

「情報サービス・ソフトウェア産業維新～魅力あるサービス・ソフトウェア産業の実現に向けて～(案)」に関するコメント

情報処理学会情報処理教育委員会

2006.7.9

概要

情報処理教育委員会において、産業構造審議会 情報経済分科会 情報サービス・ソフトウェア小委員会が公表した文書「情報サービス・ソフトウェア産業維新～魅力あるサービス・ソフトウェア産業の実現に向けて～(案)」[文献0]を検討した。その結果、その現状分析および提言の内容について全体的には賛同するところが大きであったが、いくつかの点、とりわけ次の4点においては修正を検討していただきたいと考える。

- IT 産業側だけでなく、ユーザ企業側にも (各分野を専門としつつ) 情報技術に理解と親和性を持つ人材の十分な供給が必要であること。
- IT 人材育成にあって「(分析/設計まで含めた)プログラミング」が基盤として欠かせないことを明確にすること。
- IT 技術者育成カリキュラムについては、情報専門学科以外に、各分野の専門教育に含める情報技術教育という視点も必要であること。
- 教育カリキュラムの検討においては、科目内容以外に「到達目標」「実務的視点や問題指向などの教育方法」等も考慮すべきこと。

1 はじめに

われわれ情報処理学会情報処理教育委員会は、わが国の情報教育全般の現況に危機感を持ち、「日本の情報教育・情報処理教育に関する提言 2005」[文献1](以下「提言」と記す)、「2005 年後半から 2006 年初頭にかけての事件と情報教育の関連に関するコメント」[文献2](以下「コメント」と記す)などの文書を公表してきた。

今回、産業構造審議会 情報経済分科会 情報サービス・ソフトウェア小委員会がわが国の情報サービス・ソフトウェア産業のあり方と将来に向けての提言を含む文書「情報サービス・ソフトウェア産業維新～魅

力あるサービス・ソフトウェア産業の実現に向けて～(案)」[文献0](以下「実現提案」と記す)を公表されたことは、関係各位にわが国の情報技術の将来の検討を促すきっかけとなる重要な活動であったと考える。まずこの点に敬意を表したい。また、「実現提案」の基本姿勢である「取引内容及び価値の可視化」およびその達成に向けての具体的提言についても、その趣旨については大いに賛同する。

ただし、主として人材育成に関わる事項の一部については、初等・中等・高等段階の情報教育・情報処理教育充実による国民全体の「情報水準」の向上を「提言」で主張しているわれわれの立場からすると、現状認識および目標達成手段などでの記述に修正が望ましいものが含まれていると考える。本文書はこれらの点を指摘し、可能な修正のいくつかを提案するものである。

繰り返しになるが、われわれとしては「実現提案」を公表された活動およびその全体的内容について高く評価するものである。その成果をより確かなものとするために、本文書に挙げるわれわれの提案をご検討いただき、適宜取り入れて頂けることを切に希望するものである。

2 本文書の構成

「実現提案」は次の4章から構成されている(第5章「まとめ」は除く)。

- 第1章 情報サービス・ソフトウェア産業の進むべき方向性
- 第2章 産業構造・市場の高度化(情報サービス産業競争力強化策)
- 第3章 イノベーションの高度化(ソフトウェア産業の国際競争力強化)
- 第4章 情報サービス・ソフトウェア産業の構造改革を支える高レベル人材の育成(「IT人材高度化戦略」)

このうち、我々が多くの知見を持ち、また注意を促したい点は第4章に多く含まれている。これは我々がこれまで情報教育を中心とした検討を行って来たことを考えれば自然である。ただし、第1～3章についても、現状認識や目標とすべき点について意見を述べたい箇所がいくつかある。このため、以下では各章ごとに分けて、特に注意したい点や「実現提案」の修正を検討いただきたい点について述べて行く形を採ることとする。

