

コンピューティングカリキュラム 2005

概要報告

つぎの学士プログラムを扱う
コンピュータエンジニアリング
コンピュータ科学
情報システム
インフォメーションテクノロジー
ソフトウェアエンジニアリング

コンピューティング カリキュラム シリーズ

コンピューティング カリキュラム 2005 の合同タスクフォース
3 学会共同プロジェクト
ACM(米国計算機学会)
AIS(米国情報システム学会)
IEEE-CS(米国電気電子工学会コンピュータソサエティ)

2005 年 9 月 30 日

Computing Curricula 2005 – The Overview Report

**Copyright © 2006 by ACM and IEEE.
All rights reserved.**

Copyright and Reprint Permissions: Permission is granted to use these curriculum guidelines for the development of educational materials and programs. Other use requires specific permission. Permission requests should be addressed to: ACM Permissions Dept. at permissions@acm.org or to the IEEE Copyrights Manager at copyrights@ieee.org.

ISBN: 1-59593-359-X

ACM Order Number: 999066

IEEE Computer Society Order Number: R0236

Additional copies may be ordered from:

IEEE Computer Society
Customer Service Center
10662 Los Vaqueros Circle
P.O. Box 3014
Los Alamitos, CA 90720-1314

Tel: + 1 800 272 6657
Fax: + 1 714 821 4641
<http://computer.org/cspress>
csbooks@computer.org

IEEE Service Center
445 Hoes Lane
P.O. Box 1331
Piscataway, NJ 08855-1331
Tel: + 1 732 981 0060
Fax: + 1 732 981 9667
<http://shop.ieee.org/store/customer-service@ieee.org>

IEEE Computer Society
Asia/Pacific Office
Watanabe Bldg., 1-4-2
Minami-Aoyama
Minato-ku, Tokyo 107-0062
JAPAN
Tel: + 81 3 3408 3118
Fax: + 81 3 3408 3553
tokyo.ofc@computer.org

ACM Order Department
P.O. Box 11405
New York, NY 10286-1405
1-800-342-6626
1-212-626-0500 (outside
U.S.)
orders@acm.org

Cover art by Robert Vizzini.
Printed in the United States of America

スポンサー

本報告は以下の学会の金銭的支援により実現の運びとなった

ACM**IEEE-CS**

コンピューティングカリキュラム2005 合同タスクフォース

Russell Shackelford : CC2005 タスクフォースの議長。ACM 教育委員会の前議長。スタンフォード大学コンピュータ科学科で副学科長を、ジョージア工科大学コンピューティング学部で学部生指導教官を務めた。CC2001 タスクフォースの共同議長。

James H. Cross II : オーバーン大学で Philpott-Westpoint Stevens Professor とコンピュータ科学およびソフトウェアエンジニアリングの学科長を兼任。IEEE コンピュータ・ソサイエティの教育活動委員会 (EAB) の副会長を務めた。CC2001 タスクフォースのメンバー。

Gordon Davies : 40 年間教鞭を執った後に最近引退。後半の 20 年間は英国のオープンユニバーシティで活躍した。最近では、ACM 職業開発センターの設立に貢献している。現在は、英国コンピュータ学会での認定活動に積極的に関与している。CC2001 タスクフォースのメンバー。

John Impagliazzo : ホフストラ大学コンピュータ科学科の教授。ACM 教育委員会の認定委員会議長を 20 年間努めた。現在は、『Inroads - The SIGCSE Bulletin』の編集長、コンピューティングの歴史に関する IFIP ワーキンググループ 9.7 の議長、IEEE 歴史委員会のメンバーおよび会計係を兼任。CE2004 タスクフォースのメンバー。

Reza Kamali : パデュー大学 (カリユメット, ハモンド, インディアナ) コンピュータ情報システムおよびインフォメーションテクノロジー科学科長と準教授を兼任。SITE (ACM の SIGITE の前身) の設立メンバー。現在、SIGITE の教育担当役員を務める。IT2006 タスクフォースのメンバー。

Richard LeBlanc : ジョージア工科大学コンピューティング学部コンピュータ科学科の教授を最近退任。現在は、サザンカソリックカレッジ学務部副部長。過去には ACM 教育委員会の議長と副議長、IFIP ワーキンググループ 3.2 (大学レベルの情報科学教育) のメンバー、ABET のコンピューティング資格認定委員会チーム議長、工学資格認定委員会のソフトウェアエンジニアリングプログラム評価担当を歴任。SE2004 タスクフォースの共同議長。

Barry Lunt : ブリガムヤング大学インフォメーションテクノロジー科の準教授。SITE (ACM の SIGITE の前身) の設立メンバー。IEEE コンピュータ・ソサイエティ、IEEE 通信ソサイエティ、ASEE のメンバーを兼任。IT2006 タスクフォースの議長。

Andrew McGettrick : ストラスクライド大学 (グラスゴー) コンピュータおよび情報科学科教授、ACM 教育委員会の共同議長、英国コンピュータ学会副議長を兼任。最近、英国でコンピューティングに関する学士および修士の学士プログラムのベンチマークを開発したグループの議長を務めた。CC2001 タスクフォース、CE2004 タスクフォース、SE2004 タスクフォースのメンバー。

Robert Sloan : イリノイ大学 (シカゴ) コンピュータ科学科の準教授。IEEE コンピュータ・ソサイエティの教育活動委員会のメンバーでもある。CC2001 タスクフォースと CE2004 タスクフォースのメンバー。

Heikki Topi : ベントレーカレッジ (マサチューセッツ州ウォルサム) コンピュータ情報システム科の準教授と CIS 科の学科長を兼任。米国情報システム学会 (AIS) のメンバーであり、北米 IS コミュニティでカリキュラム開発と資格認定活動に携わっている。IS2002 タスクフォースのメンバー。

要旨

コンピューティングは、科学、工学技術、ビジネス、そして人類が試みる他の多くの分野の進歩に劇的な影響を及ぼしてきた。今日の世界では、ほとんどすべての人がコンピュータを使用する必要があり、多くの人々が何らかの形でコンピューティングを学びたいと考えている。コンピューティングは、継続的にやり甲斐のあるキャリアのチャンスを与え続けてくれるであろう。そして、コンピューティングに働く人が、未来を形作るのに重要な役割を果たすであろう。

重要なことは、コンピューティング領域が幅広い層の優秀な学生を魅了し続けることであり、有能かつ信頼できるプロフェッショナル・科学者・技術者へと学生を育成することである。カリキュラムに関する指針の作成等、米国基盤のプロフェッショナルの協会・学会は、長年にわたって高等教育に対してさまざまな支援を行ってきた。カリキュラム指針の作成もそうした支援の一つである。コンピューティングカリキュラムの指針を定めた報告、そしてその改訂報告が、過去40年にわたって出されてきた。最近の活動は、国際的に共同に重きが置かれている。これは、基幹となるプロフェッショナル組織は、その範囲においても責任の面においても真にグローバルになる必要があるからである。

コンピューティングカリキュラム2001(CC2001)作成の初期の段階で判明したのは、1990年代にコンピューティングが劇的に拡大したことによって、以前から報告が作られていた領域に対してだけカリキュラム報告の改訂版を作成するというのでは不十分であるということであった。CC2001は、増大したコンピューティング領域を覆う一連の報告を用意するとして、まずは、コンピュータ科学(computer science)、情報システム(information systems)、コンピュータエンジニアリング(computer engineering)、ソフトウェアエンジニアリング(software engineering)に対して独立の巻を用意するとした。また、新しいコンピューティング領域がいずれ生まれるだろうということは明らかであった。CC2001の発行後、インフォメーションテクノロジー(information technology)がコンピューティングの一領域として加わり、現在そのカリキュラム報告の巻も必要となっている。

報告CC2001は、また、それぞれの領域固有の報告内容を要約した概要報告の作成を提唱していた。この文書は、その概要報告の第一版である。その目的は、大学関係者に俯瞰を与えて、主要なコンピューティング領域それぞれが何であるか、それぞれの学士プログラムが互いにどう違いどう補完しあっているのかを理解する手助けをすることである。この報告では、各々の主要なコンピューティング領域に対して学士プログラムに対する知識体系を総括し、それらの共通点および相違点を浮き彫りして、それぞれの学士プログラムの卒業生の業務遂行能力について説明する。この報告の作成に当たっては、それぞれの学部教育カリキュラム指針を調査し、必要に応じてコンピューティング専門家の意見を聞き、その他必要な情報を参考にした。この報告では、大学院教育やコンピューティング研究の位置づけについては焦点を当てていない。この報告は、大学レベルの教員・経営管理者を対象としており、彼らが学士プログラムを整えて大学構成員にとっても社会にとって有用なものにしようとする際に直面する問題や課題を概説するものである。

概要報告の発行に続き、より簡便な連携報告コンピューティング学士プログラム案内が合同タスクフォースによって発行される予定である。この案内は、より幅広い読者を手引きするもので、たとえば、入学志望者、その両親や生徒指導員など、高校から大学へ進む生徒が直面する選択について気を配る理由がある人などが対象となる。ここでは、それぞれのコンピューティング領域の特徴を手短に説明し、生徒がコンピューティングの学習領域を選択する際に考慮することになる事柄を概観する。この案内は、独立した文書として広く配布される予定である。

この報告は、基幹のコンピュータ学協会の間での、そして主要なコンピューティング領域の間での、前例のない共同作業によって生まれた。それは、コンピューティングカリキュラムシリーズの5つの領域別報告巻を調査し分析して作られた。これらの文書のほとんどが、米国・カナダの高等教育を指向したものであることから、この報告も潜在的に北米中心主義のものとなっている。タスクフォースは、将来、これらすべての巻の改版がより国際的な視点をもって発行されることを望んでいる。それまでの間、他国の人にとっては、現時点でのそれぞれの報告巻をどのように使用することがよいかを知るのに、この報告が助けとなる。コンピューティングでは物事が急速に変化してゆくから、どの報告も頻繁に改版される必要がある。この報告の最新版、およびコンピューティングカリキュラムに関する他の報告の電子版は、つぎのURLによって入手可能である。

<http://www.acm.org/education/curricula.html>

<http://computer.org/curriculum>

第1章 序文

1.1 この報告の目的

この報告は、カリキュラム標準が現在利用可能、あるいは、間もなく利用可能となる、現行のさまざまなコンピューティングの学士プログラムの概要を説明する。教員、管理経営者にとっても、学生、両親にとっても、この報告は必要なものである。コンピューティングが、数学、科学、工学、経営の境界をまたがる広範囲にわたる分野であり、また、プロフェッショナルの実務能力の基礎となる重要な能力を包含しているからである。コンピューティングには、複数の領域があり、多くの名高い大学では、コンピュータ科学、コンピュータエンジニアリング、情報システム、インフォメーションテクノロジー、ソフトウェアエンジニアリング等、いくつもの領域に対して学士プログラムを提供している。これらのコンピューティング領域は、相互に関連しているものの、全く異なるものである。コンピューティングのさまざまな学士プログラムがあることで、学生、教育者、管理者、他の地域指導者は、注力したいことに応じて選択することが可能になっている。

自然といくつかの疑問がわきあがってくる。これらコンピューティングの異なる学士プログラムはそれぞれ何なのか。それらはどれほど類似しているのか。どう違っているのか。その名称が何を意味するのかはしたらわかるのか。この地域の大学ではどのようなプログラムを提供すべきか。以上はすべて妥当な疑問である。しかし、コンピューティングの範囲の広さを知らない人にとって、これらの質問に対する回答を作るのは難しい。この報告は、その答えを得る手助けとなるかもしれない。

この報告は、コンピューティングのさまざまな学士プログラムの特徴を説明して、特定の目的や状況に最も適したプログラムを決める際の助けとなることを目的としている。この報告は、より幅広いさまざまな読者の役に立つよう意図しており、つぎの人たちにとって有用であろう。

- 大学において、コンピューティング関連プログラムの立案およびカリキュラム作成を担当している教職員・管理者、また、そのようなプログラムの認定を指導する立場の人
- 公教育に責任を有する関係者、すなわち、教育委員会委員、官僚、代議士、あるいは公共の利益を代表しようとする人たちなど。

さらに、我々は、間もなくコンピューティング学士プログラムに関する案内（以下、案内という）に取り掛かる予定である。案内は独立した文書として一般の人を対象とし、広く配布されるであろう。案内はつぎのことがらを対象とする。

- どの分野のコンピューティング教育が自分の興味や目標に合うかを決めようとしている学生
- 親、教師、生徒指導員等、学生の意思決定を支援しようとしている関係者
- 急速に変化し活力に満ちた現場で、どのように教育を続けていくかを検討している専門家
- 現在利用可能な、学部生向けのコンピューティングの学士プログラムがなぜこんなにも多様なのか理解しようとしている人

1.2 この報告の範囲

コンピューティングに関する学士プログラムにはさまざまな種類がある。異なる種類のコンピューティング学士プログラムの数について、信頼に足る情報を入手することは困難であるが、学生が利用可能なコンピューティング学士プログラムの種類は、ここ10年ほどで劇的に増加している。すべてを列挙し、ひとつひとつについて特徴づけることは我々の目的ではなく、その素養もない。本レポートでは、今日重要な5つの領域、コンピュータエンジニアリング(CE)、コンピュータ科学(CS)、情報システム(IS)、インフォメーションテクノロジー(IT)およびソフトウェアエンジニアリング(SE)に焦点を当てる。これら5つの領域は、この報告に取り上げる規準を満たすものである。すなわち、その学部カリキュラム指針の最新版が、国際的なプロフェッショナル協会または学会の一つまたは複数によって出版されているか、出版予定となっている。これら5つの領域は、また、コンピューティングを専攻する米国の学部生の圧倒的多数を引き付けている領域でもある。

将来的に追加されるコンピューティング領域も、この規準を満たすことを期待している。そして、その場合に限って、その領域はこの報告の改訂版に含まれることになるだろう。将来の改版時に候補となる領域は、指針がまだ存在していない新しい領域(例えば、生物情報科学)であるかもしれないし、指針が存在してはいないものの確立された領域(例えば、コンピュータエンジニアリング技術)であるかもしれない。

この報告は、上述の5つのコンピューティング関連領域の学部学士プログラムに対して存在している一連のカリキュラム標準を基にしている。5つの領域別カリキュラム報告は、それぞれ関連するプロフェッショナル団体・学術団体(学会)・教育関連団体の最善の判断を提示したものであり、これらの学士プログラムがどのようにあるべきか、どのように行われるべきかを定義している。

これらの報告の何点かは、改版が予定されている。しかし、タスクフォースは、その内容を更新することにはいっさい踏み込まなかった。実際、それは、タスクフォースの使命外のことであり、権限外のことであるからである。タスクフォースは、5領域の既存のカリキュラム報告に書かれていることをそのまま受け止め、その内容を相互に比較し、各領域の記述として重要な情報、相互比較するに重要な情報を組上げた。カリキュラムの全5巻には、この報告に取り上げていない詳細な情報が大量に含まれている。取り上げた5つの領域のいずれかについて、より詳細な情報を望む読者は、原書を参照されたい。コンピューティングカリキュラムの各報告は、つぎのURLによって入手可能である。

<http://www.acm.org/education/curricula.html>

<http://computer.org/curriculum>

この報告の基盤としてこれら5つの報告を使用するにあたっては、必要に応じて、コンピューティングのプロフェッショナルの意見や、その他の必要な情報を参照した。他の領域のコンピューティング学士プログラム、大学院のコンピューティング教育、コンピューティングの研究団体には焦点を合わせていない。また、ベンダ固有の認定プログラムに連動して提供されているような、非伝統的なコンピューティング教育に関する情報やコメントも含んでいない。このような領域も査定されるべきではあるが、このプロジェクトの範囲外である。

このレポートの残りの部分にはつぎのものが含まれる。

- 第2章では、コンピューティングの5つ主要な領域を特徴付ける。
- 第3章では、5種類の学士プログラムそれぞれの特徴を具体的に説明し、相互に比較する。また、各学士プログラムの卒業生に備わると目される専門的能力の種類を比較し、対比する。
- 第4章では、結論として、教育者、管理者その他、責任ある人に対して、コンピューティングの新しい領域を創生する過程で出現しうる課題に対して注意を喚起する。
- 第5章では、5つの領域それぞれのカリキュラム報告をオンラインで取得する方法を述べる。また、その使用方法を説明する。

この報告の発行に続いて、タスクフォースは、コンピューティング学士プログラム案内の作成に取りかかる。この案内は、概要報告よりも薄手の独立文書であり、広範囲に配布することを意図している。案内は、優秀な学生とその助言者に十分な情報を提供して、存分の判断が下せるようにする。

コンピューティングは、それ自身発展し続けるであろう。それに加え、新規のコンピューティング関連領域が出現すると考えられる。既存の領域別報告は更新され、新しいコンピューティング領域に関しても報告が出るだろうから、いずれこの報告も改訂版が作られるだろう。手にしている報告（CC2005 概要報告）がコンピューティングカリキュラムに関する概要報告の最新版であるかどうかを確認するには、つぎのところにアクセスされたい。

<http://www/education/curricula.html>

<http://computer.org/curriculum>

より新しい版が存在するかどうかは、これらいずれのサイトにおいても確認することができる。もしより新しい版が存在すれば、最新版をいずれのサイトからでもダウンロード可能である。

1.3 背景

ここ 40 年の間、米国の 4 つの主要な組織が、専門学校や大学におけるコンピューティング カリキュラムの指針を作成してきた。

- 米国計算機学会 (Association For Computing Machinery ; 一般に ACM または Association For Computing と呼ばれる) は、1947 年に設立された学術的なプロフェッショナル団体である。コンピューティングに関わるすべての側面に関する新しい知識の発展、共有に携わっている(名称に含まれる計算機という言葉は歴史的な産物である)。伝統的に、コンピュータの新しい使用法を考案したり、演算処理やそれを可能とするソフトウェアの根底にある科学と理論を進歩させてきたコンピュータ科学者のプロフェッショナルとしての本拠地となっている。ACM は CS に対するカリキュラム勧告の発行を 1968 年に (仮版は 1965 年に) 開始し、IS に対するカリキュラム勧告の発行を 1972 年に開始した。
- 米国情報システム学会 (Association For Information System ; 一般に AIS と呼ばれる) は、1994 年に設立された、情報システムを専門とする学者の国際的な組織である。AIS の学会会員のほとんどが、ビジネス・経営の学校関係者である。AIS は IS に対するカリキュラム勧告を、ACM および AITP と共同で 1997 年に発行し始めた。
- 米国インフォメーションテクノロジープロフェッショナル協会 (Association For Information Technology Professionals ; 一般に AITP と呼ばれる) は、1951 年に全米機械会計士協会 (National Machine Accountants Association) として設立された。1962 年にデータ処理管理学会 (Data Processing Management Association ; DPMA) となった。1996 年に現在の名称となった。AITP はコンピューティングの専門的な側面に焦点を当て、企業やその他の組織におけるニーズを満たすための、コンピューティング技術の利用者サイドの組織である。1995 年に最初に IS に関するカリキュラムを勧告した学会である。
- 米国電気電子工学会コンピュータソサエティ (IEEE-CS または Computer Society と表記されることが多い) は、1946 年設立のアメリカ電気学会 (AIEE) の大型計算装置部会、および 1951 年設立の無線技術者協会 (IRE) を起源とする。AIEE と IRE は 1964 年に合併し、IEEE となり、2 つのサブユニットが結合してコンピュータソサエティとなった。コンピュータソサエティは、エンジニアリング的観点からコンピューティングに焦点を当てた、IEEE 内部の技術ソサエティである。今日、コンピュータソサエティのメンバーにはコンピュータ技術者 (engineer)、ソフトウェア技術者、コンピュータ技師 (technologist)、コンピュータ科学者が含まれる。コンピュータソサエティは、カリキュラム勧告の発行を 1977 年に開始した。近年では、ACM とコンピュータソサエティのメンバーは大きく重複している。

1990 年代以前は、各学会は各々のカリキュラム勧告を作成していた。やがて、共同作業の利点が明らかになってきた。今日では、各学会はカリキュラム標準の策定において協力しあって、コンピューティングの社会に単一のメッセージを発信している。多くの研究者や教職者が一つ以上の学会に所属している。

ACM 及び IEEE-CS は、1980 年代後半になって共同してコンピューティングに関するカリキュラム報告を作成し始めた。1991 年に発行されたコンピューティング カリキュラム 1991(CC'91)は、コンピュータ科学及びコンピュータエンジニアリングに対する、4 年制学士プログラムのカリキュラムの指針を示した。1990 年代を通して、コンピューティングの教育におけるその他のプログラムに向けてカリキュラム指針を作成するためのさまざまな取り組みが行われた。ACM は、1993 年までに 2 年制準学士プログラムに対し合計 5 つの報告をまとめた。コンピュータ科学、コンピュータエンジニアリング技術、情報システム、コンピュータサポートサービス、他分野でのコンピューティングの 5 つである。また、1993 年には、高校カリキュラム勧告を作成した。1997 年には ACM,AIM および AITP (AIS) が共同して情報システムにおけるカリキュラムモデルおよび 4 年制学士プログラムに対する指針を発行した。1990 年代にはまた、ソフトウェアエンジニアリングなど、米国で重要性を増す新しいコンピューティング領域が現れた。

1990 年代末には、コンピューティング各領域が急速に成長しているのみならず、さまざまな方向に成長していることが明白になった。コンピューティングでの学士プログラムの種類が増大して、多くの人々を惑わせた。コンピューティング学士プログラムの種類の増え方は、惑うのも当然といえるほどの勢いであった。コンピューティングの学士号が多様である問題は、1990 年代にコンピューティングの影響が急増する以前には顕在していなかった。新しい問題であるがゆえに、突然にそこかしこに現れたような選択肢を調整し単純化するための確立された方法がなかったのである。

1990 年代後半に ACM と IEEE-CS が再び共同して CC'91 を最新のカリキュラム報告に更新しようとした時には、もはや問題を看過するわけにいかなくなっていた。最初は、2 学会で共同してタスクフォースを作って CC'91 報告を更新する作業を行う予定であった。ACM および IEEE-CS は、共同タスクフォースを作り、コンピューティング カリキュラム 2001 という、さまざまなコンピューティング学士プログラムに対するカリキュラム指針を提示する単一の報告を作成することを目指した。しかしながら、タスクフォースのメンバーは即座に新しい現実を認識した。コンピューティングがさまざまな方向に成長してしまったため、この分野に対する単一の視点などは作りえないという現実である。コンピューティング分野がコンピュータ科学、コンピュータエンジニアリング、情報システムの 3 つだけで構成されていた時代は終わったのだ。多数のコンピューティング領域が織りなす豊かさ・幅広さに呼応して、コンピューティング カリキュラムがどうあるべきかを定めるのに新しい方法が必要となったのである。

CC2001 タスクフォースは、つぎの4つの重要な決断を下してこの課題に対応した。

1. コンピュータエンジニアリング、コンピュータ科学、情報システム、ソフトウェアエンジニアリングを含む、各主要なコンピューティング領域に対してカリキュラム報告(巻)がなければならない。
2. コンピューティング関連領域の数は増えそうである。カリキュラム報告の構成は、その時点で確立していた4つのコンピューティング領域のみならず、生まれてくる新しい領域に対応できるものでなければならない。
3. 増加の一途をたどるコンピューティング領域の数は、当然、混乱の原因となってしまう。そこで、それぞれの領域別の巻に加え、概要報告を用意してそれら学科別巻の実用的・包括的な手引きとする。
4. コンピューティングにおける変化が十分に速いので、学協会においては10年に1度といわずもっと頻度高くカリキュラム ガイドラインを見直すプロセスを確立しておかなければならない。

当時のタスクフォースは、そのメンバーが基本的にコンピュータ科学者であり、コンピュータ科学に限って報告作成の資格があるのだと認識していた。そこで、ACM, IEEE-CS, AIS およびその他の専門学協会に呼びかけて、コンピュータエンジニアリング、情報システム、ソフトウェアエンジニアリングに関する巻を作成する作業に合同で当たることとした。このタスクフォースによる作業結果は、コンピューティング カリキュラム 2001 (CC2001) として2001年12月に発行された。CC2001 報告は、つぎの2つの重要な要素を含んでいる。

- 上述のタスクフォースによる決定に従ったコンピューティング カリキュラム指針の新しい構造(以降、CC2001 モデルと呼ばれている)
- コンピュータ科学に関する学士プログラムに対する詳細なカリキュラム指針

CC2001 報告は、CS カリキュラム指針を包含しているので、コンピュータ科学の内容を指して参照する人はこれをCS2001 と見るかもしれない。CC2005 レポートの発行以降、“コンピューティング カリキュラム 20xx” という表題は、概要報告に限って用いられる。CS カリキュラム報告の新版は“コンピュータ科学 20xx” と呼ばれる。いずれにしても、“20xx” は発行年を示す。

