する機能（計算機制御によるいわゆる CTC）
と、列車の運行が乱れたとき、それを回復するための
運転整理の機能とが必要とされるが、前者は従来の研
究成果が余り問題なく開発に移される見通しはあった
が、後者については多くの問題があった。当時、グラ
フィックディスプレイがやっと実用しうる段階とな
り、研究室では新幹線の運行乱れの回復を行うための
意志決定活動を、これによってマンマシン方式で行
うという考えを打出し、研究をすすめていたので、是非
これを実施に移し度いという強い要望をもっていた。
その実用性の可否をめぐって、当時の新幹線総局長を
はじめ、電気局、運転局の関係者と多くの議論をし、
また、研究所ではグラフィックディスプレイを設置し
て、列車運行のシミュレーションを行い、運行の乱れを
整理する運転ダイヤをディスプレイ上に表示し、マン
マシンのやりとりでこれを手直して運転整理ダイヤ
を決定して行く実験を行って、その過程のデモンスト
レーションも行った。このような努力にも関わらず、シ
ミュレーションというものの弱さから、皆が容易に納
得してくれるところまでは行かなかったが、本社関係

者の中に新技術を推進することに理解をもった方がおられ、とに角開発実施にこぎつけた。私自身はこのシステムの開発直後、大学に移ったが、あれだけ論争したグラフィックディスプレイの効果について、今でも関心をもってその後のなり行きに注目している。最初の実システムでは、当初の分析や設計の不備もあって、必ずしも満足すべき機能を発揮しなかったようであるが、新幹線が博多まで延長された時にシステムは更新され、問題の運転整理機能も大幅に改善されたときいている。これには故福田氏がプロジェクトの担当者として大変苦労されたことも伺っている。私は今でもシステムに余分な技術をおしつけたとは思っていないが、研究と開発の関係のむつかしき一面を、そして実用ということのきびしさを見る思いがしている。

この二つのシステムの明暗の差は、システム開発に先立つ研究や分析もさることながら、もっと大きいのは、国鉄における貨物輸送と新幹線の現状と将来への展望に依存していたことに疑いを入れない。新幹線の展望では、すべてが前向きに考えられ、したがって新しい技術もある程度思いきりよく試みる、という姿勢となつてあらわれたというべきで、技術の進歩はこのような積極性が必要であり、数学で表わせる利益に還元してはじめて進められる、というものでは必ずしもないわけである。

当時、本社の人たちからよくいわれたことは、鉄研の研究はすぐ開発・実用に結びつくので、研究者としてこんな幸福なことはないではないか、というのであるが、そう簡単に研究と開発・実用と結びつくわけではない。当然のことながら、見通しのよい研究をしなければならず、かといって、すぐ役立つということで、“たわし”のようなものの研究で満足するわけにも行かない。先進的な技術の研究は、ものにもよるが、一朝一夕には行かず、長い目立たぬ研究が必要であり、しかもなお、実用になるかどうかの判断は、そのときの環境条件に大きく依存する。

なお、これらに関連して、特にお話しておきたいことは、応用情報システムの研究・開発をやっていると、ある意味でその応用分野に没入して仕事することになり、はじめはかなり先進的な技術を意識してやっても、開発は既知の技術ですすめるものなので、数年のシステム作りに専念した後では、世の中の技術が思いがけず大きく進展していることに気づくことである。私自身はシステム作り直後、大学に移って見て、純粹に研究という立場からあれこれ勉強し見直

して、その感を深くした。特に、このような大規模システム開発の困難さを体験して、大規模システム開発の方法論に何か良い考えはないか、と思いつつなす所なく過しているうち、構造的プログラミングとかソフトウェア工学とかの考え方が打ち出されはじめたのを知って、その感を深くした。そこで、これらについては私自身のかねてから関心のある方向とも合うため、昭和48年頃から、勉強しながら大学院でポツポツと講義もはじめたわけであるが、これに関連したことを後で述べてみたい。

