

解 説



大学等における情報処理教育の諸問題†

—平成元年度の調査研究を中心として—

野 口 正 一†† 中 森 眞理雄†††

はじめに

我が国の将来にとって最も重要な政策は、高度の科学・技術を確立し、21世紀に向けて確固たる高度産業の基盤を構築することである。

事実、産業界では、技術革新の急速な展開に対し、ハイテクノロジーの分野で常に他に先駆けて世界をリードする状況を築いてきた。しかしながら、このことは、ハイテクノロジー産業を支える最も重要な基礎分野、特に情報科学・技術の分野で我が国が世界をリードしていることを意味しているわけではない。

21世紀が高度の情報化社会であり、情報処理の広い研究・開発の諸分野において、今後我が国が世界の先駆的な地位を占めないかぎり、我が国の将来はきわめてきびしいものとなろう。特に深刻な問題は、我が国が目指すべき知的所有権による立国を支える人材がこの分野できわめて不足していることである。そのような人材を養成・確保することこそが我が国の科学・技術・教育政策の中心課題であろう。

このような状況のもとで、我が国では文部省の政策により多くの大学などにおいて、情報系専門学科が数多く設置されるようになってきた。事実、文部省の統計によれば図-1に示すように学生数の増加も著しい。しかしながら情報系専門学科の教育を担当する真の専門家の数はきわめて少なく、またカリキュラムを初めとする今後の教育のための根本の問題について、多くの重要な問題がほとんど未解決のまま残されているのが実状である。

今こそ、今後の日本の将来のために最も重要な情報教育の問題を根本から考えるときではないであろう

か。このため文部省では、我が国の今後の一層の発展を支えるため、長期的展望の上に立つ新しい情報処理教育の理念のもとに具体的な教育内容について検討することを目的とした「情報処理教育の改善のための調査研究」を1989年から二年間にわたって情報処理学会、及び文部省科学研究費補助金の総合研究(A)によって行うことを決定した。研究題目は「大学等における情報処理教育のための調査研究」である。これに先立って、1988年度に本学会は文部省から同じ題目で研究委託を受けており、コンピュータと教育研究会(有山正孝主査)内に「情報処理教育の改善のための委員会」を設けて調査研究を行った^{1),2)}。そこで得られた知見は前記調査研究に継承されている。

1989年度からは「大学等における情報処理教育検討委員会」を設けて調査研究を行っている。当委員会のメンバは、情報処理学会の理事会より5名、大学より17名、企業より10名、合計32名の委員をもって構成され、また科学研究費によるメンバはこの中の大学の

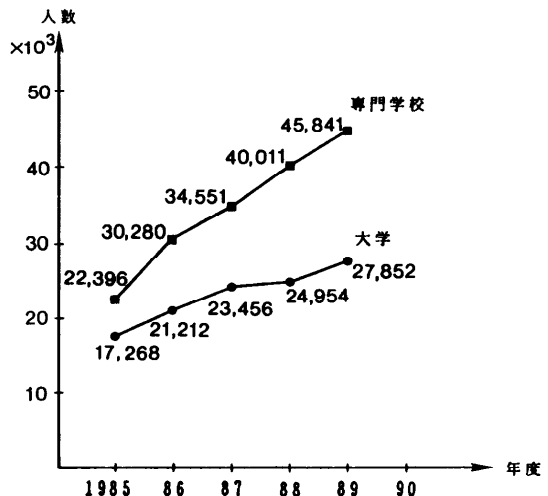


図-1 情報系入学者数の変化(文部省調査)

† Problems in the Curricula for the BA Degree in Computer and Information Sciences in Japan by Shoichi NOGUCHI (Research Center of Applied Information Science, Tohoku University) and Mario NAKAMORI (Department of Computer Science, Tokyo University of Agriculture and Technology).

†† 東北大学応用情報学研究センター

††† 東京農工大学工学部

委員をもって構成されている。本稿の著者の一人は当委員会の長の職を務め、もう一人はCSワーキンググループ(後述)の幹事を務めるとともに報告書のとりまとめなどの諸事務を担当した。

当委員会は上記のテーマについて精力的に討議を行い、その初年度分の成果を中間報告としてまとめた³⁾。

同中間報告書は、コンピュータサイエンス(CS)の教育を中心とした第一部と、インフォメーションシステム(IS)の教育を中心とした第二部からなっている(CS, ISの定義については1.2参照)。第一部はCSワーキンググループ(牛島和夫主査)の、第二部はISワーキンググループ(國井利榮、藤野喜一両主査)の努力に負うところが大きい。なお、情報処理教育には広範囲の分野の学生に対して行う一般情報処理教育もあるが、この問題は次年度に新たに発足した一般ワーキンググループ(大岩元主査)の成果を待つことになっており、中間報告書では論じられていない。

この中間報告書は次年度の最終報告書に到る中間に位置するものであり、その内容については、まだ多くの討議すべき点が残されているが、この報告書の内容に則り、筆者らの考えに従ってまとめたのが本稿である。本稿をまとめるにあたり、当委員会の一致した意見を得るに到っていない問題点についても代表的で重要と思われる意見をいくつか併記するという方法をとることにした。これにより、当委員会の雰囲気や委員たちの熱意が伝わるものと期待している。なお、当委員会における議論の詳細や委員以外の識者たちによる種々の提言は本誌に今後約1年間にわたって連載形式でとりあげられることになっている。

本稿は四つの章(1. 2. 3. 4.)からなる。1章では情報系専門学科のカリキュラムを定めるための基本的な考え方と現状の問題点を述べる。2章では情報系専門学科のカリキュラムのあり方を、特にコンピュータサイエンスを中心に論ずる。ついでに今後の大学における情報系専門教育のための制度的問題について述べる。3章では2章で述べたCSの具体的なカリキュラムについて詳細に論ずる。4章では情報系専門学科を特徴づける実験と演習について論ずることとする。

1. 情報処理教育の問題点

本章では、今後我が国が世界をリードできる体制を作るために情報のあらゆる分野における新しい情報処理教育を考えなければならないことと、情報処理教育の基本分野としてCS, IS, ISEがあることを述べ、

我が国の大学などにおける情報処理教育の問題点をあげる。

1.1 高度情報化と情報処理

今後の情報処理の進展を考えたとき、次のことが結論される。すなわち第一が情報処理の基幹となるコンピュータサイエンスの一層の基礎研究の発展、第二が広範囲にわたるコンピュータを中心とする巨大システム構築のための開発技術の発展である。

コンピュータサイエンスの分野では特に従来の形式的情報を中心とした処理から、新しい知的情報処理に向けて、今後多くの重要な研究が行われ、その成果は情報処理技術の多くの分野に適用されることになるであろう。

一方、巨大システムの構築技術はあらゆる分野の技術のインテグレーションから構成され、この分野で要求される技術者の数は今後指数関数的に増加することが予想される。このことは過去から現在に到るこの分野の進展の状況をみると分かりやすい。すなわちコンピュータは当初、学術研究環境及び各企業の財務、経理などのビジネス環境を中心として発展してきたが、高度通信システムの急速な発展にともない、バンキングシステムやVANに代表される巨大なコンピュータネットワークシステムが出現するようになった。このようなネットワークシステムはさらに経営戦略の上で必要なSISや多くの社会活動を支援する巨大なシステムとして発展していくことになる。このように社会全体にきわめて多くの影響をもつ巨大システムが生まれつつあるが、このようなシステムを構築するためには、システムが処理する全ての分野の知識が必要となることは言うまでもない。たとえば財務管理システムを考えれば、コンピュータに関する知識はもちろんのこと、財務、金融、経理、意志決定などの数多くの専門分野についての深い知識が必要となってくる。そしてこのようなシステムを開発し、設計、構築するための技術体系は現在では未完成の段階にある。

いずれの分野においても、我が国が、知的産物(学問上の概念、ソフトウェア製品、など)の輸入超過という状態を改め、輸出する側に立てるように発想を転換することが望まれる。これらの分野は、我が国の高い国民性と相俟って、我が国が世界に貢献する可能性の大きい分野である。ただし、それには、基礎的研究に従事する人材を多数養成することが不可欠である。

以上のような状況を十分に認識した上で、われわれは、情報の基礎分野から広く応用の分野において今後

我が国が世界をリードできる体制を作ること为目标とした新しい情報処理教育の問題を考えなければならない。

1.2 情報処理教育において対象となる基本分野

前節で述べたように、今後の情報処理展開の状況から判断したとき、情報処理教育として次の三つの基本分野を考えることが重要である（中間報告書第二部参照）。

第一は、将来を展望したコンピュータサイエンスの分野であり、この上における新しい中心的な情報処理教育である。

第二は、巨大システム構築の分野であり、このための体系的なシステム構築技術を中心とする情報処理教育である。

第三は、情報処理システムをベースとし、この上に具体的な利用システムを構築し、高度な利用を図る分野であり、このための情報処理教育である。

次に各分野の内容を今少し具体的に示し、その相互関係を示そう。

第一の分野であるコンピュータサイエンス/エンジニアリング (CS) 分野では、次の世代のための新しい知的情報処理、ソフトウェア、コンピュータシステム、アーキテクチャを目指し、研究開発が行われる。

