

## 解説



## 情報処理専門教育について

## 理工系情報専門学科における コアカリキュラムについて†

牛島 和夫††

### 1. はじめに

情報系専門学科の新設は、1980年代に入りいったん落ち着いていたが、この3～4年来、急に盛んになってきた。これは社会が情報関連学科の卒業生を求めていることの反映であり当然の現象といっていよい。しかし、そこでの教育を担当できる専門家の数はきわめて限られており、情報系専門学科は作ったけれども、カリキュラムの設計や教員の確保等解決しなければならない多くの問題が残されている。

このような状況の下で、情報処理学会は文部省の委託を受けて「大学等における情報処理教育検討委員会」を設けて「情報処理教育の改善のための調査研究」を1989年度から2カ年にわたって実施した。初年度の活動状況については、すでに情報処理学会誌上で報告されている<sup>1)</sup>。この委員会は、三つの分科会に分かれて活動した。筆者は、情報系専門学科におけるコアカリキュラムを検討するコンピュータサイエンス分科会（以後CS分科会と略す）の主査を務めた。この分科会の結論は「提言」としてまとめられ、親委員会の承認を得て本誌に同時に掲載されている<sup>2)</sup>。本稿は「提言」を補うことを目的として、「提言」には陽に現れていないCS分科会における討論を紹介する。

### 2. 情報系専門学科のアイデンティティ

情報工学・情報科学は情報処理の基礎と応用に関する系統的な学問であり、問題のモデル化・定式化、問題解決の手順（アルゴリズム）、仕掛け

の構造（アーキテクチャ）、システムの構築方法等を不可欠の内容とするものである。これらの基盤は学問としてのコンピュータサイエンス（以後、CSと略記することがある）にある。我が国の情報系専門学科の問題点は専門性についての共通の理解が教員の間で不十分な点にある。原因の一つはCSと情報工学あるいは情報科学との間にある解釈や認識の違いである。

我が国の国立大学理工系学部の情報系専門学科が初めて誕生したのは1970年であった（東工大、電通大、山梨大、京成大、大阪大の5校）。その後、毎年数校ずつ設立されて1980年には30校を超えた。専門学科の誕生と並行して専門学科の必要性や内容、設備等が文部省を中心に検討された。そこで次のように述べている<sup>3)</sup>。

「情報科学はそれ自体、広範な学問であって、そのための専門学科を画一化することは好ましくない。特に大学における専門教育を単一の基準で律することは困難であるから、ここでは、性格の異なる4つの学科を想定して、それぞれに対する平均的なカリキュラムを提案し、講義、演習、実験等の内容について概略を示した。しかし、これらはいくまで平均的は例示であって、標準あるいは基準と解すべきではなく、各大学、学部の事情に則して弾力的に組織すべきである。」

現在の視点でこの文章を読むと情報や計算機を専門とする学科を既存の学科と不協和を起さずに新たに作るための当時の苦心がしのばれる。

ここであげている4つの学科とは、

- (1) 基礎理論を中心とする情報科学科
- (2) ハードウェアを中心とする情報工学科
- (3) ソフトウェアを中心とする情報工学科
- (4) 情報システムを中心とする情報工学科

となっている。それぞれの学科のカリキュラムは、

† Core Curriculum in Computer and Information Sciences in Japan by Kazuo USHJIMA (Department of Computer Science and Communication Engineering, Kyushu University).

†† 九州大学工学部情報工学科

- (1) 情報科学の基礎理論に関する科目群  
 (2) 計算機技術およびハードウェアに関する科目群  
 (3) プログラミングおよびソフトウェアに関する科目群  
 (4) 情報システムおよびシステム設計に関する科目群

をそれぞれの学科が学科と同じ番号の科目群を中心として、他の番号の科目群から適当に選んだ科目を配して編成することを推奨している。上記の引用文とここで推奨されているカリキュラムとを併せて仮に「S 46 推奨」と名づけることにする。