CC2001 モデルに対応して、領域別分冊の作業も追って開始された。

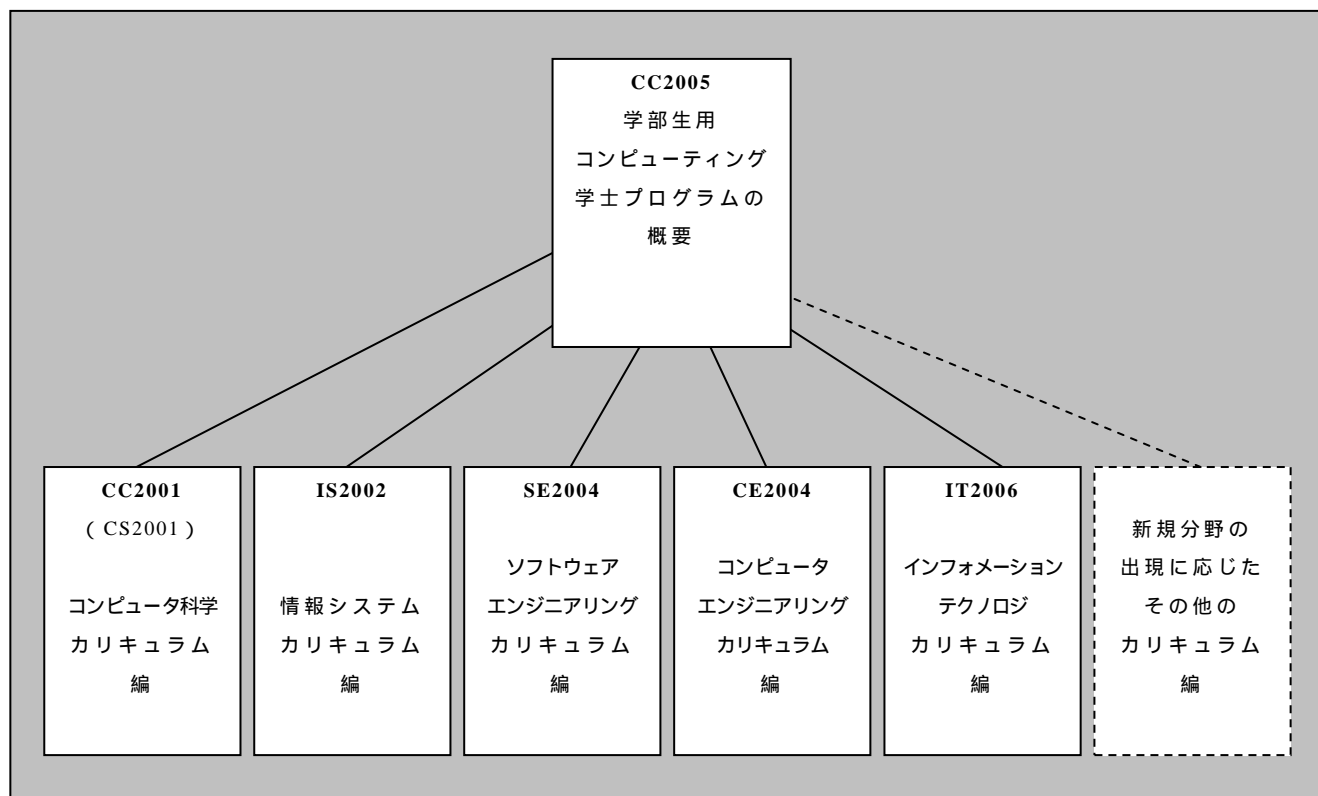
- 情報システム関係者が改版 IS2002 報告を2002年に発行した (IS2002)。
- ソフトウェアエンジニアリング関係者が SE2004 レポート初版を2004年に発行した (SE2004)。
- コンピュータエンジニアリング関係者が CE2004 レポートを2004年に発行した (CE2004)。
- CC2001 が新しいコンピューティング領域が生まれていくと予測したのは正しかった。インフォメーションテクノロジーに対して、その学士プログラムに関する報告が作成作業中である。2006年には発行され、IT2006 と呼ばれることになるだろう。

コンピューティング カリキュラムの指針および標準を提供する継続的な取組みは、コンピューティング カリキュラム シリーズと呼ばれることになった。図 1.1 は、その活動範囲を示している。最上部の CC2005 と書かれた枠はこの概要報告を表している。下部にある左の 5 つは既存のコンピューティング領域のカリキュラム報告を表している。6 つ目は、将来発行される報告が入る枠であり、新しいコンピューティング領域の出現に伴い必要となったときに埋められることになる。コンピューティング カリキュラムの各巻のオンラインコピーは、つぎのところにある。

<http://www.acm.org/education/curricula.html>

<http://computer.org/curriculum>

図 1.1 コンピューティングカリキュラムシリーズの構造



1.4 ガイドラインの原則

このレポートの作成にあたっては、つぎの5原則を採用した。

1. コンピューティング領域の数が劇的に増加し、その総体として社会に影響を与えているから、これらのコンピューティング領域総体としてのアイデンティティを明確にする必要がある。社会に対するコンピューティングの重要性に鑑みて、コンピューティングに携わっている者が何を行っているのかを、社会が理解できるように手助けする責任が我々にはある。コンピューティングに数種類の大学教育プログラムが存在するという事実は、コンピューティング分野の大きな強みであり、機会でもあるが、同時に、我々が、これらの領域に共通するものがないか、その中のそれぞれの領域がどんなものであるのか、また学生や先生そして社会が直面している選択の意味するものが何であるかに関して、実際的なビジョンを提示する責任を負っていることにもなる。この報告の目的は、領域に共通するアイデンティティを明らかにし、各コンピューティング領域独自のアイデンティティを明らかにし、そして学生、先生および社会に用意されている選択肢を明らかにすることである。
2. それぞれのコンピューティング領域に関与してもらって、これらのアイデンティティおよび選択肢をこの報告で明確に定義する。各コンピューティング領域は、自らのアイデンティティを明確に記し、他の領域のアイデンティティを認知し、コンピューティングに共通するアイデンティティを担わなければならない。
3. この報告は、幅広い読者を対象とする。技術的背景をもつ人だけを対象とするのではない。すでに1.1で議論したとおり、この報告の読者は、大学でのコンピューティング学士プログラムについての知りたい何らかの理由をもつ人たちである。読者の大部分はコンピューティングの教育者ではない。我々の目的は、簡潔且つ有用な図の形で、学生自身や学生の教育に関する選択に責任ある人々が直面している選択肢を照らし出して見せることである。この目的は、学士プログラムのカリキュラム指針を定義する報告の目的とは根本的に異なる。したがって、この報告は、より簡潔のものでなければならず、できる限り技術専門用語を使わないようにしなければならない。技術的な背景をもつ読者には、技術者向けの文書に見られるような区別や技術的な強調を避けている点にご理解をいただきたい。
4. コンピューティング領域の学部レベルの教育を特徴付けるに当たっては、それぞれの領域の最新のカリキュラム報告にある知識・スキルの体系を参照する。コンピューティング領域に共通する特徴を定義することは、これまでに例を見ないが、実現しなければならない課題である。タスクフォースは、それぞれの領域で発行されているカリキュラム報告に出ている知識・スキルの体系に注目した。その教育論や教科・教程には踏み込まなかった。教育論やコンピューティング科目の設定などは、コンピューティング領域をまたがって多くの聴衆の役に立つだろうし、時宜を得たものでもあるかもしれない。しかし、われわれタスクフォースは、これらの事柄をこの報告で扱うのに適してはいないと判断した。このことは今後の前例とするべきものではなく、以後の報告の執筆者にはこの件を再検討して欲しい。
5. この報告は、単なる詳細調査を超えて、対象とする読者にとって有用なまとめを提供するものでなければならない。この報告に記載することがらは、現時点での各領域別カリキュラム巻に含まれる知識体系を精査した結果に基づいているが、たんなる調査報告の域を超えて読者にとって意味深く有用な形にまとめることに努めた。その作業を行うのには、各領域の代表者に判断を下してもらいながら、全領域にわたる見通しの効いた概要を納得づくで作成していった。

第2章 コンピューティングの領域

世界中に、コンピューティングに関するさまざまな学士プログラムが存在する。さらに、これらのプログラムには、さまざまな名称がつけられている。これらは、コンピュータ関連の領域に対応する。この報告では、現時点で主要な5つの学科に焦点を当てる。コンピュータエンジニアリング、コンピュータ科学、情報システム、インフォメーションテクノロジー、ソフトウェアエンジニアリングの5つである。これら5つを選んだ基準は、ひとつ以上の国際的なプロフェッショナル団体または学術団体・学会が承認した学部カリキュラムの指針が発行されているか、近く発行されることになっている、というものである。将来的には、他のコンピューティング領域もこの基準を満たすかもしれない。その暁には、この基準を満たした新たな領域を、この報告の改訂版に含めることになるだろう。

2.1 コンピューティングとは何か?

一般的には、コンピューティングは“コンピュータを必要とする活動、コンピュータからの恩恵を受ける活動、またはコンピュータを作成する活動”と定義することができる。よって、コンピューティングには、ハードウェアおよびソフトウェアのシステムの設計と製造が含まれることになる。そのシステムの用途も広範囲であり、種々の情報の処理・構造化・管理、コンピュータを使用した科学研究、知的に振る舞うコンピュータシステムの構築、通信メディアやエンタテインメントメディアの創作・利用、特定目的のための情報検索・情報収集などが含まれる。このリストは事実上終わりのないものであり、可能性はほぼ無限である。コンピューティングは、その用語が使用される文脈次第で、もっと具体的な別の意味をもつこともある。例えば、情報システムの専門家は、コンピューティングという言葉に、ソフトウェア技術者とは若干異なった意味をみている。文脈はどうあれ、コンピューティングをうまく行うのは、複雑で困難なものになることがある。社会は、人々がコンピューティングをうまく行うことを必要としており、そのためコンピューティングを専門職としてだけでなく、ひとつの学習領域としても考えなくてはならないのである。

コンピューティング業界に入るための準備として、学生は一般的に、コンピューティング領域のひとつで学士号を取得する。コンピューティングには、前述したように多くの選択肢があるため、それら領域のすべてに熟達することは不可能である。ゆえに、コンピューティングの専門家になることを望む個人は、自分の職業を絞り込む必要がある。コンピューティングには、5つの主要な学士プログラムあり、それぞれがコンピューティングに異なる焦点を当て独自の観点をもっている。2.2では、これら5つのコンピューティング領域について説明し、各領域が焦点を当てているものや解決対象としているものなどを比較する。

2.2 コンピューティング学科の様相

コンピューティングとは、単一の分野ではなく、関連する領域の総称である。1990年代に、コンピューティング・通信技術およびその社会へ与える影響に重要な変化が起こり、これら一群の領域に変化をもたらした。

2.2.1 1980 年代以前

コンピューティング関連領域の学士プログラムが現れたのは 1960 年代である。当初は、わずか 3 種類のコンピューティング関連の学士プログラムが北米にあった。コンピュータ科学、電気工学、情報システムの 3 つである。これらの学科は、それぞれ独自に、明確に定義されたコンピューティング領域にかかわっていた。これらは、コンピューティングの重要な領域であり、それぞれが独自の研究範囲・勢力範囲をもっていたから、学生が学士プログラムを選択するのは容易であった。ソフトウェア開発の専門家を目指す学生や、コンピューティングの理論の専門家を目指す学生は、コンピュータ科学がその明確な選択肢であった。ハードウェアを扱う仕事をしたい学生は、電気工学がその明確な選択肢だった。ビジネス上の問題を解決するためにハードウェアおよびソフトウェアを使いたい学生には、情報システムが正しい選択であった。

これら 3 つの領域のどれもが、独自の専門領域をもっていた。しかし、コンピューティング分野を構成するものであるという共通意識はなかった。実際のところでは、コンピュータ科学者と電気工学者は、共に新しい技術の開発に携わっていたから、協力して作業にあたることがあったし、大学の同一区画を占めていることもしばしばだった。また、彼らは時々、互いの助けを必要ともした。情報システムの専門家は、ビジネス スクールと関係をもっていたが、コンピュータ科学者や電子工学者との交流はあまりなかった。

1990 年代より前にこの眺望に米国内で生じた主な変化といえば、コンピュータエンジニアリングの発展だけであった。マイクロチップを用いたマイクロ プロセッサの発明以前は、コンピュータエンジニアリングとは、電気工学内の特殊化した領域のひとつであった。1970 年代中期のマイクロ プロセッサの登場に伴って、コンピュータエンジニアリングは電気工学から抜け出し、それ自体がひとつの学問領域となり始めたのである。しかしながら、工学業界にかかわりの無い人々にとっては、電気工学とコンピュータエンジニアリングの違いは明確ではなかった。したがって、1990 年代より前に生徒がコンピューティング関連の学士プログラムを選択しようとして調べたら、図 2.1 の上半分のように見えたことであろう。この図で、各領域の間の距離は、ある領域の人が他の領域の人とどれくらい密接に働いているかを示している。

2.2.2 1990 年代における重要な発展

1990 年代には、いくつもの発展があり、北米でのコンピューティング領域の眺望がすっかり変わってしまった。世界のいくつかの地域では、これらの変化はもっと以前に起きていた。

- コンピュータエンジニアリングは、電気工学からの分離を確固たるものにした。コンピュータエンジニアリングは、1970 年代後期から 1980 年代にかけて電気工学から抜け出したものであるが、コンピュータ チップがほとんどの電気装置や多くの機械装置の主要な構成部品となるのは、1990 年代に入ってからである。(例えば、最新の自動車は多数のコンピュータを実装しており、それらは運転者が気づかないところで業務をこなしている。)コンピュータエンジニアは、チップの設計とそのプログラム作りを行って、多くの装置のデジタル御を可能にしている。マイクロチップを用いたデジタル論理に依存する装置の種類が飛躍的に増えたことが、コンピュータエンジニアリングがその強固な地位を確立することに一役かった。1990 年代には、かつてないほどの人数の学生がコンピュータエンジニアリングプログラムに応募した。北米以外の地域では、こうしたプログラムの名称がコンピュータ システム工学などとなっていることがある。

- コンピュータ科学は急速に成長し、学問の1分野として受け入れられるようになった。米国のほとんどの専門大学や総合大学でコンピュータ科学がひとつの学科として始めて登場したのは、1970年代のことである。当初は、コンピュータ科学が正当な学問の1分野であるかどうかについて、かなりの論争が展開された。支持者側は、それが独自性を持つ正当な学問分野であると強く主張したし、批判する側は、それは技術者のための職業上の専攻であるとか、数学者のための研究用のプラットフォーム、またはコンピュータプログラムのための擬似的な教科であるとして却下した。1990年代までには、コンピュータ科学が理論から実践にいたる相当の研究、知識、革新を生み出したことで、この正当性に関する論争は幕を閉じた。また、1990年代には、コンピュータ科学領域は未曾有の需要に直面することとなった。コンピュータ科学の学士の資格を持つ卒業生への産業界のニーズは、その供給を何倍も上回るものであった。コンピュータ科学プログラムへの入学者数は飛躍的に増大した。コンピュータ科学は、その短い歴史の中で既に入学者数の増減を経験していたものの、1990年代における入学者数の急増は、コンピュータ科学領域の学科に、その大量の学生に対処できるように強い圧力を与えるほどの規模であった。教育・研究の要求の高まりに伴って、多くの大学でコンピュータ科学の教員数が大幅に増えることとなった。
- ソフトウェアエンジニアリングは、コンピュータ科学のひとつの領域として出現したものである。コンピューティングがより広範囲の複雑な問題に取り組むのに使われるようになるにつれて、信頼性のあるソフトウェアを作成することがますます困難になってきている。大きく複雑化したプログラムでは、そのプログラム全体をひとりで理解することはできない。また、プログラムのさまざまな部分が、予測できない方法で相互作用していることがある。(例えば、プログラムのある個所の虫をとると、別の箇所に新しい虫が生まれることがある。)コンピューティングは、安全性第一の仕事にも用いられることがあり、そこでは1個の虫といえども負傷や死亡につながることもある。やがて、優れたソフトウェアを生み出すことが、非常に困難なことであり、費用がかさむことであり、そして非常に必要とされていることであることが明らかになった。これにより、ソフトウェアエンジニアリングが生み出されることとなった。この用語は、1968年ドイツのガルミッシュで開かれた、NATO後援の会議で初めて使われた。コンピュータ科学は、他の科学と同様に、新しい知識の創造に焦点を当てている。ソフトウェアエンジニアリングは、他の工学分野と同様に、目的の作業を確実に行うものを設計・制作する厳密な手段に焦点を当てている。1970年代には、ソフトウェアエンジニアリングについての主要な会議が開催されるようになり、1980年代にはいくつかのコンピュータ科学の学士プログラムは、ソフトウェアエンジニアリングコースを含むようになった。しかしながら、米国では1990年にはいるまで、多くの機関でソフトウェアエンジニアリングがコンピュータ科学研究の重要な構成要素となっていないのを目にすることがなかった。

- ソフトウェアエンジニアリングは、それ自体がひとつの領域として発展し始めた。元々、このソフトウェアエンジニアリングという用語は、工学の伝統的な概念を、ソフトウェア作成に関する問題に適用するために導入されたものである。ソフトウェアエンジニアリングが成熟するにつれて、そのめざす課題の範囲が明確になっていった。ソフトウェアエンジニアリングはコンピュータ科学の基盤に加えて、人間のプロセスも含んでいる。これは、その本質からして、コンピュータ科学における論理的抽象化よりも形式化することが困難なものである。コンピュータ科学カリキュラム内のソフトウェアエンジニアリングコースが多くの人々に証明したのは、そのような形でのコースは、学生にソフトウェアエンジニアリング領域について教えることはできるが、ソフトウェアエンジニアになる方法を教えることには成功しないということである。多くの専門家が、後者の目的を達成するためには、一連の講義に加えて実地のプロジェクトの経験が要求されると結論づけている。これはコンピュータ科学のカリキュラムに付け加えてできるようなものではない。ソフトウェアエンジニアリングの学士プログラムは、1980年代に英国およびオーストラリアで行われるようになった。これらのプログラムは草分け的なものだった。米国では、ソフトウェアエンジニアリングの学士プログラムとして、コンピュータ科学のカリキュラム内で提供できる範囲を遥かに超えた、徹底的な基礎訓練を提供できるように設計されたものが、1990年代になって生まれはじめた。
- 情報システムは、膨れ上がってきている課題と取り組まなくてはならなかった。1990年代より前に、多くの情報システムの専門家が焦点を当てていたのは、1960年代以来実業界が直面しているコンピューティングの必要性であった。経理システム、給与システム、在庫管理システムなどである。1990年代の終わりまでには、ネットワークに接続されたパーソナルコンピュータが日用品となっていた。コンピュータは既に専門家だけのものではなくなっていた。それらは職場の必需品となり、組織の全階層の人が使用するものになっていたのだ。コンピュータが果たす役割が広がり、コンピュータ技術によって組織はかつて無いほどの情報を利用することができるようになり、また組織内の活動はますます増えていった。情報を管理する際の問題は複雑さを極め、組織を効率的・効果的にサポートするための情報や技術の適切に使うということが、組織の命運に関わる非常に重要な課題となった。こういった要因により、情報システムの専門家が直面する問題は増加し、複雑化し、そして重要性が増していった。加えて、領域としての情報システムは、コンピューティング技術を組織内のコミュニケーションおよび意思決定の手段として利用することに、一段と注意を払うようになっていった。
- インフォメーションテクノロジープログラムは、1990年代後期に行われるようになった。1990年代は、組織のほとんどの階層でコンピュータが仕事上・業務上の必須のツールとなった時期であり、ネットワーク化されたコンピュータシステムが組織の情報基幹となった。生産性が上昇する一方で、コンピューティングインフラストラクチャにおける問題が起きると従業員の作業能力を制限してしまうという、職場におけるコンピュータへの依存状態を生み出した。企業やその他の機関でのIT部門は、組織内のコンピューティングインフラストラクチャが適合していること、確実に機能していること、その組織内の人々からのコンピューティングに関連した要求が満たされていること、問題が解決されることなどを確実にするという新しい仕事を引き受けた。1990年代の終わりまでには、こういった本質的な要求を満たす知識と技能を備えた卒業生を生み出す学士プログラムが無いことが明白となった。大学側は、この重大な空洞を埋めるために、インフォメーションテクノロジーの学士プログラムを開発した。

これらの発展は、全体相まってコンピューティング領域の眺望を一変させることになる。すべての工業化社会で途方も無い人的資源がインフォメーションテクノロジーの活動に注ぎ込まれた。その要因には、ワールド ワイド ウェブの爆発的な成長、Y2K 問題などがあり、ヨーロッパにおけるユーロの流通開始などもあった。

2.2.3 1990 年代後

コンピューティング学士プログラムの新しい眺望は、コンピューティング分野全体と新世紀の諸問題を取り扱うまでに成熟したことを反映したものとなった。米国においては、コンピュータエンジニアリングが電気工学から明確に区別され、ひとつの学科領域としての地位を固め、コンピュータ ハードウェアおよびそれに関連するソフトウェアに関して主要な役割を担うこととなった。ソフトウェアエンジニアリングが、ソフトウェア システム作成についてまわる重要な課題、つまり信頼性と費用に関する問題を取り扱うために生まれた。インフォメーションテクノロジーが、他のコンピューティング領域によって十分に解決できない空隙を埋めるためにどこからともなく現れた。

こういった成熟や進展は、学生にも教育機関にも、より大きな可能性を与えることとなった。コンピューティングの教育プログラムは多様化していき、学生は 1980 年代以前には経験しなかったような不明瞭な選択を迫られるようになった。図 2.1 の下半分は、優秀な学生にとって現状で与えられている選択肢がどのように見えるかを示したものである。波線の楕円は、優秀な学生がそれぞれの領域の第 1 の焦点として受け止めるであろうものを示している。

ハードウェアを勉強したい学生が行くべきところは明確である。コンピュータエンジニアリングは、電気工学から生まれ、デジタル置の設計に関わるハードウェアおよびソフトウェアの問題に取り組んでいる人々の拠点である。しかし、これら以外に関心がある学生にとっては、選択は簡単なものではない。1990 年代より前には、ソフトウェア開発の専門家になりたい学生はコンピュータ科学を学んだものだった。1990 年以後は、意味ある選択肢が増えた。コンピュータ科学、ソフトウェアエンジニアリング、それにコンピュータエンジニアリングもそうだ。これらは、いずれもソフトウェア開発に関する独自の展望を持っている。実際にこの 3 つの違いをあげるとこうなる。CE では、扱うソフトウェアがハードウェア装置に焦点を当てたものである。SE では、実世界の要求に応える堅固なソフトウェアの作成に重きが置かれている。そして CS では、ソフトウェアとはアイデアの表現手段であり、広範囲なコンピューティングの問題とその応用を探求する手段なのである。こうした違いは、優秀な学生にもわからないかもしれない。普通の学生にとっては、これら 3 つの領域がソフトウェアに重点を置いている点が共通で、それ以外には区別がないように見える。

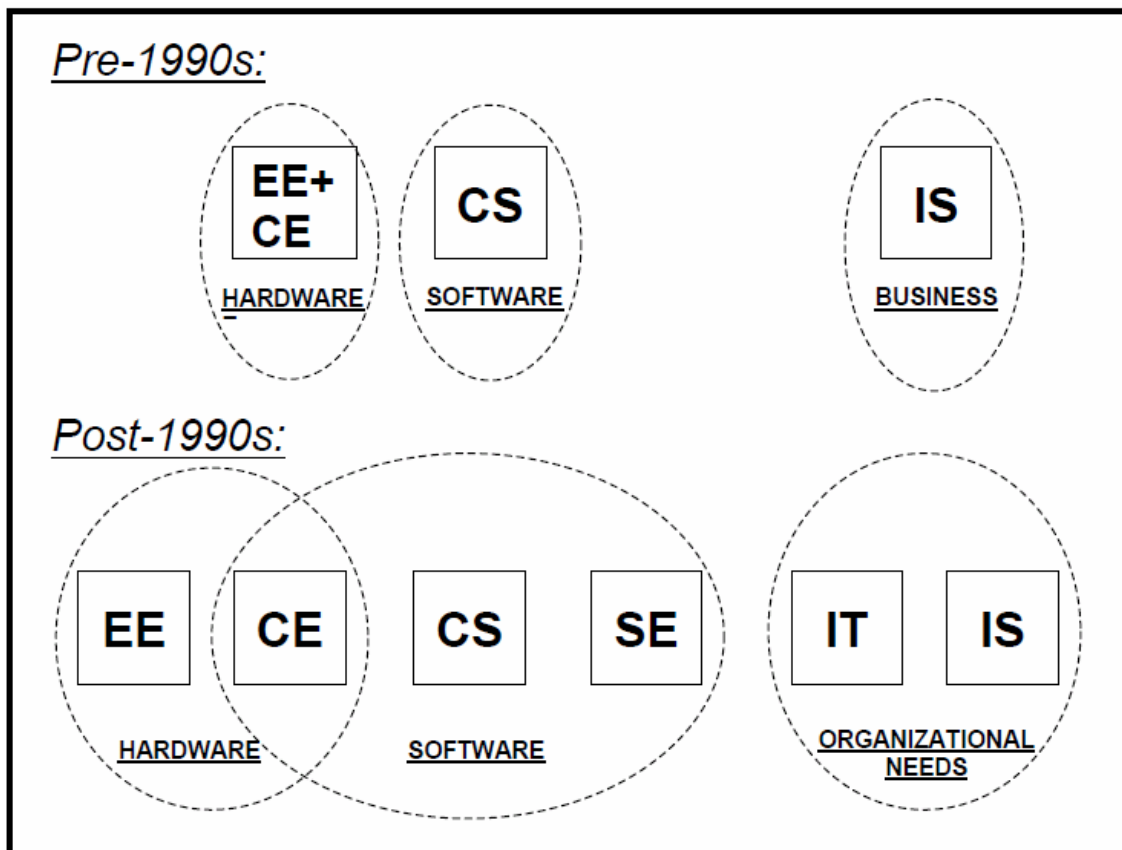


図 2.1 困難な選択: 志願者が各領域に抱くイメージ

同様に、1990年代に入る以前は、実世界の問題にコンピューティングを適用するといえばまず第一にビジネスであった。そして情報システムがそうした仕事の本拠地であった。実世界利用の範囲はビジネスからすべての組織へと広がり、学生は、情報システムを選ぶかインフォメーションテクノロジーを選ぶかできるようになった。IT領域もIS領域も、ソフトウェア・ハードウェアへの焦点を含んでいるものの、どちらもこれらそのものには力点をおかない。どちらも、これらの技術を組織のニーズに対処するための重要な道具として位置づけているに過ぎない。ISは情報の生成と活用に焦点を当て、ITは組織の情報基盤の適切性・信頼性を保証することに焦点を当てる。優秀な学生でも、こうした重要な違いに気がつかず、ISもITもともに技術活用組織のニーズにコンピューティングを用いることを目的とする領域であるとしが見えないかもしれない。

2.3 主要コンピューティング領域の説明

このセクションでは、コンピューティングの主要5領域の特徴を説明する。コンピューティング領域名と特定の学士プログラム名との間に存在する重要な差異を理解するための更なる情報は、3.4および3.5を参照されたい。

2.3.1 コンピュータエンジニアリング (CE: Computer Engineering)

コンピュータエンジニアリングは、コンピュータおよびコンピュータを利用したシステムの設計と構築に携わっている。ハードウェア、ソフトウェア、通信、およびそれらの相互作用の研究も行う。そのカリキュラムでは、伝統的な電気工学と数学の理論・原理・実践に焦点を当て、それらをコンピュータおよびコンピュータ組み込みの装置を設計する問題に応用する。

コンピュータエンジニアリングの学生は、通信システム、コンピュータ、コンピュータ内蔵機器などのデジタル ハードウェア システムの設計を学ぶ。彼らはソフトウェア開発も学ぶが、その焦点は、デジタル機器のためのソフトウェアおよびデジタル機器と利用者や他の機器とのインタフェースとしてのソフトウェアにある。CE の学習は、ソフトウェアよりハードウェアに重きを置いていることもあれば、両者のバランスが取ることにも重きを置いていることもある。CE は、工学色が強い。

現時点においてコンピュータエンジニアリングで主要となっているのは、組み込みシステムである。すなわち、ソフトウェアとハードウェアを内蔵した機器の開発である。例えば、携帯電話、デジタル オーディオ プレイヤ、デジタル ビデオ レコーダ、警報システム、X 線機器、レーザー手術用機器などといった機器はすべて、ハードウェアと内蔵ソフトウェアの統合を必要として、ことごとくコンピュータエンジニアリングの産物である。