第二の分野であるインフォメーションシステム (IS) の分野では、コンピュータシステム、コミュニケーションシステムを融合し、大規模な情報処理システムを構築するための研究開発が行われる。また、この分野では情報処理システムそのものの開発とともに、目的とする利用者の要求に十分満足できるシステム構築の技術開発も重要である。

第三の分野であるインフォメーションサイエンス/エンジニアリング (ISE) の分野ではすでに与えられている情報システムを活用し、具体的に利用者が要求している問題を十分に解決できる総合情報システムの開発の技術が中心となる。

第二、第三の分野はシステムエンジニアリング (SE) として定義づけられ、第二の分野がシステム開発の上流工程とみられている。一方、第三の分野の研究開発のためには、ただ単に従来の情報工学の分野の知識だけでなく、システム構築が対象としている専門分野の知識、たとえば OA システムの構築には財務、金融などの深い知識が必要となる。特にこの分野での研究開発者のニーズは今後急速に増加することが予想される。以上述べた CS, IS, ISE の三つについて、それぞれの分野の包含関係を、深さには着目しないで示

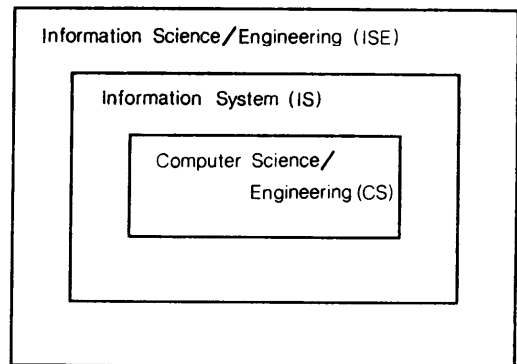


図-2 ISE, IS, CS の包含関係

せば、図-2 のようになる。この図からも分かるように、情報処理教育の根幹として最低限度の CS は (IS, ISE に従事する者にとっても) 不可欠である。

なお、ISE を、コンピュータの広範囲な分野への応用を念頭においてヒューマンマシンインタフェースや人工知能の観点から捉える考え方を主張する委員もいた。この考え方によれば、CS, IS, ISE の関係は図-2 のような包含関係とはならない。

1.3 情報系専門学科の現状と問題点

本節では、我が国の情報系専門学科における問題の所在を中間報告書第一部第3章に即して述べる。

1.3.1 教育内容について

教育内容について、当委員会で指摘された最大の問題点は、我が国では情報系専門学科とは何かという点について、共通の理解が教員の間になく点である。これは大学の学科組織についての我が国固有の伝統的な考え方にある原因があるが、“computer science” の訳語である情報科学/工学が拡大解釈されていることも無視できない。

コンピュータサイエンスという学問が存在することは、日本では当事者以外にはほとんど理解されていないが、20年前に米国で ACM CURRICULUM '68 (3.1 参照) が発表されているように、大学教育として確立した、成熟した内容をもっている。学問自体の定義は当事者によりニュアンスがやや異なるが、大学教育としての内容は大筋においてかなり一般的なものであり、その本質は次のとおりである。

“問題を論理的な対象として定式化し、それを解決する論理システムを構築する能力のある人材を教育すること”。

なお、文献 4) の定義も参照されたい。

コンピュータサイエンスという言葉には「計算機を対象とする」という意味が明確に定まっているが、情報科学/工学という言葉には多様な解釈が可能である。大別すると、計算機それ自体を研究・教育の対象とする立場と、計算機を道具として使って情報処理の応用を行う立場とがある。

情報系専門学科を日米で比較すると、日本には後者の立場のものが圧倒的に多いことが特徴である。後者の学科の教育における最大の問題点は、コンピュータサイエンスの教育・研究の含まれ方が少ないことである。そこでは、必然的に、コンピュータサイエンスのフィロソフィの核をなすアルゴリズムやアーキテクチャなどの基本的な概念を教えることが少なく、計算機の操作方法の教育が中心となる。たとえば、コンパイラの作り方を教える代わりにプログラム言語の文法を教え、オペレーティングシステムの設計方法を教える代わりにコマンドの操作方法を教えても、コンピュータサイエンスのフィロソフィを教えたことにはならない。自動車の運転を教えても機械工学を教えたことにはならないのと同じである。操作方法中心の教育の問題点は明らかであろう。

一方、卒業生を受け入れる産業界では、計算機を使える可能性のある者なら（専門能力は問わず）だれでも採用してソフトウェア生産の現場に投入せざるをえないという現実がある。しかし、この結果は、能力の不足が新たなソフトウェア要員の需要を引き起こすという悪循環となって現れてきており、ソフトウェア技術者 97 万人不足説に代表されるように、単に人員不足の解消をなんとかしてほしいという形での産業界の要求となっている。

しかしながら、真の問題は技術者の専門能力という質にあるのであって、決して量にあるわけではない。現在教育を受けている学生が卒業して社会の中堅として働く時期には、どんな新しい問題が待ち受けているか分からない。そのような局面にぶつかったときに、彼らがそれを克服して新しい視界を開く能力をこそ大学教育の場で養わなければならない。卒業して即戦力になる人材を供給するだけでは、卒業生の数をいくら増やしたところで焼け石に水であろう。この点の認識が大学においても産業界においても十分でないところに問題がある。

1.3.2 教員について

冒頭に述べたとおり、我が国の大学にはコンピュータサイエンスの専門的な訓練を受けた教員が少ない。

そこで、専門教育を行える人材をどのようにして育てるかが、今後の教育体制を考える上での最重要課題となる。

これこそが当委員会の最大の課題であり、中間報告書にはいくつかの提案が示されているが、それは次の3点に集約される。

(1) 情報系専門学科の教育内容を再検討すること。これにはモデルとなるコアカリキュラムを作成することや、真にコンピュータサイエンスを教育内容とする学科を新たに設立することも含まれる。

(2) コンピュータサイエンスのための十分な研究環境を大学に整えること。これはコンピュータサイエンスの優秀な人材を大学に確保するためである。現在は企業のほうが魅力ある研究環境を提供できる状況にある。

(3) 教員の再教育機関を設立すること。近年、国立大学では学科の再編成にともない、他分野の研究者が情報系学科に移りコンピュータサイエンスの学部教育を担当する必要に迫られているためである。

大学の教員の資格審査は（新設・再編時以外は）学部の教授会に委ねられている。ここで、学位の有無、論文の数などを中心とした「客観的な」基準を（分野を問わず）一律に適用することには問題がある（2.4 参照）。

1.3.3 設備上の問題点

情報系専門学科のアイデンティティは、コンピュータサイエンスのカリキュラムを実験と演習に基づいて実施している点にある。このカリキュラムを実施するには計算機システムを必要とする。この点に関して、中間報告書では、以下の問題点を指摘している。

- (1) 計算機システムの老朽化の問題
- (2) 計算機システムのブラックボックス化の問題
- (3) ネットワークの維持管理の問題
- (4) ソフトウェアの費用の問題

以下、それぞれの問題について述べる。

第一の問題は、計算機システム自身の問題である。初期の国立大学の情報系専門学科の計算機システムは買取りであった。しかし、現在の国立大学の予算のシステムでは、機種更新の予算がつくまでには少なくとも10年はかかる。ソフトウェアの新しい概念は最新のプログラミング環境からだけ生まれるものであり、4～5年で1世代の技術革新がある計算機システムを10年も使っているのは、教育内容自体が陳腐化してしまう。そのため、レンタル制度への移行の必要性が文

部省に認められたが、現実には、移行のテンポは遅く、レンタル化が達成できていない大学の情報系専門学科は相当の教育・研究ギャップを強いられている。また、学科の学生数によらずレンタル予算が一律であることも問題である。

なお、計算機の処理能力だけを議論して、これだけの能力とこの使用実績からすれば不足するはずはないという議論もあるが、それは事務計算など定形的な業務の場合に成立することであって、新しい概念を育てるには誤った議論といわざるをえない。

第二の問題点は、計算機システムのブラックボックス化の傾向である。すなわち、著作権問題（国産メインフレームは外国機との互換路線を選択している）に加えて、あまりにも巨大な OS になってしまっているため、TSS を使ったプログラミング言語の教育をはじめ、計算機システムをブラックボックスとして使う立場の実験や演習に重点が移ってしまう。

情報系専門学科の計算機システムを学内共同利用の計算センタへ統合する動きもあるが、それは必然的にブラックボックス化を招く。

パソコンやワークステーションに関しても、ほとんどが米国製の OS をベースにしており、事情は似ている。たとえば、ワークステーションの OS として普及している UNIX は、初期の版のものは、大学にソースコードが公開され、大学で自由に研究者が移植をしたりできた。しかし、UNIX の標準化動向とともに種々の系列の OS ができあがり、現実にはソフトウェアの交流がコンピュータサイエンスの研究に寄与することは少なく、UNIX の操作技能の向上にしか役に立っていない。もっと本質的な問題は、日本語 JIS コードを対象としていないため、日本語情報処理に関する教育・研究に対して基本的に弱いことである。