3 第1章「情報サービス・ソフトウェア産業の進むべき方向性」について

「実現提案」の第1章は、わが国の情報サービス産業およびソフトウェア産業がめざすべき方向性について論じた概論である。そこでは、今後各方面で使われるソフトウェアの規模はますます大きくなりソフトウェアの重要性が高まるにも関わらず、わが国の情報サービス・ソフトウェア産業はユーザ企業から見て技術力・企画提案力とも不足であり、今後の発展に不安があることを述べ、次の3課題への取り組みが必要であるとしている。

- 透明で価値創造型の産業構造・市場の創造 (特に、情報サービス産業)
- 国際競争に打ち勝つイノベーションの促進 (特に、ソフトウェア産業)
- 高レベル人材の育成 (両産業に共通する課題)

情報サービス産業・ソフトウェア産業における高レベル人材育成の議論において、「ユーザ側の高レベル人材」の必要性にも触れているのは卓見である。今日では企業活動の各段階において情報技術は基盤的な役割を果たすようになってきているため、各部署における大部分の人材が、本来の業務知識の専門家であることに加えて、一定水準以上の情報技術に関する知識と親和性を持たなければならないからである。ただそこに、「CIO等の」という形容があるために、かえって少数のトップだけが情報技術についての知見を持てばよいかのような印象を与えているのではないかと危惧する。

4 第2章「産業構造・市場の高度化」について

「実現提案」の第2章は、情報サービス産業と産業構造・市場の変化の関連について取り上げている。そ

こでは、現状の「ユーザ企業が弱い発注コーディネーション能力しか持たない」モデルの問題を指摘するとともに、「ユーザ企業が強い発注コーディネーション能力を持つ」複数のモデルを例示し、それらへの移行を示唆している。また、それを実現する手段として(1)構造の可視化、(2)価値の可視化、(3)政府調達改革、(4)グローバル展開、(5)地域・中小情報サービス産業活性化を提示している。

これらに関してわれわれが特に指摘したいのは、ユーザ企業側が強いコーディネーション能力を持つためには、ユーザ企業側に、その企業のドメインの専門家であり、なおかつ情報技術についても一定の知見と親和性を持つ人材が豊富に存在しなければならない、という点である。そのためには今後、「実現提案」で主に検討されている『情報システム構築者向け教育』に加え、『情報システム発注者向け教育』、『情報システム利用者向け教育』という3種類の情報処理教育の構成を検討し、体系的に実施することを目指す必要があると考える。

また、(1)や(2)の可視化についても、ユーザ企業側から見て構造や価値がきちんと見えるためには、上記の人材群が不可欠である。(5)の地域・中小情報サービス産業の活性化にしても、同様の人材群が不可欠であることはいうまでもない。それには、情報系の専門教育課程にとどまらず、広範な教育課程における情報教育・情報処理教育の推進が必要となることにも留意していただくとありがたい。

5 第3章「イノベーションの高度化」について

「実現提案」の第3章は、ソフトウェア産業が直面している技術的变化を中心に取り上げ、解決策として(1)オープン化、(2)知識情報社会、(3)技術開発戦略、(4)組み込みソフトの強化の4つを挙げている。

(2)で述べられている知識情報社会を支えるインフラストラクチャも、それを活用できるレベルに多くの国民があつてはじめて意義あるものとなることにも触れていただくとありがたい。

6 第4章「高レベル人材の育成」について

「実現提案」の第4章は、情報サービス・ソフトウェア産業の構造改革を支える高レベル人材の育成について取り上げ、現状が抱える問題点について整理した後、人材育成のための方策を複数取り上げている。

以下ではこの章の内容について整理しつつコメントしていく。

6.1 IT 人材供給の現状認識

「実現提案」ではIT 人材供給の現状について、次の問題を挙げている：

- 少子化/他産業との競合のため、優秀な人材の確保が困難
- 新卒人材の能力が低下している
- IT 関連分野を専攻する学生数が低下傾向にある

さらに3番目の項目について、その原因として次のものを挙げている：

- 勤務環境が厳しい (いわゆる 3K 問題)
- 能力に応じた適切な評価/処遇の欠如
- キャリアパスの未整備
- 産業界と教育界の連携不足
- 下流工程を中心とするアジア諸国へのオフショアリングの進展
- IT=プログラミングという誤解