情報を発信しない学問分野はない。コンピュータを使うか使わないかは別にして、多くの学問分野で情報処理技術が役に立つ。しかし、情報処理技術に堪能であることと、情報科学や情報工学の専門家であることは必ずしも同じではない。「S 46 推奨」は、情報科学がそれ自体、広範な学問であると規定することによって、既存の分野との摩擦を避けて情報系専門学科をこの世に認知させるという役割を果たしたが、情報系専門学科のアイデンティティを確立するには必ずしも十分でなかった。

産業界の採用担当者から、情報工学科や情報科学学科を卒業した学生と、電気工学科や電子工学科を卒業した学生とを区別していないというようなことを聞く。しかし、一方、情報系専門学科を卒業した人はやはり違うという評価も聞く。情報処理学会の各種の賞、論文賞、研究賞、奨励賞を受けた人の出身を調べてみると、実は情報系専門学科の卒業生が高い比率を占めている<sup>4)</sup>。これは確かに他の学科との差がトップレベルでははっきり出ていることの証拠である。しかし、全体として差が出ているかという点、産業界がいうとおり情報専門の学問領域を修めてきたと感じられない場合が多い。その原因は、情報系専門学科のアイデンティティ、卒業生の専門能力について共通な尺度を確立していなかったからではないだろうか。

「S 46 推奨」で想定されている4つの学科に共通の科目を表-1に列挙した。これを見ると、たとえば、「データ構造」が共通部分から落ちていることに気づく。情報工学・情報科学の分野の進歩は特に著しいものがある。この検討がなされた

表-1 「S 46 推奨」における共通科目一覧

科目名称	内容
プログラミング言語	プログラムとプログラミング言語、コンパイラ言語、機械語とアセンブリ言語、特殊言語、会話型言語、形式言語理論との関係
システムプログラム	アセンブラ、モニタ、インタプリタ、コンパイラ、OS、同時処理、多重プログラミング、TSS
情報処理	情報処理の概念、情報の構造、通信と情報、記号およびリストの処理、IR、シミュレーション、機械翻訳、パターンの認識と合成、証明、画像および音声情報の処理
計算機システム	計算機の構成、ファイルシステム、端末とデータ伝送、システムの評価、TSS、リアルタイムシステム、ハイブリッドシステム、人間機械系
数値解析	数値計算と計算機四則、基本的数値解析各論、誤差評価、モンテカルロ法
情報数学	集合論、整数論、論理数学、記号論理学、符号理論、符号化理論、組合せ理論、アルゴリズム、帰納的関数
情報基礎論	情報理論、グラフ理論、オートマトン、数理言語理論

注)：学科によっては複数の講義に分けているものもあるが(たとえば、第1、第2等)、この表では一つの科目名で統一している。

のは、今から20年前のことであり、内容に関して言及するつもりはない。問題は、情報と名がついた学科ならば、これだけは習得しておくべき科目、言い替えればコアカリキュラムを明示しなかったことである。情報系専門学科のアイデンティティがはっきりしない一因としてあげることができるのではないだろうか。実際、「S46 推奨」では4つの学科の科目に表-1の共通部分があるということを描しているわけではない。

一方、米国の計算機学会 ACM 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ついて、1968年にカリキュラム68という提案を行った<sup>5)</sup>。これは、電気工学、物理学、統計学といった既存の学問領域に対してコンピュータサイエンスが独立な学問領域であり、独立の学科を作って教えるべき内容をもったものであるということを世の中に初めて宣言したものである。1970年ごろに発足した我が国の各大学情報系専門学科も新しい学科を作るうえで参考にしたと思われる。

1989年7月現在で理工系情報学科協議会に属している国立38学科、私立13学科の学科名を列挙すると、情報工学科24、電子情報工学科7、情報科学科5、電気情報工学科3、知能情報工学科2、情報学類、情報知識工学科、システム工学科、組織工学科、情報通信工学科、制御情報工学科、計算機科学科、情報処理工学科、計算機工学コースそれぞれ1となっており、大部分の学科が情報という名前を使用している。計算機という名前を使っているのは51学科中2学科である。学科の英語名称でみると、Information Science(s)24、Computer Science 7、Information Engineering 6、Electrical & Computer Engineering 3、(以下略)となる<sup>6)</sup>。