第三の問題は、ネットワークに対する研究・教育・運用の問題である。特に、ネットワークの維持、管理については、従来のオペレータ的な色彩の強い技術職員では十分に対応できず、OS やシステムアーキテクチャの素養のある助教授・助手・大学院生などが面倒をみないかぎり、実質的な運用は不可能である。その運用のための維持・管理業務の負荷が研究と教育の本業をはるかに上回ってしまう場合も多い。

第四の問題は、ソフトウェアの費用に関するものである。たとえば、パソコンやワークステーションを学生実験の 1 クラスの学生数に対して 1 人 1 台（全体で数十台）設置することは可能である。しかし、OS を

はじめ各種の応用ソフトウェア（ワープロソフトなども含めて）の費用は、台数効果のため、実に膨大な支出になってしまう。しかも、バージョンアップの費用も台数分になる。

大学での実際の使用は営利的でない。業界の理解が望まれる。

以上の問題点には、決定的な妙案があるわけではないが、中間報告書には解決策がいくつか示されている。

1.3.4 学科の構成について

コンピュータサイエンスのコアカリキュラム（情報系専門学科が教えるべき最小限のカリキュラム。3.1 参照）を教えるには、少なくとも 6 名の教員が必要であることが、ACM の CURRICULUM '78 では指摘されている⁵⁾。一方、我が国の学科の編成は、従来、教授 1 名、助教授 1 名、助手 1~2 名からなる講座を単位として 4 講座程度であり、小さいところが多かった（例外はある）。このような小編成の学科が単独でコアカリキュラムを実施するには、大部分の教員がコンピュータサイエンスを専門としているのでなければ困難であろう。コンピュータサイエンスのカリキュラム全体を教育し、大学院の教育・研究の指導もするためには、もっと多くの教員が必要であることは言うまでもない。

また、計算機環境を維持するための技術職員の必要性も、ACM カリキュラムでは指摘されている。これらの技術職員の技能の向上に対する配慮も必要である。実験・演習による個人指導のための助手も不足している。これについては、必ずしも常勤の助手を多数必要とするわけではないが、大学院生などを雇用できる制度（TA 制度）や十分な予算措置が必要である。これは、当の大学院生にとっても良い刺激となり教育的効果が期待できる。

近年、国立大学では大講座制（一つの講座に複数の教授、助教授、助手を置く）を採用した、いわゆる大学科制への改組が盛んに行われている。工学部の場合、情報関連の学科では、電気工学科、電子工学科、物理工学科、通信工学科、経営工学科などと情報系専門学科とを合併して、電子情報工学科などとする人が多い。その結果、入学定員 250 名という大きな学科も珍しくない（もっとも、学部全体の定員が変わるわけではないから、結果として、学科数が減る）。

しかし、現実には、改組ともなう副作用の防止など、難問が多い。最大の副作用は、当事者の利害か

ら、改組に際してコンピュータサイエンスとは無関係の分野が「情報系専門学科を強化するために」取り入れられ、その結果、これから充実しなければならない情報系専門学科の教育の専門性と質が低下することである。1.3.1 に述べたように、もともとコンピュータサイエンスの教育・研究の含まれ方が少なかった我が国の情報系専門学科が改組によりますます専門性を薄めていく傾向にあることについて、危機感さえ唱える委員がいたことを報告しておこう。中間報告書では、大学科への改組をプラスに活かすコース運営のしかたなどが提案されている。

我が国の大学では、情報系専門学科の枠（教員・学生の定員など）は文部省が認可するが、内容は基本的には大学の自治に任されている。上記のような問題点が生じた背景には各大学固有の事情もあるが、基本的にはそれぞれの情報系専門学科の責任と努力で解決すべきことがらである。大学人のモラルが問われるところである。

2. 情報系専門学科の教育

本章では、CS, IS, ISE のカリキュラムについての考え方を示し、人材を養成する制度的保証について論ずる。

2.1 情報処理教育の基本的な考え方

情報処理教育に限らず工学教育の基本問題は、それぞれの分野において、将来の日本を支えるに十分な独創性と人間性の豊かな人材を教育することである。しかしながら、たとえば独創性に関する具体的な教育の方法論となると、いまだ十分な議論が行われていない。

本稿では、これらの問題に対し、次の基本的立場から情報処理教育についての議論を行う。

これらの問題に対する本稿の基本的立場は、1.3.1 に述べたコンピュータサイエンスの教育と同一である。より詳しく述べるならば、**[基本的立場]** “与えられた実世界と、その中でグローバルに与えられた解決すべき問題に対し、まず問題を論理的に正確にとらえ、定義し、この上で与えられた問題を解くための論理的方法とこれを具現化するシステム構築のための十分な能力を養成すること”。

この教育が徹底されれば、われわれは少なくとも急速な技術の変化に対応でき、さらに新しいイノベーションを行うことのできる人材が養成できるものと思っている。

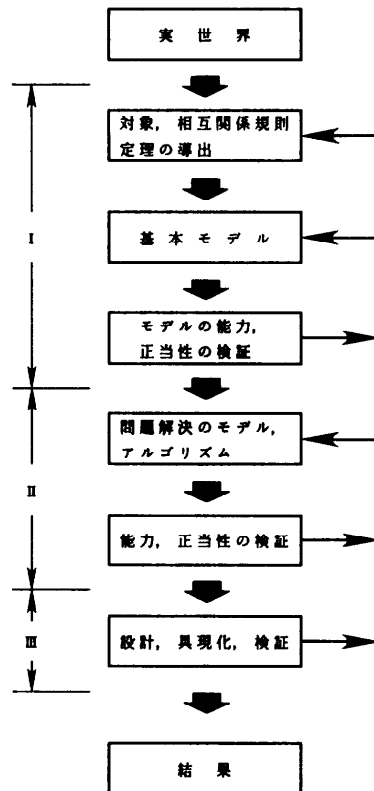


図-3 与えられた世界から問題解決まで

さて、上で述べた、与えられた世界から問題解決に至る流れを今少し詳しく論理的に整理してみれば、図-3 のように三つのフェーズに分解できる（中間報告書第二部参照）。第一のフェーズは、実世界の中から問題の本質を見だし、この問題が十分に処理可能とする抽象モデル設定のフェーズである。

このためには問題の中から問題に関与する本質的な対象を抽出し、各対象間の相互関係、対象の世界のもつ基本的な法則、規則を解析的に、またはヒューリスティックな手段で見いださなければならない。当然このフェーズでは与えられた世界と抽象モデルとの間で、このモデリングが十分正当なものであるか否かの検証が不可欠である。

第二のフェーズでは、第一のフェーズのモデルを、必要があればさらに詳細化し、この上で実際の問題を具体的に解くことのできるアルゴリズムを与えるまでのフェーズである。このとき与えられたアルゴリズムに対する正当性、効率性は十分に検討されなければならない。

第三のフェーズは第二のフェーズのモデルとアルゴリズムをベースとし、実際に存在する実システムの上に問題解決を具現化するためのシステム構築のフェーズである。

当然のことながら、システムの情報処理に関する深い知識をもとにした、具体的なソフトウェアのインプリメントはこのフェーズに入る。またこのフェーズでは構築されたシステム全体の正当性、効率、及び外部より与えられる多くの制約が十分に満足されているか否かの検討はシステム設計の上で重要なものとなる。

われわれは以上述べてきた三つのフェーズの教育を徹底することにより、情報処理の次世代の発展に対しても十分フォローできる人材が養成できるものと信じている。

次の問題は、以上述べてきた三つのフェーズの考え方を具体的にどのようにして情報処理教育として、つまりカリキュラムの上で具現化するかということであるが、方法論として重要な考え方に次の二つのことがある。

第一は直観的なまたはヒューリスティックな立場からのモデリング、及びアルゴリズムを導くことのできるセンスを養う教育である。このためには情報処理に対する考え方の基本的な枠組みと、これまでに知られている代表的な問題解決法、及びアルゴリズムについて、十分な教育を行うこと。

第二は実際のシステムのインプリメントのための十分なセンスを養うための徹底した実験と演習による教育である。

本章では、以下、第一のポイントに主眼をおいて、カリキュラム編成の一つの考え方を示そう。

2.2 CS 教育のためのカリキュラム

コンピュータサイエンスのモデルカリキュラムは現在当委員会で検討中である。当委員会としては ACM CURRICULUM '78, ('88?) を参考としつつ、それを越えるものを目標に作業を進めている。ACM CURRICULUM については 3.1, 3.2 で紹介するので、本節では、コンピュータサイエンスのカリキュラムについて当委員会で提唱されたいくつかの見方を紹介することにする。