これらについて、われわれもおおむね同意するところであるが、「IT= プログラミングという誤解」という表現については危惧する点がある。以下にその論点を整理する。

「IT=プログラミングという誤解」という表現は[文献4]の『CS=プログラミング』という狭いイメージ(注:CS=Computer Science)を土台にしていると思われるが、その元文献では”CS=Programming”と引用符をつけた上で「その”programming”からは”programmer”が連想され、ひいてはIT業界とは単なる”coder”(注:コーディングをする職種)というつまらない仕事であるという誤解が生まれている」「専門家と同じ”programming”という言葉で設計・開発・テスト・デバッグ・文書化・保守までの一連の活動を指すということが理解されていない」と述べている。

日本では、世間一般の人々や大多数の学生は「プログラミング」がどのようなものであるか体験したことがないだけに、より一層「よく分からない記号を書き連ねて行く単純労働」というイメージが強いかもしれない。

しかし専門家にとっては「プログラミング」はより広い範囲の、創造性を多く含んだ活動を意味している。「提言」において、「手順的な自動処理」の構築、

と呼んでいるのが、まさにこの意味でのプログラミングにほかならない。

この文書では、本来の意味でプログラミングという用語を用いることにした上で、一般の誤解を避けるため括弧書きを補って「(分析/設計まで含んだ)プログラミング」という呼称を用いている。(「システム構築における、業務分析から設計、プログラミング、テストまでのサイクルの全体」のような表現がより正確かも知れないが、あまりにも長い前記のようにした。)

ここで問題は、上記のように世間一般に「プログラミング」があまり重要でない下請け作業であるかのよう考えられ、「(分析/設計まで含んだ)プログラミング」という本来の姿が知られていない点であり、その誤解を解かなければ、高度IT人材を育成しようにも十分に人が集まらず、情報サービス・ソフトウェア産業の発展にとどまらず、わが国の今後の「高度情報社会」への適応そのものに支障をきたすであろう。「実現提案」をまとめた情報サービス・ソフトウェア小委員会も、その問題認識においてはわれわれと同じであると考えている。

ただ、「実現提案」ではその対処法として、「IT=プログラミングという誤解」を解き、「ITの最大の魅力はイノベーションであること」を伝えるとしている。その記述が、結果として、プログラミングという用語に対して否定的な印象を与えることにならないように配慮願いたい。「ITの最大の魅力はイノベーションであること」を強調するあまり「IT人材育成にプログラミングはいらぬ」という誤ったメッセージを伝えることになっては逆効果である。魅力となるイノベーションも、「(分析/設計まで含めた)プログラミング」を通じて始めて実現されるものであり、上流のイノベーションを展開するにも「(分析/設計まで含めた)プログラミング」の経験と能力に裏打ちされた情報処理システムの原理解が欠かせないからである。

ただし、大学等でのプログラミング教育において、「プログラミング言語の機能を順に講義する」ような時代錯誤な形態のものも残念ながら行われていることは確かである。そのような教育はIT人材の育成という目的から見れば有害であり、そのような「コーディングだけを目的としたような」教育を避けるべき、ということは明確に打ち出すのがよいと考える。

6.2 IT 人材と大学専門教育の現状認識

「実現提案」ではわが国の情報サービス・ソフトウェア産業で働くエンジニアの多くにあてはまる問題として、次のものを挙げている：

- a. 文系出身者、理系でも非情報系出身者の数が、情報系出身者より多い
- b. 情報系の学生が受ける教育は、理論・基礎知識偏重であり、大学によるばらつきが大きい

これに関連して、システム・インテグレーション分野では体系的知識と実践的能力の不足、組み込みソフトウェア分野ではこれを専門とする人材の不足、ソフトウェア開発分野では市場知識の不足や独創的クリエータの不足を挙げている。