一方、北米(アメリカとカナダ)のコンピュータサイエンス127学科とコンピュータエンジニアリング30学科、合計157学科の学科名<sup>7)</sup>では、Computer Science(s)89、Electrical & Computer Engineering 22、Computer Science & Engineering 11、Computer & Information Science(s)10、(以下略)となる。学科名の中に現れるcomputerなどの単語の数を数えてみると、表-2のとおりで、彼我の間における情報と計算機に対する捉え方の違いを表している。

### 3. CSのコアカリキュラム

ACMは、カリキュラム68に対して、継続的に見直しを行って、ちょうど10年後にカリキュラム78を提案している<sup>8)</sup>。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たに加えられた。カリキュラム78では、コアカリキュラムと上級科目、数学科目を分離している点が重要である。

コアカリキュラム 情報系専門学科のアイデンティティを確立し卒業生の専門能力を明確にするためにはコアとなるコンピュータの基本を、論

表-2 学科名称中に含まれるキーワードの比較

	日本	北米
全学科数	51 学科	157 学科
information	31 学科 (61%)	12 学科 (8%)
computer	22 学科 (43%)	150 学科 (96%)
electrical	7 学科 (14%)	33 学科 (20%)

理のレベルからきちんと身につけさせる教育が全ての情報系専門学科の中で行われなければならないという問題意識で、カリキュラム78を当委員会で1989年度に集中的に検討した。このカリキュラムは当時のメインフレームの影響が色濃く残っているが、コアの部分はまだ十分通用する、コアの部分は情報系専門学科で教えるべき最低条件と考えるべきである、カリキュラム78から10年間の学問の進歩を反映させたカリキュラムを各学科の事情を考慮してよく吟味して作るべきであるという暫定的な結論を得た。カリキュラム78のコアカリキュラムは表-3にあげた8科目である。各科目の目的や内容、さらに1989年度の議論については、文献1)、8)を参照されたい。

現行カリキュラムの調査 CSの教育は座学だけでは不十分であり、よく計画された実験・演習と密接に連携を保つ必要がある。CS分科会では、1990年度に多くの情報系専門学科の協力を得て、各学科で実施しているカリキュラムと実験・演習科目の内容説明を提供していただき、「カリキュラム78のコア科目+10年の学問の進歩」を物差しにして、心修科目の内容、実験・演習の内容を中心に、各大学のカリキュラムを分析した。大学ごとに大きな振れがあることが分かった。

すなわち、カリキュラム78をはるかに超える内容のカリキュラムを実施している学科、現在はやや不十分だが将来が期待できる学科がある一方で、電子工学科または数学科にプログラミング教育の味付けをしたにすぎないような学科、重要科目を非常勤講師が担当ししかも集中講義になっている学科、重要科目が必修になっていないため(しかも実験・演習の内容が不十分)CS関係科目を履修せずに卒業が可能な学科等が見受けられた。また、電気、電子、応用物理等の諸学科と

表-3 カリキュラム78におけるコア科目

番号	科目名称
CS 1	プログラミング序論
CS 2	プログラムの設計と実現
CS 3	コンピュータシステム序論
CS 4	コンピュータハードウェア基礎
CS 5	ファイル処理入門
CS 6	オペレーティングシステムとアーキテクチャI
CS 7	データ構造とアルゴリズム
CS 8	プログラミング言語の構造

合併して大学科（たとえば、電子情報工学科）を作ったけれども、カリキュラム編成に無理がかかり、従来の情報工学科に比べて、CS 関係の科目が極端に減ってしまった学科、等もある。