教える対象を物理現象のレベルから知能情報のレベルに到る抽象化の軸に沿って眺める見方がある。そこでは、次のような段階が考えられる (図-4)。

- (L6) 認識、知能、思考
- (L5) アルゴリズム、計算可能性

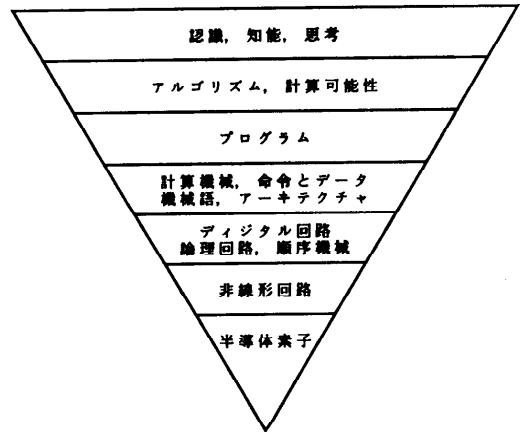


図-4 コンピュータサイエンスのカリキュラムにおける諸概念の抽象化レベル

- (L4) プログラム
- (L3) 計算機械、命令とデータ、機械語、アーキテクチャ
- (L2) デジタル回路、論理回路、順序機械
- (L1) 非線形回路
- ((L0) 半導体素子)

コンピュータサイエンスのカリキュラムはこれらのレベルを連続して含まなければならない。中間のレベルがどれか一つでも欠けると、その上下のレベルの教育が連続せず、学生はコンピュータサイエンスの全貌を把握することができない。実は、我が国の多くの情報系専門学科の教育に欠けているのは、この連続性なのである。どのレベルから上をコンピュータサイエンスのコアカリキュラムに含めるべきかについては、当委員会の委員の間に意見の違いがあった。少なくとも (L3) を含めるべきである (現実には (L4) から上だけを教えている情報系専門学科もあるようであるが) ということについては意見が一致したものの、(L1) までも含めるのは行き過ぎではないか (まして (L0) など論外) とする意見もあった (実際、ACM 88 報告書⁴⁾には (L1) は含まれていない)。一方、シリコンコンパイラの時代に (L0) の知識は不可欠との意見も強かった。

学生各人の学習の展開を時間軸に沿って眺める見方もある。そこでは、次のような段階が考えられる (図-5)。

- (T1) 問題解決手段とプログラミング
- (T2) 大規模システムのインプリメント
- (T3) 計算機と人間を含むマルチメディア環境に

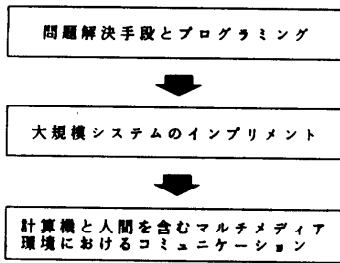


図-5 コンピュータサイエンスのカリキュラムの展開

におけるコミュニケーション

(T1) は、初心者フォンノイマン型コンピュータでできることの限界を理論的に教えるとともに実感として体得させ問題解決能力を養うことを目的とするもので、たとえば、プログラミング序論、アルゴリズムとデータ構造、計算機の基本構成と動作、コンピュータアーキテクチャ、論理設計、オートマトン理論、数値解析、数値計画法などを含む。(T2) は、(T1) の基礎の上に、オペレーティングシステムなどをインプリメントする能力を養うことを目的とするもので、コンピュータサイエンスの一つの本質が量との戦いであることを実感として体得させる場でもあり、たとえばソフトウェア工学やデータ工学を含む。(T3) はフォンノイマン型コンピュータの世界から外へ踏み出して新しい世界を築く能力を養うことを目的とするもので、たとえば人工知能、日本語情報処理、ヒューマンインタフェース、情報ネットワークなどを含む。ただし、(T3) 全体をコンピュータサイエンスのコアカリキュラムに含めて考えるということについては異論もあった。

最後に、コンピュータサイエンスの教育の基本的な構成要素として次のものをあげる見方もある。

- (a) 問題の定式化、モデリング
- (b) 問題解決の手順、アルゴリズム
- (c) 計算のしかけ、アーキテクチャ

この分類は、上記(Ln)、(Tn)とは異なり、1次元の軸の上に並ぶものではない。むしろ、平面上に三角形の3頂点のように配置するのがよい(図-6)。(a)はソフトウェア開発において最初に取り組みなければならない部分であるので、しいて順序をこじつければ、(a)だけ最上位に位置するということもできよう。

アルゴリズムやアーキテクチャが基本的な概念であることについては当委員会の委員全員の意見が一致したが、(a)の「問題の定式化、モデリング」にコンピュータサイエンスの中でどの程度の重みを与えるかに

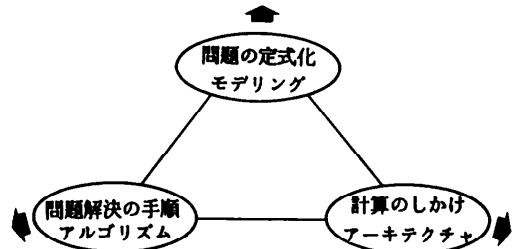


図-6 コンピュータサイエンスのカリキュラムの構成要素

ついては委員の意見が分かれた。「アイデンティティのないジェネラリスト」を育成してしまうことを危惧する意見がある一方で、当委員会の理想どおりに我が国の情報系専門学科の教育が充実した暁において次に目指すのは(a)の教育の充実であるとする意見もあった。

以上に述べたことがらは、コンピュータサイエンスのカリキュラムを多次元空間とみなしているいろいろな座標軸や平面に射影して論じたものである。どの論じ方においても、教えるべきことがらの「最小限」の設定については、当委員会に若干の意見の違いはあった。それは、各委員の専門分野の違いを反映したものであるとともに、各委員の所属する学科にどの分野の専門家がいてかを反映したものである。不可欠なことがらを教えないのは論外であるが、必要最小限の要件を満たした上でそれぞれ得意な分野を重点的に教えることにより情報系専門学科が独自性を発揮し特徴を競うのはむしろ望ましいことである。当委員会が目標としているのは、この必要最小限であるコアカリキュラムである。

2.3 IS, ISE のための教育

CS の教育とカリキュラムの構成についてはすでに前節で述べたとおりである。CS の分野はすでにコンピュータサイエンスの教育のもとに、ある程度の長い歴史と、実績があり、また教育の考え方についても多くの研究調査が行われてきた^{9)~11)}。しかしながら本節で対象とする IS, ISE の教育問題については、今まできちんとした論議が行われたことは少なく^{9)~12)}、その教育の体系の考え方についても、今後多くの議論が必要であろう。当委員会としても、この問題について討議を行っており、最終年度の報告書には基本的なガイドラインが示されることになろう。その間に得られた成果は本誌にも報告されることになる。

以上の立場から本報告では、具体的にカリキュラムの内容に立ち入ることなく、カリキュラム構成のため

の一つの考え方を提示したい。

まず基本となる IS, ISE 教育のありかたは CS の教育の理念と同じようにして次のように述べる事ができよう。

“実世界の中から利用者によって漠然と要求された問題解決についてその問題を理論的に明確にし、問題解決の方法論を的確に定式化すること。またこの上で与えられている多くの制約条件を満足する最適または準最適システム構築の具体化ができる能力を与えること”。

この立場からみると、IS の分野の教育には相当程度 CS のコアカリキュラムが必要となる事が理解できる。一方 IS 特に ISE の分野については特にユーザオリエンテッド (user oriented) の立場からのシステム構築の能力の養成が重要である。

すなわち IS, ISE の教育の上で重要なポイントの一つは利用者の立場にたった要求分析のための能力を養うことである。以下 CS の考え方と同じく、この分野の教育についても基本となるものは、モデル化、システムの具体的設計、インプリメンテーション、そして検査及び保守に至る一貫したシステム構築技術の教育である。

以上のことを総括すれば、この分野の技術者に要求される教育の考え方はまず CS の教育に基礎をおき、実際に直面する問題に対応できる、すなわち実世界の業務内容に関する深い知識をもつことのできる、能力を養う教育が不可欠のものとなる。たとえば、バンキングシステムの構築のためには、情報処理技術の上にこの分野に直接関係ある金融、財務、会計などの深い知識及びバンキングシステムのもつ社会的な背景も十分に理解できる能力がシステム構築の上できわめて重要なものとなる。そしてこのような教育を本当に達成するためには教育制度の上からみて従来とは異なる新しいタイプの教育制度を考えることが不可欠であり、これについては次節で述べよう。

いずれにしても今後のことを考えれば、われわれは IS, ISE の教育のあり方について、我が国独自の立場から、早急につめなければならない。特に ISE の技術者について言えば、この分野の急速な拡大にともない、必要とする技術者の不足がきわめて深刻な問題となりつつある。このために広く文系を含む一般情報処理教育の立場から ISE の教育を考えていくことも今後の重要な問題解決法である。