上記2点のうち a. については、現在の IT 技術者不足を考えれば、今後急に情報系出身者の人数が増大して需要を充足し切るようになるとは考えられない。このため、各大学では情報系以外の専攻においても専門教育の一環として、適切な情報処理教育を必須内容として加え、多様な専門分野の大学出身者であっても情報技術に対する一定の知識や親和性を持つようにすべきと考える(われわれが「提言」において掲げた項目の1つである)。

したがって、「実現提案」において情報サービス産業・ソフトウェア産業人材育成のための教育内容について詳細に検討されていることには敬意を表するものであるが、それが情報系専門学科の範囲にとどまるのではなく、各分野の専門教育中に含まれる形で検討頂きたい。

具体的には、「実現提案」にある、情報専門系課程においてダブルメジャーの1つとしてビジネスモデル等経営系の内容を含めることは大いに賛成であるが、それと並行して、経営系の専門課程におけるダブルメジャーの1つとしての情報系科目のカリキュラムについても検討していくべきである。

b. については「実現提案」では実務的・応用的科目をきちんと盛り込むことが主張されており、その点については大いに賛成である。ただし、それらだけに重点が置かれ、「(分析/設計を含む)プログラミング」やアルゴリズムなどの基盤的な科目が軽視されることがないように、十分に配慮頂きたい。

「実現提案」に挙げられている情報サービス産業・ソフトウェア産業の新しい教育体系案については、短期間でまとめられたことに敬意を表するものであるが、「実現提案」自体でも述べられているように、今後さらに産業界・教育界・行政の協力による検討を進めていくことが必要と考える。

また、「実現提案」で指摘されている「理論的科目の偏重」については、実際には科目内容そのものより、その目標設定の不適切さや、教育方法における旧態依然さを指すのではないかと考えられる。たとえば「プログラミング入門」などの科目において、言語の機能

を逐一取り上げるだけで、実際的なソフトウェア開発や問題解決の視点に立たないものも残念ながら存在している。そのような教育方法を改め、プログラミングに本来内在されている問題解決の視点に立ち、実務的な問題を対象とし、学生に自らの力で考えさせるような教授方法の普及が望まれる。

したがって、今後のカリキュラム策定に際しては、各科目ごとに、単なる「科目内容」だけでなく「どの水準を到達目標とするか」「どのような形で学ばせるか」まで含めた形での検討が必要だということを確認して欲しい。

6.3 高度 IT 人材養成・供給メカニズムの構築について

「実現提案」では、情報サービス・ソフトウェア産業向けの高度 IT 人材供給に必要なことがらとして標準カリキュラムの策定を挙げ、その土台として ACM/AIS/IEEE-CS の Computing Curricula 2004[文献3](CC2004)に含まれる5カリキュラム(CE — Computer Engineering、CS — Computer Science、IS — Information Systems、SE — Software Engineering、IT — Information Technology)から次のように選択を行っている。

- 情報サービス産業の人材 → IT
- 組み込みソフトウェア産業の人材 → CE
- 独創的ソフトウェア開発人材 → CS

しかし、この選択では実務ソフトウェア開発でもっとも重要と考えられる IS(情報システム)カリキュラム、SE(ソフトウェア工学)カリキュラムが含まれていないことに疑問を感じる。ここは、5つのカリキュラムから3つを選ぶという形ではなく、各人材ターゲットごとに複数のカリキュラム選択肢を挙げることも考えられる。また、CE(コンピュータ工学)のカリキュラムについては、電気電子工学を土台とした、半分程度をハードウェアの学習に充てるカリキュラム構成となっており、わが国のソフトウェア産業のための人材養成カリキュラムを検討する際には適切な取捨選択が必要であることを注意しておきたい。

また、「実現提案」では教育機関に対する認証(ア krediteーション)制度の設置について「検討すべき」としているが、すでに日本技術者認定機構[文献5](JABEE)による認証制度が情報および情報関連分野においても行われており、約20の学部/学科が認定されている。また、「実現提案」ではFD(faculty development)の必要性およびそでの実務経験者の