4. TSS とネットワークについて

計算機関係では、わが国でとくにおくれているのはソフトウェアであるといわれているが、これはわが国のTSSの実用と重要なかわりがあるように思える。というのは、MITでMACをはじめたときのMcCarthyの理念の中に、TSSで発展を見込める問題の1つにソフトウェアがあることが指摘されているにもかかわらず、わが国のソフトウェア技術の一般水準ではそれについての理解が充分でなかったように見えるし、それだからこそ一方では、国産のよいTSSを供給できなかったからである。

わが国でTSSが何故発展しなかったか、その原因はいろいろ考えられる。日本人はタイプライタのような端末操作は一般には馴れていない、というのもその1つかもしれないが、物まねの好きな国民が米国でのTSSの発展を見ながらも、なおかつ、プログラミングはバッチの方がやり易い、あるいはそれで充分である、という考えが根づきかかった。さらに、国産のTSSによいものがなかった。ということと、効率の好きな日本人にはTSSのシステムとしての効率の悪さが好まれなかった。これは、経済と効率の一本槍で進んできた高度成長時代には止むをえなかったことかもしれないが、そのことが、MITの打ち出したマンマシンによる問題解決の重要さや、バッチ処理の人間をふくめた総合的な効率の悪さを充分に認識させなかったともいえよう。そして、メーカーは当然のことながら需要のないことに力を入れてくれないわけである。

私は三年半程前に、一般学生を対象とした情報処理教育センタを計画する立場におかれ、教育方法、学内事情、要員・予算事情、また当然のことながら、計算機技術の動向などを考慮して、教育センタはTSSによる以外にない、という計画をたてたが、当時、学内の計算機利用者としての教官からTSS反対の意見が多

く寄せられた。また、京大の大型計算機センタの関係者からも同じような意見が聞えてきた。それは何故かといえば、それまで京大の TSS が機能、性能ともに、余りにも問題点があり、多くの人たちに TSS 否定の気持ちを植えつけたわけである。そして、たしかに、当時は端末装置を 60 台以上同時に操作する TSS は、国内メーカではどこも経験がないし、大学でもわずか筑波大学でそれに近いシステムが使われているにすぎなかった。

ところが、京大の若い教官で米国に留学し、向うの計算機を使って帰ってきた人たちや、少数の洞察のある教官が TSS を支持し、激励してくれたわけである。TSS の考えが出されたのが今から 20 年近くも前で、米国でそれが使えるようになったのは、15 年以上も前である。今さらわが国の大学で TSS 是非の議論をしなければならぬのは、何とも淋しい限りである。かつて、ある大学の先生が、“TSS というのは利用者は自宅に端末装置を置いて自分でプログラムを打ち込む、などという馬鹿げたことを考えているようで、……”といわれた。このような認識の中にこそおくれの本当の原因があったといえるかもしれないが、ここ数年来の TSS に対する若い層の需要の高まりを抑えていたのは、良い TSS を供給しえなかった国産のソフトウェア技術力であったことは否定しえないことである。そして、このことがわが国のソフトウェアの技術力の発展をますますおくらせていたことも指摘しておきたいことである。

ところで、最近わが国の TSS の状況に急速な変動が見られるようになった。先述の京大の情報処理教育センタが全面的な TSS により実現したが、これより前すでに九州大学その他でもやっておられ、成果を上げている。京大の大型計算機センタでも TSS が改善されて以来その需要が急速に伸びはじめ、次の計画では大幅に TSS の拡大を計画していると聞いている。これらは多くの反対のあった三年前には考えられない変りようであり、京大の情報処理教育センタでは 60 台の端末に学生が毎日溢れるように取りついている。これも、良い TSS (まだ多くの問題点がなくはないが) がメーカにより供給されはじめたからに外ならない。