2.4 情報処理教育、特に IS, ISE のための教育制度のあり方

前節で述べたように、IS 特に ISE の技術者の不足は今後 100 万人に達するであろうという推定がある。このような状況の中で IS, ISE の教育はきわめて重要なものとなるが、このことは単に数多くの技術者を養成すればよいということではなく、システム構築の上流工程の技術者としても十分対応できる人材を教育することである。

しかしながらトップレベルの IS, ISE の技術者をどのようにして教育すべきか、このためには次の二つのことを考えなければならない。

第一はこの分野における教育が十分機能するためには少なくとも教育を受ける学生にとってシステム構築の意味、問題意識をある程度明確にもっていることが必要である。

第二は IS, または ISE の卒業者として、卒業後相当早い機会に実践的に役に立つ能力を身につけていることである。

しからば以上二つの問題の解決に対して、現在の教育制度の中でこのような教育が可能かと言えば、相当に難しいことも事実である。この問題点の第一が教育制度の問題であり、第二が教員側の問題である。中間報告書第二部では、いくつかの提案を述べている。

2.5 大学院制度の改革

今までの議論は、すべての情報処理教育 (CS, IS, ISE) を対象に行ったものである。この中で IS 及び ISE の教育について、思いきった制度上の改革が必要であることを述べてきた。しかしながら、より広い立場で現在の大学院制度を考えてみると、もっと本質的に重要な制度改革が必要であろう。

大学の使命が、社会が必要とする有為な人材を教育し次の世代のための基礎分野の研究・開発を推進することであることは当然であるが、同時に次の世代のための優秀な研究者を育成する人材を育てることもきわめて重大である。そしてこの機能を果たす重要なものが大学院の博士 (後期) 課程である。しかしながら現在どの大学の博士 (後期) 課程をみても日本人学生の進学者はきわめて少なく、多くは外国人留学生によって占められているのが現状である。

このままでは、次の世代の基礎研究を支える優秀な研究者を確保していくことはきわめて難しく、同時にこのことは大学において、優秀な教育者も確保できないことを意味している。そしてこの現象は少なくとも

情報科学・工学の分野ではすでにその兆しが現れてきている。

このような状況を変えていくためには、新進気鋭の若い研究者が積極的に博士（後期）課程に進学を希望するような抜本的な制度改革が必要である。そしてこの改革のために必要なポイントとして次の二点があげられる。

第一は博士（後期）課程に魅力ある研究環境を整備すること、それに相当する待遇を与えること。

第二は経済的にも現在のわずかな奨学金ではなく、同世代の人と比較しても引け目を感じることがない十分な給費制度を打ち立てること。

日本の将来が今後の先端的産業の研究・開発にかかっているならば今こそ思い切った大学院制度のための施策をとるべきときではないであろうか。

3. コンピュータサイエンスのためのカリキュラム

コンピュータサイエンスの教育が IS, ISE を含めたあらゆる情報処理教育の基礎となることはこれまでに述べたとおりである。そのため、当委員会はコンピュータサイエンスのためのコアカリキュラムのモデルを作ることを大きな目標としており、現在その作業を進めている。それには ACM のモデルカリキュラムが参考となるという点で委員の意見が一致している。前述のとおり、当委員会では ACM を越えるモデルカリキュラムを作る意気込みである。本章では、3回にわたって改訂された ACM のカリキュラムを、当委員会での検討の経過を加えて紹介し、さらに、我が国独自のカリキュラムに必要なものについて考察する。なお、本章の内容は中間報告書第一部第4章を圧縮したものである。

3.1 ACM カリキュラムの変遷

米国の学会 ACM (Association for Computing Machinery) では、カリキュラム委員会を作り、コンピュータサイエンスの大学教育に関して、1968年に CURRICULUM '68 という提案 (Recommendations for the Undergraduate Program in Computer Science) を行った⁶⁾。この提案は、既存の学問分野に対してコンピュータサイエンスが独立した学問分野であること、これを専門に教育研究する学科を独立して作る必要があること、そこで教える内容がどのようなものであるかを明らかにしたものである。この提案は、大学などの教育・研究機関のみならず、メーカの研究

所などに対しても、計算機分野の研究を推進するためには、どのような学問素養が必要であるかを明らかにした点で、大きな影響を与えた。今日の学問水準からみれば不十分であるが、その歴史的意義は大きい。

当時は、IBM System/360 が世に出て間もなくであり、OS/360 の開発が進行中で、それまでに経験したことのない大規模なソフトウェア開発の複雑さ、納期の遅れ、信頼性の欠如などの問題が認識され始めたころである。コンパイラの最初の教科書は執筆中であつたし、FORTRAN の最初の標準化が完成したばかりであった。専門教育の場で広く使われている PASCAL も論文の形で世に出たのは 1970 年代に入ってからであった。データベースの最初の教科書が出版されたのも 1970 年代の半ばである。

ACM は CURRICULUM '68 に対して継続的に見直しを行って、10年後の 1978 年に CURRICULUM '78 を提案している⁶⁾。この提案は 1970 年代におけるソフトウェア分野と基礎分野における学問の発展を主として踏まえており、データベースやオペレーティングシステム、アルゴリズム解析の科目が新しく加わつた。CURRICULUM '78 の科目名は次のとおりで、CS1 から CS8 までがコアカリキュラムである。

- CS1. プログラミング序論
- CS2. プログラムの設計と実現
- CS3. コンピュータシステム序論
- CS4. コンピュータハードウェア基礎
- CS5. ファイル処理入門
- CS6. オペレーティングシステムとアーキテクチャ I
- CS7. データ構造とアルゴリズム
- CS8. プログラミング言語の構造
- CS9. コンピュータと社会
- CS10. オペレーティングシステムとアーキテクチャ II
- CS11. データベース管理システムの設計
- CS12. 人工知能
- CS13. アルゴリズム
- CS14. ソフトウェアの設計と開発
- CS15. プログラミング言語の理論
- CS16. オートマトン、計算可能性、形式言語
- CS17. 数値解析 (解析系)
- CS18. 数値解析 (代数系)

また、CURRICULUM '78 は基礎となる数学についての次の提案をしている。

- MA 1. 微分積分学序論
- MA 2. 数学解析 I
- MA 2 A. 確率論
- MA 3. 線形代数
- MA 4. 離散構造論
- MA 5. 数学解析 II
- MA 6. 確率と統計

これらの科目の間の関係は図-7 のとおりである。

ACM では、その後も、学問や技術の進展に合わせたカリキュラムの検討を行っており、1984 年には最初の二つの科目 CS 1 と CS 2 について検討結果を発表している^{7),8)}。

1985 年に、ACM は学問分野としてのコンピュータサイエンスを定義づけるための作業部会を充足させた。作業部会は、IEEE コンピュータソサエティと協力して作業を行い、1988 年 8 月に最終報告書を発表している⁹⁾。これによれば、分野を 9 つの副分野に分類し、各副分野を理論・抽象化 (モデル化)・設計の 3 パラダイムにわけ、結局、9×3 のマトリックスによって、全分野を定義する枠組みを与えている。同報告書は、マトリックスによって学問分野の枠組みを示しそれを構成する要素を明示したものであり、それ自体はカリキュラムではない。これらの要素を組み合わせ順序だてることによって、カリキュラムが決定されるべきものである。

CURRICULUM '68, CURRICULUM '78 に対応するいわば CURRICULUM '88 の全体像は、この報告書には提示されていない (その代わりに、最初の 3 学期に教える三つの科目の講義内容とそれと並行して行うべき実験・演習との試案を示している)。

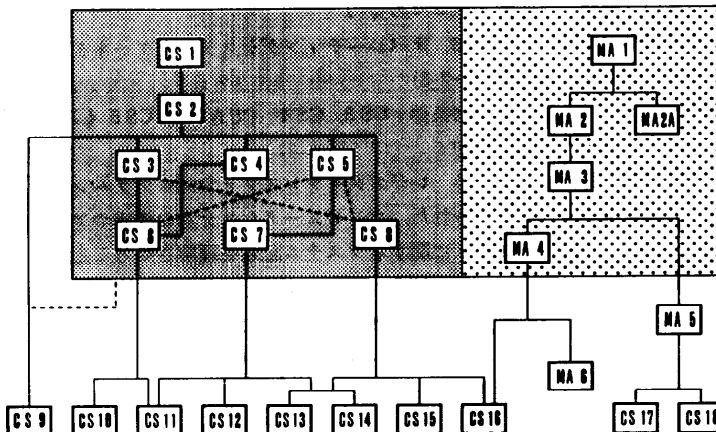


図-7 計算機科学と数学コース (ACM CURRICULUM '78 より)

当委員会は現在この ACM 88 報告書に対して検討を加えているところであり、その結果は最終報告書に述べられるはずであるが、CURRICULUM '78 のコア・カリキュラム CS 1-CS 8 (3.4 で詳述) と比べて、関連する分野は広くなり、内容的には浅くなっているように見える。しかし、講義と並行して行われる実験・演習を強調しているの、運用次第によってきわめて強力なカリキュラムを構成できる可能性を秘めているかもしれない。なお、木村泉委員による ACM 88 報告書の翻訳が本号に掲載されているので、それも参照されたい。