活用についても述べているが、JABEEの情報および情報関連分野では同内容が審査項目として含まれている。したがって、教育水準の保証やFDの実現等について、産業界の協力を得てJABEEの運用を改善し、またその有効性を高め広報していくことは、「実現提案」の目的にもかなうところと考える。

7 まとめと結言

全体として「実現提案」には、わが国の情報サービス・ソフトウェア産業の現状分析および改善の方策が的確にまとめられており、そのことは高く評価したい。ただし、とくに次の4点については、一層の検討をお願いしたい。

- ユーザ企業に対して、CIO等だけに限定されることなく、ユーザ企業側の分野を専門としつつも情報技術について一定の知識と親和性を持つ人材の十分な供給が必要だという点を明確にしたい。
- 「(分析/設計まで含めた)プログラミング」は今日の情報技術の基盤であり、イノベーションなど上流部分について重視することは当然としても、この基盤もないがしろにされてはならないことを確認したい。
- IT技術者育成カリキュラムの検討に当たっては、情報系を専門とするものに加えて、それ以外の各分野を専門とするカリキュラムにおける情報技術教育の検討も、同等に重要であることを確認したい。
- 教育カリキュラムの検討においては、科目内容にとどまらず、その到達目標、および基盤的科目であつても実務的な問題を対象とした問題解決の視点を持たせて教育するなどの、教育方法まで含めた検討が重要であることを明確にしたい。

参考文献

[文献0] 産業構造審議会 情報経済分科会 情報サービス・ソフトウェア小委員会, 情報サービス・ソフトウェア産業維新～魅力ある サービス・ソフトウェア産業の実現に向けて～ (案), <http://search.e-gov.go.jp/servlet/Public?CLASSNAME=Pcm1010&BID=595206008>, 2006.

[文献1] 情報処理学会情報処理教育委員会, 日本の情報教育・情報処理教育に関する提言 2005, <http://www.ipsj.or.jp/12kyoiku/proposal-20051029.html>, 2005.

[文献2] 情報処理学会情報処理教育委員会, 2005年後半から2006年初頭にかけての事件と情報教育の関連に関するコメント, <http://www.ipsj.or.jp/12kyoiku/statement2006.html>, 2006.

[文献3] The Association for Computing Machinery (ACM) / The Association for Information Systems (AIS) / IEEE Computer Society (IEEE-CS), Computing Curricula 2004 Overview Report A Guide to Undergraduate Degree Programs in Computing, http://as.nida.ac.th/resource/pdf/Overview_Draft_11-22-04.pdf, 2004.

[文献4] Peter J. Denning, Andrew McGettrick, Re-centering Computer Science, Communications of The ACM, vol. 48, no. 11, pp. 15-19, 2005.

[文献5] 日本技術者教育認定機構ウェブサイト, <http://www.jabee.org/>

情報処理学会 情報処理教育委員会委員長 笈 捷彦
(早稲田大学理工学術院教授)
初等中等教育委員会委員長 久野 靖
(筑波大学大学院教授)

付録1 Q&A

注記: この「付録1 Q&A」は情報処理学会内部での検討時に論点を整理するために作成したものであり、提出したパブリックコメントには含まれていない。読者の参考のためにここに掲載する。

- Q. 「実現提案」を読みましたが、大いに賛同できる場所が大だと感じたのですが。
- A. 本文冒頭に書いたように、我々も「実現提案」を高く評価しています。ただ、一部に我々の考えと相違するところがあり、その部分の再検討をお願いしたい、ということです。
- Q. システム開発全体を通して見ると、コンセプトや上流部分が重要だというのはまさに正しい主張だと思うのですが。
- A. もちろんその通りです。しかし、そのことを強調するあまり、多くの人材がプログラミングやコンピュータの原理を学ばないまま上流部分の設計を行う方向に進んでしまうと、設計したものの実現性などの部分において不安が生じるものと考えます。さらに、一般的に言えば、プログラミングなどを通じてコンピュータの動作原理に触れた後