このほか、個別の科目についていえば「情報\* \*学」、「情報\* \*論」のように頭に「情報」を付けて情報系専門学科の講義科目とする、アセンブリ言語によるプログラミング演習を課していない、等がある。また、ソフトウェア開発経験の乏しい学部学生に「ソフトウェア工学」という科目をいきなり教えても教育効果には疑問がある<sup>1)</sup>。この科目をどの学年で行っているかをみれば、カリキュラム設計に関する考え方的一端をうかがうことができる。「ソフトウェア工学」が独自の内容をもつ学問領域に付けられた名称であることを無視して、この科目の内容を単にソフトウェアの工学としている学科もある。

**専門教員の不足** CS カリキュラムの立場からみてこのような不十分な状況は、詰まるところ CS 教育を担当する教員の数の問題にゆきあたる。カリキュラム 78 のコア部分を実施するだけで 6 人の専任教員が必要だとしている<sup>9)</sup>。講義科目については、よい教科書を選定すれば専門家でなくてもある程度は実施できるかも知れない。しかし、CS 関連の実験・演習を計画し学生を指導できるためには CS に関する学問的訓練と経験が必須である。各大学の実験・演習科目の内容を調べてみると、そのような訓練と経験を有する教員がどのくらい含まれているか窺い知ることができる。実験・演習は学生数が効いてくる。プログラミング演習には学生 15~20 人増えるごとに教育助手を 1 名増やす必要がある<sup>9), 10)</sup>。

**計算設備** 実験・演習を実施するうえでもう一方の柱は計算設備（ハードウェア・ソフトウェア）の問題である。陳腐化したハードウェアやソフトウェアは、これを使って問題を解くものの思考方法を制約する。使用設備を常に使える状態に維持し、発展させてゆくには、専門的な知識を必要とする。計算設備は学科のインフラストラクチャでもあり、学科の全ての教職員が計算設備の維持・発展に協力する体制をとらなければならない。学科の中が、ユーザとサーバに分かれるのは好ましくない。現状では学科の中で一般に非専門家（ユーザ）の人数のほうが多く、計算設備の選定にまで

影響を及ぼすことがある。ユーザが選定した計算設備は、CS の専門教育にとって最適ではない。

#### 4. モデルカリキュラム案の検討

カリキュラム 78 のコアカリキュラム構成はよく吟味されたものではあるが、

(1) CS5 の内容は、当時のコンピュータシステムに依存する部分が多く、時代にそぐわない。CS 4 と CS 6 と CS 7 に解体すべきである、

(2) 情報ネットワークおよび我が国の環境でいえば日本語情報処理がまったく扱われていない、という点が 1989 年度の議論で特に指摘された<sup>1)</sup>。

**CS 5 の解体** CS 5 で扱っている話題を大別すると

- a. 記憶装置の物理層に関する話題
- b. オペレーティングシステムの内部構造に関する話題
- c. アルゴリズムとデータ構造に関する一般的な話題
- d. データベースシステムに関する話題
- e. メインフレーム系コンピュータ固有の話題

b は CS 6（オペレーティングシステムとアーキテクチャ）に、c は CS 7（データ構造とアルゴリズム）に含めることができる。a は CS 6 の中でたとえばインプリメントの必要に応じて例示すればよい程度の内容である。d は、CS 7 の中で軽く触れてあとは、CS 11（データベース管理システムの設計）で本格的に扱えばよい。e は、特定のシステムに依存したローカルな話題なので正規の科目の中で表だって取りあげる必要はないものと考えられる。

#### 日本語情報処理と情報ネットワークの扱い

カリキュラム 78 では、自然言語処理は人工知能科目の一部に組み込まれている。米国の環境ではこの科目を受講する以前から、文字列や単語列としての英語文章を演習や実験の中で、あるいはアルゴリズムやデータ構造を説明する教材の一部として扱う機会がきわめて多い。これが、自然言語処理入門の役割を果たしている。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とげ、機械翻訳研究も今や世界の頂点に立っている。日本語情報処理の基礎技術をしっかり身につけさせることは重要である。このために