3.2 Curriculum '78 の検討

CURRICULUM '78 は、誕生してから約 10 年になった。使用できる計算機も、10 年前のバッチ処理中心のメインフレームと機械語 (アセンブリ言語) 中心のミニコンから、最近では、パーソナルコンピュータやワークステーションに変わってきた。これにより、学生のプログラミング教育は質的变化を遂げつつある。たとえば、「プログラミング環境」ということばが示すように「使い勝手」は格段に向上した。

そのような現実の中で、CURRICULUM '78 を見直してみると、CURRICULUM '78 自身にも、計算機の歴史的発展から、ファイルやデータセットなどメインフレームの強い影響を受けたカリキュラム構成である点がはっきりしている。

しかし、コンピュータサイエンスのコアカリキュラムについては、ほとんど現在でも十分使用できるものであり、当委員会としては、情報系専門学科では少なくともこの程度の教育内容はきちんと行ってほしいと考える。以下、CURRICULUM '78 の提案の CS 1

から CS 8 までの内容について検討経過を含めて紹介する。なお、邦訳は諸橋委員によるものである。

CS 1 プログラミング序論
目的:

- (a) 問題解決手法とアルゴリズム化を秩序だてて教えること。
- (b) 手続き抽象化とデータ抽象化を紹介すること。
- (c) プログラムの設計、コーディング、デバッグ、テスト、文書化につき、良質のプログラム書法を用いて教えること。
- (d) ブロック構造の書ける高水

準プログラミング言語を教えること。

(e) コンピュータのハードウェア及びソフトウェアの技術の発展について熟知させること。

(f) 計算機科学をさらに学んでいくための基礎を習得させること。

CS2 プログラムの設計と実現

目的:

(a) 引き続き、ブロック構造の書ける高水準言語を用いたプログラムの設計、コーディング、テストの手順を秩序だてて教えること。

(b) データ抽象化の利用法を教えること。ここで使用するプログラミング言語があらかじめ提供している基本のデータ型以外のデータ構造（リンクされたリスト、スタック、待ち行列、木など）を、例題を用いて教えること。

(c) それらのデータ構造のさまざまな実現方法を理解させること。

(d) 探索とソートのアルゴリズムとその解析の初歩を教えること。

(e) 計算機科学をさらに学んでいくための基礎を習得させること。

[当委員会のコメント: CS1, CS2 では、計算機による問題解決能力を養うためにプログラミングの方法論とそれを支えるアルゴリズムを教えることが目的である。CS1, CS2 を FORTRAN や BASIC の教育で代用するのは論外である。

CS1, CS2 が重要であるという点では委員の意見が一致したが、教育のフィロソフィでは当委員会の中でも立場の違いがみられた。一つは“問題解決法”の教育であるという立場であり、他の一つは“ソフトウェアを作るための道具立て”の教育であるという立場である。これは各委員の専門分野のバックグラウンドの違いを反映したものと見えよう。この点に関しては、相違点を問題にするより“自分の専門分野以外はすべて道具である”ことを認識することのほうが大切であろう。残念ながら、我が国の大半の情報系専門学科のカリキュラムは、このようなフィロソフィを論争するレベルにすら達していない。]

CS3 コンピュータシステム序論 (2-2-3)

先修条件: CS2

目的:

(a) コンピュータシステムの基本概念を習得させること。

(b) コンピュータアーキテクチャの基礎を教える

こと。

(c) アセンブリ言語を教えること。

CS4 コンピュータハードウェア基礎 (3-0-3) または (2-2-3)

先修条件: CS2

目的:

(a) ハードウェアの主要な構成要素の構造や仕組みの基礎を教えること。

(b) コンピュータシステムにおける情報の流れと制御の仕組みを理解させること。

(c) 論理設計の基礎を習得させること。

[当委員会のコメント: CS3, CS4 の主眼は、計算機のアーキテクチャとアセンブリ言語を演習つきで教えることであるが、問題点は手間がかかることであり、スタッフの充実が不可欠である。架空の計算機をモデルに教えると、アーキテクチャとアセンブリ言語が分離する。メインフレームのアセンブリ言語ではアーキテクチャの全貌がつかみきれない。マクロアセンブラも好ましくない。]

CS5 ファイル処理入門 (3-0-3)

先修条件: CS2

目的:

(a) 大容量記憶装置上でのデータ構築法に関する技術やその裏付けとなる概念の基礎を教えること。

(b) 大容量記憶装置の使用経験を積ませること。

(c) データ構造とファイル処理技術の応用に関する基本を教えること。

[当委員会のコメント: コンピュータシステムに依存する部分が多い点に注意。CS5 は IBM OS/360 の影響が大きく、時代にそぐわない。CS4 とCS6 に解体すべきである。]

CS6 オペレーティングシステムとアーキテクチャ I (2-2-3)

先修条件: CS3, CS4 (できれば CS5 も)

目的:

(a) レジスタトランスファやプログラムレベルで記述された、コンピュータシステムの構成やアーキテクチャに関するシステム記述を理解させること。

(b) オペレーティングシステムを原理的にとらえ、その主要概念を教えること。

(c) オペレーティングシステムとアーキテクチャの関係を教えること。

[当委員会のコメント: アーキテクチャと OS との関連づけを教えることが大切である。シリコンコンパ

イラの時代になり大学でアーキテクチャの研究・開発ができるようになりつつあることを考えると、大学で CS6 を教える条件が整いつつあるとも言える。]

CS7 データ構造とアルゴリズム (3-0-3)

先修条件: CS5

目的:

(a) データ構造ごとに存在する非数値アルゴリズムについて解析し、設計に用いる方法を習得させること。

(b) DBMS 上でデータを扱う際にどの手法を選ぶかという問題に対して、アルゴリズム解析を行い、設計の規範を作れるように指導すること。

[当委員会のコメント: データ構造とアルゴリズムというキーワードはますます重要となってきた。科目名はもっと一般的に“情報構造とアルゴリズム解析”とすべきである。]

CS8 プログラミング言語の構造 (3-0-3)

先修条件: CS2 (できれば CS3 と CS5 も)

目的:

(a) プログラミング言語の仕組み、特に、実行時のプログラムの振舞いについて理解させること。

(b) プログラミング言語の記述と解析に関する形式論の基礎を教えること。

(c) 基礎コースで教えた問題解決手法とプログラム作成手法をさらに発展させること。

[当委員会のコメント: プログラミング言語の文法を教えるのではない。コンパイラを作らせるくらいの心構えがないと本当の目的を達せられない。]

3.3 我が国のコンピュータサイエンスのカリキュラム

我が国のコンピュータサイエンスのコアカリキュラムを検討するときに、前記の ACM のカリキュラムなどを参考にすることはもちろん必要であるが、忘れてはならないものがある。それは、第一に我が国の情報処理に固有の問題であり、第二に比較的歴史が浅く ACM の諸カリキュラムに盛り込まれていないコンピュータサイエンスの新しい分野である。前者に属するものには日本語情報処理があり、後者に属するものにはソフトウェア工学と情報ネットワークがある。以下に、これらをコアカリキュラムに取り入れるときに盛り込むべき内容を述べる。本節の内容には CS ワーキンググループの牛島主査、都倉委員、諸橋委員、市川委員の寄与が大きい。

3.3.1 ソフトウェア工学の発展とソフトウェア工学の教育

ソフトウェア工学は、ソフトウェアの品質向上とソフトウェア技術者の生産性向上とを目標にして提唱されて 20 年が経過した。そこで確立されたソフトウェア開発に関する多くの概念や知見は現在のソフトウェア開発や保守の基礎となっている。それらには次のようなものがある¹³⁾。

ソフトウェアライフサイクル

ソフトウェア品質

構造化プログラミング

ソフトウェア構造

並行性

検証

可視性

ソフトウェア開発環境

ソフトウェア開発経験の乏しい学部学生にこれらの項目を教えたとしても、単なる知識の羅列に終わってしまう。そこで、プログラミング入門教育の段階からソフトウェア教育のあらゆる場で、特に演習において、次のことがらを強調することが大切である。

(1) 問題の定義

入門コースの一人で作る小さなプログラムでも、ライフサイクルの概念を当てはめて各段階の存在を意識させ、最上流工程(発注者の要求を分析し解くべき問題を的確に定義する工程)が最も重要であることを強調する。プログラミング演習の課題を曖昧な形で与え、問題を定義できれば問題解決の大部分が終わったということに気づかせる。

(2) アルゴリズムとデータ構造の設計

データ構造が計算の手に与える影響を検討させる。データ構造の設計において、一度しか実行しない操作の時間を重視するのか、頻繁に行う操作の時間を重視するのか、といったトレードオフの感覚を養わせる。