2.3.2 コンピュータ科学 (CS: Computer Science)

コンピュータ科学は、広範囲に及ぶ。その理論やアルゴリズム基礎論から、ロボット、コンピュータ視覚、知能システム、生物情報などの最先端の開発までに広がる。コンピュータ科学者の活動は、つぎの 3 つのカテゴリに分けて考えることができる。

- ソフトウェアを設計し実現する。コンピュータ科学者は、プログラミングの仕事に自ら挑戦する。そして、他のプログラムを監督指導して、プログラムに新しいアプローチを伝え続ける。
- コンピュータ利用の新しい手法を考案する。ネットワーキング、データベース、マンマシン インタフェースなどの CS 領域の発展があって、ワールド ワイド ウェブの発展が可能となった。現在では、CS の研究者は、他の分野の科学者と協力して、ロボットを実際的で知性ある助人に仕立てたり、データベースを活用して新しい知識を創造したり、コンピュータを使って DNA を解読したりしている。
- 彼らはコンピューティングに関わる問題を効果的に解決する方法を開発する。例えば、コンピュータ科学者達は、情報をデータベースに格納し、データをネットワークで送信し、複雑なイメージを表示するための最良の方法を開発している。彼らの理論的なバックグラウンドにより、可能な最高のパフォーマンスというものを定めることが可能になり、彼らのアルゴリズムの研究はより良いパフォーマンスを生み出す新しいアプローチを開発することに役立っている。

コンピュータ科学の範囲は理論からプログラミングまでに広がっている。この広大な領域を反映したカリキュラムは、卒業生が特定の仕事の準備を整える障害となるとして批判されることがある。他の学科が、もっと仕事に直結した技能を身につけた卒業生を輩出する一方で、コンピュータ科学は卒業生が新しい技術やアイデアに適応できるようになる包括的な基盤を提供している。

2.3.3 情報システム (IS: Information Systems)

情報システムの専門家は、情報技術の解決手法と業務プロセスとを融合し、ビジネスその他の企業活動での情報ニーズに適合させて、その目標を効果的かつ効率的に達成できるようにする。この領域での情報技術に対する視点は、情報そのものに重点をおき、技術を情報の生成・処理・配信する手段として位置づける。この領域のプロフェッショナルが第一義的に携わっているのは、企業が目標を定義し達成する際に利用することができる、コンピュータシステムの提供する情報であり、企業が情報技術を使って実施し向上させることのできる処理業務である。彼らは、技術的要素も組織的要素もともに理解していなくてはならない。また彼らは、組織を補佐し、うまく業務処理に情報と技術を活用して競争力を高めることができなくてはならない。

情報システムの専門家は、組織の情報システムの要求定義に中心的な役割を果たし、またその仕様、設計、実装に活動する。結果として、こういったプロフェッショナルは、組織の原則や実務に対する十分な理解をもって、組織内の技術部門と経営部門との間の有効な橋架け役となり、彼らが調和を保って働くことができ、組織がその運営に必要な情報とシステムを確実に所有できるようにすることができなければならない。情報システムのプロフェッショナルは、技術を使った組織間対話・相互連携のためのシステムの設計にも携わる。

情報システム (IS) プログラムの多くは、ビジネス スクールに置かれている。どの IS の学士課程も、ビジネスとコンピューティングの学習を結合したものである。さまざまな IS プログラムがさまざまな名前のもとで存在している。たいていの場合、その名前はプログラムの内容を反映したものである。例えば、“コンピュータ情報システム”は最も技術重点のものであり、“経営情報システム”は IS の組織的側面・行動的側面に力点をおいたものである。とはいえ、学士プログラムの名称が常に内容にそったものであるとも限らない。

2.3.4 インフォメーションテクノロジー (IT: Information Technology)

インフォメーションテクノロジーという名称には、2つの意味がある。広い意味では、このインフォメーションテクノロジー(=情報技術)という言葉は、コンピューティング全体を示すためによく使用される。教育の場においては、この言葉は、企業、政府、医療、学校などの組織でのコンピュータ技術のニーズを満たせる学生を育てるための学部レベルの学士プログラムを示すものとして使われている。国によっては、こうした学士プログラムに別の名称が使われている。

前のセクションでは、情報システムは情報技術の情動的側面に焦点を当てていると述べた。インフォメーションテクノロジー (IT) は、それと相補的な視点をもつ。つまり、情報よりも、それを伝える技術に重点を置いているのだ。IT は、新しく、そして急速に成長している分野であり、ビジネスやその他の組織における日常のニーズに対しての草の根レベルでの対処から始まったものである。今日では、あらゆる組織が情報技術に依存している。組織は、適切なシステムを保つ必要があるのだ。こういったシステムは、適切に稼働し、安心安全に保たれ、適切にアップグレードされ、保守され、リプレースされなければならない。組織の全従業員は、IT スタッフのサポートを必要としている。すなわち、コンピュータ システムとソフトウェアを理解していて、従業員がでくわすコンピュータ関連の問題を解決することに専心しているスタッフのサポートが必要なのだ。インフォメーションテクノロジーのプログラム修了生はこれらのニーズに対処するのである。

インフォメーションテクノロジーの学士カリキュラムが現れたのは、他のコンピューティング関連の学士プログラムが、こういった非常に現実的なニーズを処理できる卒業生を十分に輩出していなかったからである。IT プログラムの存在意義は、知識と実践力をバランスよく身につけた卒業生、つまり組織の IT 基盤とそれを使用する人々の両方の面倒を見ることができる実践的な専門家を輩出することである。IT 専門家は、組織に適したハードウェア・ソフトウェアの製品を選択し、それらの製品を統合して組織のニーズとインフラに合わせ込み、そして組織のコンピュータ ユーザのためにそれらのアプリケーションをインストールし、カスタマイズし、保守していくことに責任をもつ。これらの責任の例としては、ネットワークの設置、ネットワークの管理と保全、ウェブ ページの設計、マルチメディア リソースの開発、通信機器の設置、e メール システムの監視があり、更に、技術のライフ サイクルの計画と管理を行って組織の技術を保守し、アップグレードし、置き換えていくことがある。

2.3.5 ソフトウェアエンジニアリング (SE: Software Engineering)

ソフトウェアエンジニアリングの領域は、信頼でき効率がよいソフトウェアシステム、開発も保守も費用が安く、顧客が定めた要求事項のすべてを満たすソフトウェアシステムの開発・保守を担当する。これはセクション 2.2.2 で述べたこの領域の原点を反映している。最近更に発展したのは、さまざまな状況で大型の高価なソフトウェアシステムが使われ始めたこと、また、安全性不可欠な応用でのソフトウェアがますます重要になって来たことに呼応したものである。ソフトウェアエンジニアリングは、他のエンジニアリングとは性質が異なる。それは、ソフトウェアの無形性やソフトウェア操作の非連続性に起因する。ソフトウェアエンジニアリングは、数学・コンピュータ科学の原理と、有形の物理的人工物を対象に発達してきたエンジニアリングの技法とを統合することを目指す。優秀な学生なら、ソフトウェアエンジニアリングが下記の 2 つの形で提示されているのに気付くだろう。

- コンピュータ科学の学士プログラムが、ふつう、その CS カリキュラムの一部としてソフトウェアエンジニアリングの科目を用意している。中には、CS 内にソフトウェアエンジニアリングの複数科目を組んでいるものもある。
- 多くの教育機関が、ソフトウェアエンジニアリングの学士プログラムを提供している。

コンピュータ科学の学士プログラムとソフトウェアエンジニアリングの学士プログラムには、共通する科目がたくさんある。ソフトウェアエンジニアリングの学生は、ソフトウェアの信頼性と保守についてより多くを学び、生まれつき正しいソフトウェアを開発し保守するための技法により焦点を当てる。コンピュータ科学の学生もそのような技法の重要性について耳にするはあるにはあるが、SE が提供する工学的な知識と経験は、CS が提供できるものを遥かに超えている。この事実は非常に重要であり、SE の報告巻での勧告によれば、SE プログラムでの学習の中で、学生は必ず他人に本当に使ってもらうソフトウェアの開発に加わるべきであるという。SE の学生は、顧客のニーズの見極め方やそのニーズを満たした使えるソフトウェアの開発の仕方を学ぶ。本当に有用で使えるソフトウェアを提供する方法を知ることが最優先事項なのである。

職場においては、ソフトウェア エンジニアという用語は、職務区分を表す。職務記述に用いられた場合の、この用語の標準的な定義はない。雇用主によって意味が大幅に違ってくる。コンピュータ プログラマと同義の肩書きであることもあれば、大きく複雑なソフトウェアや安全に欠かせないソフトウェアなどのプロジェクトを管理する人の肩書きであることもある。普通の人には注意して、ソフトウェアエンジニアリングという領域の名前と、雇用宣伝などでの職務の肩書きとしてのソフトウェア エンジニアという用語の曖昧な使い方とを混同しないでほしい。

2.4 コンピューティング領域の図式化

各コンピューティング領域の共通点と差異を表現するために、それらを図式化した。こうして図化されたものは、各領域が、コンピューティングに関わる問題の領域をどのように担当しているかを示す(図 2)。これらは、現状を描写したものであり、将来のあるべき姿を示したものではない。各領域に所属する学生が典型的に卒業後に行うことに焦点が当てられており、学生が勉強する主題を基本としたものではない。これらの図が示しているシナリオを超えたキャリアを持つ人もいることだろう。

水平方向は、左側の理論(theory)、原理(principles)、革新(innovation) から、右側の応用(application)、配備(deployment)、構成(configuration)へと続いている。であるから、研究所で新しいことを発明したいとか、大学で新しい原理を展開させたい人であれば、左側の空間を占めるコンピューティング領域で働きたいと思うだろう。逆に、人々が適切な技術を選択したり使用したりするのを助けたいとか、市販の製品を統合して組織に関わる問題を解決したい人であれば、右側の空間を占めるコンピューティング領域を望むだろう。これらの両極端の間に入る多くの職務があるため、両端だけを見るのではなく、それらの間にあり得るものを考慮してみることが必要である。

垂直方向は、一番下がコンピュータハードウェアとアーキテクチャ(computer hardware and architecture)で、組織関連事項と情報システム(organizational issues and information system)が一番上にある。この軸を上昇するにつれて、焦点は人間、情報、そして職場に移っていく。この軸を下降するにつれて、焦点は機器とそれらに共有されるデータへと移っていく。であるから、回路設計が好きであるとか、コンピュータの内部構造に興味があるとかという人は、この下方の部分に関心をもつことだろう。また、技術がどのようにして人々の役に立つか知りたいとか、技術が組織に与える影響に興味がある人は、この上方の部分に関心をもつことだろう。

私たちは、水平方向と垂直方向を総合的に考えることもできる。人々のために物事をうまく運ばせることに関心があり、装置よりも組織に関心がある人は、右下の空間に興味をもち、情報が組織に与える影響について新しい理論を発展させたい人は、左上の空間に興味をもつといった具合だ。

図 2.3 から図 2.7 までは、この枠組みを用いて示した、これら 5 つのコンピューティング領域が占めている概念的な空間を、この枠組みを用いて示したのである。これらは、タスクフォースの主観的な解釈を伝えるために用いた非公式な図である。緻密な定量的な基礎に基づいたものではない。さらに、これらの図はコンピューティングの主題だけを示している。コンピュータエンジニアリングのプログラムも情報システムのプログラムも、コンピューティング以外の空間に大きな注意を向けているが、それらはこの図に反映されていない。必要とされるコンピューティングおよび非コンピューティングの主題は、第 3 章に示した。

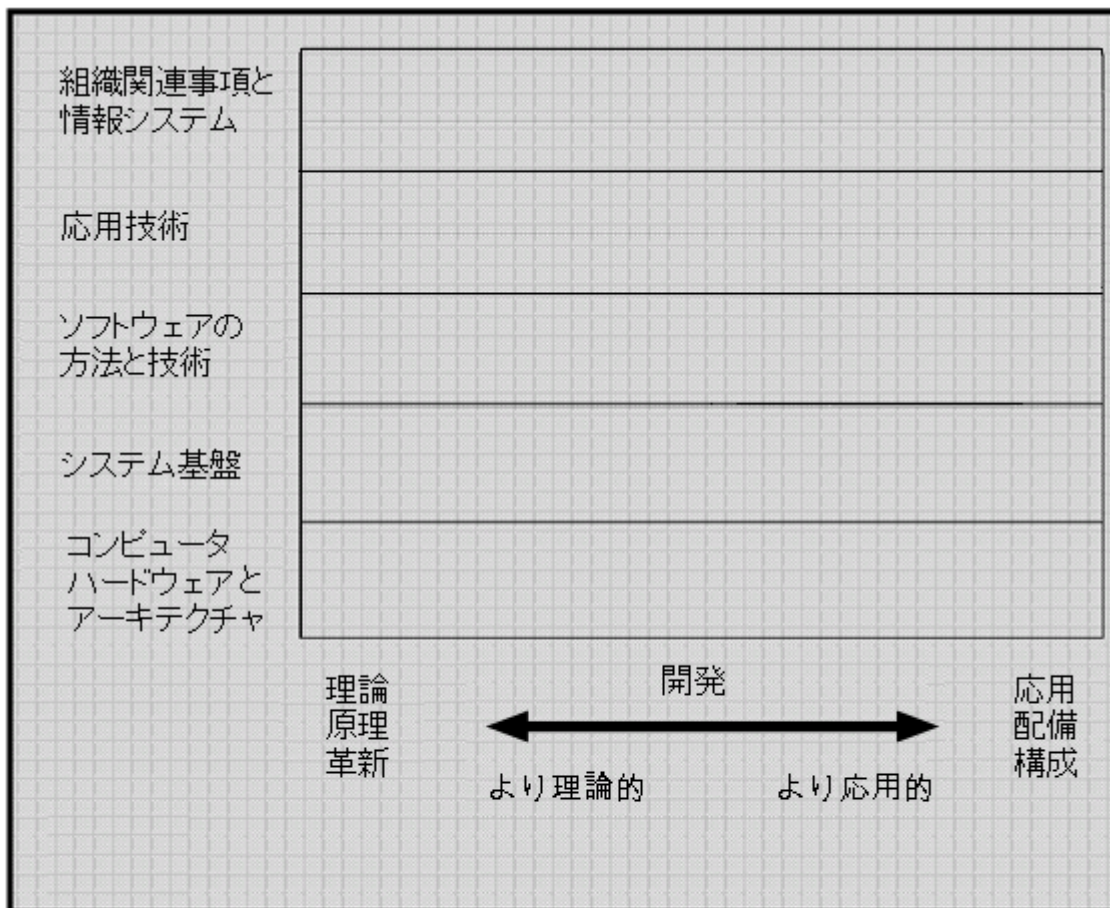


図2.2 コンピューティングの問題空間

2.4.1 コンピュータエンジニアリング

図 2.3 の影の部分が、コンピュータエンジニアリング領域を表している。図の下方で広がっている理由は、コンピュータエンジニアリングがハードウェアおよびソフトウェアを用いての製品設計や実装に関する理論・原理から実際の応用までを対象としているからである。上方に行くにつれて中央へと狭くなっていくのは、コンピュータエンジニアリングがハードウェアから離れていくにつれて関与が薄くなっていくからだためである。ソフトウェア開発のレベルまでには、コンピュータエンジニアリングの関与は、水平方向は中央部分だけになってしまうが、これはコンピュータエンジニアリングがソフトウェアに関わるのは、統合装置を開発するために必要な時だけだからである。

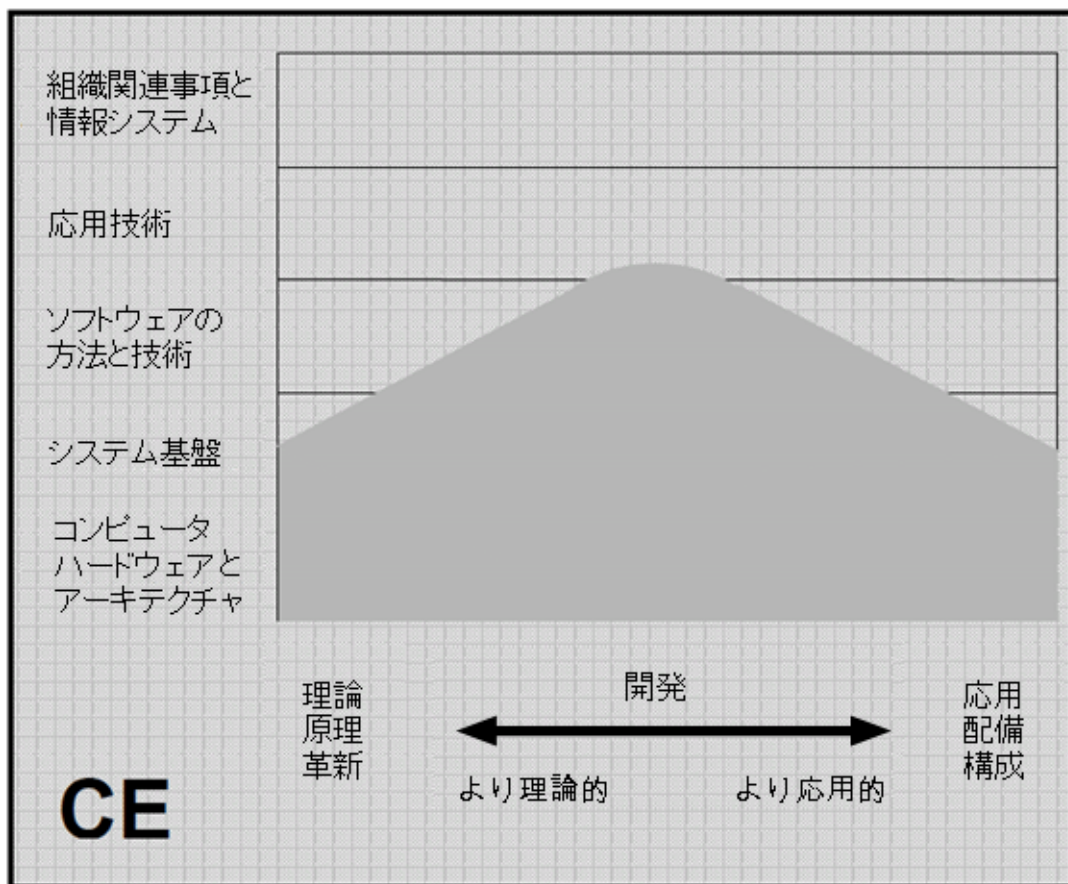


図 2.3 コンピュータエンジニアリング

2.4.2 コンピュータ科学

図 2.4 の影の部分が、コンピュータ科学領域を表している。コンピュータ科学は、上端と下端を除いて、垂直方向をほとんどすべて対象としているが、コンピュータ科学者は通常、ソフトウェアが走るハードウェアそのもの、コンピュータが提供する情報を利用する組織そのものを扱うことがない。コンピュータ科学者は、総体として、これら両極端の領域の間にあるほとんどすべて(下方はデバイスを動作可能にするソフトウェアまで、上方は組織運営を助ける情報システムまで)に関心をもっている。コンピュータ科学者は、システム基盤(オペレーティングシステム、通信プログラムなど)からアプリケーション技術(ウェブブラウザ、データベース、サーチエンジンなど)まで、あらゆる類のソフトウェアを設計・開発する。コンピュータ科学者は、こういった機能を開発するものの、それら機能の実配備までは行わない。こういうわけで、コンピュータ科学を示す影の部分は、右に進むと狭まり、ついには止まってしまう。これは、人々がコンピューティング製品を選択するのを助けたり、組織のニーズに合わせて製品の調整をしたり、そうした製品の使い方を学んだりすることに対して、コンピュータ科学者が手助けすることがないからである。

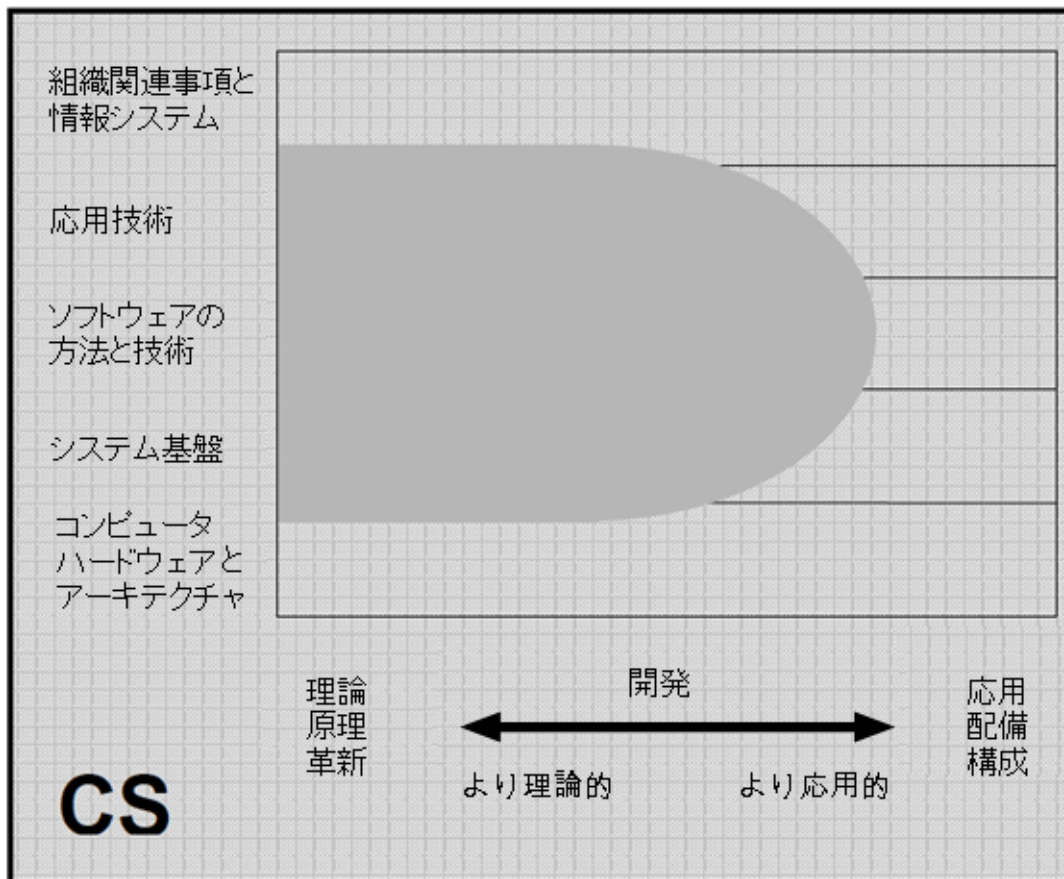


図 2.4 コンピュータ科学

2.4.3 情報システム

図 2.5 の影の部分が、情報システム領域を表している。影の部分は、上端のほとんどをカバーし、理論・原理から応用・開発までに広がっている。これは、情報システムに関わる人々が、情報システムとそれを役立てる組織との関係に携わっているからである。そして、情報システムのプロフェッショナルの多くは、システムの配備と構成、そしてユーザの訓練にも関与している。情報システムが対象としている空間は、図の右側で、ソフトウェア開発からシステム基盤まで垂れ下がっている。これは、情報システムの専門家が、企業のニーズのためにアプリケーション技術（特にデータベース）を調整することがよくあるからである。また彼らは、他のソフトウェア製品を利用するシステムを開発して組織の情報ニーズに合わせ仕事も担当する。（この図は、情報システムの教育プログラムがビジネスに主軸をおいていることを全く反映していない。コンピューティングおよび非コンピューティングのトピックについて要約した表を第 3 章に記載している。）

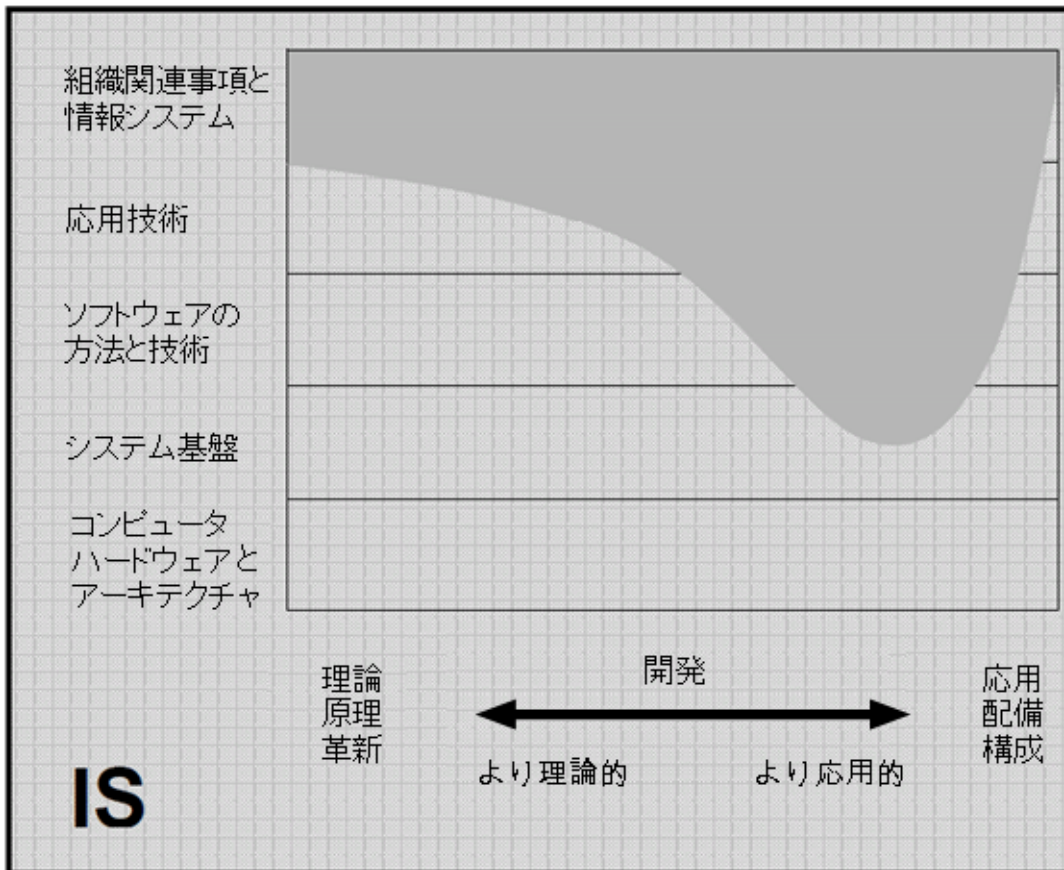


図 2.5 情報システム

2.4.4 インフォメーションテクノロジー

図 2.6 の影の部分が、インフォメーションテクノロジー(IT)領域を表している。影の部分は、右端のほとんどをカバーしている。これはインフォメーションテクノロジーが、広範な組織や人々の、応用・配備・構成のニーズに焦点を当てているからである。この範囲(組織の情報システムから、アプリケーション技術、そしてシステム基盤まで)にわたって、ITの役割には、ISと重なり合うものがある。しかし、ITに関わる人々は、コンピューティング技術に起因する人間のニーズを満たすことに特に焦点を当てている。加えて、ITを示す影は、特にアプリケーション技術の領域で、応用から理論・革新へと左側に伸びている。これは、ITに関わる人々がしばしば、ウェブ上のデジタル技術を開発して、組織が各種情報を提供するのに供しているからである。つまり、関連する原理や理論の適切な概念的基盤にも関与しているからである。