は特別な科目を設けて系統的に教える方法<sup>1)</sup>と、関連する CS 科目の中で英語の場合と同じように少しずつ教えてゆく方法とがある。英語に比べて文字セットの数が著しく大きいこと、分かち書きされていないこと、等克服すべき問題点も多い。入力手段としてのかな漢字変換、入力文から単語や語尾変化部分を認識する形態素解析プログラムを実習用に準備することができれば、これらのツールを利用して、たとえば、CS 1, CS 2 のプログラミング演習の一部として日本語文字列データの処理アルゴリズムの実習等日本語処理の基礎技術に触れる機会を与えることが考えられる。さらに、オートマトンと形式言語の応用として形態素解析プログラムを実際に作らせることも考えられよう。

一方、ネットワークについての常識がなければ、現代のコンピュータシステムを他人に迷惑をかけずに扱うことは困難である。入門時に演習や実験の中で取りあげておくことが重要である。このほか、最低限の知識として、たとえば現時点では OSI 7 階層モデルについて CS 6 (オペレーティングシステムとアーキテクチャ) の中で言及するのが適当であろう。

**モデルカリキュラム J 90** 当委員会ではモデルカリキュラムを作成するにあたり、特定のタイプの学科をモデルに高度に専門的なカリキュラムを作ることは避け、さまざまな特徴をもつ広範囲の情報系専門学科に共通に使える最小限のカリ

キュラムを示すにとどめることにした。カリキュラム J 90 は表-4 にあげた 15 の授業科目からなる。各科目の内容は、本稿と同時に掲載されている「提言」<sup>2)</sup>を参照していただきたい。このうち、コアカリキュラムは JCS 1 から JCS 7 までの 7 科目である。このコアカリキュラムを履修した学生にさらに進んだ計算機分野の授業科目を履修させるための目安として、8 授業科目を示した。カリキュラム J 90 は教えるべき内容を示したものであり、必修科目数や科目名称等の形式面を規定するものではない。

表-4 のコア科目の名称と表-3 の科目名称とは、字面の上ではほとんど変わっていないので、両者の内容の異同について若干の注釈を述べる。カリキュラム 78 の CS 1 と 2 についてはソフトウェア工学の進展を反映して 84 年に改訂が行われた<sup>9),10)</sup>。JCS 1 と 2 とは、ほぼその内容に対応している。JCS 3 では、実在する(実在した)計算機のアセンブリ言語による演習をとおして学ぶことを強調している。JCS 4 では、ハードウェアレベルでのアルゴリズムの考え方を習得させることを CS 4 に対して追加している。JCS 5 と 6 については、CS 5 の解体の項で議論した。JCS 7 では、CS 8 に加えて言語処理系を作成できる程度の素養をつけることを要求している。

次の上級科目 JCS 8 から JCS 15 までは専門性がやや強く、情報系専門学科の学生はこれらの中から選択的に半分程度を履修し、特に 1 科目を深く履修することが望ましい。しかしながら、情報系専門学科の卒業生であれば、JCS 8 から JCS 15 で取り上げられる内容全般について概括的な知識を有していることが必要である。

上級科目は、ACM 88 年提言<sup>11)</sup>の枠組みを参考にして構成している。この中で、JCS 11 (ヒューマンインタフェース)は、カリキュラム 78 には存在しなかった発展途上の分野である。したがって、大学院の授業科目とするほうが適当な場合もあろう。また、JCS 13 (ソフトウェアの設計と開発)は、いわゆる「ソフトウェア工学」に対応する。大きなプログラムを作成した経験のない学生には理解が困難である。大学院の科目とするほうが適当な場合もあろう。

上記の 15 科目の中には、多くの情報系学科で開講している「離散構造論」や「オートマトンと

表-4 CS カリキュラム J 90 の科目名称

分類	番号	科目名称
コア科目	JCS 1	プログラミング序論
	JCS 2	プログラムの設計と実現
	JCS 3	計算機システム序論
	JCS 4	計算機ハードウェア基礎
	JCS 5	情報構造とアルゴリズム解析
	JCS 6	オペレーティングシステムとアーキテクチャ I
	JCS 7	プログラミング言語の構造
上級科目	JCS 8	オペレーティングシステムとアーキテクチャ II
	JCS 9	ファイルとデータベースシステム
	JCS 10	人工知能
	JCS 11	ヒューマンインタフェース
	JCS 12	計算のモデルとアルゴリズム
	JCS 13	ソフトウェアの設計と開発
	JCS 14	プログラミング言語の理論と実際
	JCS 15	数値計算の理論と実際