先述の通り、TSS というのはソフトウェア開発のための環境として極めて有用である、といわれている。プログラミングという問題解決をオンラインで行うことの有効性を熟知して、米国ではすでに何年も

この技術の積み重ねをし、それぞれの TSS 上に多くのソフトウェアを蓄積してきた。そういう蓄積されたリソースをもった各システムを互いに他と結合して、リソースの共用をしようということになって、ARPA などの計算機ネットワークができた。わが国でも大企業や企業グループで専用のネットワークを作り活用している所はあるが、汎用のものはまだない。数年前から大学間計算機ネットワークを開発すべく N-1 ネットワークの開発グループがあり、東大-京大間は一応限定された形ではあるが実現している。しかし、全国的ネットワークはまだかなり時間がかかりそうである。

ある人は“ネットワークを作っても共用すべきリソース、特にソフトウェアリソースがあるのか”といっているが、正にこれは的を射た言葉である。わが国の各大学のシステムが TSS として充分機能を発揮し、その結果として、共用すべきリソースが蓄積されたとき、はじめてネットワークが意味をもってくる。米国で ARPA ネットワークの仕事がはじまったのが 11 年以上も前であり、わが国でははじまったのが 5 年前で、そのおくれは他分野に比して少ないといえるが、不幸にして、ノードとなるべき各計算機システムで、その機能の整備が充分できていない事情は、折角のネットワークの開発・実用をおくらせてしまうと考えられる。

主要な大学の計算センタが、従来、バッチ処理による効率のみに力をそそいできた同情すべき理由はあるにしても、たて前論や表向き議論が先ばり、実情が伴わないわが国の大学のあり方は、大いに反省しなければならない。

5. ソフトウェア工学について

ソフトウェア工学での期待とおそれ わが国でもここ二、三年来、ソフトウェア工学について各方面で論じられ、研究されるようになったのは大変よろこばしいことである。NATO の後援で最初の会議が行われてからおよそ 10 年であるが、わが国では今までメーカ側での研究・開発は相当な努力で行われていたのに、大学などでの研究はソフトウェアの限られた分野以外、余り行われていなかった、といわれていることも、これからは改善されるのではないかと期待される。そして、TSS やネットワークなどソフトウェアのための環境条件も先程述べたように整備されつつあり、すでにデータベースとその利用面では相当な進展が見られ

ている。

しかしながら、一方では期待に反するおそれもある。たとえば、“ソフトウェアは重要であり興味もないわけではないが、研究しようとは思わない”、“ソフトウェアの研究では研究費はとれない”、“プログラミングはしんどい仕事なので、研究としても近づき度くない”。など、多くのいいわけをよく耳にする。どうやらプログラミングをそれ程高級な仕事と見ていな

い、余り表に現われない理由があるようである。

さらにつけ加えておかなければならないのは、他の学問分野でもありがちなことで、ソフトウェアの研究が理論に埋没して、実用面への反映を意に介さない、あるいは意識的にさける、という傾向がありはしないか。これは大学での研究にはすでに多くの前歴があるので、そのおそれは大いにある。