ITは、非常に新しい領域であるため、その焦点は既存の概念やスキルの基盤を学生に提供する教育プログラムを開発することに当てられてきた。IT教員の多くは、彼らの分野における研究が成長して、関連領域での新しい知識を創造し発展させるようになると主張する。そうなった暁には、この陰は、大きく左側に伸びることになるだろう。しかしながら、これはまだ達成されていない。この図は、ITの現状を反映したものである。

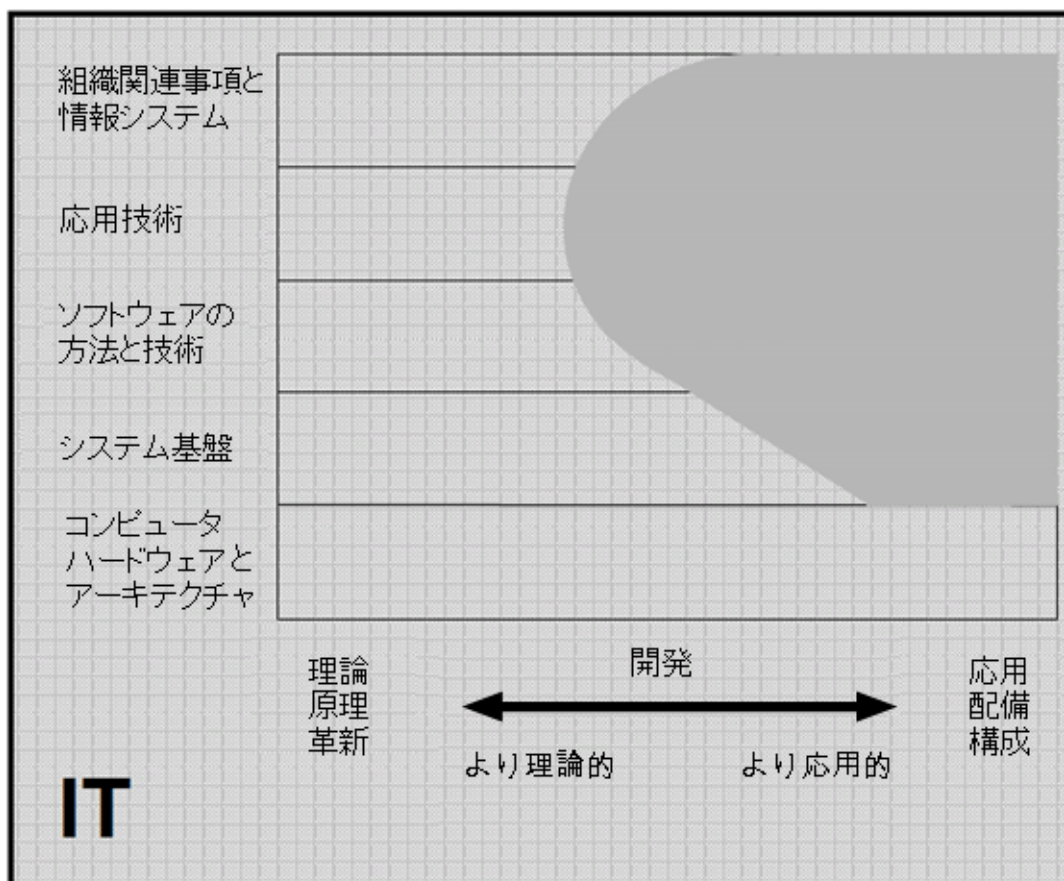


図 2.6 インフォメーションテクノロジー

2.4.5 ソフトウェアエンジニアリング

図 2.7 の影の部分部分が、ソフトウェアエンジニアリング(SE)領域を表している。SE は下部のハードウェア関連レベルで水平方向全体に広がっており、IS は上部の組織関連レベルでほぼ全域に広がっているのを見てきたが、SE は、ソフトウェアの体系的開発に関連する広い部分をカバーしている。この理由は、SE に携わる人々が大規模ソフトウェア開発の専門家に対する広範なニーズを満たしているからである。SE の主要な目標は、高品質なソフトウェアを時間どおりに予算内で作成するための体系的モデルと信頼性のある技術を開発することであり、これらは理論・原理から日々の実践へと広がっているのである。SE の影は下に向かってシステム基盤にまで広がっているが、これは堅牢なソフトウェア基盤の開発を SE の人々が行っているからである。上に向かっては組織の問題にまで入り込んでいるが、これは、顧客である組織に適した情報システムの設計と開発にも SE の人々が興味をもっているからである。

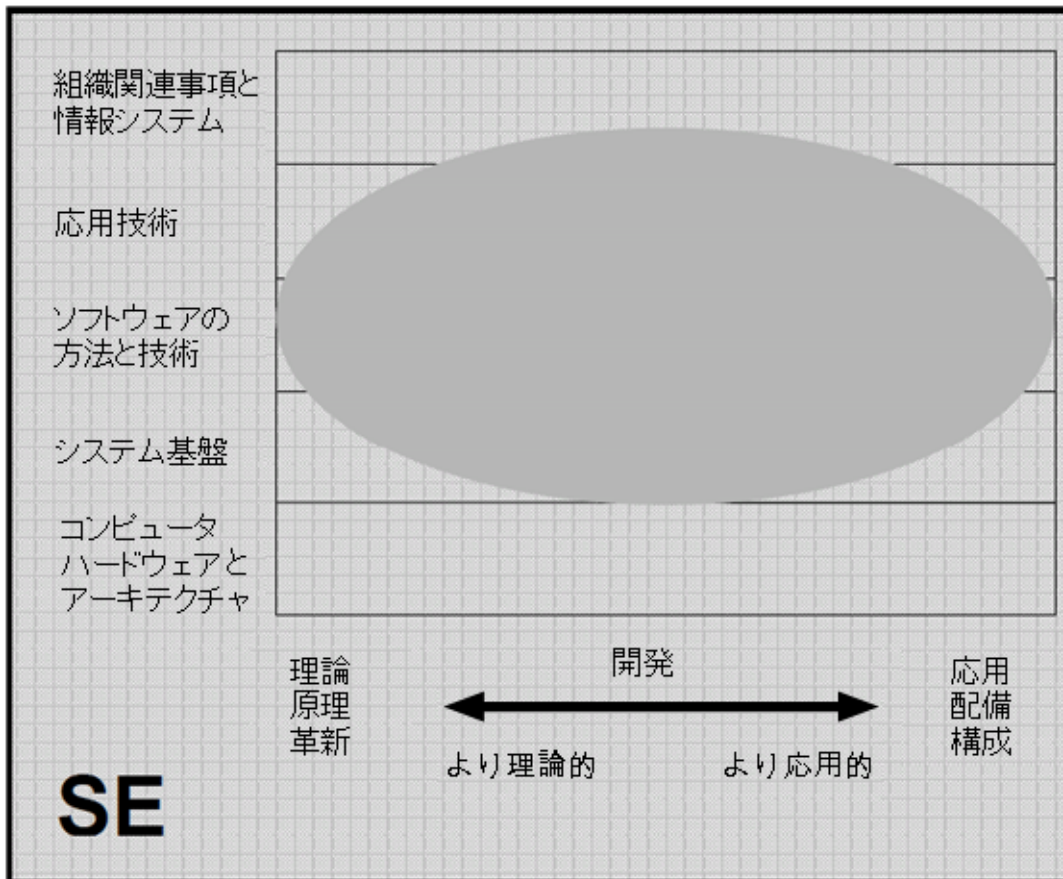


図 2.7 ソフトウェアエンジニアリング

第3章 学士プログラムとキャリア

本章では、5つの主要な領域の学士プログラムの特徴について概説し、(a) 学士プログラム内での相対的な焦点範囲と、(b) 修了者に期待できる能力の点から相互比較を行う。また、コンピューティング領域の発展に対する教育機関の対応の状況・速度について検討する。さらに、学士プログラムとプロフェッショナルとしての就業機会の関係についてまとめた上で、コンピューティング学士プログラムに共通する諸事項についてまとめる。

3.1 カリキュラム概要： コンピューティング系学士プログラムの一覧

図式化は、情報を一瞥の下に伝達するのに有効な手段であっても、その性質上、細目について不完全であり、不正確な印象を与えかねない。本節では、より詳細な内容に踏み込んでコンピューティング領域の比較を行う。

表 3.1 は、5領域の学士プログラムが対象とするものの中で“コンピューティング”のトピックに重点を置いて比較したものである。左の列は学生がコンピューティング領域学士プログラムで学ぶコンピューティングの知識と能力を示した 40 のトピックを並べあげたものである。ここに並べたものは、5 主要領域のカリキュラム報告に指定されているコンピューティングのトピックの和集合にあたる。つまり、どれかのコンピューティング領域学部レベルで履修する内容の要約となっている。これらのトピックに不案内な方は、この文書の末尾に添付した用語集を見て頂きたい。5つのカリキュラム報告は、同じトピックに関して異なる用語を使っていることがあるし、トピックをサブトピックに分割する分け方でも異なっている。結果として、表 3.1 に並べたものは、それぞれのカリキュラム報告のトピック一覧と正確に一致してはしない。というよりも、5つの学部レベルのコンピューティングカリキュラム報告に現れているトピックの要約である。

表 3.2 は、同様の表で、5領域の学士プログラムにわたる“コンピューティング以外”のトピックについて、相対的な重みを示す。5領域のコンピューティング学士プログラムのカリキュラム指針は、いずれも、コンピューティング以外のトピックをカバーするよう指示しているが、コンピューティング領域の2つは、もともとコンピューティング分野と他の分野の間に位置しているものである。コンピュータエンジニアリングはコンピュータ科学と電気工学の要素を包含している。情報システムは、コンピューティング分野とビジネス分野の境界にまたがっている。したがって、この2つの学士プログラムの学生は、表 3.2 に示す様に、コンピューティング以外のトピックの習得にかなりの時間を割いている。

どちらの表も、最左の列にトピックが並び、その他の列にトピックごとに5領域のコンピューティング学士プログラムの数値化した値が示してある。この数値は、0（最低）から5（最高）までの整数であり、5領域のコンピューティング学士プログラムがそのトピックにどの程度の重点を置いているかを相対値として近似的に示したものである。表中の数値は、説明用のものであって、各領域がそのトピックに置く重みを正確に表すことを意図したものではない。

5 領域の学士プログラムのそれぞれについて、各トピックは二つの数値をもつ。一つは“最小”の列に、もう一つは“最大”の列に記される。

- “最小”に記す値は、そのトピックに対して最低限払うべき重みであり、典型的には該当するコンピューティング領域のカリキュラム報告で決められている。ただし、“最小”値は、その領域での最低限の要求を他の領域の最低限の要求との相対値として示したものである。
- “最大”に記す値は、最大限払いうる重みであり、典型的には該当する領域のカリキュラム報告で規定された許容値内にある。それぞれの領域では、学生が専門学習領域を一定の許容範囲の中で選ぶことを許すと同時に、カリキュラム報告で定める最低限を超えて履修することを定めている。報告書は、それぞれの教育機関が5領域の報告書が定めるものより高い要求を学生に課してもよいとしている。“最大”値は、学士プログラムのその他の要件からくる制限の中で、そのトピックにどの程度まで注力可能かを近似したものである。

表 3.1 5領域の学士プログラムにおけるコンピューティングトピックの相対的な重み付け

知識領域	CE		CS		IS		IT		SE	
	最小	最大	最小	最大	最小	最大	最小	最大	最小	最大
プログラミング基礎	4	4	4	5	2	4	2	4	5	5
統合プログラミング	0	2	1	3	2	4	3	5	1	3
アルゴリズムと計算複雑性	2	4	4	5	1	2	1	2	3	4
コンピュータのアーキテクチャと構成	5	5	2	4	1	2	1	2	2	4
オペレーティングシステムの原理と設計	2	5	3	5	1	1	1	2	3	4
オペレーティングシステムの構成と利用	2	3	2	4	2	3	3	5	2	4
ネットワークの原理と設計	1	3	2	4	1	3	3	4	2	4
ネットワークの利用と構成	1	2	2	3	2	4	4	5	2	3
プラットフォーム技術	0	1	0	2	1	3	2	4	0	3
プログラミング言語理論	1	2	3	5	0	1	0	1	2	4
ヒューマン・コンピュータ・インタラクション	2	5	2	4	2	5	4	5	3	5
グラフィックと可視化	1	3	1	5	1	1	0	1	1	3
人工知能 (AI)	1	3	2	5	1	1	0	0	0	0
情報マネジメント (DB) 理論	1	3	2	5	1	3	1	1	2	5
情報マネジメント (DB) 実践	1	2	1	4	4	5	3	4	1	4
科学計算 (数値解析的方法)	0	2	0	5	0	0	0	0	0	0
法律/職業/倫理/社会	2	5	2	4	2	5	2	4	2	5
情報システム開発	0	2	0	2	5	5	1	3	2	4
ビジネス的要求の分析	0	1	0	1	5	5	1	2	1	3
Eビジネス	0	0	0	0	4	5	1	2	0	3
技術的要求の分析	2	5	2	4	2	4	3	5	3	5
SWのための工学的基礎	1	2	1	2	1	1	0	0	2	5
SWのための工学的経済学	1	3	0	1	1	2	0	1	2	3
ソフトウェア・モデリングと分析	1	3	2	3	3	3	1	3	4	5
ソフトウェア設計	2	4	3	5	1	3	1	2	5	5
ソフトウェアの検査と検証	1	3	1	2	1	2	1	2	4	5
ソフトウェアの進化 (保守)	1	3	1	1	1	2	1	2	2	4
ソフトウェアプロセス	1	1	1	2	1	2	1	1	2	5
ソフトウェアの品質	1	2	1	2	1	2	1	2	2	4
コンピュータシステム工学	5	5	1	2	0	0	0	0	2	3
デジタル論理	5	5	2	3	1	1	1	1	0	3
組み込みシステム	2	5	0	3	0	0	0	1	0	4
分散システム	3	5	1	3	2	4	1	3	2	4
セキュリティ:課題と原理	2	3	1	4	2	3	1	3	1	3
セキュリティ:実施と管理	1	2	1	3	1	3	3	5	1	3
システム管理	1	2	1	1	1	3	3	5	1	2
情報システム組織のマネジメント	0	0	0	0	3	5	0	0	0	0
システム統合	1	4	1	2	1	4	4	5	1	4
デジタルメディアの開発	0	2	0	1	1	2	3	5	0	1
技術サポート	0	1	0	1	1	3	5	5	0	1

繰り返すが、“最小”値はカリキュラム指針に基づいた必要最低限度を表す。“最大”値はその分野で選択科目を選んだ学生が一般的な場合に期待できる最大の重みであり、あるいは、その分野についてカリキュラム報告のレベルを超えて精通することを学生に要求するような教育機関の卒業生に期待される重みである。“最小”値と“最大”値は大きく異なるので、同じ領域名を付したプログラム どうしであっても、教育機関の履修要件の定め方によって、相当に違ったものになりうる。各教育機関に求められるものによって実質的には異なったプログラムとなる可能性はある。“最小”値も“最大”値も、一般的な場合に合理的に期待できるものを表している。個々の学生、個々の学士プログラムについてみれば、カリキュラム標準にかかわらず、“最小”値が0となっているかもしれないし、“最大”値が5になっているかもしれない。

3.1.1 表の値の決定過程

表 3.1 と表 3.2 は、CC2005 タスクフォースで達したコンセンサスに基づくものである。タスクフォースは、各コンピューティング領域（コンピュータエンジニアリング、コンピュータ科学、情報システム、インフォメーションテクノロジー、ソフトウェアエンジニアリング）の知識体系をその最新のカリキュラム報告巻によって調査して、この数値の合意に達した。二つの表のトピックも、この調査結果に基づいている。また、この調査結果は、トピックごとの各領域の数値を定めるのにも使われた。領域別の知識体系からは、各領域がそれぞれのトピックに要求している最低限のものを示す量的な情報が得られた。しかしながら、トピックごとに各領域がおいている相対的な重みを算出するには、何の情報も得られなかった。

表 3.2 5 領域学士プログラムでのコンピューティング外トピックの相対的な重み付け

知識領域	CE		CS		IS		IT		SE	
	最小	最大	最小	最大	最小	最大	最小	最大	最小	最大
組織論	0	0	0	0	1	4	1	2	0	0
決定理論	0	0	0	0	3	3	0	1	0	0
組織行動論	0	0	0	0	3	5	1	2	0	0
機構改革マネジメント	0	0	0	0	2	2	1	2	0	0
一般システム理論	0	0	0	0	2	2	1	2	0	0
リスクマネジメント(プロジェクト,安全性,リスク)	2	4	1	1	2	3	1	4	2	4
プロジェクト・マネジメント	2	4	1	2	3	5	2	3	4	5
ビジネス・モデル	0	0	0	0	4	5	0	0	0	0
機能的ビジネス分野	0	0	0	0	4	5	0	0	0	0
業績評価	0	0	0	0	4	5	0	0	0	0
回路とシステム	5	5	0	2	0	0	0	1	0	0
電子工学	5	5	0	0	0	0	0	1	0	0
デジタル信号処理	3	5	0	2	0	0	0	0	0	2
VLSI 設計	2	5	0	1	0	0	0	0	0	1
HW テスト法とフォールトトレランス	3	5	0	0	0	0	0	2	0	0
数学的基礎	4	5	4	5	2	4	2	4	3	5
対人コミュニケーション	3	4	1	4	3	5	3	4	3	4

それぞれのトピックに関して各領域が設定している相対的な重みについての有用な指標を得るのに、タスクフォースは、最大限の知恵をしぼって、さまざまなハードな要素やソフトな要素を統合して意味ある計量を導き出した。ハードな要素とは、領域別の知識体系に示されている数値的仕様である。ソフトな要素には、つぎのものがある。

- それぞれのコンピューティング領域がもつ俯瞰的特性の差。各コンピューティング領域は、2.2, 2.3 で特徴づけたようにそれぞれに固有の観点と課題をもっている。
- 同一と思われる用語のもつ意味の差。各コンピューティング領域は多数のコンピューティングのトピックを扱っているが、その領域固有の観点から同じ用語であっても異なる意味をもたせることになる。たとえば、どの領域でも数学的基礎、プログラミング基礎、ネットワーク、オペレーティングシステムなどの履修を求めているが、それぞれの科目に設定する知識とスキルの正確な範囲は領域によって異なるし、時に大幅に異なる。この問題は、3.2 と 3.3 および表 3.3 でさらに検討する。こうした不整合な用語使用を考慮すると、特定のトピックがそのコンピューティング領域において何を意味しているかを明らかにするには、その領域のカリキュラム報告巻を調査するほかはない。
- トピック選択の許容範囲の差。5 領域のカリキュラム報告は、国際的に適用可能なものにしようという努力が払われているものの、多くは暗黙のうちに米国の学士プログラムを定める米国のシステムに応じたものとなっている。米国のシステムでは、多くの分野において、その学士プログラムで要求できる専門科目学習時間は、すべての分野共通する一般教育に相当量の学習時間が要求されることからの制限を受けている。この制限は、どのコンピューティング領域にも及んでいる。この米国での一般的な制約に加え、コンピューティング学士プログラムによっては、この制限された時間の内の多くを特定のトピックに当てなければならず、その結果、その学士号取得に要求されていないコンピューティング系トピックを履修する自由度が制限されることになる。たとえば、コンピュータエンジニアリングの学士プログラムは、(a)エンジニアリングプロフェッションのためのトピックと (b)CE が重視するハードウェア関連のトピックの履修に多くを割かなければならない。こうして、CE の学生のコンピューティング系の選択トピックに充てる時間は小さいものになる。情報システムの学士プログラムでも、同様の制限を受ける。IS の学生はビジネスと組織に関するトピックを履修しなくてはならないからである。どのコンピューティング領域も同じ自由度をもっているわけではないという事実から、学士プログラムについて正確にそのトピック別の学習状況を“最小”・“最大”の数値に変換するのは難しいという制約が生じている。

これらの要素の広がりとその本性から、表 3.1 と表 3.2 に示した数値は、ハードな要素、ソフトな要素をタスクフォースが総合して出したものである。したがって、ファジーな値である。各トピックの各領域の数値は、その領域の代表者が納得しているだけでなく、他の領域の代表者も納得しているという点で、現実に応じたものであるとあっていいだろう。つまり、各数値は、その領域の代表者に受け入れがたいほど低くもなく、他の領域の代表者に不当だと思われるほど高くもないものである。

3.1.2 テーブルの使用法: 関連する 2 例

表 3.1 の使用方法を示すのに、表の 4 番目と 5 番目にある“オペレーティングシステムの原理と設計”および“オペレーティングシステムの構成と利用”を取り上げる。どちらのトピックもオペレーティングシステムに関するものである。用語について不案内な方は、用語集を参照されたい。

- オペレーティングシステムが何であるかについて詳しくない読者も、用語集を見ることで、つぎのことがわかる。オペレーティングシステムとは、特定種類のソフトウェア（たとえば、Windows, Linux, UNIX, Mac OS など）であり、人間であるユーザがコンピュータとやりとりをするためのものである。オペレーティングシステムは、また、コンピュータがそのリソース（メモリ、ディスク、モニタ画面、ネットワークインタフェースなど）を管理して、ユーザの実行指示に従って種々のアプリケーションプログラム（ワープロ、表計算、Web ブラウザなど）を実行できるようにする働きをする。つまり、オペレーティングシステムは、裏方として動作して、コンピュータが役に立つ形で動作するようにするためのプログラムである。
- 用語集を見ると、“オペレーティングシステムの原理と設計”とは、オペレーティングシステムが担わなければならない役割を学生が理解するための基礎的な知識であると知ることができる。そこには、オペレーティングシステムが利用している様々な戦略と戦術も含まれており、その戦略と戦術を実現するためにオペレーティングシステムの設計者が使用するメカニズムや広く使われるアプローチの利点と弱点なども含まれている。オペレーティングシステムの役割の基礎知識の習得に加え、学生は 0 からオペレーティングシステムを作成する、あるいは、オペレーティングシステムに重要な改良を加える、といった大規模なプログラミングプロジェクトを仕上げることになることと期待してよい。
- 同様に、用語集から、“オペレーティングシステムの構成と利用”は、主要なオペレーティングシステム製品の取り扱いについて実用的に精通することがその主旨としていえることがわかる。このトピックでは、オペレーティングシステムを設計したり実装したりするための基本コンセプトとか原理とかではなく、現行のオペレーティングシステムで提供されている様々な機能を使いこなす能力を身につけることに重きが置かれる。このトピックの目的は、2 つ以上の主要なオペレーティングシステムについてその強みと制限を理解し、システムのもつ特質が、組織のポリシーや個々のユーザの要求とどの様に関連するかを理解し、さらに、ユーザの要求と組織のポリシーを実現するためのオペレーティングシステムの機能の使用法を理解している学生を育成することである。

表 3.1 を見ると、それぞれの領域がこの2つのトピックにどれほどの重きを置いているかを比較することが出来る。

- “オペレーティングシステム(OS)の原理と設計”についてみると、情報システム・インフォメーションテクノロジーのおく重みがコンピュータエンジニアリング・コンピュータ科学・ソフトウェアエンジニアリングのおくものよりも低くなっているのがわかる。IS と IT の典型的な学士プログラムでは、“最小”・“最大”とも1となっている。つまり、一般的には基本概念と用語に触れる程度で、学生が OS の原理や設計を何にがしか履修するという事はない。一方、CE, CS, SE のプログラムでは、より大きな“最小”・“最大”の値となっている。つまり、学生の習熟度は、最低標準も高く、その上限も高い。レベルは高く、CS と SE の“最小”値（それぞれ3）が CE の値（2）よりも高くなっているのは、CS と SE のカリキュラムが CE のカリキュラムよりも OS の原理とデザインについてより多く扱っていることを示唆している。CS の“最大”値（5）が CE と SE の値（4）よりも大きいのは、CS カリキュラムが希望者には OS の原理と設計に関してより深い範囲の学習を行わせる余地をもっていることを示している。
- “オペレーティングシステムの構成と利用”についてみると、学士プログラムそれぞれに異なったパターンとなっている。どの学士プログラムでも OS の利用と構成について経験できるようになっているが、IT プログラムが最も高い“最小”・“最大”の値（それぞれ3と5）を示している。これは、IT プログラムが OS の構成と利用に焦点を当てており、この分野に関して学生に特筆できる能力を持たせようとしていることを示唆している。他の領域プログラムの“最小”値は、いずれも低いもの（2）となっている。CS と SE（“最大”が4）には、CE と IS（“最大”が3）と比べて、習熟度を高めるための選択肢を用意する余地が多少ある。しかし、そうであっても IT プログラムには及ばない。

表 3.1 による比較から、OS の原理と設計を理解しようとする学生は、一般に、IS と IT のプログラムでは十分にはその希望を満たすことができず、CE, CS, SE のプログラムの方がより満足できそうであり、詳細に学ぶ最も良い機会が得られるのは CS プログラムであると結論づけることができる。一方で、実際の OS の構成や利用について興味のある学生には、IT プログラムが最も適している。他の学士プログラムでは細部にわたり習熟できる機会は少ないと思われる。両方のトピックを追求したい学生は、CS か SE かの学士プログラムに惹かれるだろうが、そこでは実際的な利用についての深さを犠牲にして、原理と利用の両面をバランス良く学ぶことになる。

3.2 学士の成果：プログラム修了者の期待値の比較

前節では5つのコンピューティング領域での学習の重み付けを比較してきた。本節では、それぞれの学士プログラムの修了者に期待される遂行能力について比較してみる。前節では学生が何を履修するかをまとめたが、本節では、修了者に期待できることをまとめている。