で、次第に上流部分について理解を深めて行く、というカリキュラム構成が有効であるとも考えています。繰り返しますが、我々は「上流が重要ではない」とは主張していません。「上流も重要であるが、それを的確にこなすためには、「(設計/分析まで含めた)プログラミング」も重要であるので、そのことを明記して欲しい」ということが我々の要望です。

Q. 「実現提案」でプログラミングを否定的に捉えているというのは誤読ではないのでしょうか?情報サービス・ソフトウェア小委員会がそのような意見を持つとは思われないのですが。

A. もちろん、情報サービス・ソフトウェア小委員会はソフトウェア開発において上流(設計・分析)から下流(実現)まですべて重要だという適切な認識をお持ちだと思います。しかし「実現提案」本文の中では、「プログラミング」という言葉は100解を正す」という文脈でのみ現れており(ほかに「プログラミング入門」「プログラミング言語」など複合語の一部としての出現はあります)、また科目名でも「プログラミング」を含むものは否定的に言及されているように読めます。このために、「実現提案」を読んだ人が「プログラミングを重視しないでよい」というメッセージを受け取る可能性が大いにあります。この点をご配慮頂き、教育上、(上流工程も重要だが)プログラミングをきちんと学んでおくことも重要だ、というメッセージを入れて頂きたいというのが我々の要望です。

Q. 「プログラミング」という用語に多義性があること、すなわち一般のほとんどの人が「プログラミング=コーディング」だと思っているのに対し、専門家の多くはプログラミングが分析/設計まで含むと考えていることが問題なのではないでしょうか。

A. まったく同感です。そのために、本文では「プログラミングの本質はコーディングではない」ことを明記し、また意識的に「(設計/分析までを含んだ)プログラミング」という言い方をしています。より詳細に「ワークフロー」「モデリング」などの用語を入れてはという考えもありますが、それらの専門用語を導入して読み手に理解してもらうのは難しそうだと考え、上記のように選択しました。

Q. 「実現提案」で「IT=プログラミング」と書かれている「プログラミング」はコーディングのことなのだから、「コーディング」と書いてもらうよ

うにしたらよいのではないのでしょうか。

A. 確かに文面としてはその方が正確になると思います。しかし、「コーディング」「狭義のプログラミング」などの用語に差し替えても、その内容はよほど詳しく説明しないと読み手に分かって貰えず、実効性がないのではと危惧します。さらに、そのようにして「プログラミング」という用語を削除することが世間一般における「プログラミング=単純作業」という誤解を解き、きちんとしたプログラミング教育を実施することにつながると思えません。このため、そのような訂正ではなく、プログラミングの本質的内容をより正確に記述して頂くことを要望するという立場を取りました。

Q. 「手順的な自動処理」すなわちステップを経て行いたいことが実現される、というのは不正確ではないのでしょうか。つまり、手順を取捨し抽象化することで、望まれる処理内容を宣言的に記述し、それを実現することが最終的には重要だと考えます。

A. おっしゃる通り、システム全体として上流から下流まで通して考えた場合はその方が適切だと考えます。ただ、我々が「手順的な自動処理」を取り入れることを主張しているのは、初等中等教育において、コンピュータの動作の本質を体験させ、コンピュータに何ができて何ができないかの直観的理解を与えることを目的としてのことです。この段階では、コンピュータの動作原理との対応性が必要なので「手順」に注目することは適切と考えています。このため本文でも「手順的な自動処理」は初等中等教育と結びつけて述べているつもりです。高等教育後半以後はご指摘通り、手順やプログラミングを離れて上流部分の考え方、宣言的な考え方を学ぶのが適切だと考えます。

Q. 今日ではプログラミングを学ぶことは「読み書き計算」に代わる「読み書きプログラミング」のようなものであり、したがって、プログラミングは産業のためにあるのではなく、基本的素養として捉えていただきたいのですが。

A. はい、我々も「提言」ではそのような立場を明確に述べているつもりです。今回の「実現提案」が情報サービス・ソフトウェア産業の改革を主題としており、本文がそれに対してコメントしているため、上記の印象を持たれたのではないのでしょうか。本文でも国民全体の「情報水準」の向上を明確に主張しているつもりです。