形式言語」等の科目が陽にあげられていない。これに対するCS分科会の見解は次のとおりである。

**数学関連科目に関する補足** カリキュラム 78 では数学系列の科目が別に列挙されている(表-5 参照)。MA 4 以外の科目は、教養課程の伝統的な数学科目として、多くの大学で開講されている。伝統的な数学科目であるから、数学の教員が担当して開講している場合が多いだろう。しかし、離散構造論(離散数学と名付けている場合もある)はやや性質が違う。この種の講義を教養課程の科目として開講している大学は少ないだろう。したがって、数学の教員に依頼するのではなく、専門課程で情報関係の教員が講義している例が多そうである。CS 分科会の議論では、この科目を、情報系専門学科の専門科目としてではなく、専門科目を学ぶための基礎科目と位置づけている。モデルカリキュラム J 90 の中に離散構造論が含まれていないのは、この理由であって、教養課程で開講されていない場合には、専門課程の科目として開講する必要がある。

**オートマトンと形式言語** この講義の特徴や利点をあげると次のようになる。

(1) 有限オートマトンの状態遷移図的な捉え方、正規表現、文法的な捉え方、導出木の概念等、情報関係のいろいろな分野で思考の道具として有用である。

(2) コンパイラはソフトウェアの中でも設計理論が整備された典型例である。オートマトンや形式言語の理論は、その基礎理論として縦横に使われる。コンパイラだけではなく、その他のソフトウェアを設計したり問題を把握したりするときにも有効に使えることがある。

(3) 会社に就職してからこのような基礎的な事柄を勉強する機会はほとんどないというのが、卒業生からしばしば聞く。やるなら大学しかない。

表-5 カリキュラム 78 に併記されている数学科目

番号	科目名称
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

学生に問題を解かせる際に、

(1) 正解かそうでないかが明確に分かる。

(2) 論理的な思考力が重要であるが、単に無味乾燥な論理展開ばかりというのではなくアイデアが重みをもっている。問題が明確に設定できるので、学生にとって目標がはっきりしており、解こうという意欲が高まる。

(3) 一般に解は一つしかないというのではなく、いろいろの別解が可能で、アイデアを出す楽しさを増してくれる。

(4) 証明を書くという段階で正確な論証と正確な文章を書くという訓練に適している。

この科目を開設するうえで留意すべき点は次のとおりである。

(1) 内容が抽象的であるため低学年では頭に入らない。オートマトン理論にはほとんど予備知識がいらぬ。だからといって、低学年で講義できるということにはならない。ある程度の成熟が必要である。

(2) 動機付けをきちんとしないと、何の話をしているかさっぱり理解してもらえないことがある。コンパイラとの関連、システム分析や設計との関連、日本語の形態素解析プログラムとの関連等について、教える側が意識していないと学生の側は、定義・証明の続く面倒くさいものと受けとめることになる。理論のための理論に陥ってしまう危険がある。

(3) 論証する訓練になるといいながら、その効果が上がるまで講義することはかなり困難であり、実効が上がりにくい。よい教科書や演習の課題等の開発が必要であろう。

モデルカリキュラムでは、上記の議論を踏まえて、これらの理論が実際に使われる場面との関連から、JCS 7, JCS 12, JCS 14 の中で取り扱っている。各大学の事情に応じて「オートマトン理論」とか「オートマトンと形式言語」という講義を独立して設けることも考えられる。