表 3.3 は、11 のカテゴリから約 60 の遂行能力を選んで並べてある。それぞれの能力に対して、各領域に 0～5 の数値を配している。0 は全く期待値がなく、5 は相対的に最も高い期待値があることを意味する。表 3.1、表 3.2 の値と同様に、これらの値はファジーな値である。(3.1.1 の終わりに述べたようにして決定されている。) 表 3.3 はつぎのことを示している。

- コンピュータエンジニアは、ハードウェアとソフトウェアを統合したシステムの設計と実装を行うことができる
- コンピュータ科学者は、論理的な仕事からソフトウェア開発まで幅広いレンジでの仕事を行うことができる。
- 情報システムの専門家は、情報処理の要求とビジネスプロセスを分析することができ、組織の目標に沿ったシステムの仕様化や設計を行うことができる。
- インフォメーションテクノロジーのプロフェッショナルは、組織のコンピュータインフラの計画、導入、構成、および保守を効果的に行うことができる。
- ソフトウェアエンジニアは、大規模ソフトウェアシステムのライフサイクルの、どの局面における作業であっても、適切に遂行し管理することができる。

表 3.1、表 3.2 が学士プログラムの学生に入力されるものをまとめているのに対して、表 3.3 は、出力に、すなわち修了者に期待できる相対的な能力をまとめている。

表 3.3 コンピューティング領域の修了者の相対的遂行能力

学習領域	遂行能力	CE	CS	IS	IT	SE
アルゴリズム	理論的結論の証明	3	5	1	0	3
	プログラミングの問題に対する解の開発	3	5	1	1	3
	実証デモプログラムの開発	3	5	3	1	3
	更に高速化可能かどうかの判断	3	5	1	1	3
アプリケーション	ワープロプログラムの開発	3	4	1	0	4
	ワープロの機能を使いこなす	3	3	5	5	3
	ワープロユーザの教育・支援	2	2	4	5	2
	表計算プログラムの開発	3	4	1	0	4
	表計算機能を使いこなす	2	2	5	5	3
	表計算ユーザの教育・支援	2	2	4	5	2
コンピュータプログラミング	小規模なプログラミング	5	5	3	3	5
	大規模なプログラミング	3	4	2	2	5
	システムのプログラミング	4	4	1	1	4
	新ソフトウェアシステムの開発	3	4	3	1	5
	安全性第一のシステムの作成	4	3	0	0	5
	安全性第一のプロジェクトの管理	3	2	0	0	5
ハードウェアと装置	組み込みシステムの設計	5	1	0	0	1
	組み込みシステムの実装	5	2	1	1	3
	周辺機器の設計	5	1	0	0	1
	複雑なセンサシステムの設計	5	1	0	0	1
	L S I チップの設計	5	1	0	0	1
	L S I チップ用のプログラミング	5	1	0	0	1
	コンピュータの設計	5	1	0	0	1
マンマシンインタフェース	ソフトウェアユーザインタフェースの作成	3	4	4	5	4
	グラフィックまたはゲームプログラムの作成	2	5	0	0	5
	人に優しい装置の設計	4	2	0	1	3
情報システム	情報システムの要求の定義	2	2	5	3	4
	情報システムの設計	2	3	5	3	3
	情報システムの実装	3	3	4	3	5
	情報システムの利用者の教育・支援	1	1	4	5	1
	情報システムの維持・改良	3	3	5	4	3
情報管理 (DB)	データベース管理システム(Oracle 等)の設計	2	5	1	0	4
	データベースのモデル化・設計	2	2	5	5	2
	情報検索ソフトの実装	1	5	3	3	4
	データベース製品の選定	1	3	5	5	3
	データベース製品の設定	1	2	5	5	2
	データベースの管理	1	2	5	5	2
	データベース利用者の教育・支援	2	2	5	5	2
	IT リソースプランニング	企業情報計画の策定	0	0	5	3
コンピュータリソース計画の策定		2	2	5	5	2
リソースアップグレードの計画・予算化		2	2	5	5	2
コンピュータの設置とアップグレード		4	3	3	5	3
ソフトのインストールとアップグレード		3	3	3	5	3
知的システム	自動推論システムの設計	2	4	0	0	2
	知的システムの実装	2	4	0	0	4

学習領域	遂行能力	CE	CS	IS	IT	SE
ネットワークと通信	ネットワーク構成の設計	3	3	3	4	2
	ネットワーク装置の選定	2	2	4	5	2
	コンピュータネットワークのインストール	2	1	3	5	2
	コンピュータネットワークの管理	3	3	3	5	3
	通信ソフトの実装	5	4	1	1	4
	通信資源の管理	1	0	3	5	0
	モバイルコンピューティングシステムの実装	5	3	0	1	3
	モバイルコンピューティング資源の管理	3	2	2	4	2
システム開発から統合	組織のWebを管理する	2	2	4	5	2
	Eコマースソフトの構成・統合	2	3	4	5	4
	マルチメディアソリューションの開発	2	3	4	5	3
	Eラーニングシステムの構成・統合	1	2	5	5	3
	業務システムの開発	1	2	5	3	2
	サーチエンジンの新形式の評価	2	4	4	4	4

3.3 国際的な相違点

この報告も、参照しているコンピュータティン グ カリキュラム シリーズの5報告巻も、国際的な入力、特に英国からの入力に負うところがある。タスクフォースは、国際的にも意味をもった助言と解説を行うことを目標としている。今後の活動のなかで国際的に拡大していく事になるが、ここまでも、いくつかの点について興味深い比較が得られたという成果があった。たとえば、違いが見つかったものに、学年度の構造、学士プログラム内でのコンピュータティン グ履修の配分（たとえば、英国ではコンピュータティン グ学士プログラムのほとんどすべての科目がコンピュータティン グに特化したものである）、および、品質管理の仕組み（たとえば、4.5 で述べるように、認定に対する期待と実施）などがある。加えて、学士プログラムの焦点を定義する方式もさまざまである。

- 米国では、それぞれの領域がそのコアを非常に強く意識している。領域のコアとは、その領域にとっての基幹部分となる一連の要素であって、その領域に学ぶ学生すべてが身に付けるべきものとして指定されている。このような方式には少なくとも2つの利点がある。一つは卒業生に対して期待できる能力について共通の理解を形成できることであり、もう一つは教育機関の間での移動を可能にするということである。コアをどのように定義するか議論を経ての合意形成は、その領域の確固たる定義を与えるのに役立つし、学士号の意味するところを明瞭に理解する助けともなる。
- コアという考え方は、英国などの国では主流でない。英国では、学士号は就職機会を生み出すためのものと見られている。このため、米国流に比べ、英国にはより多様なコンピュータティン グ学士プログラム名がある。それぞれの学士号は、異なる履修配分を反映し、異なるキャリア機会を反映したものである。英国では、学士号に関する指針と品質保証を、基準の設定と一群の仕組みによって扱っている。これらについては3.5で説明する。

タスクフォースは、こうしたモデルに関するその他の国の情報を十分には得ていない。世界各国の様々な方式についての徹底的な評価は、今後の重要な課題である。

3.4 大学の変化の速さ：領域と学士プログラム

前述したように、1990年代のコンピュータティン グの爆発的な増加により、コンピュータティン グ学士プログラムの景観は著しく変化してきている。コンピュータティン グの分野も進展して、新しい学士プログラムが生まれ、コンピュータティン グ プロフェッションとコンピュータ依存の社会とがともに一体として直面している課題に焦点を当てるようになるまでに至った。

2章で述べたコンピュータティン グの発展は、多くの場合、提供されている学士プログラムの種類に反映されていない。北米では、専科大学・総合大学でごく少数が主要5領域のすべての学士プログラムを提供しているに過ぎない。比較的新しい2種類の学士プログラム、ITとSEは、確立されたCE、CS、ISに比べるとまだまだ少数である。高等教育機関は、慎重で保守的な傾向があり、大学での学士プログラムが本来もつ複雑性により、大幅な変化を急激に起こすことが難しい。したがって、教育機関の中には、コンピュータティン グ学士プログラムの選択肢が、図2.1の1990年代以降のものではなくて、1990年代より前の様相を呈している所もある。これは、コンピュータティン グ分野の変化が非常に速いにもかかわらず、教育機関の変化が一般的に非常に遅いためである。

この教育機関の本性的な遅れは、自らの興味や目標に適合するようなコンピューティング学士プログラムを選択しようとしている学生にとって、問題となっている可能性がある。また、教育機関にあってその構成員にとって最も有益な最新の学士プログラム群構成にしようとしている教育者にとっても、問題となっている可能性がある。多くの教育機関では、この遅れを最小に抑えようと教員が努力して、学士プログラムの最終年度に専門分野の最先端の学習や調査に費やす機会を設けている。

この本性的な遅れを別にしても、重要かつ根本的な変化が起きている徴候が広範囲にわたって見られる。過去 20 年余りに及ぶコンピューティングの発展を見てみると、重点を置く対象が劇的に変化していて、重点は相互作用に向かって移動しており、その分、おそらくはアルゴリズム学習への重みが減ってきている。相互作用への移行は、コンピューティングの成功を示す重要な表象である。初期のころに比べてみれば、はるかに広い範囲の人々がコンピュータを利用し、はるかに深い度合いでコンピュータと情報のやりとりを行っている、という事実が浮き彫りにしている。新しい教育プログラムはこの事実を反映するものであり、IS プログラムの大多数や最近の IT プログラムの出現が、これの現れであることは極めて自然なことである。プログラムがより豊富で多岐にわたるものとなる余地は大きい。そして、コンピューティングにおける変化が持続していくことが期待されている。

教育機関の遅れは何らかの結果を招いているはずであるから、読者は注意して、そうした結果の存在に気付き、与えられた教育機関がそれを克服する手段を講じている兆候を見つけ出すことが肝要である。一般論として、教育機関の遅れが特定の教育プログラムや学科にとって深刻な問題とはなっていないことを示す、品質の測量法とか目印とかといえるものがある。そうした指標として、つぎのものがある。

- 活発な顧問団があつて、産業界からのアドバイザーの声を聞いている。
- プログラムを点検したり変更の目的を定義したりする委員会に学生を参画させている。
- 教育機関の質を制御する委員会があり、積極的に外部の専門家のアドバイスを求めてそれを受け止めている。
- 卒業生の雇用統計をとり、プログラム修了者および教育機関そのものの評判を示す指標としている。
- 学士プログラムの認定（詳細は 4.5 参照）を受けていて、品質の表示をするとともに、その教育プログラムの内容が時代に即応できていることなどの点検を行っている。

ここからは、それぞれの領域固有の課題をまとめる。

3.4.1 コンピュータエンジニアリング

1990年代よりも前のコンピューティングの見方から1990年代以降の見方への変化の典型に、コンピュータエンジニアリングが電子工学の内から生まれて完全に独立したことがある。一番多いのは、この二つのプログラムが一つの学科に置かれていることであるが、コンピュータエンジニアリングプログラムがコンピュータ科学とコンピュータエンジニアリングの合同学科に置かれていることも多い。国際的には当てはまらないが、米国では、コンピュータハードウェアを学ぶ者の拠点は常に唯一に定まった。かつては電子工学であったし、現在ではコンピュータエンジニアリングである。ほとんどの大学では、1990年代以前のものから1990年代以降のものへのこの変化はすでに起きてしまった。しかし、一流大学の中にも、独立したコンピュータエンジニアリングプログラムを提供していない大学がある。電子工学とコンピュータ科学とが同じ学部にあって、コンピュータエンジニアリングとは、これら2つの領域の教員の中で興味を一にする者が自然に醸し出していることのできる大学もある。その教育機関の学科構成にコンピュータエンジニアリングが陽には見受けられなかったからといって、かならずしも劣っているとは限らないのである。

コンピュータエンジニアリングのプログラムは、コンピュータ科学や情報システムのプログラムに比べると著しく少ない。これは、米国では単科大学や総合大学の多くがエンジニアリングプログラムを何一つ設けていないことに起因する。米国でのエンジニアリングプログラムは、偏って存在し、比較的大きな大学や工学専門大学で、エンジニアリングの学士プログラムの認定に足る要件を満たしている大学にだけ存在している。この状況は、将来的に著しく変わることはないであろう。

3.4.2 コンピュータ科学

米国においては、コンピュータ科学の学士プログラムが他のコンピューティング領域プログラム全部よりも多く存在する。単科大学も総合大学も、そのほとんどがCS学士プログラムを設けている。これは、ある程度は歴史的な経緯によるものである。大学でのコンピューティングの学士プログラムが1970年代に出現した際、コンピュータ科学は、ソフトウェア開発に明示的に焦点を当てた、唯一のコンピューティング領域であった。ほとんどの大学がコンピューティング学士プログラムを設立した際、コンピューティング科学は、数学、科学および／または工学と強い結びつきをもった唯一の選択肢であった。(ISプログラムも同時期に発展したが、主に、ビジネス学部との結びつきが強かった。)より規模の小さな大学では、CS学士プログラムが“数学およびCS”学科に含められることもあるが、これらの学科のほとんどが2つの明確に区別された学士プログラムを設けている。

1990年代以降に目の当たりにしたコンピューティング領域の多様化は、コンピュータ科学に影響を及ぼす領域にほぼ限定されている。新しいコンピューティング領域が対象とするキャリア区分は、もともとはCSプログラムの卒業生によって埋められていたものである。

現在、議論が続いているのは、コンピュータ科学プログラムで教えていることと、コンピュータ科学の卒業生の多くがそのキャリアの中で実際に行っていることとの関係についてである。この議論を理解するには、2.3.2で述べたコンピュータ科学の特徴づけを復習する必要がある。コンピュータ科学者の仕事は、3つのカテゴリに分けられる。ソフトウェアを設計し実装すること、コンピュータの新しい利用方法を考案すること、そして、コンピューティングの問題を解く効率的な方法を開拓することである。これらそれぞれのキャリアパスに何が含まれているのかを見てみよう。

● キャリアパス 1: ソフトウェアを設計し実装する

これは、ソフトウェア開発の仕事を目指す。この仕事は Web 開発、インタフェース設計、セキュリティ関連案件、モバイル コンピューティング等を含むまでに拡大している。このキャリアパスは、最も多くのコンピュータ科学の卒業生が選ぶ道である。このようなキャリアパスに入るには、一般に学士号の取得で十分である一方で、多くのソフトウェア専門家が再び大学に戻って修士号を取得している。(博士号は滅多に必要とならない。) キャリアの機会が、広範囲に存在していて、大規模・小規模のソフトウェア開発会社、大規模・小規模のコンピュータ関連サービス会社、そして、多種の大規模組織(産業界、政府機関、銀行、医療等)がその対象になる。ソフトウェア工学の学士プログラムもまた、学生に対してこの種のキャリアパスに対応した教育を行う。

● キャリアパス 2: コンピュータの新しい利用方法を考案する

これは、コンピュータ技術の応用に関する革新をいう。この分野のキャリアパスには、大学院に進んで更に学習し、その上で研究大学や産業界の研究開発機関に席を得て仕事することや、1990年代のドットコムブームに見られたような起業して活動することや、あるいはその両方の組合せが含まれる。

● キャリアパス 3: コンピューティングの問題を解く効率的な方法を開拓する

これは、コンピュータ科学の理論やアルゴリズムの知識を応用したり開発したりして、コンピューティング集約的な問題に対する最善の解を保証する仕事をいう。実際問題として、新しいコンピュータ科学の理論の発展にかかわるこのキャリアパスには大学院博士課程での研鑽を必要とするのが普通であり、その後には研究大学または産業界の研究開発機関に席を得て仕事をするようになる。

コンピュータ科学プログラムは、一般的に、上記3つのキャリアパスのいずれかを学生が歩むための準備をすることを意図している。これに加え、CS プログラムが目的とはしていないものの、多くのコンピュータ科学の卒業生がたどる4つめのキャリアパスが存在する。

● キャリアパス 4: 組織的な技術基盤を計画し管理する

これは、新しいインフォメーションテクノロジー(IT)プログラムがその学生教育の目的とする仕事をいう。

上記4つのキャリアパスのうち、キャリアパス2および3は、コンピュータ科学のアイデンティティの重要な要素であり、多くのコンピュータ科学の教員が学生に選択してもらいたいと考えているようなキャリアパスである。しかしながら、実際的な観点から言うと、コンピュータ科学の学士号取得者のうちのごくごく少数の者だけがこのようなキャリアパスを選択している。この少数派にとって、教育機関の対応の遅れは問題ではないのだ。権威あるコンピュータ科学の学士プログラムを終え、続いて大学院で学ぶ（おそらくは博士号レベルまで）というのが、彼らにとる選択なのである。

キャリアパス1および4が議論の焦点である。これらのキャリアは、コンピュータ科学の卒業生の圧倒的大多数を引き取っている。これらのキャリアは、また、新しい領域（ソフトウェアエンジニアリングとインフォメーションテクノロジー）が焦点を当てているものでもある。これらの領域は、学生がこのようなキャリアパスに進むための準備により照準を合わせて、コンピュータ科学プログラムに代わる選択肢として誕生したのである。さらに、情報システムの多くの卒業生が、長年にわたり、これらのキャリアパスに極めてよく似た組織上の職務を選択してきた。コンピュータ科学のプログラムとソフトウェアエンジニアリング・インフォメーションテクノロジーのプログラムの相対的価値に関する論議は、未だ解決を見いだしていない。しかし、論議の中で提起された課題は、それぞれ、下のソフトウェアエンジニアリング・インフォメーションテクノロジーの項で明らかにし議論することにする。キャリアパスの1および4については、コンピュータ科学プログラムの価値を評価するに際しては、2つの新しいコンピューティング学士プログラムが出現したことを考慮して、それらとの比較の下で行う必要がある。

3.4.3 情報システム

情報システムの役割における変化は、情報技術の役割がすべての組織内において拡大したことに呼応している。歴史的には、情報システムプログラムは、給与管理、売掛金管理、在庫管理等のビジネス応用に対応できるように学生を教育し準備させるものであった。技術的側面でいうと、ISの学生に期待されているのは、伝統的なビジネス分野におけるコンピュータ応用、特にデータベース管理システムに精通していること、さらに、表計算その他の市販ソフトウェアなど、ビジネス界の人々が広く実用しているものに精通していることであった。

現代のISプログラムは、広い範囲の企業における、ITで可能となった情報活用やビジネスプロセスの拡大した役割に焦点を合わせた上で、経営学部との密接な結びつきを保っている。企業はどのような情報を必要としているのか。情報はどのように生成されるのか。情報は必要としている人に届けられているか。届けられる情報はすぐに利用できる形になっているか。組織の構成は技術を効果的に利用できるものになっているか。組織のビジネス・プロセスは適切に設計されているか。ビジネスプロセスは情報技術が作り出す好機を完全に活かしているか。組織は情報技術による会話と協調の可能性を活かし切っているか。組織は変化する外部環境に十分迅速に適応することができるか。これらは、重要な事項であり、さまざまな企業がますますIS出身者にその取り組みを依存しようとしていることである。

IS プログラムにとっては、伝統的な役割が今なお存在するものの、それだけではもはや十分でないのである。こう問うてみるとよい。「IS プログラムは、その守備範囲を広げ、企業がもつ、複雑な情報の要求と、IT 利用ビジネスプロセスへの高度な依存とに対する総合的視点も包含するまでになっているのか。」と。ウェブを基盤として流通した技術は、地球規模の結びつきを持つ組織に対して基礎構造を提供するものとなった。現代の IS プログラムはそのような組織のニーズに対応しなければならない。IS の学生が学ばなければならないことがらには、組織的な情報のニーズをどのようにして見積もり評価するのか、情報要求をどのようにして特定するのか、こうした要求を満たす実用的なシステムをどのようにして設計するのか、などがある。プログラムがその焦点を絞って、狭い機能範囲のアプリケーションの設計・開発や個人の生産性にかかわるツールの使い方に限定していたとすると、そのプログラムは IS プログラムの主流には徹底的に遅れてしまっている。

これらの懸案事項に加え、IS は IT プログラムが出現したことも考慮に入れなければならない。伝統的に IS プログラムの多くの卒業生が果たしてきた役割は、IT プログラムがその学生に準備させる対象としている役割に非常に類似している。IT プログラム数が増加するにつれて、多くの IS 学科が、そのコアとする構成要素をどのように定義し、どのように教えるべきかを考えることになるだろう。

3.4.4 インフォメーションテクノロジー

ここ数年の間に、インフォメーションテクノロジー(IT)の学士プログラムが出現し発展して、コンピューティング学士プログラムの議論といえば必ずその重要な部分を占めるほどまでになった。2.3.4 でまとめたとおり、IT プログラムは、幅広い環境においてどうやって情報技術を機能させるか、に精通している卒業生を出すことに焦点を合わせている。全種類の組織がネットワーク化されたコンピューティング基盤に依存するようになっており、この基盤なしには機能しえない程までに至っている。IT の学生は、こうした基盤の選択・管理・維持を行って組織のニーズを充足させることができるように教育される。また、この基盤のためのデジタルコンテンツを作成し、これを利用する個人の IT 支援に対するニーズに対応する。

IT プログラムの出現は、地域社会と学生双方の現実のニーズに対応しようとするコンピューティング教育者による草の根運動を象徴するものである。IT プログラムが生まれたのは、コンピュータ科学領域や情報システム領域のプログラムがその役目を果たしていないというのではなく、これらの領域が自らの役目を別のものに定めているからなのだ。IT プログラムの存在は、コンピューティングにおけるキャリア機会の変化の一部を反映している。

ほんの数年前まで、米国のコンピューティング教育者には IT 学士プログラムになじみがなかった。他国では、類似のプログラムがすでに何年にもわたって存在していた。今日では、こうしたプログラムは数多く存在しているし、今後もその数は増えてゆくものと思われる。2001 年になって初めて北米の大学レベル IT 教育者が組織作りを開始し、それ以降の短期間でプロフェッショナル団体が形成され、何度か会議を開き、IT 学士プログラムのカリキュラムと認定の指針を策定するところまで進んだ。IT プログラムが爆発的な勢いで米国社会を席卷したといっても過言ではない。

IT プログラムは一時的な流行なのではないかという疑問をもっている人がいる。IT プログラムは学術的領域としての地位を受けるには技術的すぎるのではないかという人もいる。30 年以上も前には、コンピュータ科学に対して同じような疑問が投げかけられたが、北米の大学の大多数が CS 学士プログラムを設置し始めた。IT に関しても同じような結末を見るかもしれない。IT 学士プログラムは、社会全般に広がっている重要なニーズに適うものである。コンピュータ技術に依存する組織に至っては、IT 分野が重要な役割を果たすのである。

ここで、重要な課題が 2 つある。

- 厳密性 組織の IT 基盤を設計し管理するという仕事は、困難かつ複雑なものであり、応用コンピューティングに関する確たる基礎が要求され、管理能力や人間力も要求される。IT 領域の人には特別なスキルが必要であり、とくにネットワーク・システムがどのように構成され構築されているか、その長所および短所は何か、などを理解する力が必要である。ソフトウェアシステムに関しては、目的に応じた信頼性、安全性、有用性、有効性、能率性などの重要な事項が存在し、いずれもが不可欠である。これらの学習項目は、いずれも難しく、相当の知力を必要とする。
- 受入れ 米国において、IT 領域は新入りであり、その結果、すでに確立された領域からはなかなか受け入れられない。これは自然な現象であり、すでに確立されたコンピューティング領域の人々が、IT 領域のもつ価値を評価し、認識するには時間がかかり経験が必要となるだろう。IT は、自らを領域として確立しようと、独自の知識コア、厳密なカリキュラム、そして認定の指針を整備しつつある。このような挑戦を乗り越えたとき、IT は自然と受け入れられ、敬意を受けるようになるに違いない。

多くの教育機関において、運営陣は、既設の IT プログラムが社会のニーズに応え、入学志望者により多くの選択肢を与えていることに気付き始めた。教育機関が IT プログラムを設置するにあたっては、そのプログラムが適正に実現されるよう、細心の注意を払う必要がある。IT 学士プログラムに責任をもつ人たちが IT の重要性を認識していること、IT 学生に対する高品質な教育経験を作り出すことに意欲をもっていることが不可欠である。

3.4.5 ソフトウェアエンジニアリング

ソフトウェアエンジニアリング(SE)プログラムが発展したのは、ソフトウェアシステムを適切に理解し開発することのできる卒業生を輩出する学士プログラムの不足という現実的な問題に対応するためであった。CS プログラムは、その卒業生として基礎プログラミングの確実なスキルをもった人材を輩出していた。しかし、多くの人は、CS プログラムが、基礎プログラミング以上にエンジニアリングの専門的技術を必要とする複雑なソフトウェアシステムについて効果的に仕事ができるような卒業生を輩出してはいないと考えている。1990 年代以降の世界においては、多くのソフトウェアプロジェクトは大規模かつ複雑となっている。そして、ソフトウェアの信頼性が高く、ユーザの目的とニーズに合致し、予定通りに予算内で開発されることを実現するような専門的な実務を行うことのできるソフトウェアエンジニアに対する切迫したニーズがある。SE プログラムとは、学部生に対する教育を、ソフトウェア専門家としてのキャリアのための適切な知識とスキルをもった学生を輩出しようとする努力の結果、生まれたものである。

実際問題として、CS と SE の学士プログラムには多くの共通点がある。いずれの領域も、学生がこの分野全体を理解できるような単一の学士プログラムなど存在しないまでにコンピューティングの対象が拡大していることを認識している。30 年前であれば、CS の卒業生がすべてを学んだと考えても間違いではなかった。今日では、4 年間（もしくは 5 年間）の学習プログラムに納まるにはあまりに多くの学習対象が存在している。

一般的に、CS と SE のカリキュラムでは、プログラミングの基礎および CS 理論の基本が必修となっている。コアをなす要素の先で何に焦点を合わせるか、という点で二つは枝分かれしているのである。CS プログラムはコアを小さく保ち、CS が集積している学習域（システム、ネットワーク、データベース、人口知能、理論等）の上級コースを学生に選ばせようとする傾向にある。これとは対照的に、SE プログラムは、概して、SE の基本方針に不可欠な題材（問題のモデル化および分析、ソフトウェア設計、ソフトウェアの検証および妥当性確認、ソフトウェアの品質、ソフトウェアプロセス、ソフトウェア管理等）に学生を集中させようとする。一般的に、CS、SE の両プログラムとも学生にチームプロジェクト活動を経験させているが、SE プログラムの方がより多く携わらせている傾向にある。これは、効率的なチームプロセスが SE の実践には不可欠だからである。さらに、SE カリキュラムの指針に明記されている重要な要件として、顧客にとって真に実用的で使用可能なソフトウェア、また、定められた要件をすべて満たすようなソフトウェアを、どのように構築するかを学ばなければならない、とある。