ソフトウェア工学研究動向調査 わが国のソフト

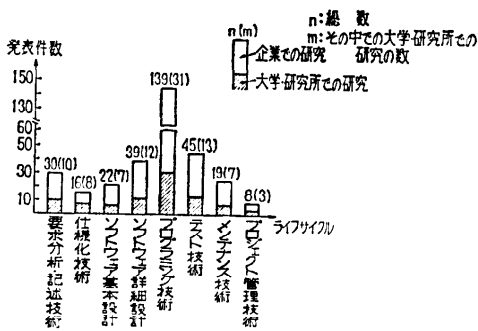


図-1a ライフサイクルによる分類—国外

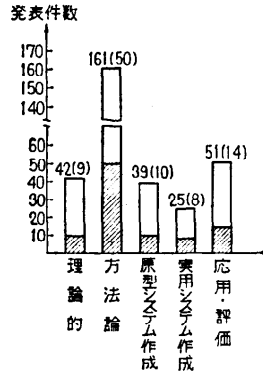


図-1b 研究対象範囲による分類—国外

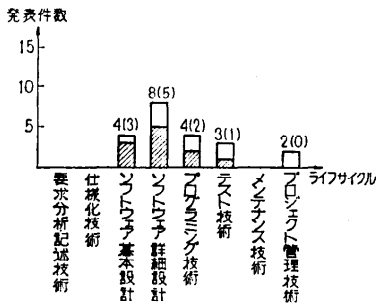


図-1c ライフサイクルによる分類—国内(論文・研究会資料)

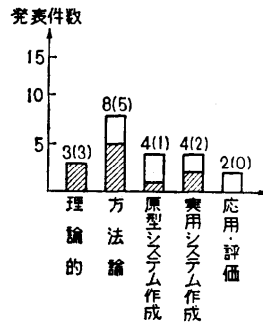


図-1d 研究対象範囲による分類—国内(論文・研究会資料)

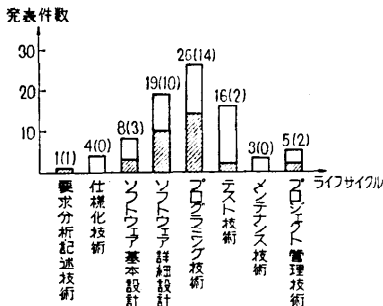


図-1e ライフサイクルによる分類—情報処理学会全国大会

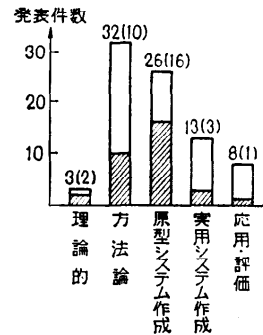


図-1f 研究対象範囲による分類—情報処理学会全国大会

図-1 ソフトウェア工学の研究動向

表-1 ソフトウェア工学研究動向調査対象資料
(1977~1978年分)

国外	CACM, JACM, IEEE Trans on S/E, COMPSAC, IBM System Journal, IBM Journal R & D, Software-P&E, SIG-PLAN, DATAMATIOM, 3rd UJCC Proc.
国内	情報処理, 電子通信学会論文誌, 情報処理学会ソフトウェア工学研究会資料, 企業報告誌(日電, 日立, 富士通, 東芝, 三菱)

ウェア工学の研究と米国などの外国のそれとを、傾向がどう異なるかを見るために、最近2年間だけに限って調べて見た。調査対象は表-1に示すような雑誌報告類からソフトウェア工学に関するものを拾った。簡単な調査で対象範囲も狭いので、これだけで正確に動向をつかめたとは考えていないが、参考にはなるであろう。調査としては、発表された研究がソフトウェアのライフサイクルのどの段階に主として対応するのか、その研究が理論、方法論、プロトタイプ、実システム、応用などのどれに当るのか、を計数した。

この調査で気付いたことは、ライフサイクルの各段階に対する考え方が論文によって異なり、また、どの段階のことを研究しているのか、あいまいなものがあったため、論文の表題と本調査の計数場所とがくいちがっているものが、特にわが国のものに多かったことである。結果は図-1-a~fに示す。

これらの結果を見て、まずわが国の研究が正式に論文として発表されたものが、いかにも少ない、ということ、ライフサイクルの初期の要求や仕様の段階に関する研術がほとんどないことである。そこで、本学会の全国大会での発表中、ソフトウェア工学的研究がどうなっているかについても調べた。そうすると、その数は適当な多さで分布しているのである。このように、わが国のソフトウェア工学の研究はかなり行われているが、正式の論文としては発表のされ方が少ない、という理由は何か。考えられるのは、研究内容が充分高度でない、単に外国でやったことのまねである、良い仕事であっても論文にはならない、などが考えられる。これは、ソフトウェア工学が提唱されて十年たった今も、わが国では本格的な研究が充分には始まっていない、といえよう。