(3) 検証とテスト

入門コースの一人で作る小さなプログラムでも、トップダウン設計と段階的詳細化法とに基づく構造化プログラミングのパラダイムを経験させ、要求仕様に基づく機能テストと、プログラムの論理に基づく構造テストを実践させる。

プログラムの正当性の検証とその例も示す。

(4) 品質評価

出来上がったプログラムの品質を評価する習慣が大

切である。ソフトウェアのマトリックスはソフトウェア工学の今日的な課題であり、教育の段階でそのまま適用できるものは多くない。したがって、プログラムに対する教員の批評（教員の感性や美意識とも関係する）が重要である。

性能評価（特に時間と実行回数の計測）だけはただちに実施できる。測定によって系に乱れを生じさせることがプログラムの場合にもあることに気づかせる。性能に関係するのはプログラムテキストのごく一部分であることに気づかせる。性能だけが唯一の品質というわけではないことを断わっておく。

(5) 利用の手引き

発注者でも開発者でもない利用者の存在を常に念頭においておくこと、そのような利用者のために入力データの与え方や出力結果の読み方の手引きが必要であることを強調する。

演習のレポートには、プログラムの仕様やアルゴリズムの解説や作成過程の説明のほかに、そのプログラムの利用の手引きを（たとえ数行でも）必ず付けさせる。

(6) 日本語文章を書く能力

ソフトウェアはプログラムだけから成り立っているわけではなく、開発のあらゆる段階にドキュメンテーションが必須であることを強調する。

必ずしも整理されているわけではない内容を論理的に整理して日本語で相手に分かりやすく説明する能力を養う。

3.3.2 情報ネットワーク学

ACM のカリキュラムに欠けているが重要なものに、情報ネットワークの理論がある。コアカリキュラムとしては、階層化アーキテクチャを中心にデータ伝送方式、通信方式、交換方式など、ネットワーク構築に必要な基本的技術について教え、また、システム評価に必要な待ち行列理論の基礎についても教える。

内容は

- (1) 情報ネットワークとは
- (2) データ通信システムの構成
- (3) 階層化アーキテクチャ
- (4) 物理層
- (5) データリンク層
- (6) ネットワーク層
- (7) ネットワークアプリケーション
- (8) 性能評価

評価手法（待ち行列理論の基礎）、及びこれによる簡単なネットワークの設計手法

3.3.3 日本語情報処理

漢字を含む日本語テキストの処理技法は、英文テキスト処理で確立された技術においてアルファベットを漢字にするだけで解決するものではない。我が国の日本語処理技術は、日本語ワープロの普及とともに急速な発展を遂げ、今や世界の頂点に立っている機械翻訳研究においても重要な役割を果たしている。今後ますます重要となる日本語処理技術をしっかり身につけさせるためには、コンピュータサイエンスのコアカリキュラムにおいて独立した科目として取りあげるべきであると思われる。

なお、自然言語を扱うという意味では、人工知能の科目とも強い関連がある。

(1) 日本語情報処理

- A. 文字の特質とコード
- B. 語を単位とする情報処理
- C. 日本語テキスト処理
- D. 日本語の文法の特徴と構文解析アルゴリズム

(2) 自然言語処理

- A. 言語とオートマトン
- B. 構文解析アルゴリズム
- C. 言語理解モデル
- D. 言語理論、構文論と意味論
- E. 計量言語学

4. 情報系専門学科の実験・演習

ACM のモデルカリキュラムにおいて演習の重要性は随所に指摘されていたが、本章では、情報系専門学科のカリキュラムから実験や演習のあり方だけを取り出して検討し、委員会で出された問題点を整理して述べる。本章の内容には、CS ワーキンググループの都倉委員の寄与が大きい。

4.1 理工系学科における実験・演習

理工系の学科では単に講義をするだけでなく、実験や演習を課している。

しかし、同じ実験・演習とは言っても、伝統的な自然科学系と歴史の浅い情報系とでは、あり方がかなり違うのではないか。この疑問が、中間報告書で提起されている（第一部 5.1 節）。

その違いは、前者では観測行為を通じて先人の経験を追体験させることを主要目的としている¹⁰⁾のに対して、後者では製作過程を通じて（初心者といえども）前人未踏の新規性や獨創性を体験させることを主要目的としていることに起因すると考えられる。後者の

「実現方法が学生に任されている」、「評価が単純でない」などの特徴は理学部の数学科の演習にも共通と思われる。数学の研究では、問題を作り、それを部分問題に分割する、という繰り返しが基本であろう。ソフトウェアの生産性の高い人はモジュール分割が上手であることを連想させる。また、自然系に対する「人工系」という考え方が提唱されているが¹⁵⁾、コンピュータサイエンスは人工系の極端なものと言えよう。中間報告書では、以上の認識から、「体得教育」という概念を提唱し、体得教育に要する負荷（教職員、学生）、実施方法などについて、検討と試案を述べている。

4.2 特徴ある実験・演習の例

中間報告書では、CS ワーキンググループの委員の所属する大学の情報系実験・演習のテーマ例も紹介している（この部分は、最終報告書では、より多くの大学の事例を提供していただき、より充実したものになるはずである）。中間報告書の実験・演習の例を網羅的に紹介することは紙面の都合で不可能であるので、本稿では、筆者の一人が所属する大学の実施例を紹介することにしよう。

(1) 機械語プログラミングの演習の重視

これは前章でも ACM モデルカリキュラムに関連して触れたが、我が国にはこれを実施していない情報系専門学科もあるので、あえて強調しておく。機械語プログラミングはコンピュータサイエンスのコアカリキュラムの根幹である。

機械語プログラミングは計算機アーキテクチャの概念の基礎を与えるために実施するものである。機械語プログラミングの訓練を経ず、コンピュータアーキテクチャの概念の基礎としての位置づけが確立されていないと、「コンピュータのメモリは買い足せば無制限に増やせる」、「文字はキーボードの座標として入力される」などの見当違いな誤解を生ずる。

対象計算機は実在する（実在した）ものであるべきである。筆者の一人の大学では、EDSAC（事実上、世界初のプログラム内蔵計算機）を用いている。架空の計算機のアセンブリ言語は、「低水準言語」の教育には使えても、アーキテクチャの概念を把握させる目的には使えない。

(2) 自分が計算機になったつもりで手を動かしてみる演習

筆者の一人は、手回し計算機を用いて数値計算問題を解いて収束や桁落ちなどの数値的現象を体験させたり、多数の伝票を手作業で整列（ソート）させて計算

量の概念を体得させたりする演習・実験を担当している。一見、前近代的な教育であるが、手ごたえがあるという意味で、他の方法には換えがたい。

(3) メタ工学の実践

問題解決は、問題の把握とモデル化から始まる。モデル化のための理論的基礎には、数理論理学、離散数学、グラフ理論、数理計画法など多数の理論があるが、理論とは別に、問題を把握する実践的訓練も必要である。これを仮にメタ工学とよぼう。メタ工学の演習は、できるだけ専門外の漠然としたテーマを課するのがよい。筆者の一人の大学では、メタ工学の演習として、次のものを実施している。

(3-1) KJ 法 発想法の訓練としての見地から実施しているものである。4人ぐらいで1チームを作り、テーマを決めて、KJ法による問題解決のための提案の発表と文章化を行う。同じ言葉でも、個人個人によって相当違うという事実やコミュニケーションの重要性が認識されるなどの副産物も得られる。「もし自分が当学科の教官であったらどのような教育をするか」、「いかにして楽しく大学生活を送るか」などは学生がしばしばとりあげるテーマである。発表は他チームの学生全員と担当教官の前で行う。

(3-2) 学外での実地調査 実際のテーマを5人ぐらいのチームで調査・報告（発表会と報告書の作成）させるものである。基本的には図書館で調べるのではなく、直接学外（企業、研究所、各種ショーの会場、など）へ出かけて調査する。たとえば「十万円前後のワードプロセッサの性能評価」というテーマで、メーカーや展示場へ出かけたり、ワードプロセッサをメーカーから借りてきたりして、全員で評価項目を作成し、手分けして評価作業を実施した。また、「機械翻訳の程度」というテーマで、ベンチマークデータを作成し、企業、研究所、大学などを訪問して入力・翻訳させてもらったりデータを郵送し翻訳結果を返送してもらったりして評価した。

学外の機関を訪問させるときは、先方に迷惑とならないように、また、十分な協力が得られるように、教官も同行し指導している。したがって、担当教官は5～6名が必要である。

発表は他チームの学生全員と全教官の前で行う。

(4) 製品として完成させてみる

これも筆者の一人の学科で実施しているもので、教官から提示され自分たちが選んだテーマ（ハードウェア、ソフトウェアの製作）を、約2カ月かけて実現

し、発表させる。前記メタ工学の実践の延長上にあるものである。2～5名のチームを編成し1名の教官が指導する。したがって、ほとんど全教官が担当する。たとえば、「音声応答タイム」、「論理シミュレータ」、「(簡単なプログラム言語の)コンパイラ」、「日英・英日翻訳システム」などのテーマが考えられる。発表会は口頭発表と「製品」の実演を含み、他チームの学生全員と全教職員が出席するものとする。製品として完成させることを義務づけているので、発表会の1～2週間前は学生実験室は不夜城となる。ここで「どの程度の仕事ならどの程度の労力でできる」という目安と自信が得られる。講義で教わったことがらの意味が製作実験によってはじめて理解できたという学生も多い。