SE プログラムがどのように発展するかについて問う、つぎの 2 つの質問に対する答えはまだ出されていない。

- SE 学士プログラムは CS 学科の中で、あるいは中から生まれたが、伝統的な CS 学士プログラムの代替物として隣り合う形でどこまで進みうるのか。

そのようなトレンドは不可避であると言う人もいる一方、そのような発展は必要ないという人もいる。学生を競争力あるソフトウェアプロフェッショナルにするために学部生向けの教育で必要とされる徹底した SE 経験の質と量に関しては、教員の間でも意見が分かれている。

- エンジニアリングという用語を付していることの結果として、どの程度の米国の単科大学や総合大学が SE 学士プログラムに異なる名称を付けることを選択してしまうだろうか。

米国においても、他のいくつかの国においても、エンジニアリングプロフェッショナルの集団は、自らのアイデンティティを守ろうとするし、エンジニアリングという用語もそのアイデンティティを示すものとして守ろうとする。米国やカナダには、エンジニアリングという用語の使用が規制されている州すらある。ソフトウェアエンジニアリングは、（ソフトウェアが無形物であることや、物理法則よりもむしろ人的プロセスに焦点を合わせていることなどから）伝統的なエンジニアリング領域とは異なるものである。伝統的なエンジニアリングの規律（信頼のおけるものを作り出すための確固たる手法）は、依然 SE 教程のコアを成すものである。ソフトウェアエンジニアリングという名称の一部にエンジニアリングという用語を用いる限り、米国においては、認定に限定がつき、そして教育機関にとっては大学で SE を置ける場所に限定がつくことになる。この結果、米国のプログラムの中には SE という名称を採用しないものがでてくるかもしれない。英国では、このような問題は数十年前に解決済みである。

ほとんどの CS の学生は、卒業後のプロフェッショナルキャリアがソフトウェアの開発を含むものになると思っているようである。もし多数の SE 学士プログラムが存在していれば、CS の学生の多くはそれを選択したのではないかと思われる。現時点では、そうした選択の余地は少ない。米国内には、今、約 30 の SE 学士プログラムが存在する。IT プログラムの増加に見られたような増加率は望めないにしても、この数は増加するであろう。

CS 学士プログラムについてみると、そこで行われている SE 教育の程度はさまざまである。最新の CS 用カリキュラム指針 (CC2001) では、最低限行うべき SE 教育を定めている。ほとんどの CS 学士プログラムはこの最低限の範囲を超えて、SE 科目を 1 科目以上設けている。CS プログラムの中には、数個の CS 専門学習域の一つとして SE を設けているプログラムもある。そのような選択肢を設けている CS プログラムを評価する際には、その選択肢でどの程度まで厳格な教育を行っているかを見定めることが大切である。

3.5 職場の変化の速さ：学士号とキャリア機会

教育機関が適応するのに時間がかかっているのと同様に、産業界その他のコンピューティング関係の職場が適応するにも時間がかかる。その結果、雇用機会のある部分は、図 2.3 の上部に示されているような 1990 年代より前のモデルを反映したものになっている。換言すれば、新規のコンピューティング領域は新しいものであるがゆえに、その資格をもっていることが未だ要求されていないのである。

CE に関して言えば、新しい領域は問題とはならない。ハードウェアやそれに関連するソフトウェアを専門とする者がどこに行けばよいのかは明白だからである。CE はその役割を EE から受け継ぎ、変化は大部分において完了している。CE は、EE がそうであったように、エンジニアリングプロフェッショナル集団の一部であり、米国においては、これは CE の学士号をどこで取得できるかも規定しまっている。すべての大学がエンジニアリングの学士号を出しているわけではない。CE のバックグラウンドとする人材を探している雇用者は、認定された CE プログラムの卒業生を求めるのである。

ビジネスとコンピューティングの情報ニーズ間のインタフェースを専門とする人材がどこで見つけられるか、ということもまた明白である。1990 年代の前にも 1990 年代以降も、IS がその役割を果たしている。この進路を選ぼうとする人は、IS プログラムの中でも、ビジョンを継続的に更新し、自らの興味に合致するものに重点を置いているものを探し出すようにしなければならない。IS の志望者にとって、適切なプログラムを見つけるのは、そのようなプログラムにはさまざまな名称が用いられているために、手間のかかる作業となる。

現在、米国で本格的なソフトウェアのエンジニアとして働いている人々のほとんどは、SE ではなく CS の学士号を持っている。これは主に、30 年もの間、CS の学士号が広い範囲で取得することが可能であった一方、SE の学士号はそうではなかったからである。IT 専門業界で働く人々の経歴についても同様である。ほとんどの IT 専門家は CS または IS のプログラムで授けられたコンピューティング学士号をもっている。ソフトウェアエンジニアや IT 実務家として働こうと志望している者が、新規領域の学士プログラムを卒業しないと仕事のチャンスがないのではないかと心配するのは時期尚早に過ぎるのである。一般的に、定評あるプログラムで取得した CS の学士号は、もっとも適応性のある学士号であり、CS、SE および IT、そしてときには CE の専門的業界への扉を開くことが可能である。また、定評ある IS プログラムで取得した学士号は IS と IT の両方のキャリアに進むことを可能とする。

この状況は、どの領域を学ぶかを決めようとしている学生や、どうやってプログラムを関係者にとって最良のものにしていくかを決定しようとしている教育者に、あいまいさを残してしまっている。教育者が考慮すべき事柄は、第 4 章に述べる。学生が考慮すべき事柄は、手引きに述べる。手引きは、この概要報告の第 2 部にあたるもので、単独の冊子として用意する予定である。

メディアがアウトソーシングや海外への外注、業務移管に関心を向けたことにより、多くの人々がコンピューティング関連キャリアの将来について懸念することとなった。このような問題は、この報告の対象外のものである。英国に影響を及ぼす問題としてこのような問題を扱った英国コンピュータ協会 (British Computer Society) の報告は、つぎのサイトで見ることができる。

<http://www.bcs.org/BCS/News/PositionsAndResponses/Positions/offshore/offshorereport.htm>
ACM の業務移管タスクフォース (Job Migration Task Force) が、2005 年の第 4 四半期に報告を発行する予定である。ACM のこの報告は、米国中心の観点ではなく、国際的な観点を反映することを意図している。作成が完了し次第、つぎのサイトで見ることが可能となる予定である。

<http://www.acm.org/jmtf>

3.6 共通のアイデンティティ：コンピューティング学士号に共通の要件

これまで見てきたとおり、主要なコンピューティング領域にはそれぞれ特徴がある。それぞれに、重点を置いている課題、目標、卒業生の能力などが、他のものとは少しずつ異なる。しかしながら、なお共通点も多い。評判のコンピューティング学士プログラムとなるには、つぎの各要件を満たさなければならない。

- 1) その領域の本質的で基幹となる土台。抽象的になるが、例えば、数学に根ざした形式的な理論、専門的価値や原則に関するもの等。土台とは、その形式や対象にかかわらず、技術的变化に直面しても不変であるような、その領域の本質的な側面を浮き彫りにするものでなければならない。領域の基幹とは、時間や環境を超越した試金石となるものであり、その教育的使命に永続性と安定とを与えるものである。学生はその基幹にしっかりと根を下ろさなければならない。

- 2) コンピュータプログラミングの諸概念やスキルの基礎。この基礎はつぎの5つの層から成る。
 - a) アルゴリズムおよびデータ構造が果たす中心的役割の知的理解と正しい評価。
 - b) ソフトウェアの観点からのコンピュータハードウェアの理解。例えば、プロセッサ、メモリ、ディスクドライブ、画面の使用等。
 - c) ソフトウェアにおけるアルゴリズムやデータ構造の実装を可能とする、プログラミングの基礎的なスキル
 - d) アルゴリズムおよびデータ構造を用いた規模の大きい構成単位や、このような構成単位が通信を行うのに利用するインタフェースを設計し実装するのに必要とされるスキル
 - e) ソフトウェアの実装が厳密で、信頼性が高く、対象ユーザにとって適切であることを確実にするための、ソフトウェアエンジニアリングの原則および技術
- 3) コンピュータ技術（ソフトウェア、ハードウェア、ネットワーク）には何ができて、何ができないのか、その可能性と限界に対する理解。これにはつぎの3つのレベルがある。
 - a) 現在の技術では、何ができて、何が達成できていないのかに関する理解。
 - b) コンピューティングが本質的に実現しえないこと、それに対して将来の科学や技術によって達成しうることを含む、コンピューティングの限界に関する理解。
 - c) 技術的な解決策や介入による個人、組織、社会に対する影響。
- 4) 各段階（設計、開発、展開、進化）の意義を含む、ライフサイクルの概念、コンピュータ関連システムのすべての側面（ソフトウェア、ハードウェア、人間・コンピュータ間のインタフェースを含む）の開発にかかる意味合い、およびライフサイクル管理と品質との関係についての理解。
- 5) 重要な概念であるプロセスについて、少なくとも、つぎの2つの意味の理解。
 - a) プログラム実行やシステム運用など、コンピューティングに関するプロセス。
 - b) 製品の開発期間中の適切な人員配置と製品品質との関係の作用など、専門的な活動に関するプロセス。
- 6) 学生が分野の最先端を見て理解することができるような上級のコンピューティング関連トピックの学習。一般的に、これは、最先端の進歩が浸透した、学生を初歩的なトピックから上級のトピックやテーマに導くような学習体験によって達成される。
- 7) 技術的スキル以上のスキルの検証、習得。そのようなスキルには、各学科の必要に応じて、対人コミュニケーションスキル、チームスキル、管理スキルが含まれる。価値あるものにするためには、学習体験はこれらのスキルを確立するものでなければならない（重要なことだとただ伝えるだけではだめである）。また、新しい状況に転換可能なスキルを教えるものでなければならない。
- 8) 適切な範囲のアプリケーションやケーススタディに触れさせ、大学で学んだ理論とスキルを実際社会の出来事に結びつけて、その妥当性と実用性を理解させること。
- 9) 専門的、法的、倫理的な事項に注意を払わせることによって、専門職業での倫理的器量と立場を尊び、守り、強化する態度と優先付けを、学生に体得させ、磨かせ、示せるまでにする事。
- 10) それぞれの学生が学部生として経験した種々の要素を統合して、卒業プロジェクトを遂行し、完成し、発表する成果提示。

第4章 教育機関の検討事項

4.1 コンピューティング学士プログラムの進化と現状

多くの教育機関は、そのコンピューティング学士プログラムを発展させていく最適な方法について重要な判断を迫られている。しかし、すべての単科大学・総合大学が同じ問題に直面しているわけではない。幸いにも、少数のプログラム特性を特徴付けし、北米の単科大学・総合大学の大多数がそれらのパラメタに基づいて各々の状況を認識できるようにすることができる。まず、コンピューティング専門領域を順に取り上げながら、機関にかかわる属性を特徴付けしていく。

コンピュータ工学 (CE) CE プログラムを提供するかしないかのジレンマに直面している教育機関はほとんどない。エンジニアリング学部 (工学部) (もしくは類似の部門) をもつ教育機関のほとんどが既に何らかの形で CE プログラムを提供しているからである。ほとんどの場合は明確に CE プログラムという形をとっているが、一部の教育機関では CS プログラムと EE プログラムが協力して提供する形で同じ内容のプログラムを提供している。2004 年の半ばにおいて、米国ではコンピュータエンジニアリングに約 500 個のプログラムが存在しているし、さらに何千ものものが米国以外に存在している。エンジニアリングプログラムを一つももっていない北米の教育機関が認定対象となる CE プログラムを設立する可能性は薄い。実質的な支援管理体制がなくては認定の要件を満たすことが困難だからである。

コンピュータ科学 (CS) 事実上、CS プログラムを提供するかしないかのジレンマに直面している教育機関はほとんどない。ほとんどすべての単科大学・総合大学が既に提供しているからである。CS は、コンピューティング学士プログラムで最もユビキタスなものである。最も一般的なシナリオとして、1970 年代にコンピュータ科学は数学科の中から生まれ出た。しばしば、そうした学科は “数学・コンピュータ科学” 学科と姿を変えて 10 年 20 年存続して 1990 年代に入って数学科と CS 学科とに分れたが、現在もなお “数学・コンピュータ科学” 学科が多数残っている。電気工学プログラムをもった大学では、CS は電気工学科から生まれるか、数学科から生まれるか、あるいは両学科から生まれるかした。これらのシナリオが代表的なものではあったが、別のシナリオも多数存在していた。1990 年代になると、少数ながらコンピューティングのもつ広範な影響力に気付いた教育機関が、コンピューティング学部とかコンピューティング単科大学とかといった自立した組織を開設した。これは洞察力と先見性を示した動きであった。教育機関が CS プログラムをどこに設置したかによらず、おおくの CS プログラムは、1970 年代・80 年代にあっては周りの学科・学部の認知を得るのに苦労を重ね、1990 年代になってようやく確立し名の知られたプログラムとして受け入れられた。ほとんどの教育機関で、CS プログラムは、互いに密接に関係しあった、つぎの 2 つのことがらに関して決断を迫られている (そうなるはずである)。(1) 狭い意味での CS アイデンティティは、3.2 で議論した課題から見て、妥当なものであるか。そして、(2) 新たなコンピューティング領域の出現への対応を、3.2 で議論した課題から見て、どのように行うべきか。

情報システム (IS) 2005 年の半ばにおいて、米国には約 1000 個の IS プログラムが存在する。しかし、それぞれの具体的なプログラム名は、情報システム、経営情報システム、コンピュータ情報システムなどとさまざまである。また、世界中には更に何百という IS プログラムが存在する。その名称も、また、多様である。コンピュータ科学と同じく初期のプログラムは 1960 年代に出現し、ほとんどは 1970 年代及び 1980 年代に出現した。また、CS プログラムと同様に、現在 IS プログラムは IT プログラムとの新たな競争に直面している。一部の IS プログラムの教員は、自分たちの関心が薄い分野の責任を新しい IT プログラムが取ってくれることに喜び、その他の教員は影響力を失い自分の芝生を失うのではないかと恐れている。IT プログラムの出現に対するこうした IS の個別の反応に関わらず、IS と IT のコミュニティが協力しあう機会には豊富に存在している。

情報技術 (IT) 2005 年の半ばにおいて、約 70 のアメリカの教育機関が IT 学士プログラムを提供している。こうしたプログラムの伸びは急激なので、現時点でいくつ存在するかを把握するのはとても難しい。現存の多くのプログラムはまさに先駆者である。これらは、海図もない領海に作りあげられた品質の高いプログラムであり、生まれつつある新領域の特別なニーズに合わせた新しい科目群を提供する。残念ながら、中には品質が低く、学生やコミュニティに責任ある対応ができていないものもある。後者のグループは、入学者数を増やしたり、地域のニーズにすぐ応えるというイメージを与えたりすることが目的で、作った IT プログラムが他の領域の既存の科目を組み直しただけに過ぎない。ほとんどの教育機関はまだ IT プログラムを提供していないが、その多くがプログラムを始めるかどうか判断している最中である。始めることに決めた教育機関は、特別に注意を払って高品質な IT プログラムのリーダーシップ例にならうように努めることが大切で、間違っても低品質の搾取的なプログラム例にならってはならない。IT2005 カリキュラム報告巻を作ろうとするに至ったのは、高品質な IT プログラムを開発しようと試みている教育機関に対して実質的な指針を提供したいという願いからである。

IT プログラムを検討している教育機関への鍵となる質問は、“その IT 計画は教育機関の設置目的に合っていますか？”である。IT プログラムを作ると決めることは、よく訓練された IT 実務家が欲しいという地域のニーズに応じてキャリア志向に徹したプログラムを作ることの意味する。これは、伝統的に学術に焦点のある文系理系のプログラムとは違うことを意味するから、つぎのような基本的な疑問を伴うことになる。質問を暗に示している。わが教育機関の設置目的は何であるのか？わが教育機関は県内対象か域内対象かそれとも全国対象か？わが教育機関が県内あるいは域内に対して負っている責任は何か？

ソフトウェア工学 (SE) 2005 年の半ばにおいて、米国では 30 強の教育機関だけが SE 学士プログラムを提供している。(イギリス及びオーストラリアに存在するものより、はるかに少ない)。これらの多くが、先駆者的なプログラムとして、学生を訓練して大規模なソフトウェアや安全性第一のソフトウェアの開発に参画できるようにすることのニーズを認識したものであった。これらの教育プログラムは、こうしたニーズに応えた訓練を学生に与えることにおいて、CS や CE のプログラムに期待できるレベルをはるかに超えたものになっている。SE プログラムが増えることを期待しているが、IT プログラムが増加する勢いほどに達することはないだろう。SE プログラムの設置は、教育機関にとって、学術の面でもプロフェッショナル養成の面でも他との差別化を示すものとなる。

SE 学士プログラムを提供する米国の教育機関の中には、エンジニアリングという単語を含まない学士号を選んで使うことがある。北米では、エンジニアリングという単語をエンジニアリング学部外におかれた学士プログラム名に使うと摩擦を生じることがある。その結果、高品質のプログラムながら SE とは別の名前がついたプログラムを見ることになるのだ。場合によっては、SE 風プログラムは単独の学士プログラムの形をとる。また別の場合には、CS 学士プログラム内において特別にまとまった科目群として提供される。したがって、プログラムの学士号ではなく、プログラムの内容こそが、そのプログラムがこの報告で述べた意味での高品質な SE 教育を提供しているかどうかを決めているのである。

SE プログラムを検討中の教育機関にとって鍵となる質問は、“SE の専用プログラムが必要か？”である。これは、プロフェッショナルのソフトウェアエンジニアのキャリアに対して十分な準備を CS の教程が学生に与えているかどうかについて、地域の世論がどうなっているかに依存する。明示的な SE プログラムが必要だというのが地域の世論であり、かつ、資格をもつ教員が適切にそろっているのであれば、残る問題は、どのような構造で実現するかだけである。CS 内においた数種の選択群の一つとするか？あるいは、教程と必修指定の点で異なるプログラムとするか？

4.2 ポートフォリオ戦略

コンピューティング学士プログラムの種類が増えたことは、重要なチャレンジや機会を与えてくれる。コンピューティングの教育者は、選択肢が増えた分、厳しい判断に迫られる。ニーズに応えようとするれば、適応と変化が必要となるが、教育機関の常として、これはとても難しい。

財政上の制約があると、コンピューティング学士プログラムを変更したり新設したりする提案を却下するほかはないが、行動しないことによる悪影響が出る可能性がある。コンピューティングプログラムの新たな広がりには、教育機関にとってのまたとない機会であり、学士プログラムを改編して、学生や、社会や、その構成員のニーズに最も効果的に合わせていく好機となる。教育機関は、その指導性、先見性、そして対応性を最も効果的な方法で示すことができるのだ。

コンピューティング学士プログラムの新たな広がりへの大きな可能性の一つは、大学にとって、学生や地域社会の中に存在する多様なニーズに見合う形にコンピューティング学士プログラム群を整えることができるということである。かつて多くの教育機関にとっては、CS プログラムを技術部門に用意し、IS プログラムをビジネス部門に置く以外選択肢がなかった。現在の大学は、つぎに示す学士プログラム群のポートフォリオを提供して、多様な学生ニーズにより適切に応えることができるのだ。

- **CS** プログラム コンピューティングの総合職を目指す学生、または、大学院進学とか研究職や学際的革新とかを熱望する学生を対象とする。
- **SE** プログラム ソフトウェア開発者としての知的及び技術的な才能に秀でた学生、または、大規模なソフトウェアを開発したり、チーム一丸となって顧客のニーズに見合った堅固な製品を生産したりしたい学生を対象とする。
- **IT** プログラム 技術にだけ焦点を当てるのではなく、技術的側面・人間的側面の両面に対応するコンピューティングキャリアを求める学生、および、様々な組織や状況の中に生じている IT プロフェッショナルへのニーズに惹かれる学生を対象とする。
- **IS** プログラム 他のビジネスプログラムとの協力の下で提供するプログラムであり、組織の情報ニーズに焦点を置くキャリアを求める学生、および、第一義的にはこれらのニーズを満たすための手段として技術に興味をもった学生を対象とする。
- **CE** プログラム エンジニアリングプログラムを提供している大学にあって、コンピュータを使った装置を開発することに焦点を当てたキャリアを求める学生を対象とする。

多様性と多角化の根源は同じである。そして、コンピューティング入学における近年のパターンは、一貫性のない価値への投資を防止するためにも有益な多角化戦略が、予測不可能な入学のスイングから教育機関のコンピューティングプログラムを守る為にも同様に有益であるかもしれないと提案している。最近のイベントはタイムリーな実例となっている。CS プログラムには長らく、循環的入学パターンがあった。最近では、ドットコムブームにより、前例のない CS 入学数を見た。バブルが崩壊したとき、CS の入学数は急落した。更に最近では、CS 入学者数が次第に増加するであろうという期待の中、多くのプログラムではその正反対の状況に陥った。つまり、CS への入学者数は更に減少していったのである。これらの事実は、コンピューティング学士プログラムの人気が下がったということの意味してはいない。CS の入学者減少に関する幅広い報告がされている一方で、様々な教育機関は SE 入学者数が以前と変わらず一定であり、また、IT 入学者数は継続的に増加していると報告している。CS プログラムに加えて IT 及び SE プログラムを提供しているいくつかの教育機関では、集合的なコンピューティング学位への入学者の急激な減少については報告せず、むしろ CS の減少にもかかわらずコンピューティング入学者が純増加していることを報告している。

これらは、証拠ではなく単に事実に過ぎない。多くの要素が絡んでいるだろうし、確実な証拠といえるものはまだない。しかし、これらの事実は考えてみるのに十分な材料である。ポートフォリオ戦略をとることは教育機関が生徒のニーズに近づく一助となる、と思うに足る十分な理由がある。

- 入学生数が学生の興味と能力をなにがしか示しているものとする、コンピューティング学士プログラムの種類を増やすことは、教育機関が学生のニーズの広がりによく対応していくことになる。
- 多くの教育機関にとって、学生数を維持することは大きな懸案事項である。単科大学も総合大学も、出発時点で CS を選択した学生の 50%以上が次第に断念して去って行く、と毎年繰り返し報告している。生徒学生の興味・才能と利用可能な学士プログラムとがよく合っていれば、それだけ学生数の維持率も高くなると期待できるだろう。
- 維持率とは別に、コンピューティングの教育者は（そして、その他の人も）、コンピューティング分野に興味をもつ学生のプロファイルが比較的狭いことを長らく懸念してきている。ポートフォリオの幅を広げることで、より広い学生を引き付けられるだろう。例えば、IT プログラムを提供している教育機関の中には、女性や少数者の入学が増加したと報告しているものがある。これも証拠ではなく単に事例に過ぎない。
- 大学なり学部なりが一群の品質の高いコンピューティング学士プログラムを提供する場合、その教員団は、おのずとコンピューティングの問題や挑戦に対する様々な見方を反映した構成になる。このような状況は、そこに集まった専門領域間で有用な学際的アイデアを育む場となり、ひいては、教育と研究の両方の創造と革新に役立つに違いない。

多くの大学教員は、今、プログラムの多角化について間違いなくこう自問するようになっていく。“どのようなポートフォリオでコンピューティング学士プログラムを提供するのがよいか？”この問いには、教育上の重要な問いでありながら、新たなコンピューティング領域が出現するまでほとんど問われることがなかった、つぎの問いが含まれている。“どのようなコンピューティング教育がこの大学が対象とする生徒やコミュニティに一番合っているのか？”また、このポートフォリオの問いかけから、意志決定権者は、費用対効果を測らざるを得なくなる。明白な費用には、新たなプログラムを作成するための初期費用に加えて複数プログラムを運営していくための管理運営費がある。

費用と効果をどのようにして見積もって比較するか。これには標準的な回答はなく、それぞれの教育機関がその設置目的と環境・状況を踏まえて判断するほかはない。教育機関によって、答えが簡単に出せるところもあれば、難しい判断を迫られるところもあるだろう。鍵となる質問は明白である。“この大学の学生、コミュニティ、そして将来に最も見合った選択は何か？”リソースの制限からプログラム改編に直ちに踏み切ることができない教育機関であっても、鍵となる人々を集めてこの重要な問いを検討することならできるのだ。

プログラムの多角化を真剣に検討すれば、決まっていくつかの課題に直面する。次節では、直面する可能性の高い、鍵となる要素について論じる。

4.3 多様性に対する教育機関の取組み

単科大学でも総合大学でも、コンピューティング学士プログラムの新しい多様性を受け入れるには、少なくともつぎの3分野において、強いリーダーシップと変化への意志が必要となる。教員の能力開発と適応、組織構造、カリキュラム構造である。これは、その性質から、いずれも教員間に意見対立を招く課題を内包している。強い意見相違を引き起こす課題につきもののものであるが、顕在する課題の下には価値観の違いが潜んでいるのである。教育機関のコンピューティング学士プログラムを誠実かつ綿密に立案するには、これらの課題そのものに取り組むだけでなく、これら3分野それぞれでの意見相違の裏にある根本的な価値観の違いを調べてみる必要がある。

4.3.1 学部の能力開発と適応

現在CSの学士号を出している教育機関が、多角化を決め、その設置目的を広げてSEやITのプログラムも用意しようとする、まず適切な教員団を見つけるのに苦勞する可能性が高い。結局のところ、多くの教育機関にとっては、CSの教員だけがコンピューティングの教員なのである。CS教員の多くは、当然のことながら、CSの使命を指向する。それこそが、自らが専門家として成長し能力開発してきたものだからである。彼らにその使命範囲を広げてSEやITの教程も網羅するようにと示唆しただけで、つぎに示す線に沿った反応の入り交じったものを生む可能性が高い。

- 正統性 CS教員の中には、SEやITは未発達で学問領域とみなすことができる段階に至ってはいないという見解をもつ人もいる。SEは、CS専攻の中の1,2科目分に値するに過ぎず、独立した専攻とする内容を担保したり必要としたりはしていない、という人もいる。ITは、その教程が職業訓練的に過ぎる、という人もいる。他のCS教員は、科学者(CS)、エンジニア(SE)、そして実務家(IT)という、それぞれのプロフェッションがもつ社会的役割の重要性に気が付いて、コンピューティング教育の教程を広げる必要性を知りたいと思っている。このグループの教員は、懐疑的な教員に対して、SEの核となるエンジニアリングの視点やITの核となる実務家の視点の正当性・重要性を説いて説得するという重要な役割を担うことができる。
- 準備 ほとんどのCS教員は、CSと実質的に差をつけられるSEとITの科目をすぐに担当できるようなバックグラウンドをもっていない。これは、彼らのバックグラウンドがCSであってSEやITではないという事実からして自然の結果である。大切なことは、こうした課題を隠してしまわないことであり、教員団の力不足を隠して全体の損害を招いてしまわないことである。CS教員はほとんどのコンピューティングトピックについて知識豊かであるだけに、こうした危険があるのだ。CS教員の多くは、そのCS指向ゆえにSEやITの学生のニーズに応えられないのである。例えば、CSの学生もITの学生もともにコンピュータネットワークについて学ばなければならない。教材の多くはCSとITの両方にとって同じくらいに重要であるが、ある時点を過ぎると2つの教程は違ってくる。CSは根本的なモデルや本質を重要視し、ITはネットワーク管理やセキュリティに関連する実践的な応用スキルを重要視する。実践面でいえば、ほとんどのCS教員はネットワーク管理を行ったことがなく、また、ネットワークセキュリティの維持に責任を負った経験もない。同様に、ほとんどのCS教員は、SEに関するCSの科目を担当することができても、学生がソフトウェアエンジニアになるための、徹底したSE教程を担当できるほどの知識と経験を得る機会をもったことがなかった。最低つぎの二つの側面に注意する必要がある。