要求定義と仕様 わが国でライフサイクル初期の段階の研究がほとんどないのは、要求定義や仕様についての理解が充分でないからではないか。Ramamoorthy等は、各段階の結果は次の段階の要求になっている、したがって、要求定義は各段階に対応した各レベルのものがある。としているが³⁾、それはそれで正しいとしても、そのように要求定義を考えた所で、研

究の上ではむしろ混乱をまねくだけであろう。要求というものはライフサイクルの初期のものに限定した方が議論は明解となる。わが国の研究論文の中で、たとえば仕様とっているものが、機能仕様なのか、設計仕様なのか、ライフサイクルのどの段階の仕様なのかあいまいなことが多い。これは一方ではライフサイクルを明確にしていけない所からくることかもしれないし、そのための議論も少ない。

プログラム言語に関係しているある研究者と話をしたとき、米国で提案されている要求定義技術は全く役に立たないのではないか、たとえば、ISDOSのPSLなどでは、普通いっているソフトウェアの要求定義はできないのではないか、といていたが、一般にわが国では要求定義は自然言語でもっと多様なことを記述している、と考えているようである。要求定義はもともと、システムをブラックボックスと考え、そのシステムと現実社会とのインタフェース条件を記述したものである。現在提案されている要求定義言語は、その記述を出来る限り形式化して、一貫性、追従性、完全性などの検証を計算機によってなるべく行うようにしようとしている。したがって、提案されているどの言語でも現実社会の無限に近い多様性に対しては極めて限られたものにはちがいないが、形式化すべき関係は形式化してその範囲で可能な検証を行い、その意味は自然言語でコメントを補うというやり方で対応しているようである。これはこれで納得できる手法であり、役に立つかどうかはその考え方を理解して如何に利用するかである。

この分野はプログラム設計やプログラミングなどのように限られた世界の形式化された後の取扱いとは考え方に大きなちがいがあり、細い議論を行う“完全主義”的な研究者には一部の否定的な面ばかり見えて、敬遠されているように思える。

基本原理と文学論について 先述の清野先生の論考の中にもあったが、アートかサイエンスかという議論がよくいわれるが、私共がほん訳した E. Yourdonの著書の序文には、アナリストやプログラマがコンピュータの内部のことをよく判っていないのは、コンピュータの分野が科学であるという誤った印象をもっているからで、クラフトであると考えの方がより適切である。という主旨のことが書かれている⁴⁾。これは、たしかに実際面ではその通りで、私自身も今でも一面でその信念をもっているが、一方これによしとしては余り進歩は望めないと考える。ソフトウェア工学が、ソ

ソフトウェアの問題に科学的知識や方法を使って、ソフトウェアの開発や維持に何か良い結果をもたらさし度、という願望をもって提唱されはじめたのは、そこに理由がある。本当に巧く行く保証は必ずしもないが、そのような考えが出てきたきっかけの中には、構造的プログラミングやその他の若干の研究成果が指摘できよう。とはいえ工学というからには、そこに基本となる原理があって、その上に組立てられるものであるべきだが、そこが必ずしも明確でない。

D. T. Ross 等は、ソフトウェア工学の基本原則は、モジュール性、抽象化、局所化、隠ぺい、一様性、完全性、確認性の七項目であるとしている⁴⁾。これら個別について本稿で説明はしないが、現在の所これは不易の原理であるより、むしろ最近の研究で判ってきた当面の指導原理といった所ではないかと考えている。これに関連して、プログラムの理論をやっている研究者から、ソフトウェア工学では、プログラムをいかに (how) 作るかの前に、プログラムの本質は何か (what) の研究がなければならない、といった主旨のことをいわれたが、たしかに正論であるにちがいない。しかしながら、他の分野の工学でも似た事情はあるが、われわれはプログラムの本質に関する理論ができ上がるまで、何もしないよといふわけにはまいらない。工学としては判らないながらも how をやらざるを得ないことになり、アートといわれる所以となるが、多少なりと利用できる理論があったら、そこから方法論や手法を展開することは、工学としてはこう定まるものである。