(5) プレゼンテーションの効果

上記の実地調査、製品製作などの発表会では、OHPなどを使ったプレゼンテーションが行われる。発表会に他チームの学生全員と教官が出席することは前述のとおりであるが、上級生や大学院生にも出席することを奨励している。プレゼンテーションは、当該学生にとって自分の研究の価値を世に問う重要な場であるだけでなく、担当教官にとっても指導の力量が問われる場であるため、必然的に各教官ともその指導にも力を入れることになる。このように、教官相互のチェックは教育の質を維持する上で欠かせない。

補章 委員会の構成と本稿の成立について

当委員会の構成は次のとおりである(所属は委員在任当時のものである)。

- 委員長 野口 正一(東北大学)
 - 正幹事 堂下 修司(京都大学)
 - 副幹事 遠藤 誠((株)日立製作所)
 - 副幹事 竹井 大輔(鉄道情報システム(株))
- (以下、五十音順)
- 青木雄太郎(新日鉄情報通信システム(株))(1990年度)
 - 有澤 博(横浜国立大学)
 - 有山 正孝(電気通信大学)
 - 市川 照久(三菱電気(株))
 - 稲垣 康善(名古屋大学)
 - 乾 侑(長岡技術科学大学)
 - 牛島 和夫(九州大学)
 - 榎本 彦衛(慶應義塾大学)(1990年度)
 - 大岩 元(豊橋技術科学大学)

- 岡野 寿夫(新日鉄情報通信システム(株))(1989年度)
- 岡本 彬(富士通(株))(1989年度)
- 川合 慧(東京大学)
- 木村 泉(東京工業大学)
- 木村 正行(東北大学)
- 園井 利泰(東京大学)
- 清水 武明(長岡技術科学大学)
- 菅谷 修((株)日立製作所)
- 高橋 延匡(東京農工大学)
- 武貞 照雄((株)東芝)
- 田中 義明(NTTデータ通信(株))
- 都倉 信樹(大阪大学)
- 所 真理雄(慶應義塾大学)(1989年度)
- 中森真理雄(東京農工大学)
- 林 圭吉(富士通(株))(1990年度)
- 藤野 喜一(日本電気(株))
- 堀越 彌((株)日立製作所)
- 御牧 義(電気通信大学)
- 村岡 洋一(早稲田大学)
- 望月 徹英(鉄道情報システム(株))
- 諸橋 正幸(日本IBM(株))
- 吉津 英男(日産自動車(株))

また、本研究の委託元である文部省高等教育局専門教育課からは、次の諸氏にオブザーバとして参加していただいた。

- 草原 克豪(課長)(1989年度)
- 若林 元(課長)(1990年度)
- 今野 雅裕(課長補佐)
- 白間竜一郎(専門職員)
- 外嶋 義広(庶務係長)

本稿は当委員会の討議の内容や中間報告書を筆者らの考えによってまとめたものであるが、記載された個々の事例・提言は上記のメンバから報告・提唱されたものである。それらの中には委員会の一般的認識となっているものもあるが、発想のオリジナリティは各提唱者に帰属すべきものである。

ま と め

以上、「大学等における情報処理教育検討委員会」の平成元年度の活動と中間報告書の内容を紹介し、今日の日本における情報系専門学科の問題点を述べ、解決の方向をさぐった。当委員会は平成2年度に最終報告書をまとめ、情報系専門学科のコアカリキュラムの

モデル案や今後の情報処理教育のための組織についての提言を発表する予定である。最近、情報系専門学科の設置が相次いでおり、それらの学科の参考となることも目指したいと思う。本学会では過去にはカリキュラムなどについての組織的な調査研究は行っていない。教育やカリキュラムについての問題は一意解が存在するようなものではなく、当委員会の報告書以外の考え方があってもよいはずである。当委員会の活動がきっかけとなって、本学会においてカリキュラムなどに関する活発な論議がおこれば望外の喜びである。

謝辞 「大学等における情報処理教育検討委員会」の活動に対して、委員の先生方には多大な協力を賜ったこと、本稿を本誌に執筆することを了承していただいたことに感謝を申しあげる。CS ワーキンググループの牛島和夫主査、木村泉委員、高橋延匡委員、都倉信樹委員、諸橋正幸委員、市川照久委員、榎本彦衛委員、および特別参加の堂下修司正幹事、有澤博委員、大岩元委員（一般教育ワーキンググループ主査）、川合慧委員、藤野喜一委員（IS ワーキンググループ主査）、御牧義委員、村岡洋一委員には、多忙中にもかかわらず泊まり込みを含む検討会で密な討論に何回も加わっていただいた。有山正孝委員には、コンピュータと教育研究会の主査として 1988 年度の調査研究のまとめと、当研究会の活動へのスムーズな移行に、ご協力いただいた。國井利泰委員には IS ワーキンググループの一方の主査をお願いするとともに数々のアイデアをいただいた。菅谷修委員には当委員会の議事録のまとめを分担いただいた。乾侑委員には本問題の提起に関し貴重なご意見をいただいた。文部省高等教育局専門教育課の今野雅裕課長補佐と白間竜一郎専門職員には多数の資料とデータを提供していただいた。以上の方々に重ねて感謝申しあげる。本調査研究は文部省高等教育局専門教育課の研究委嘱によるものであり、一部は文部省科学研究費補助金総合研究(A) 01102034 の援助によるものである。

参 考 文 献

- 1) 情報処理学会コンピュータと教育研究会内情報処理教育の改善のための委員会：昭和 63 年度教育改革の推進に関する研究委託最終事業報告書、1989 年 3 月。
- 2) 御牧 義：大学等の情報処理教育について、情報処理学会コンピュータと教育研究会報告 6-1、(1989)。

- 3) 情報処理学会 大学等における情報処理教育検討委員会：平成元年度教育改革の推進に関する研究委託中間報告書、1990 年 3 月。
- 4) Denning, P. J., Comer, D. E., Gries, D., Mulder, M. C., Tucker, A., Turner, A. J. and Young, P. R.: Computing as a Discipline, Comm. ACM, Vol. 32, No. 1, pp. 9-23 (1989).
- 5) Austing, R., Barnes, B., Bonnette, D., Engel G., Stokes, G. (eds.): CURRICULUM '78, Recommendations for the Undergraduate Program in Computer Science, A Report of the ACM Curriculum Committee on Computer Science, Comm. ACM, Vol. 22, No. 3, pp. 147-166 (1979).
- 6) Curriculum Committee on Computer Science: CURRICULUM '68, Recommendations for Academic Programs in Computer Science, Comm. ACM, Vol. 11, No. 3, pp. 151-197 (1968).
- 7) Koffman, E., Miller, P. and Wardle, C.: Recommended Curriculum for CS 1, 1984—A Report of the ACM Curriculum Committee Task Force for CS 1, Comm. ACM, Vol. 27, No. 10, pp. 998-1001 (1984).
- 8) Koffman, E., Stemple, D. and Wardle, C.: Recommended Curriculum for CS 2, 1984—A Report of the ACM Curriculum Committee Task Force for CS 2, Comm. ACM, Vol. 28, No. 8, pp. 815-818 (1985).
- 9) Ashenurst, R. L. (ed.): Curriculum Recommendations for Graduate Professional Programs in Information Systems, Comm. ACM, Vol. 15, No. 5, pp. 363-398 (1972).
- 10) Couger, J. (ed.): Curriculum Recommendations for Undergraduate Programs in Information Systems, Comm. ACM, Vol. 16, No. 12, pp. 727-749 (1973).
- 11) Nunamaker, J. F. Jr. (ed.): Educational Programs in Information Systems, Comm. ACM, Vol. 24, No. 3, pp. 124-133 (1981).
- 12) Nunamaker, J. F. Jr., Couger, J. D., Davis, G. B. (ed.): Information Systems Curriculum Recommendations for the 80s: Undergraduate and Graduate Programs, Comm. ACM, Vol. 25, No. 11, pp. 781-805 (1982).
- 13) 大野 豊：ソフトウェア工学の背景と展望、情報処理, Vol. 28, No. 7, pp. 845-852 (1987).
- 14) 宮代彰一：自然系実験、放送大学教育振興会 (1985).
- 15) 文部省高等教育局 工学教育の振興に関する調査研究協力者会議：変革期の工学教育、1989 年 12 月。

(平成 2 年 8 月 20 日受付)

訂 正

本誌第31巻10号(1990) pp. 1388, 右段11
行目の武貞照雄は武貞照男の誤りでした。訂正
いたします。