1. CS で扱っている共通のトピックを SE や IT の科目だと装うのは、きわめて不適切である。
2. CS 教員に SE 科目や IT 科目を担当するための能力開発を要求するのなら、十分なリソースを与え、エンジニアリングの観点や実務家の観点をもった非常に異なる教程に彼らが備え、適応できるように支援することが必要である。

正統性と準備の両方についていえば、核心となる課題は、新たなプログラムへ向けて教員の協力を引き出すこと、そして、これに応じて要求があったことには教員に対して支援を惜しまないことである。より広範なコンピューティング教程にするかどうかの高度な決断が必要になることもあるだろうし、それが不満足なものに終わることもあるだろう。教員の中には、適応し意欲をもって出された難題に立ち向かう者もいるだろう。努力を払って、新たな能力を開発したいと望んでいる教員を見つけだし、それらの教員がある種の自己学習を行ない準備を行って成功するように支援を与えることが成功につながる。

4.3.2 組織的構造

多角化するには、一連のコンピューティング学士プログラムをどこにどのように設けるかの選択をしなければならない。専門領域それぞれを単独の学科とするのか？すべてのものをまとめて多角的な 1 個の単位、たとえばコンピューティング学科、とするのか？伝統的な学科構造を全く無視して、新たなプログラム群を収容するセンターを作るのが一番なのか？これらのやり方それぞれに対して支持者がいる。どれが一番良いかという問いには絶対的な解答がない。教育機関ごとに自らの状況と使命に照らして判断するほかはない。

コンピューティング領域は、それぞれに明確な性格と焦点をもっている。それでいながら、その学習・教育の教程は異なったものというにはほど遠く、多くのトピックやスキルがすべての学士プログラムに関係しているのである。この事実から生じるのは、独立した個々の学科を置く方式と、単一コンピューティング学科で異なる学位を出す方式のどちらに利があるかの綱引きである。前者の方式は、各学士プログラムがそれぞれの任務に専念でき目に見えた妥協など必要としないことを保証しやすい。後者の方式は、すべての領域がもつ共通性を挺子にして、カリキュラム、コンピュータ実習室、教育負荷などの重複を省くことができるという、スケールによる経済性を得やすい。

構造変更をにらんでいる人は、変化と革新が継続して起きていくと期待されていることに留意して欲しい。SE と IT の学士プログラムは比較的新しい現象であるけれど、これらが新しいコンピューティング学士プログラムの最後のものであるわけではない。教育機関の指導者が“SE や IT をどうするか？”という問いに答えるときには、同時にまた、次の新たなコンピューティングプログラムが現れたら、それがどんなものになるかはわからないながらも、どうするかについても自問すべきである。新たな組織構造を作るといふのであれば、その目的を明確にすべきである。当面のジレンマを解決するためなのか、それとも、長期に渡っての土俵づくりのためなのか。

4.3.3 カリキュラム構造

コンピューティング学士プログラム群が同じ学事単位に置かれているのか、あるいは別の学科に置かれているのかによらず、コンピューティング領域にまたがった共通性から、さまざまな科目、特に入門科目の役割と目的共通項は様々なコース、特に導入コースの役割や目的について疑問がわく。相互に関連する二つの課題がある。

- フィルタ方式とじょうご方式 入門レベルでは、科目設定の基本的な考え方に二通りがある。フィルタ方式とじょうご方式である。フィルタ方式では、意欲的に集中した形の領域特有の教程を最初の科目から組み上げたカリキュラムを求める。フィルタ方式のカリキュラムでは、入門科目を使って、その領域の基礎を築いたり、最初から厳密な能力標準を確立し、その標準に達しない学生を直ちに濾し落としたりするのに当てる。対照的に、じょうご方式では、当該領域を専攻し先に進む学生だけでなく、より広い範囲の履修学生を対象とするカリキュラムを求める。じょうご方式のカリキュラムでは、入門科目を使って、学生にその学習経験を通して当該の領域に進むのに適しているかどうかの判断ができる情報を与えるのに当てる。

フィルタ方式は、コンピューティング領域ごとに一連の入門科目を並行して設けることを求める。フィルタ方式を採る主な動機は、諸般の要因から、選んだ領域に対する十分な準備を学生にさせるだけの時間余裕が教員には与えられていないと教員が感じていることにある。フィルタ方式の支持者は、たいてい、彼らのプログラムでは早期に領域の基礎を強固に築かなければならず、したがって、全コンピューティング領域共通の入門科目では事足りないと信じている。一方で、教育機関としての特性（CEがおかれている伝統的なエンジニアリング学部、ISがおかれているビジネス学部）から、そのカリキュラムの枠組みに共通のコンピューティング入門科目を置く余地がないか、あってもごく狭いものになっているかしていることもある。それに加えて、フィルタ方式の支持者の中には、並行したカリキュラムの設置を主張するのに、学生は出来る限り早く区分した方が意欲をもって勉学に励むものだという人がいる。この見解によれば、並行した一連の入門科目が、それぞれの履修者群にひとつずつ、必要であるばかりでなく、初期の科目をフィルタとして使い、学生が最初に優れた適正を示さなかった科目群の学習を続けないようにすることが必要である。したがって、フィルタ方式の支持者の中でも意見が分かれる。一部の人はフィルタ方式を必要悪であるといい、他の人はその利点をもって望ましい方式であるという。モチベーションと理由付けの違いによらず、フィルタ方式の支持者は、学生が学部課程の初期段階で適切な学士プログラムを選ぶべきであり、選ぶことができると主張する。それも、選んだ上で、授業を受けたり、領域での経験を積んだりすべきであるというのである。

フィルタ方式のカリキュラムを学生に責任あるものとするには、学士プログラムは学生に合理的で十分な支援を与えるようにして、学士プログラムの情報に基づく選択が行えるようにしておかなければならないし、その時期も学生がプログラムに入る前でなければならない。若い人々に、人生を決める重要な選択を主として学士プログラムの名前や入門科目の名前に基づいて行うことを求めるのは、不合理であるし無責任である。しかし、積極的かつ効果的に学生にアドバイスを与えることは、どんな場合にも価値あることであるが、フィルタ方式を採用したカリキュラムにとってはことさらに重要である。

じょうご方式の支持者は、共通の入門科目系列を置くことを主張する。多くの学生が新生の時に専攻の選択肢を明確に理解できているとは期待しにくいと強く主張し、学生が明確に理解できるようにする最善の方法は科目履修を通じてそれぞれの専攻領域の感じを与えてやることだと主張する。この方式によると、統合した入門科目系列が欲しいというだけでなく、学生に学士プログラムの情報に基づいた選択をさせたいのならそれが不可欠であるという。英国では、第1学年に共通の入門科目を置くことが広く行われている。英国での経験によれば、専攻選択の時期を遅らすは、学生数の維持にも役立つという。

共通科目を置くことのもう一つ動機に、学士プログラムそれぞれに異なる科目群を配置するとその運営費用がかさむことがある。科目運営費用の削減に加えて、維持しなければならない科目数を削減すると、各科目を扱いやすく最適化することも可能となる。

じょうご方式では、どれも多くの学生に入門体験をさせるという点で変わりがないが、その後のじょうごの設計はさまざまである。設計方針の一つは、学生に複数のコンピューティング学士プログラムの中から一つを選択させるというものである。別の設計として、学生に用意してあるのは、特定のコンピューティング領域(米国ではほとんどの場合コンピュータ科学)で学習を継続するか、コンピューティング学習を断念するかを選択だけというものもある。どちらの場合でも、入門経験を通じて学生が情報に基づいた選択を下せるようにしている。異なるのは、情報に基づいた選択の対象となるコンピューティング学士プログラム数である。

じょうご方式では、その適用範囲に応じていくつかの重要な課題が生じる。じょうご方式は、一つの専門領域だけ、たとえばCSプログラムだけを対象とするほうが設計しやすい。その入門科目を学内の広範囲な学生向けとしてしまうのである。このシナリオでは、専攻を決める前の学生がプログラミング基礎への厳格な入門を果たすことができると同時に、CSでの学習を更に進めたいかどうかの感触を掴むことができる。この場合、じょうご科目は一つの学士プログラム向けに的を絞っているから、どんな教材に焦点に当てるのが最適かを簡単に知ることができる。

複数のコンピューティング学士プログラムにまたがったのカリキュラムの統合は、教員にとってその設計をどうするかが難題となる。目標とするのは、様々なコンピューティング領域の上級科目で扱う内容を含んだことを学生に直接経験させることである。そうできると、用意されたコンピューティング学士プログラムを選択するための可能な最良の基礎を学生に与えることができるが、そうするには、こうした科目を設計し実施する教員が井の中から抜け出し、それぞれのコンピューティング領域のニーズに、とりわけ自分の領域でない領域のニーズに、敏感にならなければならない。起こりうる最悪のシナリオは、特定の領域がこの一連の導入科目をその領域に進む学生向けにハイジャックし、フィルタとして使ってその領域では成功しない学生を他の領域に進ませるという目的に使うことである。

カリキュラムモデルの中で、5つのコンピューティング領域すべてに役立つ形のじょうご方式を採用した成功例は知られていない。しかし、多くの大学の学部では、英国の大学の例を含めて、2,3の領域を対象とする、じょうご方式の科目設定が成功したとの報告を行っている。5つの領域すべてを網羅する確固としたモデルがないのは、つぎの二つの原因を反映してものだろう。じょうご方式そのものに難しさがひそんでいること、そして、5つの領域すべてで学士号を出す経験のある学校がまだ少ないという単純な事実、の2つである。将来作られる領域別カリキュラム指針群では、入門モデルの調査・評価に基づいて、広い対象をもったじょうご方式教程の成功例が報告されることを期待したい。

カリキュラム構成要素の粒度 多くの知識項目や能力項目の中で、さまざまな学士プログラムに関連するものが、1, 2年次に見つかるが、こうした学士プログラム間の共通性はプログラムの全学年に広がっている。例えば、コンピューティング領域のプログラムは、どれも上級科目として、オペレーティングシステム、ネットワーク、データベースなどの学習域を対象とするものがなければならない。これらの学習域における教材には、すべての専門領域の学生に共通して使えるものもあれば、領域ごとに特別の扱いが必要となるものもある。共通するものが多々あるとはいえ、それらをまとめてみても伝統的な学期単位の科目に比べると小さな塊にしかならないので、伝統的な科目配置の隙間に落ちてしまう。この結果、それぞれの学士プログラムが独自に調製した科目をもつことになり、多くの共通点をもった類似科目がコンピューティング専攻にあるという事態に至ってしまう。

これらの理由により、短期間の科目を用意して、それぞれの領域が独自の組み合わせ方でそれらを利用するという方式が工夫される可能性が高い。片手で数えられるほどの短期間（多分は2, 3週の）科目モジュールが用意してあって、ネットワークのそれぞれの側面を扱うようにし、例えば、CSの学生とITの学生とに共通するモジュールもあるものの、その他のモジュールは、CS・ITのそれぞれが、その教程に則して明確に焦点を合わせたものを使う、といった状況を想像してみればよい。この方式は、明らかに便益を内在させているものの、教室繰りをどうするかといった様々な後方支援上の課題を伴っている。この方式は、堅実なアイデアであると思われるが、その実用性を判断するには相当量の実施例を待つ必要がある。

コンピューティングプログラムの多様性が新しいだけに、これらの課題を解決するには分野全体としてまだ経験が不足している。したがって、革新と実験を進める必要性がとても高い。じょうご方式のカリキュラムを好む人の多くは、もっとも緊急に必要であるのは、はっきりと区別をして生徒を正しい学士プログラムに導くことのできる入門科目の新しいモデルであると考えている。課題は、うまく内容を配置して、どの領域も特別に優先したりおとしめたりすることなく、学生に対して、各コンピューティング学士プログラムで何に出会い何をするかを予示する知識と経験を与えることである。このような内容配置を開発することは、この分野での重要な課題であり、その中から、コンピューティング全体に共通するプログラミング以外のものについての新たな知見をえることができる可能性を秘めている。現時点では、うまくこれを成功させることのできる、強力なモデルはまだない。向こう数年の間にこの領域で教育者が進歩をなして、成功したモデルによって、学生にも教育機関にも強力な便益がもたらされることを期待したい。

4.4 大学の品位と市場動向

市場の動向は、大学の教育プログラムに、さまざまな形で影響を与える。それらのいくつかは、この報告で取り上げるべき範囲を超えるものだ。たとえば、近年さまざまな証明書(certification)が普及してきている。この用語“証明書”は、重要な違いのあるものに広範囲に使用されている。いくつかの証明書は、ベンダ固有 (Microsoft や Cisco などの証明) のものだ。そして、学会 (IEEE-CS や BCS など) や、その他の組織 (ICCP など) から得ることができる証明書などもある。さまざまな形で、証明は大学の教育プログラムと競合している。証明が大きな流れであることは明白だ。物事の常として、評判の高い証明もあれば、物議を醸し出す証明もある。学位を認定する公共機関が、ベンダ固有の証明プログラムと提携することになれば、大学の品位が疑われることになるだろう。読者には、このような提携関係が大学の品位と倫理についての論争を巻き起こすことに気付いて欲しい。証明というものに関連した、こういった広範囲な問題を取り扱うのは、この報告の目的とするところではない。他のプロジェクトで証明が取り扱われたときには、この報告の改訂版で、その報告書を参照文献として載せることになるだろう。ここで取り扱うものは、学部のコンピューティング学士プログラムに限定される。

社会のニーズを満たすために新しいコンピューティング学士プログラムが誕生するのを目にしているが、これは社会のダイナミズムが学界で失われていない証拠である。新しいコンピューティング領域の登場は、学界が進化し、社会のニーズに応えることができるということを実証している。悲しむべきことであるが、新しい学士プログラムの登場という同じ現象が、悪い側面を実証するものともなる。大学の自発性が、重要な問題を取り扱うための技術革新の必要性に基づいたものではなく、政治課題、財政上の必要性、マスコミの煽りなどに動かされたものであるときがそうである。

幸い、これら 2 種類の現象を見分けるのは簡単である。新しく登場したコンピューティング学士プログラムを見たとき、高品質なプログラムと低品質なプログラムの両方の例を目にすることになるだろう。

- 高品質な教育プログラムとは、内部から推進され開発されたプログラムである。教員と管理者が手を携えて、慣習的な主題領域の境界線を越えた見通しをもち、学生や共同体が今までに無い新しいものを必要としていることを認識し、正真正銘現実の問題を解決するために刷新を断行した新しい技術を導入してきた結果である。そのようなプログラムの教員は、バックグラウンドが必然的に他の領域であるにもかかわらず、その新領域の教員であると自称する傾向にある。教員は、彼らの学生を大切に、学生や共同体のニーズを正当なものと認めて、学生をその領域にふさわしい高い水準に保とうと奮闘するのだ。
- 低品質な教育プログラムとは、外部から推進されたプログラムである。これが起こるシナリオのひとつは、権力をもつ誰かが、独断的にスケジュールを与えて新しいプログラムを作るように命令するという上位下達の構図である。教員と管理者は、そうするように言われたから、これに貢献するだけである。その指令に本源的な価値がなく、またそれが学生やコミュニティの正当なニーズに基づくものでもないことを知っている。そこで、外見だけを刷新したと装うだけで、新しいプログラムとしては他の学科の既存科目を寄せ集めたのに過ぎないものを用意することになる。そのようなプログラムの教員は、前に所属していた、もっと定評のあった領域名を名乗り続ける傾向がある。

ここでの重要な教訓は、優れたプログラムは上意下達だけでは生まれてこないということである。上意下達はきっかけとしては必要かもしれない。しかし、優れたプログラムには、教員の関与と育成が不可欠であり、教員が機会を逃すことなく可能性を秘め本源的な妥当性をもつ仕事をこなしていくことが必要なのだ。財政的圧力から、需要を満たすプログラムを作成することが魅力的な選択肢となることがあるが、大学の品位を保つためには、市場動向に対する表面的な対応以上のものが必要とされる。新しいコンピューティングプログラムを作るには、相応の開発努力が必要であることを認め、開発担当者を置き、実質のある一貫したプログラムを開発することができるに足りる十分なリソースを支援することが決定的に重要である。

4.5 認定制度とコンピューティング カリキュラム

大学認定制度とは、教育機関およびその学士プログラムの向上を支援し、学士プログラムが外部の必要条件を満たしていることを示し、もって社会的にその信頼度を向上させることを目的とするプロセスである。

いくつかの国では、認定が大学の各レベルに対して行われることがある。こういった場合、機関全体の認定は、その大学が正当な高等教育機関すべてに要求される資源（図書館など）と手続き（入学者受け入れ方針など）の最低基準を満たしていることを示す。同様な指針が、関連する分野の学士プログラムを傘下にもつ機関内単位（ビジネス スクールなど）に対しても存在していることがある。一群のプログラムを傘下にもつ学部等に対する認定も、機関認定と同様のものであるが、より特定されたものである。

認定の中で最も厳しいものは、個々の学士プログラムの評価に関係するものである。これには独立の第三者機関または政府機関が与り、個々の専門領域ごとに学士プログラムの質的標準および基準を定めている。領域別の認定（プログラミング認定ともいう）は、学士プログラムを個別に評価して、その学士プログラムが、制定されている規準を満たし、かつ継続的な改善を行う厳密なプロセスをもっていることを証明するものである。認定は、すべての専門領域に対して存在しているわけではないが、コンピューティング学士プログラムには存在している。

認定がさまざまなレベルで行われる国では、教育機関が大学認定組織の認定を得ていたとしても、その教育機関の中に置かれたコンピューティング学士プログラムが、コンピューティング学士プログラムの質の評価を行う機関から認定してもらえないことがある。たとえば、米国の大学は、大学自体としては認定されていても、その中に認定されていない学士プログラムが存在する可能性がある。大学が認定されているからといって、そのコンピューティング学士プログラムが対象コンピューティング領域の品質基準を満たしているとは限らないことを覚えておいて頂きたい。

4.5.1 領域別認定制度の利点

領域別認定制度には、プログラムおよびそれを傘下にもつ教育機関にとってつぎの2つの重要な利点がある。

- 1) 学士プログラムが、第三者機関（学会など）や政府機関が定めた最低限の質的水準を満たしていることを証明する。これは、教育機関がそのプログラムを市場に提示する際に役立つことになる。これは、一般市民や志願者に、その学士プログラムの品質に対する信頼を与えるだろう。
- 2) プログラムは、審査チームによる実地審査を受け、専門家の立場からその長所と短所を指摘してもらい、改善を必要とする事項を指摘してもらう。このやり取りを通じて、教育機関は、そのプログラムがどの程度のレベルにあるかを知り、その品質を向上させるために何を行わなくてはならないかを完全に理解することができる。

このように、認定制度によって、学生をひきつけるマーケティング上の助けを得ることができ、同時に品質向上に焦点をあてた専門家の助言を得ることができるという2つの便益が得られる。前者の便益を必要としない、または望まない教育機関もあるだろう。そうした教育機関でも、認定プロセスが教育プログラムの品質向上に役立つことの便益を目的として認定を受けるところもある。それでも結果としてその教育プログラムの評判は強固なものになるだろう。いくつかの国では、教育機関に認定を受ける受けないの選択権がない。そうした国では、認定は、教育プログラムが存在するための必須条件なのだ。

領域別認定制度のプロセスは、学士プログラムが特定の基準を満たしているかどうかを判断する。認定制度は、教育プログラムに資格を持った教員が十分な人数いてその教育負担が許容範囲内に収まっているかどうかを判断するだけではない。そのプログラムが教材や課題をどのように与え、課題や試験をどのように評価しているか、そしてどのように継続的な点検と改善を行っているかについても判断する。

プロフェッショナル団体の中には、この認定制度を利用して、認定された学士プログラムの修了をもってそのプロフェッションにおける会員資格の保証としているものがある。さらには、特定のプロフェッションにおいて実務を担当するには、認定された学士プログラムを卒業していることが必要条件となっていることもある。これは、そのプロフェッションで仕事をしたい学生は、ただ適切な専門領域で学士号を取るだけでは十分ではなく、認定された学士プログラムで学士号を取らなくてはならないということを意味している。

認定がそのようなプロフェッショナルな意味合いをもつかどうかを決めるのは学士プログラム自体ではない。認定プロセスは、その国の、その専門領域に対する慣行によって決まっているのだ。たとえば、米国では、エンジニアリングの教育プログラムの認定にはこのプロフェッショナルな意味合いが含まれている。その他の学士プログラムにはたいていその様な意味合いは含まれていないし、ほとんどの文系（Arts）と理学系（Sciences）の分野は全く認定制度をもっていない。

認定に関する最大の誤解は、教育機関が認定マークを得るためだけにプログラム認定を達成しようとしているのだ、という考えである。領域別認定制度を良く知らない人々は、認定プロセスが果たす重要な役割が理解できていない。認定プロセスは、教育プログラムが提供する内容、そしてそれが排出する卒業生の品質を向上させるために何を行わなくてはならないかをあぶり出す助けとなるのだ。

4.5.2 認定と品質

領域別認定制度は、学士プログラムが第三者による品質基準を満たしているかどうかを証明する手段であるが、その基準が意味するところはさまざまである。その厳密さは、認定機関の規定と適用される政府の条例によって定まる。場合によっては、認定は、学士プログラムが最低限の品質を満たしたことを示す。場合によっては、最低限の基準と、より高い基準の両方の認定が存在することもある。

領域別の認定制度はプログラムの品質に関係しているとはいえ、認定と品質との関係について根拠のない結論に飛びつかないようにしていただきたい。適切な結論を出すには、対象の領域および国別の事情をよく理解してはならない。

認定されていないということは、その教育プログラムの品質が低いことを示すのか？

その答えは、その領域とその国の事情次第である。たとえば、英国のほとんどのコンピューティングプログラムは認定されている(4.5.4を参照のこと)。米国での状況はさまざまである。信頼できるすべてのCEプログラムは認定されている。ほとんどのCSプログラムは認定されていない。そしてIS,IT,SEの認定制度はまだ始まったばかりである(4.5.5を参照のこと)。

教育プログラムが認定されているということは、それが高品質であることを示すか？

その答えは、国の事情次第である。いくつかの国では、認定されるということは、その教育プログラムが許容できる最低限の品質を満たしたということしか示さない。また別の国では、認定がさまざまな品質レベルに対して行われている。

すべての認定プロセスは同じ手順で行われるのか？

そうではない。国ごとに品質保証のやり方が異なっている。たとえば、米国では、認定チームは実地訪問時に、個別科目の課題からハイレベルなプログラム上の問題まで、その教育プログラムの全レベルを査察し査定しなくてはならない。対照的に、英国での品質保証の実務として、外部の試験員が参加することが日常的に行われている。英国のシステムでは、他の教育機関の教員によって、事前に、すべての科目での課題や試験問題、およびそれらの評価基準の査察と査定が行われる。つまり、英国の学生は、計画されていない課題や試験、あるいは適切な評価基準が無い課題や試験を受けることが無いのだ。課題・問題やその評価基準が外部の試験員が満足するものでない場合は、それらを試験員が満足するものになるまで改善しなくてはならない。これらの品質保証の実務は認定に関与していないとはいえ、認定チームは、外部試験員が作成した評価票を利用することができる。これにより、認定チームは、時間をかけて得られた評価データを手にすることができるので、低いレベルの調査から解放される。結果として、認定チームは、高レベルな教育プログラム上の品質問題に集中することができるのである。

認定以外で、教育プログラムの品質の高低を示すものは?

高い品質には、多くの側面がある。たとえば、良い教員の存在、適度な教育負担で教員が担当科目に集中できてかつその分野で現役であり続けられること、十分なインフラがあること、などである。調べるべき最も重要なもののひとつは、品質をモニタし改善していくための厳格な手順が採られているという証拠である。大切なことは、どこに改善が必要かを見つけだすプロセスが働いていること、そして、品質を示す情報を実際に使って目に見える形で改善が行われていることである。

優れた教育プログラムでは、品質のモニタプロセスが品質改善の発動と組み合わせられて継続的なサイクルを構成していることである。つまり、改善活動の効果をモニタし、新しい改善活動を計画し実践する。そしてまたその効果を評価し、このサイクルを繰り返していくということである。これを適切に行うことは難しくはないが、かなりの貢献、修練に加えて情報の共有・活用が必要となる。

学士プログラムの内部運営の当事者でもない人がプログラムの実情を知ることは難しい。入手のしやすさという面から見れば、次のような項目を問い合わせてみるのがよい。

- プロフェッショナル諮問委員会: それは存在しているか? 誰が参加しているか? それは、プログラムの教育面に焦点を当てているか? それとも、教員の研究プログラムにだけ関わっているのか?
- 学生から入手した情報の利用: その教育プログラムでは、学期末に学生の満足度アンケートからデータを集める以上のことをしているか? 学生は、大学のさまざまな委員会に参与しているか? 卒業しようとする学生から、プログラムの長所と短所に関する意見を得るために、修了時の面接を行っているか? その教育プログラムの卒業生が社会で数年働いた後に、彼らに意見を聞いているか? その様な情報が、実際の改善を引き出すために使用されている証拠があるか?
- 評価・改善のための体系的なプロセス: 各科目の実効性の評価を行う手順が整備されているか? 教育プログラムを修了した学生の強みと弱みを調査するためにどのような手法が使用されているか? どのようなプロセスによってそれらの情報を実際に使用することを担保しているか?