ソフトウェアは、ある言語によって何かを表現するもの、であるから、文学での文章論や小説作法といった類のソフトウェア技術論や理念論があってあたり前で、それらはむしろアートという言葉よりソフトウェア文学といった方がよいのではないか。このような文学論はたしかに必要である面で有用だが、ソフトウェア屋がおち入りやすい趣味の文学論ではこまるのである。少なくともソフトウェア工学が目指しているのは文学ではなくて、超高層ビル程ではなくても大規模なソフトウェア開発の技術と方法論を確立したい、ということで、小さい趣味的な茶室を作ることではないであろう。

“役に立つ”について 先に実用の場での役に立つことのきびしさを述べたが、役に立つとは何なのか。実用して役に立ったのか、誰かが実用すれば役に立ちうるのか、実用で役に立つ積りで作ったのか、等

等といろいろな意味がある。

ソフトウェア工学で行われている研究に対し、実際面の方々から、役に立つという基準から評価する事がよく聞えてくるが、かつての筆者も今も基本的には同調している。しかしながら、早急に役に立つという評価をすることには、必ずしも賛成しない。長い間のいくつかの研究の積重ねと、それを実用するための十分な検討や準備作業が本当の実用に結びつく。後者は実用ユーザの側の責任が大きい。情報学やソフトウェア工学では、実用の検討にも値しない多くの研究があるにせよ、実用ユーザ側の不勉強が実用とまでは行かなくても、役に立つ評価を妨げていることも多い。ソフトウェア工学という新しい分野は、すぐに役立つこともさることながら、まだ多くの冒険的研究を必要としているのではないであろうか。

6. む す び

最後に結論として、わが国のソフトウェアをもっと発展させるためにはどうしたらよいか、言及しなければならないであろう。計算機のハードウェアが国の大プロジェクトで、兎に角、米国に近づくことができたことを思えば、ソフトウェアも同様の手が使える、といえるかもしれない。すでに通産省などではいろいろな手を打ち出しているようである。しかしながら、ハードウェアと異なり、わが国のソフトウェアの基礎となるレベルはむしろ低く、余り即効をねらうプロジェクトの効果は期待できない。今のところ良い方法の見通しは余りはっきりしないが、比較的効果のありそうなのは、ソフトウェアの基礎的なことから即効的なことまで、幅広いスペクトルをもった、確固とした目標をもったプロジェクトを推進することではなかろうか。これによって、特に大学におけるソフトウェアのレベルを上げ、従って教育も向上し、数年後には実際面への効果も期待しようというわけで、比較的長期の展望が必要であるように思われる。

以上、勝手な余り脈らくもなく、また“役に立ちそうもない”雑談に、各位が貴重な時間をさいて下さいましたことを深く感謝いたします。

参 考 文 献

- 1) 清野：雑談的情報科学論，情報処理，15，7，497 (1974.7).
- 2) Ohno, Y.: Le' systeme COMTRAC du Shin-kansen, Automatisme. (Fev. 1973).
- 3) Ramamoorthy, C. V. and So, H. H.: Software

Requirements and Specification: Status and Perspective", (Aug. 1977).

宮本訳: ソフトウェア要求と仕様技術, dit 臨時増刊, Vol. 10, No. 10, pp. 1103~1159 (1978).

4) Yourdon, E.: Design of On-line Computer

Systems, Prentice-Hall, Inc. Englewood Cliffs, New Jersey, U.S.A (1972).

5) Ross, D. T., Goodenough, J. B. and Irvine, C. A.: Software Engineering Process, Principles, and Goals, Computer, pp. 77-27 (May 1975).