**実験・演習の位置づけ** 都倉教授による解説<sup>12)</sup>が本号に同時に掲載されているので、参照されたい。

**コアカリキュラムの意味** 「S 46 推奨」における共通科目と、J 90 におけるコア科目との違いを再確認する。前者は、4 者の共通部分をとったものであるのに対して、後者は、コアをまず設

定し、それぞれの大学の事情や学科の設立目的に応じて、コアの回りに置くものを決めてそれぞれの学科の特色を出す。

現行の大学設置基準によれば、卒業要件として4年間で124単位以上を習得することが規定されている。しかし、工学関係学部では実際には140単位程度を卒業要件として課しており、その中で専門教育科目は80単位程度の大学が多い。ここで、コアとして掲げた7科目をいずれも講義2単位分、実験演習1単位分として実施するとすれば、合計21単位となりまだ60単位の余裕がある。上級科目として掲げた8科目のうち4科目を選択するものとして、さらに12単位が加わる。卒業研究に10単位分程度をあてるとしてもまだ40単位分残る。この中には数学や、制御工学、計測工学等工学部の共通的な講義が含まれるかもしれない。たとえば「S46 推奨」に掲げられた4つの学科のカリキュラムを学科の特色を発揮させて余裕をもって設計できるものとする。

本年2月に大学審議会が、大学教育の改善について答申した<sup>19)</sup>。その中に大学設置基準の大綱化がある。一般教育科目と専門教育科目の区別を設けない等、各大学がカリキュラムを自由に設計し得るようにするために基準を大幅に簡素化することをうたっている。正式には設置基準の改正を待たなければならないが、これに従えば、学科の特色を出す余地はさらに広がる。自由設計の余地が広がれば広がるほど、コアの存在意義がますます重要になることは明らかだろう。

## 5. おわりに

CS分科会では、モデルカリキュラムをわずか2年間の検討で作るのは無理との見通しのもとに、コアカリキュラムの存在が重要であるという考えを広めることを第一の使命とした。情報と名のついた学科の学生にはコアカリキュラムを必ず履修させる。これによって情報系専門学科のアイデンティティを確立し、情報専門学科の卒業生の専門能力の核部分を明確にすることができる。コアカリキュラムとしては、「カリキュラム78のコアの部分に10年間の学問の進歩を加味すること」を推奨し、それぞれの学科の設置目的や実状に応じてカリキュラムを設計することを期待した。

1990年度は、カリキュラム設計に当たって参

考になるとされる資料を中心に報告書をまとめることを目標に作業を行ってきた。その結果、

(1) 6つの大学で実際に行われている実験・演習の全体像やテーマ例を集めたこと

(2) カリキュラムを共通の尺度で評価する方法を開発したこと、すなわち ACM 88年提言<sup>11)</sup>を検討しそこで提示されている計算機分野の9副領域と3パラダイム(理論、抽象化、設計)からなる9×3のマトリックスの升目に当該のカリキュラムを投影することにより、その特徴や偏りを評価するもの

(3) 上記の9副領域ごとに、試験問題例を集めたこと

である。これらが最終報告書の大きな部分を占めている。これらの詳細は「提言」<sup>2)</sup>に譲る。

しかし、情報系学科は最近著しく増えており、報告書を読んだからといって、自らカリキュラムを設計開発できるスタッフを揃えているところばかりとは限らないので、検討結果を参考例ということでよから公開すべきであるという文部省専門教育課の勧めもあって、暫定モデルカリキュラム案(IPSJカリキュラムJ90)を提示することになった。暫定と呼ぶとおりの案が最終のものとは考えていない。情報処理学会では、情報処理教育に関する検討委員会を新たに設けて、カリキュラムについてさらに継続的に検討することになっている。

自らカリキュラムを設計開発し実施できるスタッフを擁している学科では、カリキュラムを議論する一つのたたき台として、また、そのようなスタッフが不十分な学科では、カリキュラム設計の一つの出発点として参考にしていただければ幸いである。そこでの検討結果をご批判ご意見として提出していただき、よりよいモデルカリキュラムの確立に寄与できればよいと考えている。目標とするところは、情報系専門学科のアイデンティティを確立し、卒業生の専門能力を明確にすることであり、それによって我が国の情報工学・情報科学分野の学問レベル、技術レベルを向上させるという目的に近づくことである。