品質の大幅な改善のためには、成績評価点の平均を計算したり、学期末の学生の満足度アンケートからデータを集めたりするだけでは不十分である。これらの基本的データの収集以外にほとんど何も行っていないとすれば、その学士プログラムおよびそれが輩出する卒業生の質を向上させるために組織ぐるみで取り組んでいるかどうか、はなはだ疑問である。

4.5.3 国民性と国際協力

多くの国が、認定制度を採用している。枝葉末節に違いがあっても、それらには共通の特徴がある。分野を代表する専門家の一団が、制定されている標準や判定基準に照らして教育プログラムの品質を評価するのである。

国によって、認定が必須であったり、強く要請されていたり、または任意であったりする。いくつかの国では、厳格なプログラム基準が存在し、あらゆる単科大学・総合大学が提供しているすべての教育プログラムに認定基準が適用されることになっている。また別の国では、認定は任意である。認定が任意である国でも、認定が期待される程度は異なっている。

また、認定プロセスの管理方法もさまざま。国によって（オーストラリア、カナダ、英国など）は、プロフェッショナル団体がおのこの対象領域の教育プログラムの認定を執り行う。別の国（米国など）では、指名された組織が認定制度の監視や活動を行う。また別の国（エストニアやアラブ首長国連邦）では、政府が認定プロセスを執り行う。

コンピューティング領域の中には、認定機関が国を超えて協力し合うこともある。たとえば、ABETは米国における認定活動に加えて、他国の教育プログラムの評価を20年以上も支援し続けている。評価・認定プロセスの相互承認は、一連の国際合意の発展を促してきた。ワシントン アコード、シドニー アコード、ダブリン アコード、ヨーロッパ諸国技術協会連盟 (FEANI)、国際専門技術者登録制度 (IRPE) などである。これらの合意には、多くの国が調印し、国境を越えたプロフェッショナルの活動を促進するという共通目標を共有している。たとえば、ワシントン アコードは、オーストラリア、カナダ、香港、アイルランド、日本、ニュージーランド、南アフリカ、英国、米国のエンジニアリングの学士プログラムを認定する機関の間で交わされた合意である。この合意は、これらの組織が認定したプログラムが実質的に同等であることを承認し合うものである。

4.5.4 英国におけるコンピューティングの認定制度

英国のエンジニアリング評議会 (ECUK) は、エンジニアリングの学士プログラムの認定に責任をもつ。その責務には、エンジニアリング学士プログラムの認定のための標準（権限と委任について）を設定すること、そして詳細な認定を代行する組織を指名することが含まれている。英国コンピュータ協会 (BCS) は、エンジニアリング評議会に代って、コンピューティング分野のほとんどを内包している情報システムエンジニアリングの学士プログラムの認定を執り行う。英国電気学会 (IEE) は、電気工学の学士プログラムの認定を執り行う。コンピュータエンジニアリング内の各学士プログラムは、どちらの学協会から認定を受けてもよいが、おそらく英国電気学会から受けることのほうが多いだろう。両学協会による共同認定が一般的である。

さらに、英国政府は、品質保証局 (QAA) を利用して、すべての分野に対して独自の品質保証手順を制定した。品質保証局は、学士号の評価基準に責任がある。各教育機関は、その学士号が基準に達していることを証明する必要がある。これらの基準の一例が UKQAA 2000 であり、これは学生の知識、技能、判断力について、しきい値 (下限) と最頻値 (平均) の期待値を規定している。

4.5.5 米国におけるコンピューティングの認定制度

最近まで、米国におけるコンピューティングの認定制度は限定的で断片化したものだった。現在は一元化が進み、範囲も広がってきている。従来、2つの組織がコンピューティング関連の学士プログラムを認定していた。米国エンジニアリング系高等教育課程認定機関(ABET)が、電気工学やコンピュータエンジニアリングをはじめとする、学部レベルのすべてのエンジニアリングプログラムを認定していた。コンピュータ科学認定機関(CSAB)は、学部レベルのコンピュータ科学プログラムを認定していた。最近になって、CSABはCSAB Inc.として再編され、ABET Inc.(これも名称変更されている)の会員学会のひとつとなった。現在ではABETがエンジニアリング、テクノロジー、コンピューティング、および応用科学の教育プログラムの認定を行っている。現在、CE, CS, IS, SEの認定が行われている。インフォメーションテクノロジープログラムの認定は始まったばかりである。

法律や規則で学士プログラムの認定が求められているわけではないという意味では、米国における認定は任意である。実際には、コンピューティング領域によってその任意性に程度の差がある。エンジニアリングにおいては、プロフェッショナルの団体としての意識が強く、エンジニアの国家免許取得には、ABET認定のプログラムでエンジニアの学士号を取得していることが必要となることがある。結果として、すべてのコンピュータエンジニアリングプログラムには認定されていることが期待されており、認定されていないコンピュータエンジニアリングプログラムの信用性は疑わしい。こういった状況は、ソフトウェアエンジニアリングプログラムにおいてはもっと曖昧である。というのも、ソフトウェアエンジニアリングは生まれたばかりであり、またその性質上、伝統的なエンジニアリングとの結びつきがはるかに少ないからである。

対照的に、コンピュータ科学の団体は、実務にあたるプロフェッショナルの強固な組織体というよりも、科学者、研究者、プログラムの漠然としたネットワークである。これまで、コンピュータ科学プログラムの認定に対して、プロフェッショナルからの強制的な圧力は無かった。要求は、認定を求める大学側からのものであった。現在は、米国のコンピュータ科学プログラムで認定されていないものは10%未満である。英国では、すべてのコンピューティング関連のプログラムが認定されている。最近では、コンピュータ科学プログラム認定への要求が高まっている。情報システム領域は、独自の認定制度を持つ経営系の学校と関係していることがよくある。インフォメーションテクノロジーは、まだ新しく、そしてほとんどのインフォメーションテクノロジーのプロフェッショナルが主に技術的能力によるものであるから、インフォメーションテクノロジープログラムに認定が要求されることになっても驚くにあたらない。

ABETの認定に関する哲学は、規範的な基準に基づくものからアウトカム(学習成果)に基づいたものにシフトしてきている。この新しい方式の下では、各プログラムはその目標を定義し、その目標が達成されていることを確認するプロセスを示すことが必要とされる。

第 5 章 次の段階

以下、近々予定されているカリキュラム関連の進展についてまとめる。

- IT2006 レポートの発行

現時点における草案は、つぎのサイトにて閲覧しコメントすることが可能である。

<http://www.acm.org/education/curricula.html>

最終版の発行は 2006 年に予定されている。

- *The Guide to Undergraduate Degree Programs in Computing* (学部生向けコンピューティング学士プログラムの手引き) の発行

この手引きでは、概要報告の小型付属文書である。大学進学志望者、その親や指導カウンセラー、その他高校から大学に進学する学生の進路選択について関心をもっている人など、より広範囲の読者を対象としている。この手引きではコンピューティング領域それぞれを大まかに特徴付け、志望者がコンピューティング学習の領域を選択する際に検討すると考えられる要点の概略を描き、独立した文書として広く配布される予定である。タスクフォースは、その草案を閲覧しコメントを付けられるようにするために、2006 年初めにはこれを公開し、2006 年半ばに最終版の発行ができるように予定している。

- コンピューティング カリキュラム シリーズ各編の改版に関する新しいスケジュールの導入

以前は、カリキュラム報告はより狭い範囲のコンピューティング領域に限定されていたし、これらの指針は 10 年から 12 年に一度見直されるだけであった。コンピューティングにおける急速な変化を勘案すると、より頻繁な指針の見直しが必要である。対象の進展に左右される必要に応じて改版を発行できるようなプロセスを整備することを長期的な目標としている。

適切なプロセスを整備する一方(以下を参照のこと)で、適切なスケジュールの当初の概算として、以前の改版サイクルを半分に短縮し、10 年から 12 年ごとではなく 5 年から 6 年ごとに各コンピューティング領域の改版を実施する予定である。したがって、2005 年半ばにはコンピュータ科学編の改版(CS2007 の作成)のためのタスクフォースを編成し、これに続いて IS2002 改版(IS2007 または IS2008 の作成)のためのタスクフォースを編成する。特定分野の編を改版する際には、本概要も改版するものとする。

- コンピューティング カリキュラム シリーズ各編(CE, CS, IS, IT, SE および概要編を含む)のフィードバックを継続的に行うための新しいプロセスの始動

過去、コンピューティング関係者がカリキュラム推奨に関して意見を述べたり、フィードバックを行ったりする機会は限られていた。SE2004 編から始まり、本編および IT2006 プロジェクトへと続いている中では、オンラインのツールを通じて、誰もがどこでも提案や批評を送信できるような機会を提供している。これらのツールを改良し、より広範囲のコンピューティング関係者に認知され、継続的に利用できるようにするつもりである。レポートの作成期間だけというのではなく、各報告が存続している限り、やる気ある人々の経験や意見が役立てられるようにすることを目標とする。

- 一部にはそのようなフィードバックによって支えられているような、コンピューティング カリキュラム シリーズの各編の妥当性を継続的に評価するための新しいプロセス（各編において変更が必要な箇所を特定するプロセスを含む）
カリキュラム・レポートが 10 年に 1 度以上更新されなければならないことは明白であるが、任意に各編の改版を発行することは避けたい。改版によって現場レベルにおける変更が必要となるが、各変更にはコストが伴うからである。改版はスケジュールによって決定するのではなく、むしろ必要に応じ、その領域における重大な変更や発展に応じて発行することが理想である。これを実現するためには、各領域における発展を監視し、カリキュラム改版の必要性を判断し、必要に応じて改版を発行するためのプロセスが必要となる。現在カリキュラムの作成に携わった関係者が、この重要な課題に取り組むためのタスクフォースの編成を進めている。
- カリキュラムのより頻繁な改変に対する支援の改善
必要に応じてカリキュラムの各編を更新する能力を整備するのみならず、各編をまたいで見渡すための、効果的な手段が必要である。現在実践されている方法では、領域別の編に比較的独立した取り組みを行っており、これには多大な労働力を要する。また、コンピューティングに関する変化は、領域の境界を超えて適用されることが多い。現在、コンピューティング オントロジー プロジェクトがコンピューティング各領域にまたがるすべてのコンピューティング関連テーマを形に表わすためのフレームワークを作成している。その目的は、コンピューティングの問題領域の上位に位置する集合体を確認し、持続的なコンピューティングの発展に対応して、その共通領域内の特定のものを更新し、この発展によってどの領域が影響を受けるのかを認識し、カリキュラム関連課題に対する注意を促すことを可能とするようなスキーマおよびツールを作り出すことである。さらに、適切なスキーマおよびツールが存在することによって、新たに出現するコンピューティング領域を評価し、特徴づけ、またその発展において既存の関係やインフラをより有効に活用することが可能となる。コンピューティング オントロジープロジェクトはかなり軌道に乗っている。将来、出版物や会議における発表などで、その進展を耳にすることがあるかもしれない。

当面は、つぎのサイトからコンピューティング カリキュラム シリーズの各編にアクセス可能である。

<http://www.acm.org/education/curricula.html> および <http://computer.org/curriculum>
このサイトでは、また、現在実施中の、または近々実施される予定のさまざまなプロジェクトに関するニュース、フィードバックが可能なリンク、さらにカリキュラム関連の発表に関するニュースを見ることが可能である。

参考文献

[ANSI/IEEE Standard 729-1983] IEEE Standard Glossary of Software Engineering Terminology, The Institute of Electrical and Electronics Engineers, Inc., NY, 1983.

[AssocDeg] ACM Two-Year College Computing Curricula Task Force. Computing Curricula Guidelines for Associate-Degree Programs. New York, NY: ACM Press, 1993.
(<http://www.acm.org/education/curricula.html>)

[CC91] ACM/IEEE-CS Joint Curriculum Task Force. Computing Curricula 1991. Association for Computing Machinery and the Computer Society of the Institute of Electrical and Electronics Engineers, 1991.
(<http://www.acm.org/education/curricula.html> または <http://www.computer.org/curriculum>)

[CC2001] ACM/IEEE-Curriculum 2001 Task Force. Computing Curricula 2001, Computer Science. IEEE Computer Society Press and ACM Press, December 2001.
(<http://www.computer.org/curriculum> または <http://www.acm.org/education/curricula.html>)

[CE2004] IEEE/ACM Joint Task Force on Computing Curricula. Computer Engineering 2004. Curriculum Guidelines for Undergraduate Degree Programs in Computer Engineering. IEEE Computer Society Press and ACM Press, December 2004.
(<http://www.computer.org/curriculum> または <http://www.acm.org/education/curricula.html>)

[HS] Pre-College Task Force Committee of the Education Board of the ACM. ACM Model High School Computer Science Curriculum. Communications of the ACM, 36(5): 87-90, May1993.
(<http://www.acm.org/education/curricula.html>)

[IEEE Std 610.12-1990] Standard Glossary of Software Engineering Terminology. The Institute of Electrical and Electronics Engineers, Inc., NY, 1990.

[IS97] IS '97 Model Curriculum and Guidelines for Undergraduate Degree Programs in Information Systems. Association for Computing Machinery and Association for Information Technology Professionals, 1997. (<http://www.acm.org/education/curricula.html>)

[IS2002] ACM/AIS/AITP Joint Task Force on Information Systems Curricula. IS2002 Model Curriculum and Guidelines for Undergraduate Degree Programs in Information Systems, Association for Computing Machinery, Association for Information Systems, and Association for Information Technology Professionals, 2002.
(<http://www.acm.org/education/curricula.html> または <http://www.computer.org/curriculum>)

[IT2006] The ACM SIGITE Task Force on IT Curriculum. Information Technology 2006, Curriculum Guidelines for Undergraduate Degree Programs in Information Technology. 本レポートの執筆時点では、ITカリキュラムレポートは作成中である。レビューおよび開発プロセスは2006年に完了する予定である。現時点で最新の草案は“IT2005”であり、一般からのレビューとコメントを募集中である。ITレポートへの最新のリンクは、
<http://www.acm.org/education/curricula.html> である。

[K-12] ACM K-12 Task Force Curriculum Committee. A Model Curriculum for K-12 Computer Science. ACM Press, 2004. (<http://csta.acm.org/Curriculum/sub/k12final1022.pdf>)

[SE2004] IEEE/ACM Joint Task Force on Computing Curricula. Software Engineering 2004, Curriculum Guidelines for Undergraduate Degree Programs in Software Engineering, IEEE Computer Society Press and ACM Press, August 2004.

(<http://www.computer.org/curriculum> または <http://www.acm.org/education/curricula.html>)

[UKQAA, 2000] Quality Assurance Agency for Higher Education Computing. A Report on Benchmark Levels for Computing. Southgate House, Gloucester, England, April 2000.

(<http://www.qaa.ac.uk/academicinfrastructure/benchmark/honours/computing.asp>)

用語集

アルゴリズムと計算量 – 問題に対するコンピューティングの解（アルゴリズム）。入力サイズが大きくなっていくときの，実行時間と入力の関係（時間計算量）およびメモリ使用量と入力の関係（空間計算量）。

ビジネス的要求の分析 – 情報システムやソフトウェアアプリケーションの開発プロジェクトで，個々のユーザやユーザ組織が掲げるビジネス上の目標に基づいて目的のシステムやアプリケーションの最適な能力を決定するプロセス。

技術的要求の分析 – コンピューティング開発プロジェクトで，個々のユーザやユーザ組織が掲げる目標に基づいてコンピューティング用および通信用のハードウェアとソフトウェアを決定するプロセス。

ビジネス・モデル – 企業やその他の組織が，外部の主な利害関係者（顧客や納入業者など）とやり取りする方法を整理し，大きな目標（利益の最大化など）を達成するために使用する様々な構造やプロセスなどの機構。

回路とシステム – コンピューティング用・通信用のハードウェア・ソフトウェア。コンピューティングのプロジェクトや解を構成するもの。

コンピュータのアーキテクチャと構成 – デジタルコンピュータのコンポーネント（プロセッサ，レジスタ，メモリ，入出力装置を含む）の形態，機能および内部構成と，それに関連するアセンブリ言語の命令セット。

コンピュータシステム工学 – 北米よりも欧州で重視されているコンピューティング分野。CE, CS, SE の各領域を統合し，コンピュータのハードウェアとソフトウェアの緊密な統合が要求される複雑なシステムの開発に焦点を当てる。特に重点を置く領域は，組み込みシステムとリアルタイムシステムの設計と実装，コンピュータシステムの仕様に対する形式的手法の利用，特定用途向け回路へのシステムの実装などがある。

決定理論 – 特定の結果をもたらすことが知られている（またはそう考えられる）様々な選択肢から意思決定者が選択する作業を支援する知識と解析モデルを開発する研究分野。

デジタル論理 – コンピュータのハードウェアに適用する順序論理および非順序論理。適用対象には，回路や，コンピュータの基本構成がある。

デジタル媒体の開発 – 持ち運び可能なデジタル情報の格納媒体を扱うコンピューティング分野。

デジタル信号処理 – デジタルフィルタ，時間変換・周波数変換などの，アナログ信号を処理するデジタル手法を扱うコンピューティング分野。

分散システム – 相互に協調する複数の独立したコンピュータシステムの理論と応用。

E ビジネス – 情報・通信の技術による解を用いて，ビジネス・モデルを実装し，組織内外のビジネスプロセスを実装すること。狭義には，インターネット技術を利用した，企業間（B2B），企業と消費者の間（B2C）または消費者間（C2C）の取引を指すことが多い。

電子工学 – コンピューティング用・通信用の回路を構成するハードウェアのうち、電子信号に直接作用するか、ソフトウェアの実行を介して電子信号に作用するもの。現在、コンピューティングと通信の分野は、電子工学に完全に依存している。

組み込みシステム – 大規模システムのコンポーネントを構成するハードウェア・ソフトウェアのうち、ユーザによる介入が最小限で動作するもの（自動車の走行制御システムなど）。

ソフトウェアのための工学的経済学 – ソフトウェアエンジニアリングでのライフサイクルに対するコストモデル（ソフトウェアシステムの開発、保守、退役を含む）。

ソフトウェアのための工学的基礎 – ソフトウェアシステムに適用する、エンジニアリングとしての設計、プロセスおよび測定。

業績評価 – 組織が目標達成度を判断する活動。

機能的ビジネス分野 – 会計、財務、マーケティング、人事管理、製造および物流など。各々が、企業による特定の機能上の目標（社内外のビジネスパフォーマンスを信頼できる方法で測定する会計基準の設定など）の達成を支援する各種ビジネス活動に対して責任を負う。

一般システム理論 – 人間の振り舞いや自然科学の様々な領域におけるシステムの一般的な特性を研究する分野。特に複雑性とシステムコンポーネントの相互依存性に注目する。一般システム理論は、物理学、生物学および工学から派生したものであるが、経済学、組織理論、哲学、社会学、情報システムといった多数の分野にも応用されている。

グラフィックスと可視化 – コンピュータで生成したグラフィックスおよびデータや情報のグラフィカルな表現の理論と応用。静的技法、動的技法、アニメーション技法を含む。

ハードウェアのテスト法とフォールトトレランス – ハードウェアテストの迅速化、効率化およびコスト削減を扱う研究分野（電子工学および回路とシステムも参照）。フォールトトレランス（ハードウェアやソフトウェアの障害時にも仕様どおりの動作を継続する機能）を向上する方法も扱う。

ヒューマン・コンピュータ・インタラクション – 情報技術ソリューションとユーザ間の対話を設計し実装するのに用いるプロセス・手法・ツールに焦点を当てた、組織内の慣行および学術分野。

情報マネジメント（DB）の理論 – 情報の表現、格納および処理の論理的モデル。

情報マネジメント（DB）の実践 – 組織内の情報資源（データベース、データウェアハウス、知識管理システムなど）の分析、設計、実装および管理に関連する活動。

情報システム開発 – 新たな情報システムの開発につながる人間の活動（要件分析、論理設計と物理設計、システム導入など）。

統合プログラミング – プログラミングの基礎を利用して異種のハードウェアおよびソフトウェアを統合し、個々のシステムの能力を超える単一のシステムを円滑に構築すること。

人工知能（AI） – 人工知能理論に加え、ルールベースのシステム、遺伝的および進化論的計算、自己組織系などの技法に基づいたコンピュータ応用分野。

対人コミュニケーション – コンピューティングを学習する学生が口頭および文書によるコミュニケーションスキルを伸ばすのを支援する研究分野。チームワーク、プレゼンテーション、顧客や情報提供者とのやり取り、文書作成、営業およびマーケティング活動などに役立てることができる。

法律／職業／倫理／社会 - コンピューティング分野の職業専門人が、倫理に関する情報を入手したうえで、法体系および職務行動規範に沿った意思決定を行うことを支援する研究および実践分野。

情報システム組織のマネジメント - 組織向けの情報システムの開発・保守・管理・運営を主目的とする組織に所属する従業員と業者とを組織し管理するのに用いるプロセスおよび構造。

数学的基礎 - コンピューティング分野の基礎をなす数学。コンピューティングとの関連性が高い数学分野は、コンピューティング分野によって異なる。例えば、代数（線形および抽象）、微積分、組合せ理論、確率、統計などである。この用語は、数学とコンピュータ科学にまたがる研究分野（離散数学、グラフ理論、計算量理論など）を指すこともある。

ネットワークの原理と設計 - コンピュータ通信ネットワークの概念とプロトコル、マルチメディアシステム、ウェブの標準と技術、ネットワークセキュリティ、ワイヤレスコンピューティングとモバイルコンピューティング、分散システムといった広範なトピックが含まれる。

ネットワークの利用と構成 - ネットワーキング技術の選定、調達、実装、構成および管理に関連する組織活動。

オペレーティングシステムの原理と設計 - 全ハードウェア資源（プロセッサ、メモリ、外部記憶装置、入出力装置など）を管理し、アプリケーションソフトウェアと本体のインタフェースを提供するシステムソフトウェアの基礎を成す原理と設計。

オペレーティングシステムの構成と利用 - 1台以上のコンピュータへのオペレーティングシステムの導入、構成および管理。

組織行動論 - 組織内の個人およびグループの振る舞いに焦点を当てるビジネス管理の研究分野。コアトピックには、個人とグループによる意思決定、問題解決、研修、インセンティブの構成、目標設定などがある。

機構改革マネジメント - 組織の従業員が組織変更（内部要因によるものか、外部環境に起因するものかを問わない）を管理して対処することを支援するトピックに焦点を当てるビジネス領域に関連する研究分野。

組織論 - 組織の構成に焦点を当てるビジネス管理の研究分野。これにより、管理者は採用すべき組織構成を判断し、構成タイプの優劣が発生する理由を理解することができる。主なトピックは、権限の集中と分散、調整および制御を行う機構の選択と利用、指示系統の幅と厚み。

プラットフォーム技術 - 全アプリケーションプログラムを実行するコンピューティング用ハードウェアとオペレーティングシステムを扱う研究分野。

プログラミング基礎 - 手続き型プログラミングの基本概念（データ型、制御構造、関数、配列、ファイルおよび実行、テスト、デバッグの仕組みなど）とオブジェクト指向プログラミングの基本概念（オブジェクト、クラス、継承、多相性など）。

プロジェクト・マネジメント - プロジェクト作業に最良の成果をもたらす管理手法、組織の構成とプロセス、ツールと技術に焦点を当てる組織内の慣行および学術分野。

リスクマネジメント（プロジェクト、安全性、リスク） - プロセス、管理手法、リスク識別技術、重大性レベル判定技術、リスク対処方法の選択および導入技術に焦点を当てる、組織内の慣行および学術分野。

科学計算（数値解析的方法） - 連続数学に関する問題解決に利用する離散近似を計算するアルゴリズムと関連手法。

セキュリティの課題と原理 - コンピュータシステムとそこにある情報へのアクセスを制御するための、理論と応用。

セキュリティの実施と管理 - セキュリティプロセスと IT 基盤およびアプリケーションに関する技術の選定、調達、実装、構成および管理に関連する組織活動。

ソフトウェア設計 - 要求モデルを詳細モデルに変換する活動。詳細モデルは、ソフトウェアとしての解を表すものであり、通常は、アーキテクチャの設計仕様および詳細設計仕様からなる。[ソフトウェアエンジニアリングでは、指定の要件を満足するように、ソフトウェアアーキテクチャ（構成）、コンポーネント、モジュール、インタフェース、テスト方法、およびソフトウェアシステム用データを定義していくプロセス [ANSI/IEEE Standard 729-1983]をいう。]

ソフトウェアの進化（保守） - (1) 納入済みのソフトウェアシステムまたはソフトウェアコンポーネントを改修して、障害を解決し、パフォーマンスなどを改善し、環境の変化に対応するプロセス。(2) ハードウェアシステムまたはハードウェアコンポーネントを保守して、要求される機能を実行できる状態を維持したり回復するプロセス。[IEEE Std 610.12-1990]

ソフトウェア・モデリングと分析 - 顧客が本当に必要としていることを把握し、ソフトウェアで解決すべき真の問題を定義する目的で、顧客の要件と制約をモデル化する活動。

ソフトウェアプロセス - (1) 任意の目的で実行する一連の処置（ソフトウェア開発プロセスなど）。(2) オペレーティングシステムのスケジューラが管理する実行単位。(3) データを操作すること。[IEEE Std 610.12-1990]

ソフトウェアの品質 - (1) 製品が技術要件に準拠していることを保証する計画的かつ体系的な活動パターン。(2) 製品の開発や製造を行うプロセスを評価する一連の活動。[IEEE Std 610.12-1990]

ソフトウェアの検査と検証 - システムやコンポーネントの要件の完全性と正確性、各開発フェーズの成果物が前フェーズの要件や条件を満足しているか否か、完成したシステムやコンポーネントが指定の要件に準拠しているか否かを判断するプロセス。[IEEE Std 610.12-1990]

システム管理 - コンピューティング資源と通信資源（ネットワーク、データベース、オペレーティングシステム、アプリケーション、ウェブ配信など）の管理を扱う研究分野。管理問題には、導入、構成、運用、保守などがある。

システム統合 - コンピューティング資源と通信資源を組み込むことによって個別の要件を満足するシステムを構築することを対象とする研究分野。トピックは組織内の問題、要件、システムのアーキテクチャ、調達問題、テスト、品質保証など。

技術サポート - コンピューティング用途や通信用途の導入済み製品またはシステムに関するエンドユーザの問題解決を扱う研究分野。

プログラミング言語理論 - プログラミング言語の原理と設計。文法（シンタクス）、セマンティクス、型システム、各種の言語モデル（宣言型、関数型、手続き型、オブジェクト指向型）など。

VLSI 設計 - コンピューティングおよび通信に関する問題や要件への「電子的な」対処法の開発を扱う研究分野。トピックにはカスタム集積回路（IC）の設計（マイクロプロセッサやマイクロコントローラを含む）、特定用途向け IC 設計（標準セルやゲートアレイを含む）、プログラム可能なハードウェア（FPGA、PGA、PAL、GAL を含む）などがある。