**補足** CS分科会の構成は次のとおりである。

主査：牛島和夫(九州大学)、幹事：中森眞理雄(東京農工大学)、委員：榎本彦衛(慶応義塾大学)、木村泉(東京工業大学)、都倉信樹(大阪大

学), 高橋延匡 (東京農工大学), 諸橋正幸 (日本アイ・ビー・エム), オブザーバ: 御牧義 (電気通信大学).

本稿の内容は CS 分科会の討議の一端を紹介したものであるが, 上記メンバによって報告・提唱された個々の事例・提言に関するオリジナリティは各メンバに帰属すべきものである. 文責は筆者にある.

また, 有澤博 (横浜国立大学), 乾侑 (長岡技術科学大学), 大岩元 (豊橋技術科学大学), 川合慧 (東京大学), 堂下修司 (京都大学), 藤野喜一 (日本電気), 村岡洋一 (早稲田大学) の諸先生には当分科会で行った合宿形式の勉強会に参加していただき熱心に討論していただいた.

本研究の一部は文部省科学研究費補助金 (総合研究(A)「大学等の専門学科における情報処理教育カリキュラム実施の実体についての調査研究」)の援助を受けたことを付記する.

### 参考文献

- 1) 野口正一, 中森眞理雄: 大学等における情報処理教育の諸問題, 情報処理, Vol. 31, No. 10, pp. 1373-1389 (1990).
- 2) 野口正一, 牛島和夫, 榎本彦衛, 木村 泉, 高橋延匡, 都倉信樹, 諸橋正幸, 中森眞理雄: 大学等における情報系専門教育の改善への提言, 情報処理, Vol. 32, No. 10, pp. 1079-1092 (1991).
- 3) 情報処理教育に関する会議, 設備部会: 情報処理専門学科のカリキュラム及び設備について, 文部省資料 (昭和 46 年 10 月 7 日).
- 4) 木村 泉: 情報関係専門学科卒業生の貢献度に関する一注意, 平成元年度科学研究補助金・総合研究(A)研究成果報告書「2000 年時点における情報技術者の需要予測モデルの構築等に関する基礎的研究」(代表者・乾 侑), 第 2 部, pp. 39-42(1990).
- 5) CURRICULUM Committee on Computer Science: Curriculum '68, Recommendations for Academic Programs in Computer Science, Comm. ACM, Vol. 11, No. 3, pp. 151-197 (1968).
- 6) 理工系情報学科協議会: 理工系情報学科専攻要覧 (1989 年 7 月).
- 7) Gries, D. and Marsh, D.: The 1987-1988 Tablee Survey Report, Comm. ACM, Vol. 32, No. 10, pp. 1217-1224 (1989).
- 8) Austine, R., Barnes, B., Bonnet, D., Engel, G. and 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).
- 9) Koffman, E., Miller, P. and Wardle, C.: Recommended Curriculum for CS1, 1984—A Report of the ACM Curriculum Committee Task Force for CS1, Comm. ACM, Vol. 27, No. 10, pp. 998-1001 (1984).
- 10) Koffman, E., Stemple, D. and Wardle, C.: Recommended Curriculum for CS2, 1984—A Report of the ACM Curriculum Committee Task Force for CS2, Comm. ACM, Vol. 28, No. 8, pp. 815-818 (1985).
- 11) 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) (木村 泉訳: 学問としての計算機分野, 情報処理, Vol. 31, No. 10, pp. 1351-1372 (1990)).
- 12) 都倉信樹: 情報処理教育における実験・演習, 情報処理, Vol. 32, No. 10, pp. 1101-1108 (1991).
- 13) 大学審議会: 大学教育の改善について, 大学審議会答申 (平成 3 年 2 月 8 日).  
(平成 3 年 5 月 15 日受付)

著者紹介 (p. 1092 参照)